Crowdsourced Mobile Data Transfer with Delay Bound

NGOC DO, University of California, Irvine
YE ZHAO, University of California, Irvine
CHENG-HSIN HSU, National Tsing Hua University, Taiwan
NALINI VENKATASUBRAMANIAN, University of California, Irvine

We consider a crowdsourcing system where mobile devices form a local community or marketplace to share network access and transfer data for each other. The designed crowdsourcing platform incorporates algorithms to enable: (i) mobile clients to select and exploit (one or more) mobile hotspots in its vicinity for data transfer, and (ii) mobile hotspots to open their cellular connectivity to admit and serve delay-bounded requests from mobile users for a fee. The evaluations of our system on an Android testbed and a packet-level simulator indicate that: (i) mobile clients can tune preferred trade-offs between cost and delay through a control knob, (ii) mobile hotspots comply with all delay bounds, and (iii) the system ensures stable and efficient transfer.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms: Algorithm, Design, Performance

Additional Key Words and Phrases: Internet Access Sharing, Network Optimization, Crowdsourcing, Cellular Networks, P2P Wireless Networks

ACM Reference Format:
N. Do, 2016. Crowdsourced Mobile Data Transfer with Delay Bound *ACM Trans. Internet Technol.* 1, 1, Article 1 (May 2015), 29 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Mobile devices, such as smartphones and tablets, have become an important part of our daily lives. Cisco reports indicate that in early 2014 [Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014], more than three out of five mobile subscribers in the U.S. own a smartphone. Today, people use mobile devices connected via cellular networks for a plethora of tasks, e.g., telephony, Internet access, capturing and sharing their surrounding context using social media services. While cellular network capacity has significantly improved over the past few years with new technologies such as 4G, mobile users are still far away from being fully satisfied with cellular network services due to (i) significant network bandwidth variation (mobile users experience different data rates at different places and times, or even connection losses), (ii) limited network coverage (this is especially true for roaming users who may

This effort was funded under the US National Science Foundation Awards CNS1450768, CNS1059436 and IIS1447720. This work was also supported by the Ministry of Science and Technology of Taiwan under the grants 102-2221-E-007-062-MY3, 104-3115-E-007-001, and 104-3115-E-007-004.
Author's addresses: N. Do, Y. Zhao, and N. Venkatasubramanian, Donald Bren School of Information and Computer Sciences, University of California, Irvine 92697, USA; C. Hsu, No. 101, Section 2, Kuang-Fu Road, Hsinchu, 30013 Taiwan.

DOI: http://dx.doi.org/10.1145/0000000.0000000

have to pay high cost to maintain services while traveling), and (iii) limited number of data plan choices.

In particular, the lack of flexible data plans is a key factor in usage and satisfaction. There is an increasing number of service providers worldwide who are moving from unlimited data plans to tiered mobile data packages. In fact, over the past 3 years, the percentage of tiered plans compared to all data plans increased from 4% to 55%, while unlimited plans dropped from 81% to 45% [Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014]. This results in very few options in terms of monthly traffic quotas for mobile users, which in turn leads to low utilization of traffic quotas. Ericsson reports that the average unused, *wasted*, monthly traffic quota is up to 61% [Traffic and Market Report 2011], causing frustration for mobile users [China Mobile Hong Kong Subscribers Can Sell Unused Data Capacity 2013]. The same report also indicates that up to 32% of mobile users exceed their monthly traffic quotas, and are charged at much higher rates. Three observations can be made from the above statistics: (i) light mobile users may find the fixed-term contracts not appealing, (ii) heavy mobile users may accidentally exceed the traffic quotas and be charged at higher rates, and (iii) other mobile users could end up with residual monthly traffic quotas.

An approach to addressing the above issues is allowing mobile users to form a localized community - this enables users to experience shorter delay of data transfer by receiving support from nearby mobile users with cellular network connectivity. In this paper, we refer to mobile users who need on-demand network connections as *mobile clients*; similarly, we refer to mobile users who are willing to transport data for mobile clients as *mobile hotspots*. Mobile clients and hotspots communicate with each other using local (one-hop) wireless networks, such as WiFi Direct [Wi-Fi Certified Wi-Fi Direct 2013] or Bluetooth. Using such a trading mechanism has many advantages, for example: (i) it provides connectivity to unconnected mobile clients and (ii) it shortens the delay of data transfer through the cellular connections of nearby devices. Simultaneously, mobile hotspots can potentially be compensated for unused data quotas. We believe such an approach is practical, especially in situations where mobile users can gather and form groups such as in buses, airports, and restaurants. Additionally, smartphone technologies today allow easy and fast local network formation. At high level, this approach allows more efficient utilization and management of data plans. Last, we mention that a mobile user may be a mobile client at one point (time and location), but may become a mobile hotspot at another point.

At the first glance, the approach seems straightforward. However, building such a marketplace-inspired crowdsourcing framework has several challenges. First, the system should support incentives for mobile hotspots to share their connectivities. Second, a mobile hotspot must be capable of serving multiple clients while ensuring good connection quality for the concurrent users - this is essential to attract mobile clients to participate in the transfer. Third, schemes that enable mobile clients to leverage multiple mobile hotspots for faster data transfer are not necessarily straightforward. Other features include the ability of the system to handle connection losses between mobile devices, save energy consumption at mobile clients and hotspots for access sharing, and enable secure data transfer. In this paper, we develop a system that addresses the challenges. The main contributions of the paper are:

— We develop a crowdsourcing system in which mobile users form a local community to share network access for data transfers (Section 3). Mobile clients who are in need of network access issue requests to buy bandwidth for data transfers while mobile hotspots sell bandwidth by accepting and serving requests.
— We propose a delay-bounded admission control algorithm using a Lyapunov optimization framework for mobile hotspots to admit data transfer requests from mobile

clients to maximize revenue while servicing each request with a worst-case delay bound, which provides guaranteed quality of services for mobile clients (Section 4).
— We develop a mobile hotspot selection algorithm for mobile clients also based on the Lyapunov framework, which allows mobile clients to exploit multiple hotspots for fast data transfers. A key aspect of this framework is the design of a control knob that captures the trade-off between delay and cost (Section 5).
— We evaluate the proposed algorithms and system using both an Android testbed and a commercial packet-level simulator (Sections 6 and 7).
— We look into the practical implementation issues for larger-scale deployments (Section 8).

2. RELATED WORK

Modern mobile phones incorporate multiple networking interfaces and can form opportunistic networks using WiFi Direct [Wi-Fi Certified Wi-Fi Direct 2013], WiFi Tether [Android WiFi Tether 2012] and Bluetooth. Opportunistic networks complement infrastructure networks with new capabilities such as network coverage improvement [Luo et al. 2003; Luo et al. 2007; Bhatia et al. 2006], network throughput enhancement [Law et al. 2010; Kim and Shin 2005; Ananthanarayanan et al. 2007], or rich content distribution [Keller et al. 2012; Do et al. 2013]. Research efforts to improve network coverage have been proposed in [Luo et al. 2003; Luo et al. 2007; Bhatia et al. 2006]. For example, Luo et al. [Luo et al. 2003; Luo et al. 2007; Bhatia et al. 2006] employ an ad hoc network to relay data downloaded over a device's cellular network interface to a nearby device which does not have a cellular connection. A mobile device which wishes to download data looks for a mobile proxy which is actually a nearby mobile device with the highest cellular data rate, and establishes an ad hoc path from the proxy to itself. Data is unicast from the Internet via a cellular network base station to the proxy and then to the requesting device along the path. These works address efficient relayer and route selection problem for achieving high throughput.

Another interesting research direction [Law et al. 2010; Kim and Shin 2005; Ananthanarayanan et al. 2007] uses multiple networks for enhancing network throughput. Kim and Shin [Kim and Shin 2005] develop a multipath TCP protocol to speed up data download from a server through the support of nearby mobile devices. More specifically, they design a system that handles TCP's data and acknowledgment traffic going through nearby mobile devices and propose a congestion control mechanism that handles packet loss at server side. Ananthanarayanan et al. [Ananthanarayanan et al. 2007] develop a decentralized system for high speed download where mobile devices collaborate to download data and earn credits. Law et al. [Law et al. 2010] theoretically evaluate the maximum capacity of hybrid cellular and ad hoc networks under an assumption that multicast is enabled in cellular networks for data transfer. Sharma et al. [Sharma et al. 2009] develop a system to increase the throughput by leveraging nearby mobile device's connection with an emphasis on energy consumption.

Recent research has also proposed solutions [Do et al. 2013; Keller et al. 2012] to efficiently distribute videos to mobile devices. For example, Keller et al. [Keller et al. 2012] separate a video desired by a group of nearby mobile devices into small chunks and distribute the chunks to the group over cellular and ad hoc networks. Do et al. [Do et al. 2013] study and develop a system for streaming live videos encoded with layered coding to mobile devices such that video quality perceived at mobile devices is maximized.

There also exist both free and commercial software that enables the tethering feature on mobile devices for sharing Internet access. Some examples are Open Garden [Open Garden 2014] and FoxFi [FoxFi 2014]. However, most of the existing works assume a cooperative environment in which mobile devices collaboratively distribute

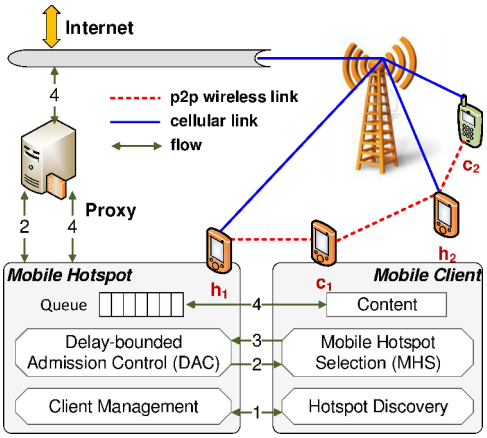


Fig. 1. System architecture and software modules.

contents. It is unusual for a mobile user to share data connections with other mobile users, unless they are family, friends, or interested in the same content. Chakravorty et al. [Chakravorty et al. 2005] propose a framework for forming open marketplaces to share wireless services. However, the framework is developed to support the sharing of unused resources in general, and cannot be applied directly to share unused data quotas because critical characteristics, e.g., cost and performance, are not well studied. More recent work [Iosifidis et al. 2014b; 2014a; Yu et al. 2014] focuses on sharing mobile Internet access using crowdsourcing and data plan sharing. Yu et al. [Yu et al. 2014] propose an incentive mechanism to encourage mobile users who have plenty of data quotas in data plans to download data for nearby users to earn credits that will be used later by themselves when they lack data quotas. The amount of credit earned by a mobile user depends on how active the user participates in downloading data for other users in history and the amount of data transferred. This work is limited in focusing mostly on designing the incentive strategy. This strategy is incomplete as it does not consider the cost of the data plan. It also excludes many other important parts in the system such as quality of services and mobility. Iosifidis et al. [Iosifidis et al. 2014b; 2014a] develop a more robust approach for mobile Internet by applying crowdsourcing concept for mobile Internet access. Specifically, they provide a game theory based approach that encourages mobile user's participation and performs efficient and fair data transfer. However, this work is heavily theoretical, and hard to implement and deploy in practice.

