On the Capacity-Performance Trade-off of Online Policy in Delayed Mobile Offloading

Han Deng and I-Hong Hou

***Abstract*—WiFi offloading, where mobile users opportunistically obtain data through WiFi rather than 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 users' movements are unpredictable, we focus on online scheduling policy where APs have no knowledge about 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 such that when the requested data by each user is very large, the policy can offload $(e-1)/e$ as much data as the offline policy, where e is Euler's constant.

We further study the case where the service provider can increase the capacity of WiFi so as to provide some guarantees on the amount of offloaded data. We derive a lower-bound on the trade-off between capacity and the amount of offloaded data, and propose a simple online policy that achieves this lower-bound. In addition, we show that our policy only needs half as much capacity as current mechanisms 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 offloading [6]–[8]. In delayed mobile offloading, a large amount of mobile users need to obtain delay-tolerant data,such as Dropbox synchronization and App updates, from service providers. Each mobile user sets a deadline for its data, and opportunistically obtains these data through WiFi whenever it is connected to WiFi access points (APs), so as to reduce traffic on cellular networks. Due to its own mobility patterns, a mobile user may only have intermittent WiFi connections. If a user fails to obtain all its data by the deadline, it downloads the remaining data through cellular networks.

When multiple users are connected to one WiFi AP, the AP makes decision on which user to serve. We aim to design scheduling policies for WiFi APs that maximize the amount of data offloaded to WiFi. We focus on WiFi offloading because it has been extensively deployed. However, all our results can be directly applied to other means of mobile offloading, such as offloading through femto cell networks.

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 when the requested data amount is large , where e is Euler's constance.

A fraction of $\frac{e-1}{e}$ of data offloaded to 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

Han Deng is with Department of ECE, Texas A&M University, College Station, Texas, 77843-3128, USA. Email: hdeng@tamu.edu

I-Hong Hou is with Department of ECE, Texas A&M University, College Station, Texas, 77843-3128, USA. Email: ihou@tamu.edu

This material is based upon work supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-15-1-0279 and NPRP Grant 8-1531-2-651 of Qatar National Research Fund (a member of Qatar Foundation).

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 $\frac{1}{2(\beta-1)}$. The value of β is chosen by the service provider based on its required offloading guarantees. On the other hand, even when APs only use a fixed transmission rate, the other commonly-used round-robin, max-weight, and proportional fair policies need 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 further prove that no policy can guarantee offloading $\frac{1}{\beta}$ data with less than $\frac{1}{2(\beta-1)}$ capacity, and therefore our policy achieves the optimal trade-off between capacity and performance.

Theoretical analysis only shows that the worst-case performance of our policies is better than that of the three commonly-used policies. 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 other three on average. The fact that our policy is significantly better than these widely used policies in WiFi scheduling, in terms of both theoretical bounds and simulation results, further highlights that delayed WiFi offloading is fundamentally different problem from traditional WiFi scheduling.

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 studies the performances of three commonly-used policies. Section VIII compares the competitive ratio of our proposed policies with the three commonly-used policies. Section IX provides some practical implication of our policy. Section X provides the simulation results. Finally, Section XI concludes this paper.

II. Related Work

Many experimental studies have shown that mobile offloading is promising. Gass and Diot [9] 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. [10] 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. [11] 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. Additional work such as [12], [13] study the AP side. Dimatteo et al. [12] study how many APs are required to cover a metropolitan area offloading. Trestian et al. [13] propose to upgrade the network capacity in a selected number of locations, called Drop Zone. They design infrastructure placement algorithm which tries to reduce the AP number. It shows that by upgrading less than 1000 infrastructures across US will upload 50% of data. However, there is still no research on upgrading the APs to guarantee the offload data ratio.

Surveys in [14], [15] provide some results on how much time users are willing to wait for different applications. Lee et al. [16] analyze how much economy benefit can be generate by delayed offloading and uses real traces for numerical analysis. Mehmeti and Spyropoulos [6] use a queueing model for delayed mobile offloading and analyze the mean delay as a function of number of users and AP availability. Cai and etc. [7] propose an mechanism to encourage users to participate in delayed WiFi offloading by reward. Thus the delayed mobile offloading problem is actually worth considering.

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 [17]–[19]. Cheung and Huang [8] study the WiFi offloading problem by formulating the problem as a finite-horizon Markov decision process by using the prediction in [19]. Li et al. [20] study using a small set of mobile users to offload data, and propose a policy based on sub-modular optimization. Whitbeck et al. [21] consider using offloading to reduce the burden in broadcasting messages. Hou et al. [22] propose a transport layer protocol to integrate 3G and WiFi networks for vehicular network access. Barbieri et al. [23] propose a system design for mobile offloading with pico base stations. Bennis et al. [24] and Singh et al. [25] consider the problem of network self-organizing for offloading traffic. Bilgir Yetim and Martonosi [26] propose offline scheduling policies for WiFi offloading. These studies assume that user mobility follows some well-defined random process. In real life, user mobility may be non-ergodic, and these studies cannot be applied. In contrast, our work aims to maximize the total offloaded data without any assumptions on user mobility.

