Cooperative Internet Access using Helper Nodes and Opportunistic Scheduling

Pouya Ostovari, *Member, IEEE,* Jie Wu, *Fellow, IEEE* and Abdallah Khreishah, *Member, IEEE,*

***Abstract*—Having ubiquitous access to the Internet is becoming a necessity of life. Furthermore, we are witnessing a rapid increase in the amount of data requested by mobile users. Cooperative Internet access is a promising approach to address these demands, which gives the mobile devices the opportunity to receive help from other mobile devices in order to access the Internet. The helpers can download the data requested by the other users, called clients, through their cellular connections. Then, they transmit the downloaded data to the clients using WiFi or Bluetooth connections. In this paper, we consider the problem of how to share the resources of helpers among a set of clients that request their assistance. Opportunistic scheduling is an effective method that uses the dynamic channel conditions to elevate the systems' overall utilities. We propose an opportunistic scheduling algorithm in order to efficiently use the helper nodes and share them among the clients fairly. We propose a rate control and scheduling method in the case of using only WiFi connections. We also propose a solution for the case of using WiFi and Bluetooth at the same time. Through simulation results, we show the effectiveness of our cooperative downloading methods.**

***Index Terms*—Cooperative download, opportunistic scheduling, device-to-device communication, cellular network.**

I. Introduction

With the rapid development in the hardware technology of mobile devices, such as smartphones and tablets, these devices can provide their users with a convenient way to access the Internet. Smartphones and tablets use is increasing rapidly, and ubiquitous Internet access is becoming a necessity. Through cellular connections the users can browse the Internet, download data, or stream videos from anywhere.

We see a rapid growth in the cellular data traffic, and the download rate offered by the cellular networks might not be sufficient for users. Moreover, the user's cellular channel quality might not be sufficient to meet the user's demand, and the data rate of the cellular network can dramatically change over time. Consequently, the users might not get the quality of service that they expect [2]. This low data rate can have a critical effect on applications like video conferencing and video streaming.

An effective approach for addressing the increasing traffic demand of the mobile devices is cooperative download, which can provide ubiquitous Internet access [3]–[5]. Using cooperation among the users, we can use the idle resources of the users to provide Internet access to other users or improve their data rate. As shown in Figure 1(a), the helpers use their cellular connections, e.g. 4G/LTE, to download the request of the clients, and transmit the downloaded data to the clients using WiFi/Bluetooth connections. It is typical for the smartphones to have WiFi and Bluetooth technology. Additionally, the smartphones, such as the Samsung Galaxy s5 use 4G and WiFi connections simultaneously to increase the download rate.

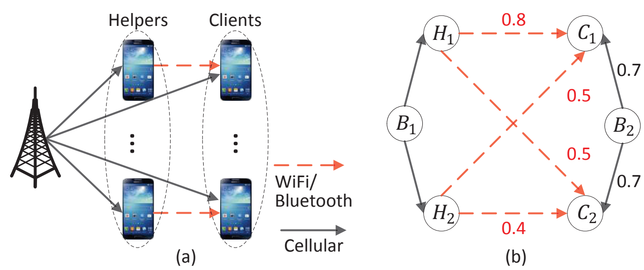

Fig. 1. (a) The system architecture. (b) Motivation example.

Consider the example in Figure 1(b). The base station, helpers, and clients are shown as B, H, and C, respectively. The helpers and the clients are connected to base station B_1 and B_2, respectively. This is similar to subscribing to different cellular carries, such as ATT and Verizon. The delivery rate of the links are shown beside the links. We assume that base station B_1 has a better coverage in the area, and the links between B_1 and the helpers are reliable. The bandwidth of the cellular links and the WiFi links equal 10Mb/s and 15Mb/s, respectively. Also, we define the utility as the total download rate of the clients.

In the case of not using the helpers, the download rate of each client and the total utility will be $10 \times 0.7 = 7$ and 14, respectively. Now consider the case of using the helpers, assume that we assign helpers 1 and 2 to clients 1 and 2, respectively. The receiving rate of client 1 from helper 1 can be up to $15 \times 0.8 = 12$. However, the receiving rate of helper 1 from B_1 is 10. As a result, the receiving rate of client 1 from helper 1 will be limited to 10. Also, the receiving rate of client 2 from helper 2 will be $15 \times 0.4 = 6$. In this case, the total utility of clients 1 and 2 will be $7 + 10 = 17$ and $7 + 6 = 13$, respectively. As a result, the total utility will be 30.

Now assume that we assign helpers 1 and 2 to clients 2 and 1, respectively. The receiving rate of each client from its helper

Manuscript received August 9, 2015; revised march 14 and Jun 7, 2016; accepted August 24, 2016. This work is supported in part by by NSF grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, and ECCS 1231461.

P. Ostovari and Jie Wu are with the Department of CIS, Temple University, Philadelphia, PA 19122 USA (e-mail: ostovari@temple.edu; jiewu@temple.edu).

A. Khreishah is with the Department of ECE, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: abdallah@njit.edu).

Part of this work appeared was presented in Resilience Week (RWS) conference, 2015 [1].

will be $15 \times 0.5 = 7.5$. In this case, the utility of clients and the total utility will be 14.5 and 29. Consequently, assigning helpers 1 and 2 to clients 1 and 2, respectively, maximizes the total utility. However, it is clear that this is not a fair resource sharing, since client 2 receives less data than client 1. In a fair solution, the helper nodes that serve the clients should be changed depending on the cumulative downloading rate of the clients in the previous time slots.

In order to have a fair resource sharing, we should use a fair matching algorithm to assign the helpers to the clients at each time slot. Moreover, it is typical to have a utility function that is not a linear function of the receiving data rate. The reason is that as the receiving data rate of a client increases, the increasing rate of its satisfaction decreases. As a result, we need an optimization algorithm to find the optimal transmission rate of the helpers. Another challenge is the time-varying channel conditions due to fading, shadowing, and interference among the links. Opportunistic scheduling [6] is an effective approach used to deal with the dynamics of channel conditions and achieve higher network performance. The idea behind opportunistic scheduling is to consider the current channel condition of the users in the scheduling. At each time slot, opportunistic scheduling transmits to the user that maximizes the system performance such that it does not violate the fairness constraints.

In this work, we study the problem of providing cooperative Internet access to a set of client users through a group of helpers. We consider the unreliability of the links and time-varying channel condition of the links in our model. In order to motivate the helpers to assist the clients, we use a credit-based incentive method. The credits that the clients need to pay the helpers depend on the effort of the helpers. In other words, instead of using the amount of data received by the clients, we consider the number of transmissions performed by the helpers to calculate the credits that they should receive. In order to use the wireless resources more efficiently, we extend opportunistic scheduling to the case of multiple relays (helpers), and use it in our cooperative Internet access. We formulate the problem as an optimization problem, and solve it in a distributed way.