To our best knowledge, our earlier work [Do et al. 2012] has been the first attempt to motivate data plan sharing via virtual marketplaces. Via data plan sharing, we provide a larger coverage area, shorter data transfer delay, and thus higher throughput. The current paper enhances [Do et al. 2012] as follows. We propose a system, and a suite of carefully designed algorithms including a new admission control algorithm at the mobile hotspot that provides a guaranteed transfer delay for any single request from mobile client and a new multihomed technique at mobile client supporting multiple mobile hotspot hiring for fast content transfer. The proposed system and algorithms result in a unified and coherent solution for mobile users to transfer data and utilize data plan more efficiently.

3. SYSTEM ARCHITECTURE AND WORKFLOW

Our system enables mobile clients to discover and hire mobile hotspots for performing and enhancing throughput for data transfers. Figure 1 depicts a sample deployment of the proposed system with two mobile clients c_1 and c_2 and mobile hotspots h_1 and h_2. Mobile clients may or may not have cellular network connections by the time they are requesting helps from mobile hotspots. Initially, mobile users register with our system, and their identity as well as other information are stored at a *proxy*. For brevity, we present our system and algorithms using upload scenarios, and mention the differences of download scenarios if necessary.

When a mobile client wants to upload content and needs support from mobile hotspots, it invokes *Hotspot Discovery* module to discover nearby mobile hotspots over a local wireless network such as WiFi Direct or Bluetooth. The module broadcasts request message with the mobile client's identity and the size of the data requested for upload. The mobile client may divide a large content into multiple segments. For each segment, the mobile client sends a request and hires a mobile hotspot to transfer the segment to the Internet (step 1). Note that energy consumed by the Hotspot Discovery module for scanning mobile hotspots may be significant, for example in WiFi networks, because the network interface must be kept up and messages are kept being sent out. Luckily, there exist some techniques [Sharma et al. 2009; Han et al. 2012] that could solve this issue efficiently.

Upon receiving the request, a mobile hotspot invokes its *Delay-bounded Admission Control (DAC)* module to decide if it would admit or reject the request. The purpose of DAC is to guarantee that if the hotspot admits the request, it will provide a guaranteed quality connection for the client. In this work, the quality refers to the delay that is measured as the time difference between the request is admitted and the last byte of the data is transferred to the destination. For each admitted request, the mobile hotspot sends a reply to the client with a segment transfer delay and the price to serve the request (step 2).

To motivate the mobile hotspot to transfer data for the mobile client, the mobile hotspot will charge the mobile client some credits that is proportional to the size of the transferred data. In future, the earned credits could be used by the mobile hotspot if it wishes to gain support from nearby mobile users for transferring its data. A possible charging strategy includes:

(1) *Data plan fee*: to cover the cost of the mobile hotspot's monthly cellular data plan.
(2) *Resource consumption fee*: to account for the consumption of local resources at the mobile hotspot, such as energy and storage.
(3) *SLA fee*: to set up a Service-Level Agreement (SLA) between the hotspot and its network provider for reselling the unused data quota.

These fees are defined by mobile hotspots as functions of transferred data mount. The first two fees motivate mobile hotspots to transfer data, and the last fee incentivizes cellular network providers to participate in our scheme. Note that our system is amenable to any charging scheme. Besides, a potential extension to the system is that mobile users can exchange credits for real currency. We would consider this extension in our future work.

The client may receive several replies from surrounding mobile hotspots. The client uses the *Mobile Hotspot Selection (MHS)* module to choose the hotspot with the most preferred trade-off between delay and cost, and sends a confirmation to it (step 3). The client next transmits data to the mobile hotspot. The incoming data to the mobile hotspot is stored in a FIFO queue Q before it can be delivered to the Internet (step 4). The mobile client repeats the above steps until the content is completely uploaded.

Mobile hotspots use the Client Management module to keep track of connections to mobile clients. To handle mobility, a segment admitted for transfer is partitioned further into chunks; a mobile hotspot only advances to the next chunk, when it completes the current chunk transfer and receives an acknowledgement from the mobile client.

The crux of the functionality of this system lies in the DAC and MHS modules. The DAC module at mobile hotspots selectively admits requests from mobile clients to maximize its earned credits from multiple requests while providing a delay guarantee for each request. The MHS module at mobile clients helps select multiple mobile hotspots, and provides a trade-off between content transfer delay and cost. Details of these modules are presented in Sections 4 and 5.

4. DELAY-BOUNDED ADMISSION CONTROL

A mobile hotspot upon receiving a request from a mobile client for data transfer needs to decide if it should admit and serve the request, based on factors such as whether it currently has some background traffic or it may be busy with serving requests from other clients. In this section, we design an admission control algorithm for mobile hotspots, and the goal of the algorithm is that mobile hotspot can maximize its earned credits while providing a guaranteed data delivery delay for each request it admits.

4.1. Admission Control Problem

We start this section by describing several network models for link capacity, incoming and outgoing traffic to and from mobile hotspot, real queue for storing data, and then move on to the problem statement.

Link capacity: Let $\mu_l(t)$ and $\omega_l(t)$ denote the amount of data transferred through link l and l's capacity at time slot t, i.e., $\mu_l(t) \leq \omega_l(t)$. We assume $\omega_l(t)$ is i.i.d. over time slots, depends on channel quality, and is bounded by $\omega_l(t) \leq \omega^* \ \forall l, t$, where ω^* is the maximum channel capacity under any channel quality.

Incoming traffic to mobile hotspot: We define the total incoming data amount to mobile hotspot h at time slot t for upload scenarios as $\mu_h^i(t) = \sum_{c \in \Gamma_h(t)} \mu_{c-h}(t)$, where $\Gamma_h(t)$ is a set of mobile clients that h serves at t and $c-h$ is the link from mobile client c to h, or $\mu_h^i(t) = \mu_l(t)$, where l is h's cellular links for download scenarios. Let's define μ^{i*} as the maximum incoming data amount. We have $\mu_h^i(t) \leq \mu^{i*} \quad \forall h, t$.

Outgoing traffic from mobile hotspot: We define $\mu_h^o(t)$ as the amount of data transmitted out of mobile hotspot h in t, i.e., $\mu_h^o(t) = \mu_l(t)$ for upload scenarios where l is h's cellular link, or $\mu_h^o(t) = \sum_{c \in \Gamma_h(t)} \mu_{h-c}(t)$ for download scenarios. We denote μ^{o*} as the maximum outgoing data amount of any mobile hotspot at any time slot, i.e., $\mu_h^o(t) \leq \mu^{o*} \quad \forall h, t$.

Real queue for storing incoming data: Each mobile hotspot uses a real queue to store incoming data packets. Let $Q_h(t)$ denote h's real queue backlog (i.e., the number of bytes in the queue) at the beginning of t at h. $Q_h(t)$ evolves over time as presented below:

$$Q_h(t+1) = [Q_h(t) - \mu_h^o(t) + \mu_h^i(t)]^+, \tag{1}$$

where $[x]^+ = x$ if $x \geq 0$ and $[x]^+ = 0$ if $x < 0$. We say h's real queue is stable[1],if it is maintained with a finite average queue backlog, i.e.,

$$\bar{Q}_h \triangleq \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Q_h(\tau)\} < \infty, \forall t \tag{2}$$

[1]More detailed definitions on queue stability can be found in [Neely 2011].

Let's define long term values for incoming data amount and outgoing data amount as $\bar{\mu}_h^i \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\mu_h^i(\tau)\}$ and $\bar{\mu}_h^o \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\mu_h^o(\tau)\}$, respectively. If $Q_h(t)$ is stable, the long term outgoing data amount is not less than the incoming data amount, i.e., $\bar{\mu}_h^o \geq \bar{\mu}_h^i$.

Requests: A mobile hotspot h may receive multiple requests and simultaneously serve several mobile clients. Let $R_h(t)$ denote a set of mobile clients requesting h to serve in time slot t, and $A_h(t)$ denote the total data amount requested to h in t, i.e., $A_h(t) = \sum_{c\in R_h(t)} s_c(t)$, where $s_c(t)$ is the size of the segment requested by client c in t. We assume that $A_h(t)$ is i.i.d. over time slot, and $A_h(t) \leq A^* \quad \forall h, t$, where A^* is the maximum possible data amount requested in a time slot. Every time slot, h admits a subset of the requesting mobile clients $r_h(t) \subset R_h(t)$. Thus, the admitted data amount is $a_h(t) = \sum_{c\in r_h(t)} s_c(t) \leq A_h(t)$.

Problem statement: A mobile hotspot h receives a set of requests for data transfer $R_h(t)$ in time slot t, and needs to determine a subset of these requests, $r_h(t)$, to admit. Its objectives are to serve each of the admitted requests (including the ones admitted previously) within a deadline and maximize the amount of admitted data (i.e., it earns as much as possible credits in long term, named as long term revenue), defined as:

$$\bar{P_h} \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in r_h(\tau)} \mathbb{E}\{P_h(s_c(t))\}, \tag{3}$$

where $P_h(s_c(t))$ is price (i.e., numbers of credits) for transfering a segment.

4.2. Delay Bounded Admission Control Algorithm

Algorithm design principles: We employ the Lyapunov optimization framework [Georgiadis et al. 2006] to design our admission control algorithm. The Lyapunov framework [Georgiadis et al. 2006] is well suited for developing distributed control techniques. Furthermore, Lyapunov based techniques can be efficiently put into practice in wireless networks.

For each mobile hotspot h, we define two virtual queues $X_h(t)$ and $Y_h(t)$, in addition to h's real queue. The virtual queues are counters, which incur very little overhead. The evolution of virtual queues are defined as:

$$X_h(t+1) = [X_h(t) - \mu_h^i(t) - g_h(t) + a_h(t)]^+; \tag{4}$$

$$Y_h(t+1) = \begin{cases} y_h(t) & \text{if } X_h(t) > 0 \text{ or } Q_h(t) > 0; \\ 0 & \text{if } X_h(t) = Q_h(t) = 0, \end{cases} \tag{5}$$

where $y_h(t) = [Y_h(t) - \mu_h^i(t) - g_h(t) + a_h(t) + \delta_h]^+$, $g_h(t)$ is the data amount which is admitted by h but gets lost due to disconnections[2] from mobile clients to h (assume $g_h(t) \leq g^*$), and δ_h is a constant which is an 'artificial' incoming rate added to $Y_h(t)$ to control $Y_h(t)$'s convergent speed. The virtual queues are stable if

$$\bar{X_h} \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{X_h(\tau)\} < \infty, \forall t; \tag{6}$$

$$\bar{Y_h} \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Y_h(\tau)\} < \infty, \forall t. \tag{7}$$

[2]Both disconnections between: (i) mobile client and mobile hotspot and (ii) mobile hotspot and final destination (or source) are included.