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 $K_{imt} = 0$. There have been some advancements in multi-homing, where a mobile user can be connected to multiple APs simultaneously. Our model can easily accommodate multi-homing by allowing a user i to have $K_{imt} > 0$ for multiple APs. We assume each user can be connected to at most H APs at any given time.

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 $0 \leq K_{imt} \leq \frac{1}{H}$ for all i, m, t. Therefore, at each time t, each client can at most obtain one unit of data.

Once client i is connected to AP m at the beginning of time slot t, the time slot is fully occupied by client i, regardless whether or not AP m is transmitting data to client i. AP m employs some scheduling policy to determine the portion of time it spends transmitting to mobile user i during time slot 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 use η to denote an online policy. We let $\Gamma_\eta(1)$ be the value of $\sum_{imt} X_{imt} K_{imt}$ under policy 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.

We summarize the notations we used in the paper in Table I.

IV. Preliminary

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

A standard form of linear programming problem (LP)

TABLE I
NOTATIONS

$\mathcal{I}$	Mobile user set
i	Single user
$\mathcal{M}$	WiFi AP set
m	Single AP
t	Time slot
K_{imt}	Normalized channel capacity between AP m and user i at time t
X_{imt}	Portion of time AP m spends on user i at time slot t
H	The maximum number of APs that a user can connect to
C_i	Data amount that user i needs
T_i	Time that user i switch from WiFi to cellular network
Y_{mt}, Z_i	Dual variables
η	Policy
R	Capacity increasing rate
Γ_{opt}	Optimal value of (1) in **Offload**
$\Gamma_{\eta}(1)$	Value of (1) in **Offload** under policy η
$\Gamma_{\eta}(R)$	Value of (5) in **Offload(R)** under policy η
β	Value of $\max(\Gamma_{opt}/\Gamma_{\eta}(R))$

is:

$$\begin{aligned}(P): Max &\sum_{i=1}^{n} c_i x_i,\\ s.t. &\sum_{i=1}^{n} a_{ij}x_i \le b_j, \forall 1 \le i \le n,\\ &x_i \ge 0,\end{aligned}$$

and its dual is

$$\begin{aligned}(D): Min &\sum_{j=1}^{m} b_j y_j,\\ s.t. &\sum_{j=1}^{m} a_{ij}y_j \ge c_i, \forall 1 \le j \le m,\\ &y_j \ge 0.\end{aligned}$$

We have the following two fundamental theorems:

Theorem 1 (Weak Duality [27]): 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:

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

Theorem 2 (Complementary Slackness [27]): 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 $c_i \le \sum_{j=1}^{m} a_{ij} y_i \le \beta \cdot c_i$ for some $\beta > 1$;
- If $y > 0$, then $\sum_{i=1}^{n} a_{ij} x_i = b_j$;

Then:

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

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

$$\begin{aligned}(D): Min &\sum_{mt} Y_{mt} + \sum_{i} C_i Z_i, &(9)\\ s.t.\ &Y_{mt} + K_{imt} Z_i \ge K_{imt}, \forall i, m, t, &(10)\\ &Y_{mt} \ge 0, \forall m, t, &(11)\\ &Z_i \ge 0, \forall i, &(12)\end{aligned}$$

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, $2K_{imt} = 2 \ge Y_{mt} + K_{imt} Z_i \ge K_{imt}$. By Theorem 2, we know $\sum_{mt} Y_{mt} + \sum_i C_i Z_i \le 2 \cdot \sum_{imt} X_{imt} K_{imt}$. Further, $\Gamma_{opt} \le \sum_{mt} Y_{mt} + \sum_i C_i Z_i$, 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 $K_{imt}(1 - Z_i)$, and delivers K_{imt} data to the client. Each time client i obtains K_{imt} data from an AP m, Z_i will be updated as $Z_i(1 + \dfrac{K_{imt}}{C_i}) + \dfrac{K_{imt}}{(d-1)C_i}$. At time t, if there are multiple APs that serve i at the same time t, which is possible under multi-homing, Z_i will be updated as $Z_i(1+\dfrac{\sum_{m:m \text{ serves } i} K_{imt}}{C_i})+\dfrac{\sum_{m:m \text{ serves } i} K_{imt}}{(d-1)C_i}$. 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:         $i^*_m \leftarrow \underset{i}{\text{argmax}}\{\sum_{m:m \text{ serves } i} K_{imt}(1 - Z_i)\}$.
 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:             $X_{i^*_m mt} \leftarrow 1$.
 9:         end if
10:     end for
11:     for each client $i$ do
12:         $Z_i \leftarrow Z_i(1 + \frac{\sum_{m:m \text{ serves } i} K_{ipt}}{C_i}) + 1\frac{\sum_{m:m \text{ serves } i} K_{ipt}}{(d-1)C_i}$.
13:         if $\sum_{p,s \le t} X_{ips} K_{ips} > C_i$ then
14:             $X_{imt} \leftarrow \frac{C_i - \sum_{p,s<t} X_{ips}K_{ips}}{\sum_{p:p \text{ serves } i} K_{ipt}}, \forall\ m$ servers $i$
15:         end if
16:     end for
17: 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.