The main contributions of our work are listed as follows:

- We extend opportunistic scheduling to the case of multiple relays, and use it in our proposed method.
- We propose a distributed algorithm to jointly solve the scheduling and rate optimization problem.
- Through empirical study, we show the effectiveness of our joint scheduling and rate optimization method.

The remainder of the paper is organized as follows. Related work is presented in Section II. Section III introduces the setting and problem statement. In Section IV, we propose our methods in the case of using only WiFi channels and in the case of using WiFi and Bluetooth simultaneously. We present our simulation results in Section V. Section VI concludes the paper.

II. Related Work

The authors in [7], [8] introduce a system for sharing the on-demand and live video streaming among a set of users in the same vicinity. It is assumed that the users are friends and they agree to share their content with each other, which might not be practical in the cases that the users are not friends. The helper nodes might not agree to help the client nodes without receiving a payment. There are many previous works on cooperative downloading that are proposed to share the same content to the users [9], [10]. However, in our work the requested data by the users can be different, and there is no restriction on the type of applications.

The authors in [2] propose a system for cooperative downloading. In their system, a client sends requests to its neighbors to form a network. The users that agree to help the client send a confirmation message to them. After forming the network, the download process is started and each helper receives a payment depending on the amount of help provided to the client. The authors propose an energy efficient method for scanning the neighbors. In contrast with [2], we consider several client nodes that are interested in using the assistance of a set of helper nodes, and focus on optimizing the total utility of the client nodes subject to fair resource sharing.

A framework for cooperative downloading is introduced in [3], in which three roles are defined for the users: client, helper, and gateway. The duty of the users can be a combination of these roles. The authors propose an incentive method in order to motivate the users to participate in the cooperation, and find the optimal payments and the transmission rates of the users using a distributed optimization. In [3], there is no limitation on the number of users that can transmit data to a common user at the same time. Also, the unreliability of the links is not considered. In contrast with [3], we limit the number of connections for each user, and use opportunistic scheduling to increase the utility of the system.

III. System Model and Problem Statement

A. System Model

We consider a set of mobile helpers $H = \{1, .., m\}$, e.g. smartphones and tablets, that are willing to cooperate in providing internet access to a set of mobile clients $I = \{1, .., n\}$. The helpers and clients can communicate with each other by forming a mesh network $G = (H \cup I, E)$, where E represents the set of links connecting them through WiFi or Bluetooth. The helper nodes get access to the Internet through a base station, e.g 4G/LTE connection and provide the clients with the Internet using WiFi or Bluetooth connections. We consider a time-slotted system, and during each time slot the channel conditions are fixed. We assume that the total number of WiFi channels are equal to m, i.e. each helper works on a different channel. Therefore, each helper node can serve only one client at a time. Moreover, each client node cannot receive help from more than one helper at each time slot. However, the helpers can receive data from cellular connection and forward it to the client nodes simultaneously. Additionally, the clients can concurrently receive data using their cellular and WiFi connections.

The clients and helpers consume energy while downloading or relaying data. The energy consumption of client i to receive one byte through cellular and WiFi connections are represented as e_i^c and e_i^w. Typically, the energy consumption of cellular connections is more than that of the WiFi connections. We represent the bandwidth of the link between helper j and client i at time t as b_{ji}^t. Also, b_i^t and b_j^t denote the bandwidth of the link from the base station to client i and helper j, respectively. Similar to any wireless links, the links in our model are unreliable. The delivery rate of the links change over time, which reflects the unreliability of the links. Delivery rate means the probability that a transmitted packet will be received by its destination. We denote the delivery rate of the link ϵ_{ji} at time t as p_{ji}^t. Also, the delivery rate of the link from the base station to client i is represented as p_i^t. We represent the transmission rate from the base station to the client i at time t as x_i^t. Moreover, x_{ji}^t is defined as the transmission rate of helper j to client i at time slot t.

In our model the links are not reliable. Therefore, some of the packets might be lost and need to be retransmitted. In order to eliminate the need for feedback messages, we use *Random linear network coding* (RLNC) [11], [12]. In RLNC, a coded packet is a linear combination of the packets. In order to code the packets, the coefficients of the linear combinations are selected randomly, and the coded packets have a form of $\sum_{i=1}^{k} a_i \times P_i$. Here, the symbols a and P are the random coefficients and the packets. If we use RLNC to code k native packets together, with a large probability any k coded packets are sufficient to decode the coded packets and retrive the original packets. In other words, with a probability close to one, any k random linear coded packets are linearly independent. In order to decode the coded packets, Gaussian elimination can be used to solve a system of linear equations.

The packets that should be transmitted to each client are partitioned into segments of equal sizes, and RLNC is performed among the packets of the same segment. The helper nodes receive coded packets from the base station, then recode the packets before transmitting them. The clients are able to decode the coded packets once they receive a sufficient number of coded packets, i.e. k linearly independent coded packets. The use of RLNC enables a flow-based model of the content, which simplifies our proposed schemes. Without RLNC, we need to decide which packets should be transmitted by the base station and which of them by the helper nodes. However, with RLNC, we just need to find the rates at which the coded packets should be transmitted. It should be noted that other types of coding schemes, such as fountain codes [13], [14], can also be applied on top of our solutions. However, for ease of description, we use RLNC in our proposed methods.

B. Problem Statement

In order to motivate the helpers to participate in the cooperation, we need to have an incentive mechanism. For this purpose, the clients pay the helpers based on the data rate that the helper nodes transfer to the clients. It should be noted that the rates at which the clients receive data might be less than the transfer rates of the helpers, which is due to the unreliable links. On the other hand, the clients consume energy to receive data from the base station and the helpers. Therefore, we define the utility of each client as a function of the data rate that it receives minus its energy consumption and the credits that the client needs to pay the helpers.

TABLE I
THE SET OF SYMBOLS USED IN THIS PAPER.

Notation	Definition
x_{ji}^t	Transmission rate from helper j to user i at time t
x_i^t	Transmission rate from base station to user i at time t
y_i^t	Total download rate of user i at time t
p_{ji}^t	Delivery rate of the link from helper j to client i at time t
p_i^t/p_j^t	Delivery rate from base station to client i/helper j at time t
z	Credit payment to a helper for transmitting one byte
b_{ji}^t	Bandwidth of link between helper j and client i at time t
b_i^t	Bandwidth of the link from base station to client i at time t
b_j^t	Bandwidth of the link from base station to helper j at time t
e_i^w/e_i^c	Energy consumption of user i for receiving one byte from the helpers/base station

The utility of client i at time t is defined as $U(i,t) = \theta_1 f(x_{ji}^t p_{ji}^t + x_i^t p_i^t) - \theta_2 e(i,t) - \theta_3 x_{ji}^t z$. Here, θ are normalizing parameters. For simplicity, we assume that θ parameters equal 1, and we do not show θ parameters in the rest of the paper. We represent the credits that client i needs to pay a helper in order to transmit one byte as z. The transmission rate of helper j to client i is equal to x_{ji}^t, and the data rate that is delivered to client i is equal to $x_{ji}^t p_{ji}^t$. Moreover, the rate of the data that is delivered to client i from the cellular network equals $x_i^t p_i^t$. We represent the energy consumption of i as $e(i,t)$, which equals $x_{ji}^t p_{ji}^t e_i^w + x_i^t p_i^t e_i^c$. Function $f(.)$ is a strict concave, non-decreasing, and continuously differentiable function of the receiving data rate. The reason for the concavity assumption relies on the fact that as the receiving data rate of a client increases, the increasing rate of its satisfaction decreases. In economics, this fact is known as the "law of diminishing returns" [15]. Since the energy consumption and the credit that needs to be paid to the helper nodes increases as the receiving data rate increases, $U(i,t)$ might reduce. Assuming that the energy consumption and credit payments are linear functions of the transition rate, $U(i,t)$ becomes a strictly concave function. Therefore, it has a unique maximum.