The reason behind introducing two virtual queues is as follows. Let's define: $\bar{g}_h \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{g_h(\tau)\}$ and $\bar{a}_h \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{a_h(\tau)\}$. If virtual queue $X_h(t)$ is stable, we have $\bar{\mu}_h^i + \bar{g}_h \geq \bar{a}_h$, i.e., if a request is admitted by h, it will be completely served unless h suffers from a disconnection to a client who issues that request (the request will be dropped out). Virtual queue $Y_h(t)$ has the same outgoing rate as $X_h(t)$, but has an additional incoming rate δ_h. $Y_h(t)$ is reset to 0 only when h completes serving all admitted requests. To stabilize $Y_h(t)$, h must stop receiving new requests at some point of time and complete all current requests before it can admit new requests. This forces h to make sure the admitted requests are completed before considering new requests. Therefore, h is able to serve admitted within the expected delay. We note that our derivations are inspired by a recent work [Neely 2011].

The admission control problem is now to maximize the long term revenue defined in (3) while stabilizing the queues. We define $\phi_h(t) = [Q_h(t); X_h(t); Y_h(t)]$. A quadratic Lyapunov function $L(\phi_h(t))$ of queue backlogs and an one-step conditional Lyapunov drift $\triangle(\phi_h(t))$ for mobile hotspot h are defined below:

$$L(\phi_h(t)) \triangleq \frac{1}{2}[Q_h^2(t) + X_h^2(t) + Y_h^2(t)]; \tag{8}$$

$$\triangle(\phi_h(t)) \triangleq \mathbb{E}\{L(\phi_h(t+1)) - L(\phi_h(t))|\phi_h(t)\}. \tag{9}$$

It is easy to see that if $\triangle(\phi_h(t))$ is negative over time slots, the queues at h are stable.

The following lemma for the Lyapunov drift for h holds in any time slot with any admission control algorithm.

LEMMA 1. *The one-step Lyapunov drift is:*

$$\begin{aligned}\triangle(\phi_h(t)) \leq \beta_h &- \mathbb{E}\{[X_h(t) + Y_h(t)]g_h(t)|\phi_h(t)\} \\ &+ \mathbb{E}\{Y_h(t)\delta_h|\phi_h(t)\} \\ &+ \mathbb{E}\{[X_h(t) + Y_h(t)]a_h(t)|\phi_h(t)\} \\ &- \mathbb{E}\{[X_h(t) + Y_h(t) - Q_h(t)]\mu_h^i(t)|\phi_h(t)\} \\ &- \mathbb{E}\{Q_h(t)\mu_h^o(t)|\phi_h(t)\},\end{aligned} \tag{10}$$

where β_h is a constant.

Mobile hotspot h attempts to both maximize the long term revenue and stabilize its queues. Let's define a control parameter V_h as a revenue weight for h, which indicates how much importance the revenue is for h ($V_h > 0$). Our optimization problem can be represented as:

$$\min : \triangle(\phi_h(t)) - V_h \sum_{c\in r_h(t)} \mathbb{E}\{P_h(s_c(t))|\phi_h(t)\}. \tag{11}$$

Next, we insert (10) into (11). Note that we cannot control β_h, $\mathbb{E}\{[X_h(t) + Y_h(t)]g_h(t)|\phi_h(t)\}$, and $\mathbb{E}\{Y_h(t)\delta_h|\phi_h(t)\}$ because β_h and δ_h are constants, and $g_h(t)$ is lost due to disconnections. Solving (11) translates to solve the maximization problems described in (12), (13) and (14):

$$\begin{aligned}\max : V_h\mathbb{E}\{ &\sum_{c\in r_h(t)} P_h(s_c(t))|\phi_h(t)\} \\ &- \mathbb{E}\{[X_h(t) + Y_h(t)] \sum_{c\in r_h(t)} s_c(t)|\phi_h(t)\};\end{aligned} \tag{12}$$

$$\max : \mathbb{E}\{[X_h(t) + Y_h(t) - Q_h(t)]\mu_h^i(t)|\phi_h(t)\}; \tag{13}$$

$$\max : \mathbb{E}\{Q_h(t)\mu_h^o(t)|\phi_h(t)\}. \tag{14}$$

Algorithm 1 Delay-Bounded Admission Control Algorithm

(1) **Request Admission** : Given a set of requests, $R_h(t)$ in time slot t, mobile hotspot h admits a subset $r_h(t) \subset R_h(t)$, to maximize:

$$\sum_{c \in r_h(t)} \left(V_h P_h(s_c(t)) - [X_h(t) + Y_h(t)] s_c(t) \right) \tag{15}$$

(2) **Congestion Control**: At time slot t, mobile hotspot h monitors its queues and stops incoming data traffic if:

$$X_h(t) + Y_h(t) < Q_h(t) \tag{16}$$

Otherwise, it allows the maximum incoming data rate, i.e., $\mu_h^i(t) = \min[X_h(t), \mu^{i*}]$.

(3) **Outgoing Data Transmission**: At time slot t, data is sent out of $Q_h(t)$ if this queue is not empty.

The proposed algorithm: We design an algorithm to perform (12), (13), and (14). The algorithm contains three procedures: Request Admission, Congestion Control, and Outgoing Data Transmission, which are summarized in Algorithm 1. The Request Admission procedure is to solve (12) by selectively admitting requests to opportunistically maximize expectations. Solving the maximization problem in (15) is simple and takes $O(n)$. Mobile hotspot h admits c in $R_h(t)$, if for c's request, $V_h P_h(s_c(t)) - [X_h(t) + Y_h(t)] s_c(t) \geq 0$. The Congestion Control procedure in Algorithm 1 is to solve (13). If h's real queue backlog is larger than the total backlog of its two virtual queues, h stops taking incoming data as h is congested. Otherwise, incoming data is transmitted to h with the highest data rate. Mobile hotspot h stops incoming data from being sent by broadcasting a stop message to mobile clients in upload scenarios, or temporarily stop downloading data from the Internet in download scenarios. The Outgoing Data Transmission procedure is straightforward: h always sends data if its real queue is not empty.

4.3. Performance Analysis of Admission Control Algorithm

We start our analysis by showing the bounds on the queues.

THEOREM 1. *At any time slot, virtual queue $X_h(t)$ is bounded by X_h^*:*

$$X_h(t) \leq \frac{V_h \rho_h^*}{2} + A^* \triangleq X_h^* \quad \forall t, \tag{17}$$

where $\rho_h^* = \max_{0 < s_c(t) \leq S} \frac{P_h(s_c(t))}{s_c(t)}$.

Theorem 1 shows that there is a finite amount of data admitted for transferring at mobile hotspot h, depending on how large V_h is selected.

We define *a service round* as a period of time at mobile hotspot h starting at time slot t_s when $X_h(t_s) = Q_h(t_s) = 0$ and ending at t_e when both these queues converge back to 0 for the first time after t_s (so, $Y_h(t)$ also turns back to 0). The conditions for a service round $[t_s, t_e]$ are: $t_e > t_s$, $X_h(t_e) = Q_h(t_e) = 0$, and at t ($t_s < t < t_e$), mobile hotspot serves at least one request, i.e., $X_h(t) + Q_h(t) > 0$. Without loss of generality, reset t to 0 when a service round starts. We have the following theorem and corollary.

THEOREM 2. *During a service round, the largest queue backlog, $X_h(t)$, at $t > 0$ when h can still admit more requests before the service round ends, defined as $X_h^b(t)$, decreases over time:*

$$X_h^b(t) = \frac{V_h\rho_h^* - t\delta_h}{2}. \tag{18}$$

COROLLARY 1. *In a service round, mobile hotspot h cannot admit more requests after t_h^s slots, which is written as:*

$$t_h^s = \frac{V_h\rho_h^*}{\delta_h}. \tag{19}$$

Clearly at t_h^s, $X_h^b(t_h^s) = 0$. These interesting lemma and corollary mean that there is a time instant such that all requests admitted by h must be completely served before h can admit new requests.

THEOREM 3. *The delay bounded admission control algorithm guarantees a worst case bound on real queue $Q_h(t)$:*

$$Q_h(t) \leq 5\frac{V_h\rho_h^*}{2} + 3A^* + \mu^{i*} \triangleq Q_h^* \quad \forall t. \tag{20}$$

Similar to $X_h(t)$, $Q_h(t)$'s worst case bound depends on V_h. This theorem is important for mobile users offering hotspot service because they can control the memory usage of the system so that device memory is not exhausted.

The following lemma shows that any request admitted by mobile hotspot h is completed within a delay bound, T_h.

LEMMA 2. *Let us denote the expected incoming data rate and outgoing data rate at mobile hotspot h as μ_h^i and μ_h^o. Our algorithm provides a guarantee on the worst case delay bound T_h for any admitted request.*

$$T_h = t_h^s + t_h^+ + t_h^- \quad \textit{slots}, \tag{21}$$

where t_h^s is defined in (19), $t_h^+ = \frac{\mu^{i*}}{\mu_h^o + \delta_h}$, and

$$t_h^- = \begin{cases} \frac{\hat{Q}_h^* + X_h^*}{\mu_h^o} & \textit{if } \delta_h \geq 3\mu_h^i - \mu_h^o \textit{ and } \mu_h^i \geq \mu_h^o; \\ \frac{X_h^*}{\mu_h^i} & \textit{if } \delta_h \geq 3\mu_h^i - \mu_h^o \textit{ and } \mu_h^i < \mu_h^o; \\ \frac{\hat{Q}_h^* + X_h^*}{\mu_h^o} & \textit{if } \delta_h < 3\mu_h^i - \mu_h^o \textit{ and } \mu_h^i \geq \mu_h^o; \\ \frac{\hat{Q}_h^*}{\mu_h^o} + \frac{X_h^*}{\mu_h^i} & \textit{if } \delta_h < 3\mu_h^i - \mu_h^o \textit{ and } \mu_h^i < \mu_h^o, \end{cases} \tag{22}$$

where $\hat{Q}_h^* = [2V_h\rho_h^* + 2A^* + \mu^{i*} - \mu_h^o t_h^+]^+$.

5. MOBILE HOTSPOT SELECTION

A mobile client with a large content requires high bandwidth for fast data transfer. In our scheme, we allow the client to aggregate bandwidth from connections of multiple hotspots nearby, if available. More specifically, the client divides the content into multiple segments, and for each segment it hires a nearby mobile hotspot to transport. Therefore, the whole content can be transferred through multiple connections at the same time (multihomed). In this section, we study the problem of selecting mobile hotspots for individual segments to transfer the whole content efficiently.