At time t, it is possible that i already obtains most of its data and only needs less than $\sum_{m:m \text{ serves user } i} K_{imt}$ data to complete its download. In this case, APs use only a fraction of a time slot to deliver all remaining data that i needs. Step 14 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,

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

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 $s + 1$,

$$\begin{aligned}
&{Z_i}^{(s+1)} \\
=&{Z_i}^{(s)}(1 + \frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)}}{C_i}) + \frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)}}{(d-1)C_i} \\
\ge&\frac{1}{(d-1)}(d^{\sum_{m,t \le s} \frac{X_{imt}K_{imt}}{C_i}} - 1)(1 + \frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)}}{C_i}) \\
&+ \frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)}}{(d-1)C_i} \\
=&\frac{1}{(d-1)}[d^{\sum_{m,t \le s} \frac{X_{imt}K_{imt}}{C_i}}(1 + \frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)}}{C_i}) - 1]
\end{aligned}$$

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

$$\begin{aligned}
&{Z_i}^{((s+1)} \\
\ge&\frac{[(d^{\sum_{m,t \le s} \frac{X_{imt}K_{imt}}{C_i}})(1 + \frac{1}{C_{min}})^{\frac{\sum_{p:p \text{ serves } i} K_{ip(s+1)C_{min}}}{C_i}} - 1]}{(d-1)}
\end{aligned}$$

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

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

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:

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

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

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

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 14 ensures that (2) holds. By Lemma 1, $Z_i^{(t)} < 1$ only when i does not receive all its data at time t. Hence, X_{imt} is updated only if the total received data of client i is less than its C_i, which is $\sum_{p,s<t} X_{ips}K_{ips} < C_i$, which makes (3) and (4) hold.

Third, we show that every time steps 7, 8, and 12 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 14 now, which will be taken into account later.

When AP m schedules i at time t, X_{imt} is increased from 0 to 1, and (1) is increased by $\sum_{m:m \text{ serves } i} K_{imt}$. Meanwhile, (9) is increased by

$$\begin{aligned}&\sum_{m:m \text{ serves } i} K_{imt}(1 - Z_i^{(t-1)}) + C_i(Z_i^{(t-1)} \frac{\sum_{m:m \text{ serves } i} K_{imt}}{C_i}\\ &+ \frac{\sum_{m:m \text{ serves } i} K_{imt}}{(d-1)C_i})\\ =&(1 + \frac{1}{d-1}) \sum_{m:m \text{ serves } i} K_{im_jt}.\end{aligned}$$

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 14. 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 14. Step 14 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 14 is invoked at most for one time slot for each client. Further, when step 14 is invoked, (1) decreases by no more than $\sum_{m:m \text{ serves } i} K_{imt} \leq 1$. Let $\Gamma_{prim,\eta}$ be the value of (1) under Algorithm 1 with step 14. 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 $d \to e$, as $C_{min} \to \infty$, 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 offload 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 exist, 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, enters the AP coverage area at time 1 and leaves at time C, while all other clients enter the AP coverage area at time 1 and stay forever. We have $T_i = \infty$ for all clients. 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. Primal-Dual Scheduling Policy

The primal-dual (PD) scheduling policy is very similar to Algorithm 1. It is described as 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 Primal-Dual Algorithm

```
Initially, $X_{imt} = 0$, $Y_{mt} = 0$, $Z_i = 0$.
$C_{min} \leftarrow \min_i C_i, d \leftarrow (1 + 1/C_{min})^{C_{min}/R}$.
for each time slot $t$ do
    for each AP $m$ do
        $i^*_m \leftarrow \operatorname{argmax}_i\{K_{imt}(1 - Z_i)\}$.
        if $K_{i^*_m mt}(1 - Z_{i^*_m}) > 0$ then
            $Y_{mt} \leftarrow K_{i^*_m mt}(1 - Z_{i^*_m})$.
            $X_{i^*_m mt} \leftarrow R$.
        end if
    end for
    for each client $i$ do
        $Z_i \leftarrow Z_i(1 + \frac{\sum_{m:m \text{ serves } i} K_{imt}}{C_i}) + \frac{\sum_{m:m \text{ serves } i} K_{imt}}{(d-1)C_i}$.
        if $\sum_{p,s\leq t} X_{ips}K_{ips} > C_i$ then
            $X_{imt} \leftarrow \frac{C_i - \sum_{p,s<t} X_{ips}K_{ips}}{\sum_{p:p \text{ serves } i} K_{imt}}$. $\forall m : m$ serves $i$
        end if
    end for
end for
```

We now study the competitive ratio of PD.