Our objective is to maximize the aggregated utility of the client nodes. However, this maximization problem might result in an unfair resource sharing among the users. The client nodes with bad channel conditions might not be able to use the help of the helper nodes in the case that the number of helpers is less than that of the client nodes. As a result, in our optimization problem, we try to provide fair resource sharing, which is discussed in the next section.

IV. COOPERATION SCHEMES

We first solve the optimization problem without considering the energy consumptions and the credits that need to be paid by the clients. As a result, the utility of the clients is a function of their received data rate, and the objective is to maximize the aggregated utility of the clients subject to the fairness constraints. We then extend our solution when considering the energy consumptions and the credit payments.

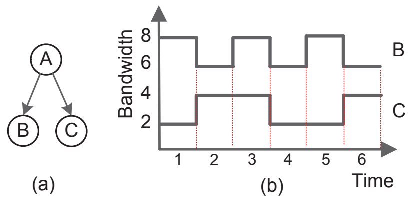

Fig. 2. Opportunistic scheduling; (a) Topology; (b) Time varying channel conditions.

A. Transmission with WiFi

Fairness is a main part of scheduling problems in wireless networks. Assume that our utility function is the total amount of received data by the users. If we do not consider fairness, we can trivially optimize the system performance by finding a bipartite matching of the helpers to the clients so that the total utilities are maximized. However, this scheduling might not be fair because the clients with good channel conditions might keep the channels forever. Thus, the clients with poor channel conditions might not be able to use the assistance offered by the helpers. It should be noted that if the utility function is $U(i,t) = log(y_i^t)$, then maximizing the system utility is correspondent to providing proportional fairness among clients.

To give an idea of opportunistic scheduling consider the example in Figure 2. The channel condition of user nodes B and C are shown in Figure 2(b). The channel condition of node B is always better than that of the node C. As a result, if we want to maximize the amount of data that the nodes receive, node A should only transfer data to node B. Clearly, this is not a fair scheduling. Now assume that we assign half of each time slot to users B. Nodes B experiences 3 slots with bandwidth 8 and 3 slots with bandwidth 6; thus, its amount of received data will be $(8/2 + 6/2) \cdot 3 = 21$. Furthermore, the total received data of node C will be $(4/2 + 2/2) \cdot 3 = 9$. However, if A assigns time slots 1, 3, and 5 to node B, and slots 2, 4, and 6 to node C, the total received data of B and C will be 24 and 10, respectively. In this case, not only the total received data of the system increases, but the total received data by each user is also enhanced.

In order to provide fairness, we can extend the idea in [6] to the case of multiple helpers. The idea is that instead of maximizing the total utility, we maximize $\sum_{i=1}^{n} \alpha_i^t U(i,t)$. Here, $U(i,t)$ is the utility of client i at time t, and α_i^t are parameters that control the fairness. Similar to [6], the idea behind parameter α is to give a chance to the client whose received utility in the previous time slots is low to use the assistance of the helpers. The α variable for the clients with large received utilities is lower than the other nodes. Thus, for each time slot, the relatively best clients are selected and assigned to the helpers. The clients with large α are the unfortunate nodes with worse channel conditions than the other clients. The calculation of α will be discussed later.

The problem of maximizing the total system utility subject

Algorithm 1 Scheduling Algorithm

$\alpha_i^1 = 1, \forall i \in I$.
At each time slot t perform the following steps:
for each client i **do**
  for each helper j **do**
    $G(i,j,t) = f(min(b_{ji}^t p_{ji}^t, b_j^t p_j^t) + b_i^t p_i^t) - f(b_i^t p_i^t)$
    if $\sum_{t'=1}^{t-1} G(i,t') \neq 0$ **then**
      $\alpha_i^t = 1/\sum_{t'=1}^{t-1} G(i,t'), \forall i \in I$
    else
      $\alpha_i^t = 1$
    end if
    Assign $\alpha_i^t G(i,j,t)$ to link ϵ_{ji}
  end for
end for
Find the maximum weighted bipartite matching using Hungarian algorithm

to fairness can be formulated as follows:

$$\max \sum_{i \in I} \sum_{t=1}^{T} \alpha_i^t U(i,t) \tag{1}$$

$$s.t.\ x_{ji}^t \leq p_j^t b_j^t / p_{ji}^t, \quad \forall i,j : j \in H, i \in I \tag{2}$$

$$x_{ji}^t \leq b_{ji}^t, \quad \forall i,j : j \in H, i \in I \tag{3}$$

$$x_i^t \leq b_i^t, \quad \forall i \in I \tag{4}$$

$$y_i^t \leq x_i^t p_i^t + \sum_{j \in H} x_{ji}^t p_{ji}^t, \forall i \in I \tag{5}$$

$$[x_{ji}^t] \in Co(R) \tag{6}$$

where, $U(i,t)$ is the utility of client i at time slot t. Also, x_{ji}^t and p_{ji}^t are the transmission rates of helper node j to client i and the delivery rate of the link ϵ_{ji} at time t, respectively. We represent the total download rate of client i at time t as y_i^t. The objective function (1) is to maximize the total utility of the client nodes. Here, we consider $U(i,t) = f(y_i^t)$, where $f(y_i^t)$ is a monotonic increasing concave function.

The receiving rate of helper j equals $p_j^t b_j^t$. Therefore, client i cannot receive data from helper j at a rate greater than $p_j^t b_j^t$. On the other hand, the receiving rate of client i equals to $x_{ji}^t p_{ji}^t$; therefore, we have $x_{ji}^t p_{ji}^t \leq p_j^t b_j^t$, which is stated as the set of Constraints (2). The set of Constraints (3) and (4) are bandwidth constraints. Constraints (5) calculate the total receiving rate of the clients. Constraint (6) implies that the transmission rate should be feasible. Here, R is the set of possible scheduling, and $Co(R)$ is the convex hull of R. In our model, a helper cannot transmit to more than one client at the same time slot. Moreover, a client cannot receive data from multiple helpers concurrently. Therefore, if for helper j and client i we have $x_{ji}^t > 0$, then $x_{j'i}^t = 0, \forall j' \neq j$ and $x_{ji'}^t = 0,\ \forall i' \neq i$.