5.1. Mobile Hotspot Selection Problem

We consider a network including a set H of mobile hotspots, a set C of mobile clients, and a set L of wireless links among the hotspots and clients. We assume that time is slotted, and use t to represent a time slot. Without loss of generality, we assume that a mobile client c has at most one content for transfer at any moment. Let f_c denote the content size. We assume that f_c is bounded by F^*. The content is divided into n segments including $n-1$ segments with size S and the last one with $f_c-(n-1)S$. A mobile client can set segment size different from other clients, but for simplicity, we assume they use the same segment size. Let us denote $s_c(t)$ as the size of the segment currently considered for transfer. Mobile client c searches in its local network for mobile hotspots to request for segment transfer. At each time slot, c requests admissions from mobile hotspots, and determines the best mobile hotspot for the current segment.

A mobile hotspot h is characterized by the pair of request delay T_h and price (i.e., numbers of credits) $P_h(s_c(t))$ for a segment requested by c for transfer. T_h (defined in Section 4) can be interpreted as the worst case delay h guarantees to complete the transfer of a requesting segment to the Internet. In our system, mobile client c will be charged by h a price $P_h(s_c(t))$ for its request, depending on the data amount $s_c(t)$ requested for transfer. It is reasonable and practical to assume that $P_h(s_c(t))$ is non-decreasing on the transferred data amount.

Mobile client c makes a decision on which mobile hotspot it should hire for transferring the current segment, based on its preference on credit and request delay. We combine both metrics into a multi-objective cost function:

$$C_c(t) = P_h(s_c(t)) + w_c T_h, \tag{23}$$

where $C_c(t)$ is the cost c pays at time slot t if mobile hotspot h is selected, and w_c is c's price and request delay weight. This weight indicates the extent to whether request delay is more important than price. If $w_c = 0$, c does not consider request delay at all while if $w_c = \infty$, price is not important to c. Such affine combination of multiple metrics in an utility function has been widely used in the literature, e.g., [Ren et al. 2012]. We define the long term cost of mobile client c as:

$$\bar{C_c} \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{C_c(\tau)\}. \tag{24}$$

The problem of mobile hotspot selection at the mobile client is to minimize the long term cost $\bar{C_c}$ while transferring the content as fast as possible.

5.2. Mobile Hotspot Selection Algorithm

Algorithm design principles: We also employ the Lyapunov optimization framework [Georgiadis et al. 2006] to design an algorithm to solve the hotspot selection problem.

A mobile client c maintains a virtual queue Z_c to control how fast the content is transferred. $Z_c(t)$ is defined as the queue backlog (i.e., number of bytes) of Z_c at the beginning of time slot t. $Z_c(t)$ is just a counter number, and evolves as follows:

$$Z_c(t+1) = [Z_c(t) - \lambda_c(t) + m_c(t) + \xi_c]^+, \tag{25}$$

where $\lambda_c(t)$ ($0 \le \lambda_c(t) \le S$) is the size of the segment considered in time slot t, $m_c(t)$ is the remaining data amount that is already scheduled to be sent to mobile hotspots in previous time slots, but added back to the virtual queue due to the disconnection between c and mobile hotspots (we assume $m_c(t) \le m^* \forall t$), and ξ_c is a constant added

to the queue every time slot. At $t = 0$, $Z_c(t) = f_c$, virtual queue $Z_c(t)$ is stable if

$$\bar{Z}_c \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Z_c(\tau)\} < \infty, \forall t \tag{26}$$

holds. ξ_c is added to the virtual queue to force c to choose mobile hotspots for fast data transfer if it wants to stabilize $Z_c(t)$. The problem of mobile hotspot selection is therefore equivalent to the problem of minimizing the long term cost $\bar{C}_c$ while maintaining stable $Z_c(t)$.

We define $\phi_c(t) = [Z_c(t)]$ and a Lyapunov function for mobile client c as $L(\phi_c(t)) \triangleq \frac{1}{2}Z_c^2(t)$. An one-step Lyapunov drift is defined as below:

$$\triangle(\phi_c(t)) \triangleq \mathbb{E}\{L(\phi_c(t+1)) - L(\phi_c(t))|\phi_c(t)\}. \tag{27}$$

The intuition is if $\triangle(\phi_c(t))$ is negative over time slots, $Z_c(t)$ will be stable.

LEMMA 3. *The Lyapunov drift at any time slot using any control algorithm satisfies:*

$$\begin{aligned}\triangle(\phi_c(t)) \leq \alpha_c + \mathbb{E}\{Z_c(t)[m_c(t) + \xi_c]|\phi_c(t)\} \\ - \mathbb{E}\{Z_c(t)\lambda_c(t)|\phi_c(t)\},\end{aligned} \tag{28}$$

where α_c is a constant [3].

Our algorithm is designed to minimize the one-step Lyapunov drift and cost. That is, we will do:

$$\min : \triangle(\phi_c(t)) + V_c\mathbb{E}\{C_c(t)|\phi_c(t)\}, \tag{29}$$

where V_c is a cost weight for c. We perform (29) by maximizing $\mathbb{E}\{[Z_c(t)]\lambda_c(t)|\phi_c(t)\} - V_c\mathbb{E}\{C_c(t)|\phi_c(t)\}$. We cannot control α_c and $\mathbb{E}\{Z_c(t)[m_c(t) + \xi_c]|\phi_c(t)\}$ because $m_c(t)$ is caused by disconnections between mobile hotspots and c due to mobility, and α_c and ξ_c are constants.

Algorithm 2 Mobile Hotspot Selection Algorithm

Given a set of mobile hotspots $H_c(t)$ admitting a request from mobile client c at t, c selects a mobile hotspot h to:

$$\max_{h\in H_c(t)} \quad Z_c(t)\lambda_c(t) - V_cC_c(t). \tag{30}$$

The proposed algorithm: Our algorithm is presented in Algorithm 2. Let's denote $H_c(t)$ as a set of mobile hotspots willing to serve c's request at t. Given $H_c(t)$, c browses through every mobile hotspot h in $H_c(t)$, calculates $\varpi_h = Z_c(t)\lambda_c(t) - V_cC_c(t)$ for h, and selects the mobile hotspot with the maximum ϖ_h (ties are broken arbitrarily) for the current segment transfer. If $\varpi_h < 0$ for all $h \in H_c(t)$, c will not choose any mobile hotspot at t. The algorithm thus runs in $O(n)$.

[3]All the proofs are given in Appendix.

5.3. Performance Analysis of Hotspot Selection Algorithm

We present our theory analysis for the algorithm.

THEOREM 4. *Assume that ξ_c is selected such that $\xi_c \leq \mathbb{E}\{\lambda_c(t) - m_c(t)\}$. The worst case content delay bound to complete a whole content transfer from mobile client c to the Internet (or from the Internet to c), denoted as D_c, is:*

$$D_c = \frac{F^* + Z_c^*}{\xi_c} + T_h^*, \tag{31}$$

where $T_h^* = \max_{h \in H_c(t), \forall t}[T_h]$ is the maximum request delay of mobile hotspots serving c, and Z_c^* is a bound on virtual queue $Z_c(t)$:

$$Z_c(t) \leq \max[V_c \eta_c^* + \xi_c + m^*, F^*] \triangleq Z_c^* \tag{32}$$

with $\eta_c^* = \max_{0 \leq \lambda_c(t) \leq S} \frac{C_c(t)}{\lambda_c(t)}$.

The above theorem shows that the content delay bound is proportional to V_c and $1/\xi_c$. Let us denote $\bar{C}_c^*$ as the minimum long term cost of the hotspot selection problem achieved by some stationary randomized algorithm.

THEOREM 5. *Our proposed algorithm achieves a bound of the long term cost:*

$$\bar{C}_c \leq \bar{C}_c^* + \frac{\alpha_c}{V_c}. \tag{33}$$

Theorem 5 indicates that our achieved long term cost is not larger than the optimal cost plus a factor of $1/V_c$. Virtual queue $Z_c(t)$ is linear with parameter V_c. Therefore, both theorems show that with the proposed algorithm, there is a trade-off $O(V_c, 1/V_c)$ between content transfer delay and cost to transfer a content from or to the Internet.

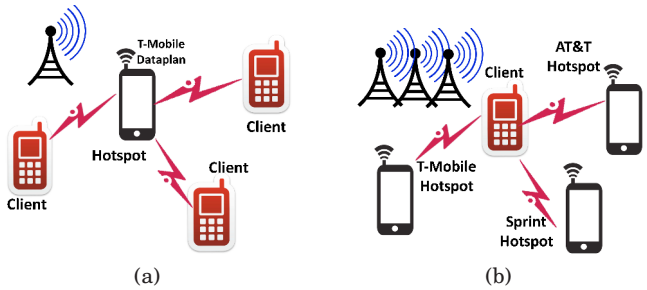
6. TESTBED BASED EVALUATIONS

6.1. Prototype and Settings

We develop a prototype system and evaluate it using upload scenarios; download scenarios with mobility are examined in the next section using simulations. Mobile devices form a local peer-to-peer wireless network using WiFi Tether [Android WiFi Tether 2012]. Mobile hotspots connect to the proxy and billing server through U.S. cellular networks. We establish two TCP connections: from mobile client to mobile hotspot and from mobile hotspot to the server. Data is buffered at mobile hotspot's real queue before being transferred to the server. The real queue is managed by the Congestion Control procedure in the DAC module.

Our implementation of the proposed system does not require time synchronization among the mobile hotspots and mobile clients. Time slot duration for mobile hotspots is 200 ms and for mobile clients is 1 s. Every time slot, a mobile hotspot runs the DAC module to select a set of mobile clients to serve, and then sends replies to the clients. It waits for confirmations from these clients in a time slot, and drops the ones from which it does not receive the confirmation out of its serving list. A client waits for mobile hotspot's replies, and invokes the MHS module to choose a mobile hotspot. The client then sends a confirmation to the selected one.

In our experiments, mobile clients do not have cellular connection while asking support from mobile hotspots. Each mobile client uploads a 10 MB content to the server. We set the segment size to be 512 KB. The client selects an appropriate hotspot to transfer each segment to the server. There are four Android phones and two laptops used in our testbed. Depending on the experiments, we configure them in one of the two scenarios: (i) one hotspot and several mobile clients as shown in Figure 2(a) and (ii) one mobile client and three hotspots as shown in Figure 2(b). The two laptops are



(a) (b)

Fig. 2. Our Android based testbed: (a) with one hotspot and multiple (up to three) mobile clients and (b) with one mobile client and three hotspots.