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

$$Z_i^{(t)} \geq (\frac{1}{d-1})(d^{\sum_{m,s\leq t} \frac{X_{imt}K_{imt}}{C_i}} - 1). \tag{14}$$

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

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

Proof: We prove PD is $(R, \dfrac{e^{1/R}}{R[e^{1/R}-1]})$-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 14 ensures (6). Further, by Lemma 2, $Z_i^{(t)} < 1$ only if $\sum_{m,s\leq t} X_{imt}K_{imt} < C_i$. 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, 8, and 12 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 14 now.

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

$$\sum_{m:m \text{ serves } i} K_{imt}(1 - Z_i^{(t-1)}) + C_i(Z_i^{(t-1)} \frac{\sum_{m:m \text{ serves } i} K_{imt}}{C_i} + \frac{\sum_{m:m \text{ serves } i} K_{imt}}{(d-1)C_i}) =(1+\frac{1}{d-1}) \sum_{m:m \text{ serves } i} K_{imt}.$$

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 14. Step 14 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 14 is invoked at most once for each client. Further, when step 14 is invoked, (5) decreases by no more than $R\sum_{m:m \text{ serves } i^*_m} K_{i^*_m mt} \leq R$, since we normalized our system such that $K_{imt} \leq \frac{1}{H}$. Therefore, throughout the system lifetime, the ratio of decrease caused by step 14 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 PD is $(R, \dfrac{e^{1/R}}{R[e^{1/R}-1]})$-competitive.

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

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

when $R \gg 1$.

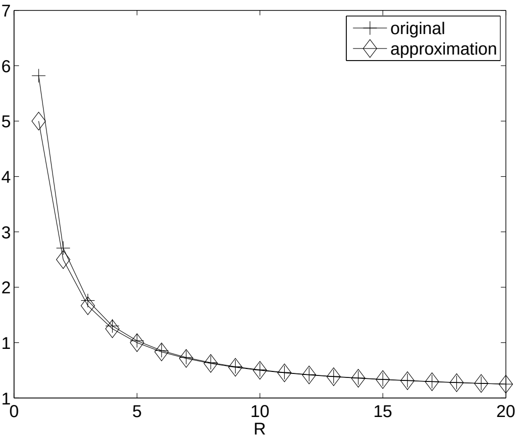


Fig. 1. Illustration of the approximation.

Thus, the competitive ratio is approximately $(R, 1+\frac{1}{2R})$. Fig. 1 plots $\dfrac{e^{1/R}}{R[e^{1/R}-1]}$ 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. ■

B. Least Progress First Scheduling Policy

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

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

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 PD makes AP m schedule the client with the largest $K_{imt}(1-Z_i)$, we can approximate PD 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 and is called *Least Progress First* (LPF). 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 LPF becomes the same as the well-known

Max-Weight scheduling policy. However, as we will show in Section VII-B, Max-Weight policy can be much worse than LPF when different clients have different C_i.

Algorithm 3 Least Progress First

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

C. 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: For any given $\epsilon > 0$, there exists a positive number R_0, such that, when $R > R_0$, no policy has better competitive ratio than $(R, 1+\frac{1}{2R}-\frac{\epsilon}{R})$.

Proof: Consider a system consisting of N clients with same file size C and one AP with On-Off channels. Assume there are k clients that will move around the area and $N-k$ clients stay in the same place. For client i, if $i \leq k$, $K_{i1t} = 1$ in time $1 \leq t \leq i \times C$, in other words, client 1 is connected to AP in time $[1, C]$; client 2 is connected to AP in time $[1, 2C]$; client 3 is connected to AP in time $[1, 3C]$; etc. If $i > k$, client i is connected to the AP forever. $T_i = \infty$ for all clients.

If the connection of all clients are known in advance, the optimal offline policy with unit capacity is to schedule the clients in the order of $1, 2, 3, ..., N$. The optimal offline policy is then able to transmit NC amount of data.

On the other hand, online policies cannot know K_{imt} in advance. Also, all clients have the same file size. 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. In the first $2C$ time slots, client 2 recieves $\frac{RC}{N}+\frac{RC}{N-1}$ amount of data.... In the first $k \times C$ time slots, client k recieves $\frac{RC}{N}+\frac{RC}{N-1}+...+\frac{RC}{N-k+1}$ amount of data. The other clients receive all their data. Thus, the AP can at best deliver $\Gamma_\eta(R) = \{\frac{RC}{N}\} + \{\frac{RC}{N}+\frac{RC}{N-1}\} + ... + \{\frac{RC}{N}+\frac{RC}{N-1}+...+\frac{RC}{N-k+1}\} + (N-k)C$ amount of data.