Our scheduling algorithm for time slot t works as follows. First, for each client i we calculate its maximum utility when it only uses the cellular connection. We then calculate the increase in client i's maximum utility in the case of using helper j in addition to the cellular connection. Since we do not consider the energy consumption and the credit payments, the

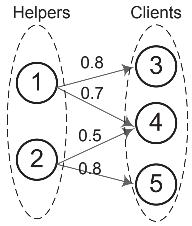

Fig. 3. Helper node assignment.

utility of each client node is an increasing concave function. Therefore, the optimal rate assignment for client i is to use the full bandwidth of both of the links from the base station and helper j. The increase in the utility of client i in the case of using helper j is equal to:

$$G(i,j,t) = f(min(b^t_{ji}p^t_{ji}, b^t_j p^t_j) + b^t_i p^t_i) - f(b^t_i p^t_i)$$

The reason for taking the minimum of the two values is that the receiving rate of client i cannot exceed the bandwidth of the link ϵ_{ji} and the receiving rate of helper j. Then, we multiply each utility enhancement $G(i,j,t)$ by α^t_i, and assign the result to the link ϵ_{ji}. We run the Hungarian algorithm [16] to find the maximum weighted bipartite matching of the helpers to the clients. In order to have a fair resource sharing, in this paper we calculate α as $\alpha^t_i = 1/\sum_{t'=1}^{t-1} G(i,t)$, and we set $\alpha^1_i = 1, \forall i \in I$. Here, $G(i,t)$ is the received utility by client i at time slot t. The details are shown in Algorithm 1.

Consider the example in Figure 3. For simplicity we assume that the bandwidth of the links from the base station to the helpers are not bottleneck, i.e. their bandwidth and reliability are much higher than the links between the helpers and the clients. Furthermore, we do not consider the direct links from the base station to the clients. The bandwidth of the links between the helpers and clients are equal to 5. The delivery rate of the links are shown in the figure. Assuming function $f(.) = \log(x+1)$, the utility of the client 4 in the case of downloading from helper 1 and 2 becomes $\log(5\times 0.7+1) = 1.5$ and $\log(5\times 0.5+1) = 1.25$, respectively. Also, the utility of the clients 3 and 5 when they download through helpers 1 and 2 are equal to $\log(5\times 0.8+1) = 1.61$. Thus, the bipartite matching that maximizes the total utility is assigning helper 1 to client 3, and helper 2 to client 5.

Assuming that the reliability and bandwidth of the links are fixed, in time slot 2 the utility of clients 3 and 5 are divided by the utility that they received at time slot 1. As a result, the weight of the links from helpers 1 and 2 to clients 3 and 5 becomes 1. The weight of the links from helpers 1 and 2 to client 4 are still 1.5 and 1.25, respectively. Therefore, the optimal weighted bipartite matching is to schedule clients 4 and 5 to receive from helpers 1 and 2, respectively.

Our proposed optimization method can be implemented in a distributed way as follows. Calculating the utility of the client for each helper assignment can be performed in a distributed way by each client (or the helpers). Moreover, the α variables are calculated by each client node separately. The products of the $G(i,j,t)$ and α^t_i variables are used as the weight of the links ϵ_{ji}. Then, we can apply a distributed version of the maximum weighted bipartite matching algorithm, e.g. [17], to find the optimal scheduling. We refer to this method as the joint scheduling and rate control (JSRC) method.

B. Transmission with WiFi Considering the Costs

1) Formulation: In the previous subsection, we did not consider the energy consumption and the credit payments in the scheduling. As a result, the utility of the client nodes was an increasing function of their received data rate. However, when we take the energy consumption and the credit payments into account, a client might prefer not to use the full capacity of the link that is assigned to it. The reason is that, as the receiving rate of a client increases, its energy consumption and the credits that need to be paid to the helper node increase as well. These increases in the costs are linear to the download rate. However, the function $f(.)$ is not linear, and $f(.)$ is strictly concave. Consequently, the whole utility might decrease depending on the receiving rate.

In the case of considering energy consumption and the credit payment to the helper nodes, we can formulate the problem as the following optimization problem:

$$\max \sum_{i\in I}\sum_{t=1}^{T} \alpha^t_i U(i,t) \tag{7}$$

$$s.t\ U(i,t) = f(y^t_i) - x^t_i p^t_i e^c_i - \sum_{j\in H}[x^t_{ji}p^t_{ji}e^w_i + z x^t_{ji}] \tag{8}$$

$$x^t_{ji} \le p^t_j b^t_j / p^t_{ji}, \quad \forall i,j : j\in H, i\in I \tag{9}$$

$$x^t_{ji} \le b^t_{ji}, \quad \forall i,j : j\in H, i\in I \tag{10}$$

$$x^t_i \le b^t_i, \quad \forall i \in I \tag{11}$$

$$y^t_i \le x^t_i p^t_i + \sum_{j\in H} x^t_{ji}p^t_{ji}, \forall i\in I \tag{12}$$

$$[x^t_{ji}] \in Co(R) \tag{13}$$

The objective function (7) is maximizing the total utility of the clients. Similar to the previous subsection, α variable is used to insure the fairness. The utility of a client node at time t is calculated from (8). The second and third terms in (8) are the energy consumption and the credit payments. We use Constraint (9) to limit the receiving rate of client i from helper j to the receiving rate of the helper. The set of Constraints (10) and (11) are bandwidth constraints. The set of Constraints (13) are constrains of the feasibility of the scheduling. The set of feasible scheduling are denoted by R, and $Co(R)$ is the convex hull of R. A helper cannot transmit to more than one client at each time slot, and each client cannot receive data from more than one helper simultaneously.

This optimization contains two sub-problems: (1) scheduling the links, and (2) finding the optimal data rates. In order to find the optimal solution, we decompose the optimization into scheduling and rate optimization. Assuming that the number of helper and client nodes is equal to n, there are $n!$ possible matchings. Therefore, the time complexity of checking the total utility of all of the possible matchings is exponential.

In this problem, there is no need for checking all of the possible matchings. In our model, each client node can download from one of the m helpers, and each helper cannot serve more than one client. Therefore, the utility of each client node only depends on the matching of the helpers to the clients, and is independent of the download rate and the utility of the other clients. Consequently, our polynomial time algorithm calculates the difference in the optimal utilities of client i in the cases of using the assistance of helper j and not using any helpers, denoted by $G(i,j,t)$. We then, assign $\alpha_i^t G(i,j,t)$ as the weight of link ϵ_{ji}, and run the Hungarian algorithm [16] to find the maximum weighted matching of the helper nodes to the clients. The Hungarian algorithm can find the maximum weighted matching of a bipartite graph in a polynomial time. It should be noted that other maximum weighted bipartite matching algorithms can be used instead of the Hungarian algorithm.