Table I. Credits charged by mobile hotspots depend on data plan, energy consumption, and SLA

Function	**Explanation**
$F^A_{plan} = 1$ credit/KB	AT&T - $50 for 5 GB per month [AT&T Data Plan 2014]
$F^T_{plan} = 0.6$ credit/KB	T-Mobile - $30 for 5 GB per month [T-Mobile Data Plan 2014]
$F^S_{plan} = 1.2$ credit/KB	Sprint - $34.99 for 3 GB per month [Sprint Data Plan 2014]
$F_{energy} = 0.3$ credit/KB	Energy consumption
$F_{sla} = 0.2$ credit/KB	SLA fee

used to generate WiFi background traffic. We examine our prototype at different locations: our lab, a cafe, and a park, where different cellular data rates are observed. If not specified, we show sample results from the lab.

Mobile hotspots earn credits from mobile clients by transferring data for clients. In general, each mobile hotspot determines its own function for charging clients. In this evaluation, we choose a common method: mobile hotspots earn credits proportionally to data size to compensate their data plan, energy consumption, and SLA fee. That is, the amount of earned credits is the sum of $F_{plan}(size)$ + $F_{energy}(size)$ + $F_{sla}(size)$, where $F_{plan}()$, $F_{energy}()$, $F_{sla}()$ are functions which return number of credits earned for covering data plan, energy consumption and SLA fee, respectively. For example, we map one cent to one credit. So, with AT&T data plan of $50 for 5 GB per month (approximately 1 cent per KB), $F_{plan}()$ is set to be 1 credit per KB. Details about credit based charging functions are presented in Table I. In our experiments, each mobile hotspot uses one of the data plans: F^A_{plan}, F^T_{plan}, and F^S_{plan}, summed with F_{energy} and F_{sla}. If not otherwise specified, the hotspots use F^T_{plan}. By default, we set $w_c = 0$ so that mobile client selects mobile hotspot solely based on price.

For security, we implement the techniques to provide confidentiality, integrity and authentication. More specifically, we adopt the ECDH (Elliptic Curve Diffie-Hellman) key agreement protocol to generate secret keys between pairs of entities. The encryption and decryption are performed using the 128-bit AES scheme while HMAC is done with SHA-1. We enhance PowerTutor [PowerTutor 2012] to measure the energy consumption of WiFi peer-to-peer and cellular networks. Energy efficient techniques for neighboring device discovery have been studied in [Sharma et al. 2009; Han et al. 2012], and will be implemented in our future work.

We consider the following performance metrics.

—*Request delay*: Delay to complete a segment transfer, i.e., the time difference between a mobile hotspot admitting a request and the last packet of that segment being transmitted to the Internet.

Table II. Benefits of the proposed system

Loc.	Content Delay (s)				Client Energy (J)			
	Cell.	1 Hotspot	2	3	Cell.	1	2	3
Lab	164.3	167.1	82.7	62.6	159.96	62.09	26.9	16.3
Cafe	112.4	113.8	65.7	62.1	110.83	41.69	18.57	16.24
Park	76.2	78.5	61.8	61.7	75.39	28.37	16.19	16.14

— *Content delay*: Delay to complete the transfer of a content (that is divided into multiple segments), i.e., the time difference between the first request being issued by a mobile client and the last packet of the content being transmitted to the Internet.
— *Revenue*: The total credits a hotspot earns.
— *Cost*: The credits a client pays for content transfer.
— *Energy usage*: The consumed energy amount.

6.2. Results

Delay and Energy Consumption at Mobile Clients. Table II reports the content delay and energy consumption of using a cellular network and our system with different numbers of mobile hotspots (see Figure 2(a)). We make several observations on this table. First, when there is only a single mobile hotspot, compared to cellular networks, our proposed system achieves a slightly longer content delay, but consumes much less energy. For instance, the content delay penalty is merely 2.8 s, while the energy saving is more than 62% in the lab. The high energy saving can be attributed to the fact that WiFi networks are more energy efficient for bulky data transfer. Second, in our proposed system, more mobile hotspots lead to shorter content delay and higher energy saving. For example, compared to cellular networks, our proposed system with three mobile hotspots reduces content delay by 62% and saves energy by 90% in the lab. Third, the above two observations are valid across different locations, which shows that the proposed system efficiently adapts to heterogeneous wireless channels and network conditions. In our experiments, the mobile hotspots consume more energy than the client because they consume energy from both WiFi and cellular networks (note that in this evaluation, we assume mobile clients do not have cellular connections and thus turn off cellular network interface). However, this is worthy because mobile hotspots get paid to do so. For example, at Lab, in the scenario in which the client uses only one hotspot, that hotspot consumes an energy amount of 223.27 J, and in the scenario in which the client hires three hotspots to serve its requests, each hotspot consumes 75.68 J.

Evaluation of the Delay bounded Admission Control algorithm. We select one device to be the mobile hotspot, and vary the number of clients in this experiment (see Figure 2(a)). Figure 3(a) presents the average request delay where there is only one client. With the same δ_h, increasing V_h leads to higher average request delay. For example, with $\delta_h = 10$, the average request delay increases from 8.8 s at $V_h = 100$ to 41.8 s at $V_h = 5000$. This is because the mobile hotspot admits and serves more requests simultaneously with larger V_h. Given the same value V_h, a larger δ_h provides lower average request delay. This is explained through t_h^s in Eq. (19), which indicates how much time in a service round h can admit new requests. That is, t_h^s is smaller if δ_h is larger, which leads to fewer admitted requests, shorter service round, and lower average request delay.

We now investigate average request delay in a scenario where there are three clients. The result is depicted in Figure 3(b). The same trend is observed in this figure. We also calculate the maximum request delay and maximum service round at $\delta_h = 10$, and report them in Figure 3(c) for different values of V_h. We observe that the maximum

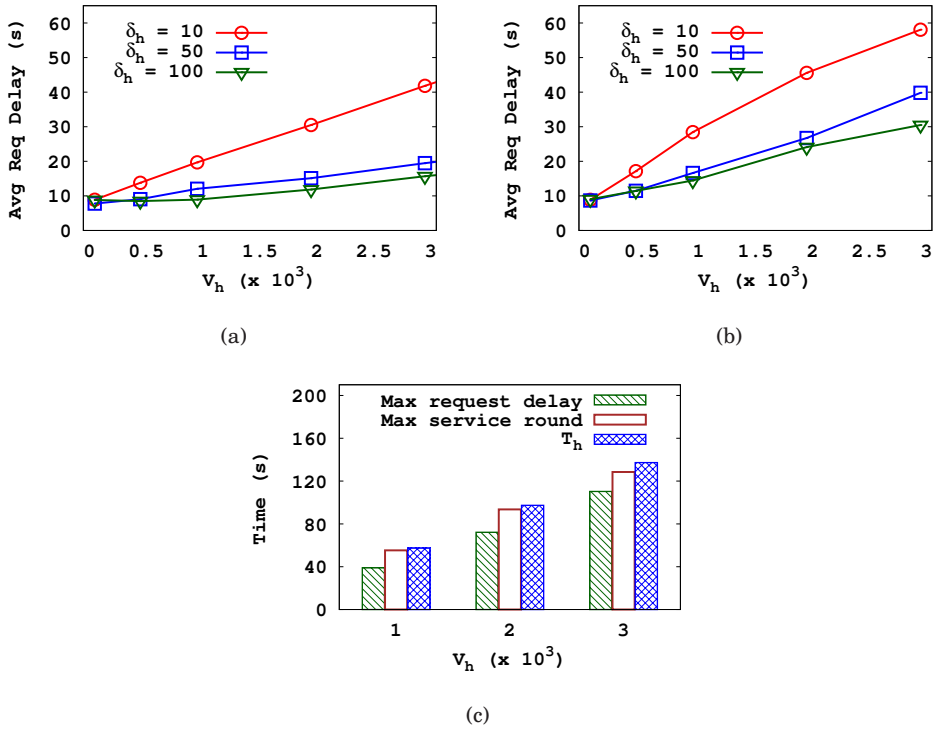


Fig. 3. Performance of the Admission Control algorithm: (a) average request delay with one mobile client, (b) average request delay with three mobile clients, and (c) maximum request delay, service round, and delay bound with three mobile clients and $\delta_h = 10$.

Table III. Performance comparisons between our algorithm and baseline

Metric	**Strict**	**DAC** ($V \times 10^3$)					**Always**
		$V = 0.1$	1	2	3	5	
Request Delay (s)	8.5	8.7	23.4	39.4	51.8	62.2	66.5
Revenue (credits)	1.65	1.61	3.22	4.83	6.44	8.06	8.06
Dropped content (%)	79	79	60	40	20	0	0

request delay is always smaller than the maximum service round, and the maximum service round is bounded by the worst case delay bound estimated from Eqs. (21) and (22). More specifically, the ratio between the maximal request delay and the delay bound is about 80%; the ratio between the maximal service round and the delay bound is about 95%. This validates the correctness of our algorithm and analysis.

We further implement two baseline schemes, *Always* and *Strict*, for the mobile hotspot to admit request. With *Always*, a mobile hotspot admits any incoming request without considering its load. With *Strict*, a mobile hotspot serves at most one request at any time. We consider a single hotspot to serve three clients. Each client uploads five 512 KB contents. The client drops the current content if it receives five rejects from the hotspot. Table III presents our results, which shows that Strict scheme achieves high service quality, but suffers from low revenue (79% of contents are dropped). Always scheme achieves the maximum revenue (all contents are accepted), but suffers from long delay: up to 66.5 s. In contrast, our DAC algorithm allows us to control the

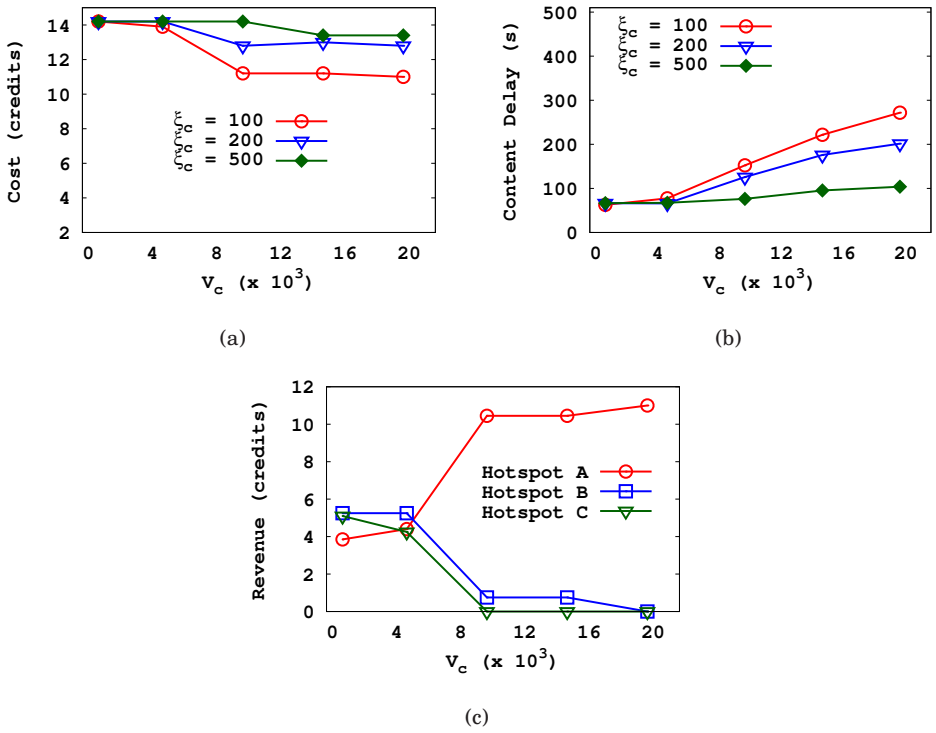


Fig. 4. Performance of the hotspot selection algorithm: (a) cost paid by the client, (b) content delay, and (c) revenue of each mobile hotspot.

revenue and the quality. At $V = 100$, DAC is close to Strict, while DAC is equivalent to Always at $V = 5000$.