Let $N = (k+1)R$. Since $\frac{R}{N} < \frac{R}{N-1} < ... < \frac{R}{N-k+1}$, the competitive ratio is:

$$
\begin{aligned}
\beta &= \frac{NC}{\Gamma_\eta(R)} \\
&> \frac{N}{\frac{R}{N-k+1}\cdot\frac{(1+k)k}{2} + (N-k)} \\
&= \frac{(k+1)R}{\frac{R}{(k+1)R-k+1}\cdot\frac{(1+k)k}{2} + ((k+1)R-k)} \\
&> \frac{(k+1)R}{\frac{R}{(k+1)R-(k+1)}\cdot\frac{(1+k)k}{2} + ((k+1)R-k)} \\
&= 1 + \frac{kR-2k}{2(k+1)R(R-1)-2k(R-1)+kR} \\
&> 1 + \frac{kR-2k}{2(k+1)R^2-2kR+kR} \\
&> 1 + \frac{kR-2k}{2(k+1)R^2} = 1 + \frac{k-2k/R}{2(k+1)R} \\
&= 1 + \frac{1}{2R} - \frac{1}{R}(\frac{1}{2(k+1)} + \frac{k}{(k+1)R})
\end{aligned}
$$

For sufficiently large k and R, we have $\frac{1}{2(k+1)} < \epsilon/2$, $\frac{k}{(k+1)R} < \epsilon/2$, and therefore $\beta > 1 + \frac{1}{2R} - \frac{\epsilon}{R}$. This completes the proof. ■

Since PD is approximately $(R, 1+\frac{1}{2R})$-competitive, Theorem 6 demonstrates that PD indeed achieves the optimal trade-off between capacity and the amount of offloaded data.

VII. Comptitive Ratios of Other Policies

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 following three policies.

A. Round Robin Scheduling Policy

With round robin policy (RR), 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: $C_1 = NC$, and $C_i = RC$, for all $i \neq 1$. $K_{11t} = 1$ for $1 \leq t \leq C_1$, and $K_{11t} = 0$ for $t > C_1$. For $i \neq 1$, $K_{i1t} = 1$ for all t. In other words, client 1 is connected to AP in time $[1, C_1]$, while all other clients are connected to the AP forever. $T_i = \infty$ for all clients. The system is shown is Fig. 2.

The optimal offline policy is first to serve client 1 from time 1 to NC. Then client 2 to N will be served. Thus 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

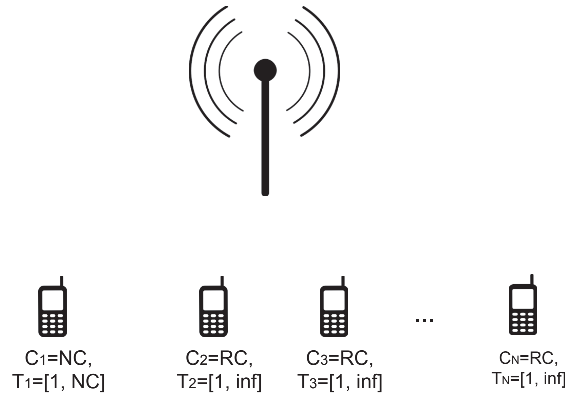


Fig. 2. System Construction for Round Robin.

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 least:

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

as $N \to \infty$. ■

B. Max-Weight Policy

With max-weight (MW) policy, of all connected clients, each AP selects the one with maximum value of the product of K_{imt} and the client's remaining data.

Theorem 8: The max-weight policy cannot have better competitive ratio than $(R, 1 + \frac{1}{R})$.

Proof: Given R, we construct a system with one AP and $N+1$ clients as follows: $C_1 = MC + C$, and $C_i = C$, for all $i \neq 1$. $K_{11t} = 1$ for all t, $K_{i1t} = 1$ for $1 \leq t \leq NC$. In other words, client 1 is connected to AP forever, while all other clients are connected to AP in time $[1, NC]$. Here we choose $N = M/R$, and $T_i = \infty$ for all clients.

The optimal offline policy is first to offload data requested by client 2 to $N+1$ in time $[1, NC]$. Then the policy will offload data requested by client1. In this case, the policy is able to deliver all $C_1 + \sum_{i=2}^{N} C_i = MC + C + NC$ amount of data with unit capacity.

With R capacity, MW policy first transmits MC data to client 1 until it has remaining data request of C, which is the same as the other clients. It takes NC amount of time to transmit MC amount of data. Since client i, $i \neq 1$, has a deadline of NC, they will no longer get data through WiFi. Then WiFi will transmit the remaining C data to client 1. Thus the total amount of data transmitted is $MC + C$. The competitive ratio of MW policy cannot be better than:

$$\begin{aligned}\beta &= \frac{MC + C + NC}{MC + C} = 1 + \frac{N}{M+1} \\ &\to 1 + \frac{1}{R}\end{aligned}$$

as $N \to \infty$. ■

C. Proportional Fair Scheduling Policy

With proportional fair (PF) policy, of all connected clients, each AP selects the one with the maximum value of $\frac{K_{imt}}{Throughput\ of\ client\ i}$.

Theorem 9: The proportional fair policy cannot have better competitive ratio than $(R, 1 + \frac{1}{R})$.

Proof: Given R, we construct a system with one AP and $N+1$ clients as follows: $C_1 = MC$, and $C_i = C$, for all $i \neq 1$. $K_{11t} = 1$ for $1 \leq t \leq MC$, $K_{i1t} = 1$ for all t. In other words, client 1 is connected to AP in time $[1, MC]$, while all other clients are connected to AP forever. Here we choose $N = MR$, and $T_i = \infty$ for all clients..