2) Optimization: In order to find the optimal rate allocations for each helper node assignment, we need to perform an optimization algorithm. Consider client node i, which is scheduled to receive from helper j at the current time slot t. The optimal transmission rates from the base station and helper j to client i can be found by solving the following convex optimization:

$$\begin{aligned} &\max\ U(i,t) = f(y_i^t) - x_i^t p_i^t e_i^c - x_{ji}^t p_{ji}^t e_i^w - z x_{ji}^t \\ &\quad s.t\ x_{ji}^t \le p_j^t b_j^t / p_{ji}^t;\ x_{ji}^t \le b_{ji}^t;\ x_i^t \le b_i^t;\ y_i^t \le x_i^t p_i^t + x_{ji}^t p_{ji}^t \end{aligned}$$

We can find the optimal rate allocation by solving the Lagrangian dual of the problem using the gradient method. In this way, we gradually update the transmission rates, based on the Lagrange variables. Since the Slater condition holds in this problem (see reference [18]), there is no duality gap between the primal and the dual problems. Consequently, the dual approach can be used to solve the problem. Let λ_1^j, λ_2^{ji}, λ_3^i, and λ_4^i be the Lagrange variables for the constraints. For simplicity, we do not show the t superscripts. The Lagrange function becomes:

$$\begin{aligned} L(x_i, x_{ji}, y_i, \vec{\lambda}) = f(y_i) &- x_i p_i e_i^c - x_{ji} p_{ji} e_i^w - z x_{ji} \\ &- \lambda_1^j (x_{ji} - p_j b_j / p_{ji}) - \lambda_2^{ji} (x_{ji} - b_{ji}) \\ &- \lambda_3^i (x_i - b_i) - \lambda_4^i \big[y_i - x_i p_i - x_{ji} p_{ji} \big] \end{aligned}$$

By rearranging the terms and removing the constants, we have:

$$L(x_i, x_{ji}, y_i, \vec{\lambda}) = f(y_i) - \lambda_4^i y_i \tag{14}$$

$$+ x_{ji}(\lambda_4^i p_{ji} - \lambda_1^j - \lambda_2^{ji} - z - p_{ji} e_i^w) \tag{15}$$

$$+ x_i(\lambda_4^i p_i - \lambda_3^i - p_i e_i^c) \tag{16}$$

The objective function of the dual problem is $D(\vec{\lambda}) = \max_{x_i, x_{ji}, y_i} L(x_i, x_{ji}, y_i, \vec{\lambda})$. The dual problem is $\min_\lambda D(\vec{\lambda})$. We can solve the dual optimization problem using the gradient method. The updates of the Lagrange variables are as follows:

$$\begin{aligned} &\lambda_1^j(\tau+1) = \big[\lambda_1^j(\tau) + \beta(x_{ji}(\tau) - p_j b_j / p_{ji})\big]^+ \\ &\lambda_2^{ji}(\tau+1) = \big[\lambda_2^{ji}(\tau) + \beta(x_{ji}(\tau) - b_{ji})\big]^+ \\ &\lambda_3^i(\tau+1) = \big[\lambda_3^i(\tau) + \beta(x_i(\tau) - b_i)\big]^+ \\ &\lambda_4^i(\tau+1) = \big[\lambda_4^i(\tau) + \beta\big(y_i(\tau) - x_i(\tau) p_i - x_{ji}(\tau) p_{ji}\big)\big]^+ \end{aligned}$$

The projection on $[0, +\infty)$ is represented as $[.]^+$. Also, β is the step size. In order to find the y_i that maximizes (14), we set the first derivative of (14) with respect to y_i equal zero. If we consider $f(y_i) = \log(y_i + 1)$, the optimal y_i becomes $y_i = 1/\lambda_4^i - 1$. For iteration τ, if y_i becomes infinity, we set y_i to $p_i b_i + p_{ji} b_{ji}$.

Algorithm 2 Calculation of x_{ji} and x_i (for client node i)

$\gamma_i = \lambda_4^i p_{ji} - \lambda_1^j - \lambda_2^{ji} - z - p_{ji} e_i^w$
if $\gamma_j^i > 0$ **then**
  set $x_{ji} = min(b_{ji}, b_j p_j / p_{ji})$
else
  $x_{ji} = 0$
end if
if $\lambda_4^i p_i - \lambda_3^i - p_i e_i^c > 0$ **then**
  set $x_i = b_i$
else
  $x_{ji} = 0$
end if

Equations (15) and (16) are linear functions of x_{ji} and x_i. Thus, in order to maximize their summation, when the multipliers of x_{ji} and x_i are greater than zero, their value should be set to the maximum possible value, which depends on the bandwidths. On the other hand, in the case of negative multipliers, we should set x_{ji} and x_i to zero. Algorithm 2 illustrates the computation of x_{ji} and x_i. Here, $\gamma_i = \lambda_4^i p_{ji} - \lambda_1^j - \lambda_2^{ji} - z - p_{ji} e_i^w$ is the multiplier of x_{ji} in Equation (15). After finishing the iterations, the final values of x_{ji} and x_i are set to the average calculated values of the iterations.

The optimal rate allocation in the case of not using the assist of helpers can be found by setting the first derivative of $f(x_i) - x_i p_i e_i^c$ respect to x_i equal to zero. Then, $G(i,j,t)$ is equal to the difference of the utilities in the case of using helper j and in the case of not using any helper. We assign $\alpha_i G(i,j,t)$ to link ϵ_{ji}, and select the maximum weighted bipartite matching using the Hungarian algorithm, and pick its respective optimal rates.

Similar to the JSRC method, our optimization in the case of considering the download costs can be implemented in a distributed way. First, each client uses the proposed gradient approach to find the optimal rate when helper j is assigned to it, calculates $G(i,j,t)$, and multiplies $G(i,j,t)$ by α_i. Then, the result is assigned as the weight of link ϵ_{ji}. Finally, the distributed version of the maximum weighted bipartite matching algorithm, e.g. [17], is run to find the scheduling that maximizes the total utility. We refer to this method as the joint scheduling and rate control with payments (JSRCP).

C. *Transmission with WiFi and Bluetooth*

It is typical for mobile devices to have both WiFi and Bluetooth technologies. As a result, the helpers and clients can use both of them to increase the transmission rates. Each helper node can serve two different clients at the same time, one client using its WiFi radio and the next client using Bluetooth. Moreover, each client can receive from two different helper nodes at the same time.