Evaluation of the Mobile Hotspot Selection algorithm. In this experiment, one device works as a mobile client, and uploads content via three hotspots with diverse prices: hotspot A is with P_T, B with P_A, and C with P_S (see Figure 2(b)). We only employ one client in this experiment for brevity, the proposed scheme is still effective, stable, and efficient with multiple clients (and multiple hotspots) as shown in a related paper [Do et al. 2012]. Mobile hotspots are set with large $V_h = 2000$ and small $\delta_h = 10$. Figure 4 presents our results. Given the same ξ_c, with higher V_c, the client pays lower cost (Figure 4(a)) but experiences longer content delay (Figure 4(b)). Figure 4(c) shows that at $V_c = 20000$ and $\xi_c = 100$, the client always picks Hotspot A, which is the hotspot with the lowest cost. This selection causes high content delay, which increases from 62.6 s at $V_c = 100$ to 271.8 s at $V_c = 20000$. These observations reveal a trade-off between cost and content delay at the mobile client.

With the same V_c, for larger ξ_c, content delay decreases. For example, at $V_c = 20000$, content delay is 103.8 s with $\xi_c = 500$, but is 271.8 s if ξ_c is set to 100. This is because virtual queue $Z_c(t)$ grows faster with larger ξ_c, which forces the client to transfer the content faster. This is inline with our analysis in Section 4.

Secured Data Transfers. To provide security and avoid fraudulence, our system leverages a central authority to issue public/private key pairs and certificates to each user. Service request and response messages from the clients and mobile hotspots respectively are always signed by the sending party as a proof of identity. Once a service agreement is made between a client c, a hotspot h, and the proxy and billing server s,

Table IV. Execution time (s) for one data segment

Cryptography Features	Client	Hotspot
Without Data Encryption	0.39	0.36
All	0.92	0.37

Table V. The impact of cryptography using Nexus One

Metric	Client	Hotspot
Execution time (s)	2.02	0.56
Energy usage (J)	1.92	0.15

Table VI. Cryptography impact without data encryption using Nexus One

Metric	Client	Hotspot
Execution time (s)	0.72	0.55
Energy usage (J)	0.51	0.14

the system sets up a secure transmission session among all parties for data transfer. In particular, a key-agreement protocol is used to create short-term shared secret keys, $K_{c,h}$, $K_{h,s}$, and $K_{c,s}$ between each pair of entities.

In order to ensure authentication, integrity and confidentiality for each original data packet d, the client encrypts d with $K_{c,s}$, the HMAC of d and that of the encrypted d with $K_{c,h}$, and sends all to h. Hotspot h, upon receiving data from c, strips and decrypts the 2 HMACs. It performs the verification using the HMAC of the encrypted d. If the verification goes well, h encrypts d's HMAC with $K_{h,s}$, and finally forwards the encrypted d and the signed HMAC of the original d to s. This security scheme protects data against various attacks from malicious hotspots and cheating clients. Timestamp and nonce can be added to avoid replay attacks. Other techniques, such as blacklisting malicious users based on reputation and serving history [Chakravorty et al. 2005], can also be used to strengthen the system.

We implement the above security scheme, and use a mobile client and a mobile hotspot to quantify the execution and energy overhead due to cryptography at both devices (see Figure 2(a)). Note that this scheme provides a secured data transfer from mobile client to the proxy and billing server (or vice versa). But we can extend it easily for a secured transfer from mobile client to application server at Internet (or vice versa) by providing confidentiality between these entities and ignoring unnecessary data encryption between client and the proxy and billing server. Table IV reports the execution overhead of each 512 KB segment on Google Galaxy Nexus. This table shows that: (i) without data encryption, the overhead is less than 400 ms, and (ii) with data encryption, the overhead on the client is slightly over 900 ms. This table shows that the proposed cryptography features are fairly efficient, even on an unoptimized Android implementation.

We also isolate the energy overhead due to the cryptography features. We use Nexus One in the experiments as PowerTutor [PowerTutor 2012] does not support Google Galaxy Nexus. The experimental results indicate that the energy overhead of each segment is 0.15 J and 1.92 J on the hotspot and client respectively. Since recent smartphones are often equipped with batteries with 2000+ mAh, the observed energy overhead is negligible. We notice that most of the overhead is caused by confidentiality, that may not be important for all content types (e.g., a sightseeing photo may not be sensitive and needed to be hidden). We further investigate the cryptography impact without data encryption. We make some slight changes in the scheme to maintain the integrity and authentication. The results are presented in Table V. We observe that the overhead at Nexus One client is significantly reduced to 0.64 s. Next, we use Nexus One's next generation, Google Galaxy Nexus, to run our experiments. While Galaxy Nexus is faster than Nexus One, it is not ranked among the latest phone models as of this writing. Table VI presents the results. It is observed that the overhead is considerably reduced, up to haft times compared to using Nexus One.

7. SIMULATION BASED EVALUATIONS

7.1. Settings

We have also implemented our system in a packet level simulator, Qualnet [Qualnet Simulator 2014], for larger-scale evaluations. To simulate mobile hotspot's data connections, we employ a WiMAX network including a WiMAX base station that connects

all mobile hotspots in the network. A WiFi 802.11b network is used to establish local peer-to-peer network among the mobile hotspots and clients. The WiMAX data rate is 500 Kbps, while the WiFi rate is 1.5 Mbps. All other system parameters are the same as the ones in Sec. 6.

We use real mobility traces [Dartmouth Mobility Trace 2012] to drive our simulator. The traces include the mobility data of 24 mobile devices moving in an area of 225 by 365 m^2 in Dartmouth University. We vary mobile hotspot density in our experiments, and keep the number of mobile clients to be 6. The mobile hotspots adopt the price functions mentioned in Section 6. Each mobile client downloads a 10 MB content. Segment size is set to 512 KB, and the chunk size is 20 KB if not specifically defined. Simulation time is long enough for mobile clients to download all contents. We repeat each scenario for 10 times and report the average results. In addition to the metrics presented in Section 6, we also consider two other metrics.

— *Credit loss*: The credits used to pay for mobile hotspots to download data that cannot be sent to mobile clients due to disconnections between hotspots and clients.
— *Overhead ratio*: The ratio between the amount of traffic transmitted in the network and the actual content size.

7.2. Results

Evaluations with real mobility traces. We vary the number of mobile hotspots from 6 to 18. We first configure all mobile hotspots to use homogeneous prices P_1 and P_4. Figure 5 shows mobile client's content delay that decreases with increasing the mobile hotspot density. As an example, with $V_c = 10000$ and $\xi_c = 200$, when there are 18 mobile hotspots, 6 mobile clients spend 78.6 s to download, compared to 180.3 s when there are only 6 mobile hotspots.

We now set different price functions for mobile hotspots. Half of them employ P_1 and P_4 while the others employ P_2 and P_4. The results are presented in Figures 6(a) and 6(b) for mobile client's content delay and cost. First we take a look at $V_c = 10000$ and $\xi_c = 200$ in Figs 5 and 6(a). Compared to the previous scenario, content delay in this scenario where half of the mobile hotspots having higher price, is higher. This is because mobile clients hesitate to select more expensive mobile hotspots. This observation is inline with our analysis in Section 4, which shows Z_c^* depends on cost and D_c depends on Z_c^*. In Figures 6(a) and 6(b), with $\xi_c = 200$, content delay is much lower than that with $\xi_c = 50$, e.g., three times lower in the network with 18 mobile hotspots. This is because larger ξ_c forces the mobile clients to care less on cost but more on content transfer delay. This is also inline with our analysis in Section 4 showing that D_c is proportional to $1/\xi_c$. Indeed, Figure 6(b) shows that the cost at $\xi_c = 200$ is higher than that at $\xi_c = 50$.

We also record credit loss and overhead in the experiments. We see that under any number of mobile hotspots in the network, the maximum credit loss is merely 49.10^{-6} credits, and the maximum overhead ratio is only 1.021. Such low overhead shows the efficiency of our proposed algorithms.

Implications of chunk size. We consider the case where peer-to-peer links are the bottleneck, and demonstrate how smaller chunk size is beneficial in such environments. We set the WiMAX data rate to 1.5 Mbps, and the peer-to-peer rate to 500 Kbps. We investigate the performance on chunk size of 2 KB, 20 KB, and 40 KB. We use Random Waypoint model (because the real trace driven has low mobility) and vary the speed. 12 mobile hotspots are used in this experiment, and they are placed in a $1000 \times 1000\ m^2$ area. Figure 7 reports our results. At the chunk size of 2 KB, the credit loss is very low even in high mobility environments. For example, at 18 m/s, the loss is 0.05 credits to download 10 MB contents. The loss becomes 0.63 credits when the

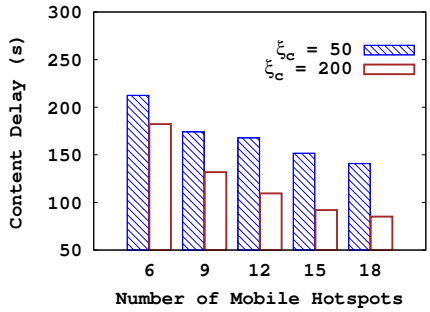


Fig. 5. Mobility simulations: Content delay under homogeneous price functions.