The optimal offline policy is first to serve client 1 in time 1 to MC. After time NC, the other clients will be served. Thus the optimal offline policy is able to deliver all $C_1 + \sum_{i=2}^{N} C_i = MC + NC$ amount of data with unit capacity.

With R capacity, from time $[1, MC]$, PF policy delivers $\frac{MCR}{N+1}$ data for each client. After time MC, all clients except client 1 are connected and PF policy will deliver the remaining data of them. Thus PF policy delivers a total amount of $\frac{MCR}{N+1} + NC$ data. The competitive ratio of PF policy cannot be better than:

$$\begin{aligned}\beta &= \frac{MC + NC}{\frac{MCR}{N+1} + NC} = \frac{\frac{N}{R} + N}{\frac{N}{N+1} + N} \\ &\to 1 + \frac{1}{R}\end{aligned}$$

as $N \to \infty$. ■

VIII. Performance Comparison

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 RR, MW, and PF is close to that of PD. However, we should 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 PD, 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 RR, MW, and PF policies still need at least $\frac{1}{\beta-1}$ capacity to achieve the requirement. In other words,PD needs about half as much capacity to provide the same guarantee as the other three policies. Thus, our policy is much more preferable to provide stringent performance guarantees. Fig 3 illustrates the capacity requirements for different β. In order to guarantee to offload at least 95% as much

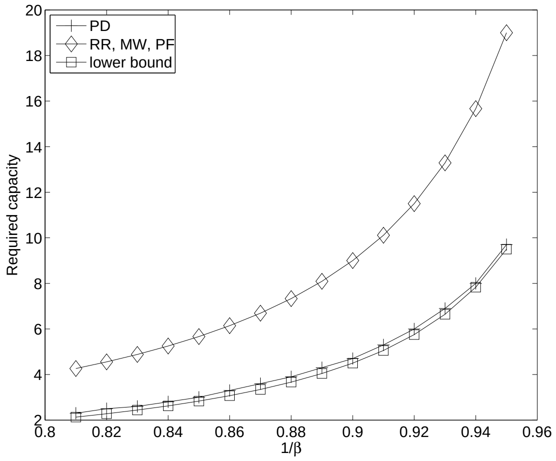


Fig. 3. Capacity requirements of different policies.

data as the optimal offline policy, PD needs to increase the capacity by 9.7 times, while the other policies needs to increase the capacity by at least 19 times.

IX. Practical Implication

In this section, we will discuss the major practical implications of our work.

First, our work can be used for system planning. When a wireless operator is to deploy public WiFi APs, it can use the macro-scale statistics of system history to estimate the amount of resource needed for the offline policy to offload a certain amount of data.

For example, suppose we have collected the mobility patterns of all clients in a specific area and the clients' demand of the past 30 days. We use K_{imt} as clients' connection status to indicate the mobility pattern. We use C_i to represent the demand. Suppose the wireless operator needs to offload a total of C_0 amount of data. Here $C_0 \leq \sum_i C_i$. With the above information, the wireless operator can then derive the minimum required resource R_0 by simply solving the following linear programming problem.

$$\begin{aligned}
&MinR_0\\
&s.t. \sum_{imt} X_{imt}K_{imt} \geq C_0, \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, t,\\
&\quad \sum_{mt} X_{imt}K_{imt} \leq C_i, \forall i \in \mathcal{I},\\
&\quad \sum_{i} X_{imt} \leq R_0, \forall m \in \mathcal{M}, t,\\
&\quad X_{imt} \geq 0, \forall i \in \mathcal{I}, m \in \mathcal{M}, t.
\end{aligned}$$

Assuming the statistics of user mobility and demand do not change too much, the wireless operator is able to estimate the average total resource requirement with the past information. However, these macro-scale statistics become much less useful when it comes to online packet scheduling. For example, when two clients enter a coffee shop with WiFi at the same time, it is very difficult to predict which of them will leave the coffee shop first. In this case, it is not guaranteed to offload C_0 amount of data with the given R_0 resource. Our derivation of performance bound is indeed based on this difficulty. On the other hand, it is possible to formulate the scheduling problem as a Markov decision process (MDP) problem using past statistics [8]. However, this formulation typically requires each AP to solve a high-dimensional optimization problem for every packet transmission, and incurs preventively high computation complexity. Therefore, in many practical scenarios, simple online policies that do not rely on past statistics are needed.

Now, lets say that the wireless operator estimates that the offline policy needs an amount of R_0 resource to offload the desirable amount C_0 of traffic. How much resource does a simple online policy need to guarantee offloading $0.95C_0$ traffic? Theorem 5 reveals that the answer is $9.5R_0$.

The second important implication concerns the comparison against other poplar policies including round robin, max-weight, and proportional fair policies. Our study reveals that our proposed policy only needs 50% as much resource as the other three to provide the same degree of performance guarantees. In other words, in view of competitive ratio, implementing our policy is as effective as doubling capacity. It is well known that 2X2 MIMO has the potential of doubling transmission rate. Therefore, our policy offers the same performance improvement as implementing 2X2 MIMO.