1) Formulation: We represent the energy consumption of client i to receive one byte through WiFi or Bluetooth channel as e_i^w. Also the bandwidth and delivery rate of the links between helper j and client i at time t are denoted by b_{ji}^t and p_{ji}^t, respectively. The optimization problem becomes:

$$\max \sum_{i \in I} \sum_{t=1}^{T} \alpha_i^t U(i,t)$$

$$s.t\ U(i,t) = f(y_i^t) - x_i^t p_i^t e_i^c - \sum_{j \in H} [x_{ji}^t p_{ji}^t e_i^w + z x_{ji}^t]$$

$$\sum_{i \in I} x_{ji}^t p_{ji}^t \leq p_j^t b_j^t, \quad \forall j : j \in H \tag{17}$$

$$x_{ji}^t \leq b_{ji}^t, \quad \forall i, j : j \in H, i \in I \tag{18}$$

$$x_i^t \leq b_i^t, \quad \forall i \in I \tag{19}$$

$$y_i^t \leq x_i^t p_i^t + \sum_{j \in H} x_{ji}^t p_{ji}^t, \forall i \in I \tag{20}$$

$$[x_{ji}^t] \in Co(R) \tag{21}$$

In our new model, a helper node can serve two clients. Therefore, the rate at which a helper receives data from the base station can be shared among two client nodes, which is reflected as the set of Constraints (17). The receiving rate of helper j is equal to $p_j^t b_j^t$, and the receiving rate of client i from helper j is equal to $x_{ji}^t p_{ji}^t$. The summation of receiving rate of the clients that are served by helper j cannot exceed the helper's receiving rate; therefore we have $\sum_{i \in I} x_{ji}^t p_{ji}^t \leq p_j^t b_j^t$. The set of Constraints (21) are the scheduling constraints, and R is the set of possible schedulings. Also, $Co(R)$ is the convex hull of R. In this extended model, each helper cannot transmit to more than two client nodes, and each client cannot receive data from more than two helpers.

In contrast with the previous model, in the case of using WiFi and Bluetooth a helper can serve two clients. As a result, the data rates that the client nodes receive are coupled together. This means that we cannot find the optimal transmission rate to each client separately. Therefore, we propose a two-phase resource sharing algorithm. In the first phase, we find a relatively fair matching of the helper nodes to the clients. Then, in the second phase, we perform an optimization to find the optimal transmission rates.

In the first round of our matching algorithm, we assign the WiFi channels of the helpers to the clients, and in the second round, we match the Bluetooth links of the helpers and clients. Before running the second phase, if the WiFi channel of helper j is assigned to user i, we remove the Bluetooth link between these two nodes. In this way, we can make sure that the same client and helper are not connected using both WiFi and Bluetooth channels. In our matching algorithm, we use $\alpha_i^t p_{ji}^t$ as the weight of the link ϵ_{ji}, and perform the Hungarian algorithm to find the maximum weighted bipartite matching. We refer to this algorithm as joint scheduling and rate control with Bluetooth links (JSRCB).

2) Optimization: For simplicity we do not show the t superscripts in the following equations. Let λ_1^j, λ_2^{ji}, λ_3^i, and λ_4^i be the Lagrange variables for the Constraints (17), (18), (19), and (20), respectively. The Lagrange function becomes:

$$\begin{aligned} L(\vec{x},\vec{y},\vec{\lambda}) &= \sum_{i \in I} \left[f(y_i) - x_i p_i e_i^c - \sum_{j \in H} (x_{ji} p_{ji} e_i^w + z x_{ji}) \right] \\ &- \sum_{j \in H} \lambda_1^j \Big(\sum_{i \in N(j)} x_{ji} p_{ji} - p_j b_j \Big) - \sum_{j \in H} \sum_{i \in N(j)} \lambda_2^{ji} (x_{ji} - b_{ji}) \\ &- \sum_{i \in I} \lambda_3^i (x_i^t - b_i^t) - \sum_{i \in I} \lambda_4^i \Big[y_i - x_i p_i - \sum_{j \in N(i)} x_{ji} p_{ji} \Big] \end{aligned}$$

where $N(i)$ and $N(j)$ are the helpers and clients assigned to client i and helper j in the matching phase. Moreover, $\vec{x}$, $\vec{y}$, and $\vec{\lambda}$ are the vector of x, y, and λ variables. By rearranging the terms and removing the constants, we can separate the variables x_i, x_{ji}, y_i as follows:

$$\sum_{i \in I} [f(y_i) - \lambda_4^i y_i] \tag{22}$$

$$+ \sum_{j \in H} \sum_{i \in N(j)} x_{ji} (\lambda_4^i p_{ji} - \lambda_1^j p_{ji} - \lambda_2^{ji} - p_{ji} e_i^w - z) \tag{23}$$

$$+ \sum_{i \in I} x_i (\lambda_4^i p_i - \lambda_3^i - p_i e_i^c) \tag{24}$$

The objective function of the dual problem is $D(\vec{\lambda}) = \max_{\vec{x},\vec{y}} L(\vec{x},\vec{y},\vec{\lambda})$, and the dual problem is $\min_{\lambda} D(\vec{\lambda})$. We can solve the dual optimization problem using the gradient method. The updates of the Lagrange variables are as follows:

$$\lambda_1^j(\tau+1) = \Big[\lambda_1^j(\tau) + \beta \Big(\sum_{i \in N(j)} x_{ji}(\tau) p_{ji} - p_j b_j \Big) \Big]^+, \forall j \in H$$

$$\lambda_2^{ji}(\tau+1) = \big[\lambda_2^{ji}(\tau) + \beta (x_{ji} - b_{ji}) \big]^+, \forall j \in H, i \in N(j)$$

$$\lambda_3^i(\tau+1) = \big[\lambda_3^i(\tau) + \beta (x_i - b_i) \big]^+, \forall i \in I$$

$$\lambda_4^i(\tau+1) = \Big[\lambda_4^i(\tau) + \beta \Big(y_i(\tau) - x_i p_i - \sum_{j \in N(i)} x_{ji}(\tau) p_{ji} \Big) \Big]^+, \forall i \in I$$

Here, $[.]^+$ is the projection on $[0, +\infty)$. In order to find the optimal $\vec{y}$, we set the first derivative of (22) with respect to $\vec{y}$ equal zero. Assuming $f(y_i) = \log(y_i + 1)$, the optimal $\vec{y}$ becomes $\vec{y} = 1/\lambda_4^i - 1$.

Algorithm 3 illustrates the computation of $\vec{x_{ji}}$. Here, $\gamma_i = \lambda_4^i p_{ji} - \lambda_1^j p_{ji} - \lambda_2^{ji} - p_{ji} e_i^w - z$ is the multiplier of x_{ji} in Equation (22), and rem is the remaining bandwidth b_j of helper node j. Equation (23) is a linear function of $\vec{x}$. Therefore, in order to maximize (23), each helper j should give a larger portion of its downloading bandwidth b_j to the client node i with a greater γ_i. Also, x_{ji} for the clients with negative γ_i value should be equal to zero. In order to find the optimal x_i, each client node i computes $\lambda_4^i p_i - \lambda_3^i - p_i e_i^c$. If the result is greater than zero, x_i is set to b_i. Otherwise, x_i is set to zero. After finishing the iterations, the values of each x_i and x_{ji} is set to its average calculated values over different iterations.