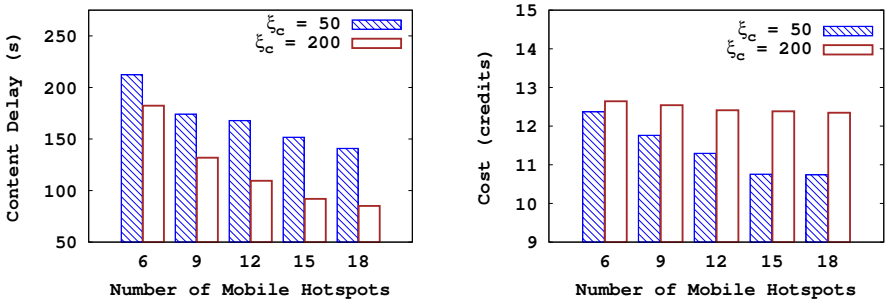


Fig. 6. Mobility simulations: (a) content delay and (b) cost - under heterogeneous price functions.

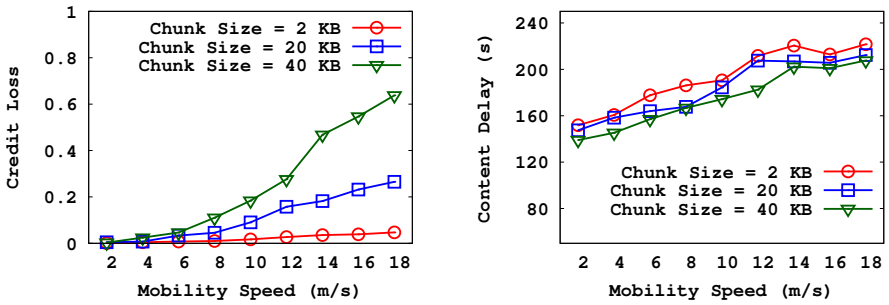


Fig. 7. Implications of chunk size: (a) monetary loss and (b) content delay.

chunk size is increased to 40 KB. The difference diminishes in low mobility environments. At 2 m/s, the credit loss is very low under any considered chunk sizes. The benefit of using large chunk size is lower network overhead due to fewer chunk reception checks. This is validated in Figure 7(b), which shows that content delay at the chunk size of 40 KB is 20 s shorter than that at the size of 2 KB. We thus recommend large chunk size in low mobility environments and small chunk size in high mobility environments.

8. PRACTICAL IMPLEMENTATION ISSUES

Several practical implementation issues need to be addressed, in order to realize broader deployments of our scheme. We first describe the concerns about business models, followed by technical challenges.

8.1. Business Models

To be sustainable, a tighter integration of the proposed scheme into cellular service providers' eco-system is needed. This will allow our scheme to be meshed with the business objectives of the telcos and service providers. This issue is perhaps not as complex as one might imagine. Cardona et al. [Cardona et al. 2014] already show that both cellular network providers and end users can benefit from data plan sharing. Moreover, today, cellular service providers, such as AT&T and Sprint, offer plans that combine the cellular access with mobile hotspot feature. Furthermore, in our proposed charging scheme, we do consider to motivate the providers to participate in our game by charging mobile clients SLA fee for additional incomes.

Note that our scheme can also be implemented in-network by cellular service providers, where the providers facilitate a marketplace in which users within partner networks can exchange unused and residual data plan minutes at low costs - this can encourage users to keep their data plan subscription, creating a win-win situation for both users and providers. In fact, the traditional economic model of cellular service providers has been emerging in the past few years [Balon and Liau 2012]. Instead of very few national providers, we see more and more *mobile virtual network operators* who buy wholesale services from cellular service providers, and sell services to mobile users. To the existing cellular service providers, our proposed marketplace is yet another (or multiple competing) smaller mobile virtual network operator(s).

8.2. Technical Challenges

There are also various technical issues that require attentions. Take energy consumption as an example. Mobile hotspot consumes energy on both cellular and peer-to-peer networks. However, the earned profit may outweigh the consumed energy, especially when mobile hotspot has sufficiently high battery level. While mobile devices may consume non-trivial energy in mobile device's discovery mode, several techniques [Sharma et al. 2009; Han et al. 2012] can be used to control this energy consumption.

Moreover, while this paper strongly focuses on *bulky data transfer*, the proposed system and algorithms are applicable to *interactive Internet access* such as web browsing, social network browsing and etc. Between these two scenarios, bulky data transfer consumes more energy and thus is more challenging to support. Therefore, our discussion thus far concentrates on bulky data transfer, despite the same mechanism also works for interactive Internet access albeit proper system parameters need to be carefully chosen and the multi-homed technique is not needed to be enabled, because mobile client does not need that high bandwidth for interactive Internet access. It should, however, be noted that the initial setup delays are not crucial in bulky data transfers, but much more important in interactive Internet access, which can be further investigated in the future.

9. CONCLUDING REMARKS

In this paper, we have developed a crowdsourcing system for mobile users to help transfer data and utilize efficiently data quotas in their data plans. In particular, we design a system and algorithms including: (i) a mobile hotspot selection method for mobile clients to choose and hire mobile hotspots for content transfers, based on a user-specified trade-off between content transfer cost and delay, and (ii) a delay-

bounded admission control method for mobile hotspots to choose transfer requests to serve with delay bounds. The system with both algorithms result in a unified and coherent solution. We have evaluated our proposed system and algorithms in an Android testbed and the Qualnet simulator. Our experimental and trace-driven simulation results show that: (i) each mobile client can select a preferred trade-off between content transfer cost and delay, (ii) mobile hotspots achieve the maximal revenue while serving all transfer requests within the promised delay bounds, and (iii) the whole system is stable under various mobility patterns and achieves better performance when mobile hotspot density is higher. The evaluation results also validate our mathematical analysis, and demonstrate the efficiency and stability of our proposed system and algorithms.

Future work. The considered marketplace is general as individual hotspots have freedom to set their own cost functions. In fact, the cost functions need not to solely depend on the content size; network conditions, battery levels, and used time durations may all be adopted by the cost functions. For example, for Web browsing, a mobile client may request for some bandwidth for a certain time duration rather than requesting for transferring a large file. Looking into the cost functions for heterogeneous applications is one of our future tasks. Our another future task is to design prediction algorithms for mobile hotspots to reserve some data plan quota and energy budget for themselves. Last, while the interference in the cellular network (among mobile hotspots and cellular base station) *is* controlled by the 3G/4G/WiMAX, the interference in the WiFi networks (among mobile clients and mobile hotspots) is not managed in our current scheme. This is partially due to the distributed nature of our decision making mechanism, but a (WiFi) interference-aware extension is among our future tasks.

REFERENCES

G. Ananthanarayanan, V. Padmanabhan, and L. Ravindranath. 2007. COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading. In *Proc. of MobiSys*. San Juan, Puerto Rico, 286–298.

Android WiFi Tether 2012. http://code.google.com/p/android-wifi-tether/. (2012).

AT&T Data Plan 2014. http://www.att.com/shop/wireless/plans/data-plans.jsp?fbid=w6awFbTp_qQ. (2014).

M. Balon and B. Liau. 2012. Mobile Virtual Network Operator. In *Proc. of International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*. Rome, Italy, 1–6.

R. Bhatia, L. Li, H. Luo, and R. Ramjee. 2006. ICAM: Integrated Cellular and Ad Hoc Multicast. *IEEE Transactions on Mobile Computing* 5, 8 (August 2006), 1004–1015.

J. Cardona, R. Stanojevic, and N. Laoutaris. 2014. Collaborative Consumption for Mobile Broadband: A Quantitative Study. In *Proc. of ACM CoNEXT*. Sydney, Australia, 307–318.

R. Chakravorty, S. Agarwal, S. Banerjee, and I. Pratt. 2005. MoB: A Mobile Bazaar for Wide-Area Wireless Services. In *Proc. of ACM MobiCom*. Istanbul, Turkey, 228–242.

China Mobile Hong Kong Subscribers Can Sell Unused Data Capacity 2013. http://www.zdnet.com/article/china-mobile-hong-kong-subscribers-can-sell-unused-data-capacity/. (2013).

Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014. Cisco report. (2014).

Dartmouth Mobility Trace 2012. http://crawdad.cs.dartmouth.edu/meta.php?name=dartmouth/outdoor. (2012).

N. Do, C. Hsu, and N. Venkatasubramanian. 2012. CrowdMAC: A Crowdsourcing System for Mobile Access. In *Proc. of ACM/IFIP/USENIX Middleware*. Montreal, Canada.

N. Do, C. Hsu, and N. Venkatasubramanian. 2013. Video Dissemination over Hybrid Cellular and Ad Hoc Networks. *IEEE/ACM Transactions on Mobile Computing* (2013).

FoxFi 2014. http://foxfi.com/. (2014).

L. Georgiadis, M. Neely, and L. Tassiulas. 2006. Resource Allocation and Cross-Layer Control in Wireless Networks. *Foundations and Trends in Networking* 1, 1 (April 2006), 1–144.

H. Han, Y. Liu, G. Shen, Y. Zhang, and Q. Li. 2012. DozyAP: Power-Efficient Wi-Fi Tethering. In *Proc. of MobiSys*. San Francisco, CA, 421–434.

G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas. 2014a. Enabling Crowd-Sourced Mobile Internet Access. In *Proc. of IEEE INFOCOM*. Toronto, Canada, 451–459.

G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas. 2014b. Incentive Mechanisms for User-Provided Networks. *IEEE Communications Magazine* 52, 9 (September 2014), 20–27.

L. Keller, A. Le, B Cici, H Seferoglu, C. Fragouli, and A. Markopoulou. 2012. MicroCast: Cooperative Video Streaming on Smartphones. In *Proc. of ACM MobiSys*. Lake District, United Kingdom, 57–70.

K. Kim and K. Shin. 2005. Improving TCP Performance over Wireless Networks with Collaborative Multi-Homed Mobile Hosts. In *Proc. of MobiSys*. Seattle, WA, 107–120.

L. Law, K. Pelechrinis, S. Krishnamurthy, and M. Faloutsos. 2010. Downlink Capacity of Hybrid Cellular Ad Hoc Networks. *IEEE Transactions on Networking* 18, 1 (February 2010), 243–256.

H. Luo, X. Meng, R. Ramjee, P. Sinha, and L. Li. 2007. The Design and Evaluation of Unified Cellular and Ad-Hoc Networks. *IEEE Transactions on Mobile Computing* 6, 9 (September 2007), 1060–1074.

H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. 2003. UCAN: a Unified Cellular and Ad-Hoc Network Architecture. In *Proc. of ACM MobiCom*. San Diego, CA, 353–367.

M. Neely. 2011. Opportunistic Scheduling with Worst Case Delay Guarantees in Single and Multi-Hop Networks. In *Proc. of IEEE INFOCOM*. Shanghai, China, 1728–1736.

Open Garden 2014. http://opengarden.com/. (2014).

PowerTutor 2012. http://ziyang.eecs.umich.edu/projects/powertutor/powertutorplus.html. (2012).

Qualnet Simulator 2014. http://www.qualnet.com. (2014).

S. Ren, Y. He, and F. Xu. 2012. Provably-Efficient Job Scheduling for Energy and Fairness in Geographically Distributed Data Centers. In *Proc. of IEEE ICDCS*. Macau, China, 22–31.