X. Simulation

In this section, we evaluate the performance of the two algorithms we proposed as well as that of the other three policies discussed in Secition VIII.

We construct a system with 9 APs (3 by 3 grid). Each AP is 1000 meters away from its nearest neighbor , and has a transmission range of 400 meters. There are 200 clients which are divided into two groups: The first group is 100 stationary clients that are uniformly distributed within the coverage area of APs. The second group is 100 mobile users whose locations are chosen uniformly at random at each time t. In each group, the i-th client has $C_i = 100$, and $T_i = 50 + 50i$, if $i \leq 95$, and $C_i = 10,000$, $T_i = 5000(i - 95)$ if $95 < i \leq 100$. Fig. 4 shows a snapshot of the locations of all clients, where the circles represent the coverage area of APs.

The channel gain is determined by both pathloss and Rayleigh fading. The pathloss factor between an AP and a client is computed by $min\{1, 1/(distance/80)^2\}$. The Rayleigh fading factor is computed as $\sqrt{a^2 + b^2}$, where both a and b are Normal random variables with mean 0 and variance 1. Finally, the channel gain is the product of these two factors.

We consider both On-Off channels and general channels. With On-Off channels, we consider the channel to

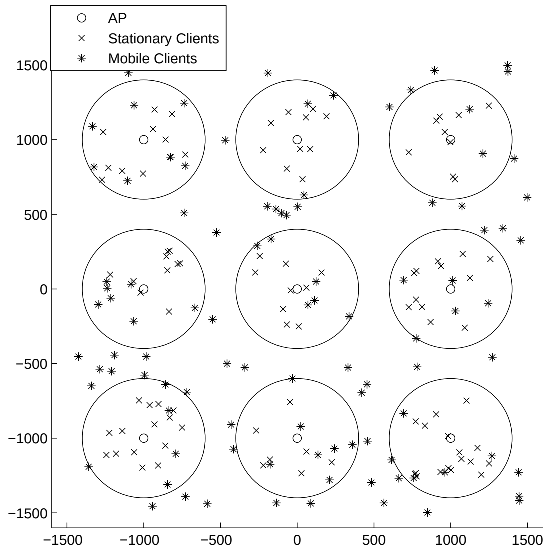


Fig. 4. Location of APs and clients.

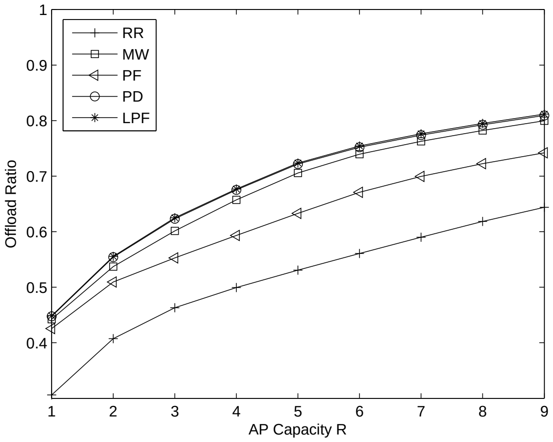


Fig. 5. Performance comparison for On-Off channels.

be ON if the channel gain is larger than $1/25$, in which case we set $K_{imt} = 1$. The threshold of $1/25$ is chosen as the pathloss factor when the distance is 400. With general channels, we set K_{imt} to be the channel gain. For each simulation run, we compute the portion of data that each policy is able to offload through WiFi. All simulation results are the average of 5 simulation runs.

The simulation results for both channels are shown in Figure 5 and Figure 6, respectively. The standard deviations of our algorithms are on the order of 10^{-4}, which shows that the deviation of our algorithm is very small. We notice that PD and LPF have almost identical performance. Recall that LPF is designed to be an approximation to PD with smaller overhead and easier implementation. These simulation results confirm that it is indeed an accurate approximation.

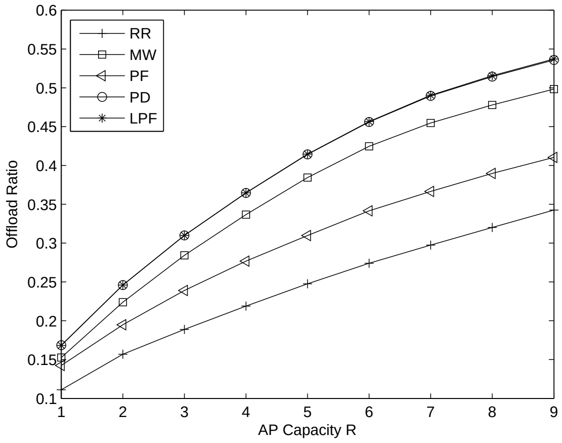


Fig. 6. Performance comparison for general channels.