V. Simulations

A. Simulation Setting

In order to evaluate our proposed methods, we develop a simulator in MATLAB environment. In the previous sections, we proposed a method for the case of using only WiFi radio,

Algorithm 3 Calculation of $\vec{x_{ji}}$ (for helper node j)

$rem = b_j p_j$, calculate $\gamma_i,\ \forall i$
for each i in descending order of γ_i **do**
  if $\gamma_j^i > 0$ and $rem > 0$ **then**
    set $x_{ji} = min(b_{ji}, rem/p_{ji}),\ \ rem{=}rem{-}x_{ji}p_{ji}$
  else
    $x_{ji} = 0$
  end if
end for

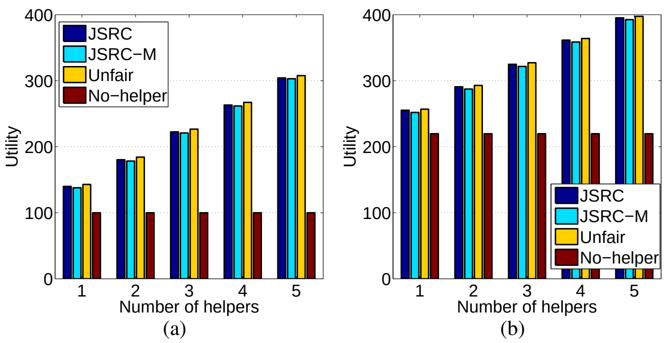

Fig. 4. Total utility of the client nodes, delivery rates $\in [0.5, 1]$, $n = 10$, $T = 50$, $b_j^t \in [2, 4]$, $b_{ji}^t \in [1, 2]$. (a): $b_i^t \in [0.2, 0.4]$; (b): $b_i^t \in [0.5, 1]$.

and extended it to the case of using WiFi and Bluetooth simultaneously. We compare each proposed method with the optimal solutions in the case of not considering the fairness, which is referred to as the Unfair method. In order to remove the fairness constraint from the optimization constraints, we set $\alpha_i^t = 1\ \forall i \in I$. As a result, at each iteration, the matching that maximizes the total utility of the client nodes is selected as the optimal solution. We also modify our JSRC and JSRCP methods by setting α_i^t to the inverse of the amount of increment in the utility of the client i in the previous time slots that is due to using the helpers. We refer to these methods as JSRC-M and JSRCP-M, respectively. Recall that in the JSRC and JSRCP methods, the inverse of the total received utility in the previous time slots in assigned to $alpha_i^t$, and we have $\alpha_i^t = 1/\sum_{t'=1}^{t-1} G(i,t)$. In other words, in the JSRC and JSRCP methods, the total utility of client i is used to calculate α_i^t. However, in the JSRC-M and JSRCP-M methods, the portion of the utility which is due to using the helper nodes is used to calculate α_i^t.

In our simulations, we consider $f(y_i) = \log(y_i + 1)$. The reason to add one to y_i is to set the utility function $U(i,t)$ equal to zero in the case that the download rate of user i from the base station and the helper nodes is equal to zero. We run each simulation 200 times and report the average results.

B. Simulation Results

1) WiFi without Credit Payments: We compare the JSRC and JSRC-M methods with the cases of not using helpers and the Unfair methods. In the first experiment, we evaluate the utility of the methods. The simulation setting is shown in the caption of Figure 4. Here, T represents the number of time slots that we run the algorithms. It is clear that the total utility without using the helper nodes should be less than the other methods, which is confirmed in Figure 4(a). Moreover, the Unfair method has the highest utility compared to the other methods. However, the utility of our proposed fair scheduling method JSRC is about only 5% less than that of the Unfair method. The figure illustrates that the utility of the JSRC-M method is about 6% lower than that of the JSRC approach. As we increase the number of helpers, more resources are provided to the client users, which increases the utility of the JSRC, JSRC-M, and the Unfair method.

We increase the bandwidth of the links between the base station and the client nodes to the range of $[0.5, 1]$ and repeat the pervious experiment. The results are shown in Figure 4(b). It is clear that the total utility of all of the methods should increase when the clients can download more data directly from the base station. In Figure 4(b), the difference between the utility of the No-helper method and the other methods is less than those in Figure 4(a). When the cellular connection of the client nodes has sufficient bandwidth, there is no need for the helper nodes.

The previous two experiments show that there is no major difference between the utility of the Unfair, JSRC, and JSRC-M methods. These results illustrate the effectiveness of our opportunistic scheduling in using the resources. In order to check if our opportunistic scheduling mechanisms can provide fairness, we compare the standard deviation of the total utility that the different client nodes receive in the case of using the Unfair, JSRC, and JSRC-M methods. Figure 5 depicts the utility standard deviation of the scheduling methods. The figure shows that the standard deviation of the JSRC and JSRC-M methods are almost equal. However, the standard deviation in the case of using the Unfair method is between 3 to 4.5 times that of the JSRC and JSRC-M methods. The utility standard deviation of the Unfair method increases as we increase the number of helper nodes. This is due to the fact that more helpers will increase the chance that some of the client nodes are connected to the helpers through channels with good conditions. Therefore, these client nodes receive much more utility than the other clients.

Figure 5(b) shows the empirical CDF of the utility standard deviation in the case of using the JSRC method to that of the Unfair method. For each run, we divide the utility standard deviation of the Unfair method by that of the JSRC method, and plot the empirical CDF. The figure shows that in 50% of the cases, the utility standard deviation of the Unfair method is up to 5 times that of the JSRC method. Moreover, in 20% of the cases, the utility standard deviation of the Unfair method is more than 6 times that of the JSRC method.

2) WiFi with Credit Payments: We repeat our first experiment to compare the utility of the methods in the case of considering the costs. Figure 6(a) depicts that the total utility without using the helper nodes is as much as 60% less than the other methods. The Unfair method finds the optimal scheduling without considering the fairness constraint, so it has the highest utility compared to the other methods. Also, the utility of our proposed fair scheduling method JSRCP is about only 5% less than that of the Unfair method. The figure shows

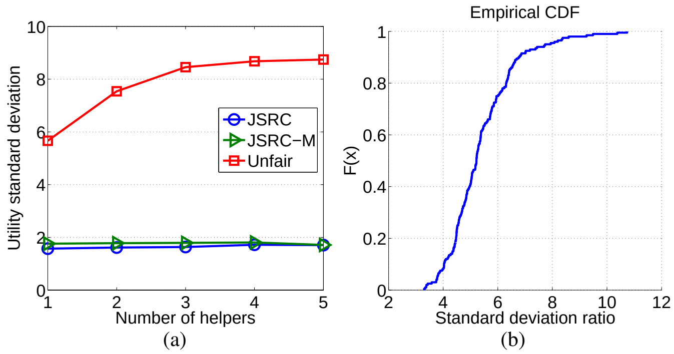

Fig. 5. Total utility of the client nodes, delivery rates $\in [0.5, 1]$, $n = 10$, $T = 50$, $b_j^t \in [2, 4]$, $b_{ji}^t \in [1, 2]$. (a): $b_i^t \in [0.2, 0.4]$; (b): $b_i^t \in [0.5, 1]$.