Further, we note that our policies outperform the other three policies in both scenarios. The theoretical analysis in Section VI only proves that the worst-case performance of our policies is better than that of the other policies. These simulation results further suggest that our policies are still more preferable on average.

With On-Off channels, we notice that the data offload with our policies at R are no less than that RR offload at $2R$. For example, when $R = 2$, our policies are able to offload more data than RR when $R = 4$. The difference between our policies and the RR policy becomes even larger when general channels are considered. 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. For both On-Off channels and general channels, our policies outperform Max-Weight policy. Proportional fair policy does not have a good performance because it tries to even out each client's portion of received data. When clients have different data request and mobility pattern, the policy does not optimize the whole system's offload data amount.

XI. 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. First, we propose PD policy and prove that it is approximately $(R, \frac{1}{2R})$-competitive and achieves the optimal trade-off between capacity and amount of offloaded data. Second, we propose an alternative LPF policy that is easier to implement and has almost identical performance as PD. We also provide that the tight bound of online policies is $(R, \frac{1}{2R})$. Our policies are compared with three commonly-used policies, including Round Robin, Max-Weight, and Proportional Fair policy, and and we prove that our policies only need half as much capacity to provide the same degree of performance guarantees under any mobility patterns. We simulate our proposed policies as well as the three commonly used policies to compare their performance in a randomly generated system. The results show that the proposed two policies have higher offloading ratio than the others.

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*, ser. Co-NEXT '10. New York, NY, USA: ACM, 2010, pp. 26:1–26:12. [Online]. Available: http://doi.acm.org/10.1145/1921168.1921203

[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*. ACM, 2009, pp. 280–293.

[6] F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? Analysis and optimization of delayed mobile data offloading," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2364–2372.

[7] S. Cai, L. Duan, J. Wang, S. Zhou, and R. Zhang, "Incentive mechanism design for delayed WiFi offloading," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3388–3393.

[8] 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*. IEEE, 2013, pp. 564–571.

[9] R. Gass and C. Diot, "An experimental performance comparison of 3G and Wi-Fi," in *Passive and Active Measurement*. Springer, 2010, pp. 71–80.

[10] 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*. ACM, 2010, pp. 209–222.

[11] 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*. IEEE, 2013, pp. 1–6.

[12] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li, "Cellular traffic offloading through WiFi networks," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, Oct 2011, pp. 192–201.

[13] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming the mobile data deluge with drop zones," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1010–1023, Aug 2012.

[14] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "Tube: Time-dependent pricing for mobile data," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '12. New York, NY, USA: ACM, 2012, pp. 247–258. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342402

[15] S. Sen, C. Joe-Wong, S. Ha, J. Bawa, and M. Chiang, "When the price is right: Enabling time-dependent pricing of broadband data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 2477–2486. [Online]. Available: http://doi.acm.org/10.1145/2470654.2481343

[16] J. Lee, Y. Yi, S. Chong, and Y. Jin, "Economics of WiFi offloading: Trading delay for cellular capacity," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1540–1554, March 2014.

[17] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "Human mobility patterns and their impact on routing in Human-Driven mobile networks," Nov. 2007. [Online]. Available: http://conferences.sigcomm.org/hotnets/2007/papers/hotnets6-final108.pdf

[18] 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.

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

[20] 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*. ACM, 2011, pp. 43–48.

[21] 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*. IEEE, 2011, pp. 1–10.

[22] 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*. IEEE, 2011, pp. 353–362.

[23] 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*. IEEE, 2012, pp. 7–16.

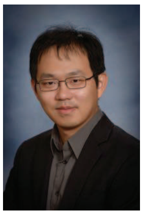
[24] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets WiFi in wireless small cell networks," *Communications Magazine, IEEE*, vol. 51, no. 6, 2013.

[25] 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.

[26] 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*. ACM, 2012, pp. 69–72.

[27] N. Buchbinder and J. (Seffi) Naor, "The design of competitive online algorithms via a primal: Dual approach," *Found. Trends Theor. Comput. Sci.*, vol. 3, no. 2&#8211;3, pp. 93–263, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1561/0400000024

Han Deng received her M.S. degree in Electrical and Computer Engineering from Oakland University, MI, USA in 2012 and her B.S. in Information Engineering in Beijing Institute of Technology, Beijing, China in 2009. She is now a PhD student at Texas A&M University, USA. Her research interests are in wireless and wired networks and optimization.



I-Hong Hou (S10-M12) received the B.S. in Electrical Engineering from National Taiwan University in 2004, and his M.S. and Ph.D. in Computer Science from University of Illinois, Urbana-Champaign in 2008 and 2011, respectively.

In 2012, he joined the department of Electrical and Computer Engineering at the Texas A&M University, where he is currently an assistant professor. His research interests include wireless networks, wireless sensor networks, real-time systems, distributed systems, and vehicular ad hoc networks.

Dr. Hou received the C.W. Gear Outstanding Graduate Student Award from the University of Illinois at Urbana-Champaign, and the Silver Prize in the Asian Pacific Mathematics Olympiad.