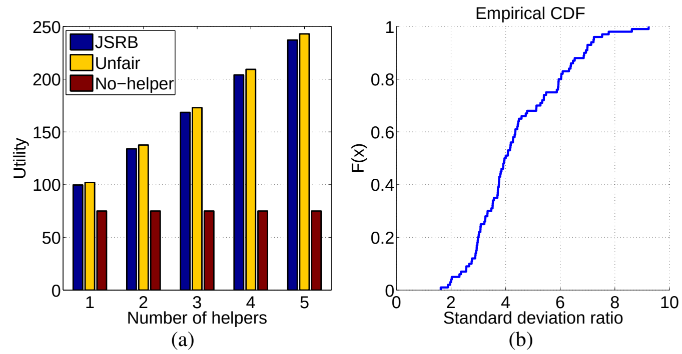

Fig. 7. Delivery rates $\in [0.5, 1]$, $n = 10$, $T = 50$, $b_j^t \in [2, 4]$, $b_i^t \in [0.2, 0.4]$, $b_{ji}^t \in [1, 2]$, $e_i^c = 0.3$, $e_i^w = 0.1$, $z = 0.1$. (a): Total utility of the client nodes; (b): CDF of the standard deviations radio, $n = 3$.

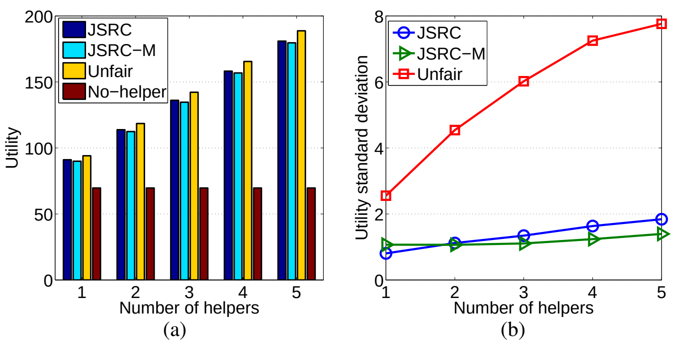

Fig. 6. Delivery rates $\in [0.5, 1]$, $n = 10$, $T = 50$, $b_j^t \in [2, 4]$, $b_i^t \in [0.2, 0.4]$, $b_{ji}^t \in [1, 2]$, $e_i^c = 0.3$, $e_i^w = 0.1$, $z = 0.1$. (a): Total utility of the client nodes; (b): Standard deviation of the client nodes' utilities.

that the utility of the JSRCP-M method is about 2% lower than that of the JSRCP approach. In the JSRCP-M method, we use the utility enhancement due to the received help from the helper nodes to calculate the α_i variable. Clearly, as the number of helpers rises, more help is provided to the clients, and the utility of the methods increases.

In order to check the fairness of the methods, we compare the standard deviation of the utility that the client nodes receive in Figure 6(b). The standard deviation of the JSRCP and JSRCP-M method are very close. However, the standard deviation of the JSRCP-M method is less than the JSRCP method, which means JSRCP-M is fairer than the JSRCP method. This is due to calculating α_i based on the utility enhancement that client i receives from the helper nodes. The standard deviation in the case of using the Unfair method is up to 4 times that of our proposed methods. From the results in Figures 6(a) and (b) we can conclude that using the JSRCP and JSRCP-M methods we can provide fairness in the cost of about 2-5% reduction in the total performance.

3) WiFi and Bluetooth: We evaluate the utility of the JSRCB, No-helper, and the Unfair methods in Figure 7(a). The utility of the Unfair method is up to 3% more than the utility of the JSRCB method, which is due to the unfair resource sharing. In addition, the utility of the No-helper methods is 26% to 70% less than the JSRCB and Unfair methods.

Figure 7(b) shows the CDF of the standard deviations. We divide the standard deviations in the case of unfair method to that of the JSRCB method and plot its CDF. As the figure illustrates, in 20% of the cases, the standard deviation of the Unfair method is up to 3 times, that of the JSRCB method. Furthermore, in 50% of the runs, the standard deviation ratio is between 4 and 9.4.

VI. Conclusion

Ubiquitous Internet connection access is becoming a requirement of our lives, as the amount of requested data by mobile users continues to rapidly increase. An effective approach to address these two demands is to use cooperative mobile Internet access. This provides the mobile users the opportunity to use help from other mobile devices to access the Internet. The mobile helper nodes download the requested data by the clients through their cellular connections, e.g 4G connection, and relay the data to the clients. The helper nodes can use WiFi, Bluetooth, or both of them to relay the data. In this work, we consider the problem of providing an Internet connection to a set of client users with the cooperation of a set of helpers. In order to increase the total utility of the client nodes, we use opportunistic scheduling to use the resources efficiently. In the first proposed method, we only use WiFi connections to transmit the downloaded data by the helpers to the clients. Then, we extend our solution to the case of using WiFi and Bluetooth connections simultaneously. The reported simulation results show the effectiveness of our scheduling and rate optimization algorithms.

References

[1] P. Ostovari, J. Wu, and A. Khreishahand, "Cooperative mobile internet access with opportunistic scheduling," in *RWS*, 2015, pp. 1–6.

[2] T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu, and S. Lu, "INDAPSON: An incentive data plan sharing system based on self-organizing network," in *IEEE INFOCOM*, 2014, pp. 1545–1553.

[3] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Enabling crowd-sourced mobile internet access," in *IEEE INFOCOM*, 2014, pp. 451–459.

[4] N. Do, C. Hsu, and N. Venkatasubramanian, "CrowdMAC: a crowd-sourcing system for mobile access," in *ACM Middleware*, 2012, pp. 1–20.

[5] M. Zakerinasab and M. Wang, "A cloud-assisted energy-efficient video streaming system for smartphones," in *IEEE/ACM IWQoS*, 2013, pp. 1–10.

[6] X. Liu, E. Chong, and N. Shroff, “A framework for opportunistic scheduling in wireless networks,” *Computer Networks*, vol. 41, no. 4, pp. 451–474, 2003.
[7] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, “Microcast: cooperative video streaming on smartphones,” in *ACM MobiSys*, 2012, pp. 57–70.
[8] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou, “Cooperative video streaming on smartphones,” in *IEEE Allerton*, 2011, pp. 220–227.
[9] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.
[10] M. Ramadan, L. E. Zein, and Z. Dawy, “Implementation and evaluation of cooperative video streaming for mobile devices,” in *IEEE PIMRC*, 2008, pp. 1–5.
[11] S. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
[12] P. Ostovari, J. Wu, and A. Khreishah, “Network coding techniques for wireless and sensor networks,” in *The Art of Wireless Sensor Networks*, H. M. Ammari, Ed. Springer, 2014.
[13] M. Luby, “LT codes,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.
[14] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
[15] X. Lin and N. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in *IEEE CDC*, 2004, pp. 1484–1489.
[16] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[17] J. Schwartz, A. Steger, and A. Weißl, “Fast algorithms for weighted bipartite matching,” in *Experimental and efficient algorithms*, 2005, pp. 476–487.
[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ Press, 2004.