A. Sharma, V. Navda, R. Ramjee, V. Padmanabhan, and E. Belding. 2009. Cool-Tether: Energy Efficient On-the-Fly WiFi Hot-Spots Using Mobile Phones. In *Proc. of ACM CoNEXT*. Rome, Italy, 109–120.

Sprint Data Plan 2014. http://shop.sprint.com/mysprint/shop/plan/plan_wall.jsp?tabId=pt_data_plans_tab&flow=AAL&planFamilyType=null. (2014).

T-Mobile Data Plan 2014. http://www.t-mobile.com/shop/plans/mobile-broadband-plans.aspx. (2014).

Traffic and Market Report 2011. http://hugin.info/1061/R/1561267/483187.pdf. (2011).

Wi-Fi Certified Wi-Fi Direct 2013. http://www.wi-fi.org/Wi-Fi_Direct.php. (2013).

T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu, and S. Lu. 2014. INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network. In *Proc. of IEEE INFOCOM*. Toronto, Canada, 1545–1553.

Received November 2015; revised March 2016; accepted May 2016

A. APPENDIX

A.1. Proof of Lemma 1

Lemma 1 is proved as follows. The one-step Lyapunov for mobile hotspot h is:

$$\begin{aligned}\triangle(\phi_h(t)) &= \mathbb{E}\{L(\phi_h(t+1)) - L(\phi_h(t))|\phi_h(t)\} \\ &= \frac{1}{2}\mathbb{E}\left\{Q_h^2(t+1) - Q_h^2(t)|\phi_h(t)\right\} \\ &\quad + \frac{1}{2}\mathbb{E}\left\{X_h^2(t+1) - X_h^2(t)|\phi_h(t)\right\} \\ &\quad + \frac{1}{2}\mathbb{E}\left\{Y_h^2(t+1) - Y_h^2(t)|\phi_h(t)\right\}\end{aligned} \tag{34}$$

Apply (1), (4) and (5), we have the following expressions:

$$\begin{aligned}Q_h^2(t+1) - Q_h^2(t) = \frac{1}{2}(\mu_h^o(t) - \mu_h^i(t))^2 \\ - Q_h(t)(\mu_h^o(t) - \mu_h^i(t))\end{aligned} \tag{35}$$

$$\begin{aligned}X_h^2(t+1) - X_h^2(t) = \frac{1}{2}(\mu_h^i(t) + g_h(t) - a_h(t))^2 \\ - X_h(t)(g_h(t) + \mu_h^i(t) - a_h(t))\end{aligned} \tag{36}$$

$$\begin{aligned}Y_h^2(t+1) - Y_h^2(t) = \frac{1}{2}(\mu_h^i(t) + g_h(t) - a_h(t) - \delta_h)^2 \\ - Y_h(t)(g_h(t) + \mu_h^i(t) - a_h(t) - \delta_h)\end{aligned} \tag{37}$$

Let's define a constant β_h for h as below such that $\beta_h(t) \leq \beta_h^*$ at any time slot:

$$\begin{aligned}\beta_h(t) &\triangleq \frac{1}{2}[(\mu_h^o(t) - \mu_h^i(t))^2 + (\mu_h^i(t) + g_h(t) - a_h(t))^2 \\ &\quad + (\mu_h^i(t) + g_h(t) - a_h(t) - \delta_h)^2] \\ &\leq \frac{1}{2}[\max(\omega^*, \mu^{i*})^2 + \max(g^* + \mu^{i*}, A^*)^2 \\ &\quad + \max(g^* + \mu^{i*}, A^* + \delta_h)^2] \triangleq \beta_h\end{aligned} \tag{38}$$

By inserting (38) into (34), we have Lemma 1 proved:

$$\begin{aligned}\triangle(\phi_h(t)) \leq \beta_h &- \mathbb{E}\{[X_h(t) + Y_h(t)]g_h(t)|\phi_h(t)\} \\ &+ \mathbb{E}\{Y_h(t)\delta_h|\phi_h(t)\} \\ &+ \mathbb{E}\{[X_h(t) + Y_h(t)]a_h(t)|\phi_h(t)\} \\ &- \mathbb{E}\{[X_h(t) + Y_h(t) - Q_h(t)]\mu_h^i(t)|\phi_p(t)\} \\ &- \mathbb{E}\{Q_h(t)\mu_h^o(t)|\phi_h(t)\}\end{aligned} \tag{39}$$

A.2. Proof of Theorem 1

We prove that virtual queue $X_h(t)$ at mobile hotspot h is bounded by:

$$X_h(t) \leq \frac{V_h\rho_h^*}{2} + A^* \tag{40}$$

At $t = 0$, of course $X_h(t) = 0$, which satisfies (40). We assume that at time slot $t > 0$, $X_h(t)$ satisfies (40). We prove that at $t + 1$, $X_h(t + 1)$ also satisfies (40).

In a service round, it is easy to see that $Y_h(t) = X_h(t)+t\delta_h$ because $Y_h(t)$ has the same output rate as $X_h(t)$ but an additional input rate δ_h in every time slot. We consider two cases at $t > 0$: $X_h(t) > \frac{V_h\rho_h^*}{2}$ and $X_h(t) \leq \frac{V_h\rho_h^*}{2}$.

(1) Case 1: $X_h(t) > \frac{V_h\rho_h^*}{2}$. Since $Y_h(t) = X_h(t) + t\delta_h$, $X_h(t) + Y_h(t) > 2X_h(t) > 2\frac{V_h\rho_h^*}{2} = V_h\rho_h^*$. Following the first procedure in Algorithm 1, Request Admission, there would be no request admitted to serve by h in time slot t. Thus, $X_h(t + 1) = X_h(t) \leq \frac{V_h\rho_h^*}{2} + A^*$.

(2) Case 2: $X_h(t) \leq \frac{V_h\rho_h^*}{2}$. It could be that $X_h(t) + Y_h(t) \leq V_h\rho_h^*$. Thus, h can admit some requests in t. The maximum data amount h can admit in a time slot is A^*. So, $X_h(t + 1) \leq X_h(t) + A^* \leq \frac{V_h\rho_h^*}{2} + A^*$.

The theorem is completely proved here.

A.3. Proof of Theorem 2 and Corollary 1

We first prove Theorem 2 by showing $X_h^b(t)$, which indicates the largest queue backlog $X_h(t)$ where mobile hotspot h is still able to admit new requests, decreases over time.

$$X_p^b(t) = \frac{V_h\rho_h^* - t\delta_h}{2} \tag{41}$$

Without loss of generality, set $t = 0$ at the beginning of a service round. Following the Request Admission procedure in Algorithm 1, we see that h can only admit new requests if

$$X_h(t) + Y_h(t) \leq V_h\rho_h \leq V_h\rho_h^* \tag{42}$$

For $t > 0$, $Y_h(t) = X_h(t) + t\delta_h$. So,

$$X_h(t) + Y_h(t) = 2X_h(t) + t\delta_h \leq V_h\rho_h^* \tag{43}$$

This leads to $X_h(t) \leq \frac{V_h\rho_h^*-t\delta_h}{2}$. This completes the proof for Theorem 2.

Now, we verify when h will stop receiving new requests. The latest time slot t_h^s when h no longer accepts new requests is at $X_h^b(t) = \frac{V_h\rho_h^*-t\delta_h}{2} = 0$. This leads to Corollary 1.

$$t_h^s = \frac{V_h\rho_h^*}{\delta_h} \tag{44}$$

For $t > t_h^s$, procedure Request Admission in Algorithm 1 does not allow h to receive any new requests no matter how small $X_h(t)$ is because $Y_h(t) > V_h\rho_h^*$. $Y_h(t)$ only goes back to 0 when $X_h(t) = 0$ and $Q_h(t) = 0$; i.e., when all admitted data is completely transferred to the Internet.

A.4. Proof of Theorem 3

To show a worst case bound on real queue $Q_h(t)$

$$Q_h(t) \leq 5\frac{V_h\rho_h^*}{2} + 3A^* + \mu^{i*} = Q_h^* \quad \forall t \tag{45}$$

we start by showing in a service round, the worst cast bound on $Q_h(t)$ at time slot t_h^s, which is:

$$Q_h(t) \leq 2V_h\rho_h^* + 2A^* + \mu^{i*} \quad \forall t \in [0, t_h^s] \tag{46}$$

If $t_h^s = 0$, expression (46) is true because $Q_h(t) = 0$. Now consider $t_h^s > 0$. In a service round, at any time slot, according to Theorem 1:

$$X_h(t) \leq \frac{V_h \rho_h^*}{2} + A^* = X_h^* \quad \forall t \tag{47}$$

During the service round, because $Y_h(t) = X_h(t) + t\delta_h$, we have:

$$X_h(t) + Y_h(t) \leq V_h \rho_h^* + 2A^* + t\delta_h \quad \forall t \tag{48}$$

Before $t = t_h^r$, it is easy to see that

$$\begin{aligned} X_h(t) + Y_h(t) &\leq V_h \rho_h^* + 2A^* + t_h^s \delta_h \\ &= V_h \rho_h^* + 2A^* + V_h \rho_h^* = 2V_h \rho_h^* + 2A^* \end{aligned} \tag{49}$$

Assume that (46) is true for $Q_h(t)$ at $t \in [0, t_h^s)$. We consider two cases at $0 \leq t < t_h^s$: $Q_h(t) > 2V_h \rho_h^* + 2A_h^*$ and $Q_h(t) \leq 2V_h \rho_h^* + 2A^*$.

(1) Case 1: $Q_h(t) > 2V_h \rho_h^* + 2A^*$. Note that data is scheduled to be sent by procedure Congestion Control in Algorithm 1 into $Q_h(t)$ if and only if $X_h(t) + Y_h(t) \geq Q_h(t)$. Because $Q_h(t) > 2V_h \rho_h^* + 2A^* \geq X_h(t) + Y_h(t)$, there would be no incoming data traffic to $Q_h(t)$ in time slot t. Thus, $Q_h(t+1) = Q_h(t) \leq 2V_h \rho_h^* + 2A^* + \mu^{i*}$.
(2) Case 2: $Q_h(t) \leq 2V_h \rho_h^* + 2A^*$. There is a possibility that $X_h(t) + Y_h(t) \geq Q_h(t)$. So, there is no stop command issued from h to prevent incoming data to h. Note that the maximum data amount sent to h in a time slot is μ^{i*}. So, $Q_h(t+1) \leq Q_h(t) + \mu^{i*} \leq 2V_h \rho_h^* + 2A^* + \mu^{i*}$.

Therefore, for $0 \leq t < t_h^s$, $Q_h(t+1)$ still satisfies (46). We complete the proof for (46).

At $t > t_h^s$, h no longer admits more data until both $X_h(t)$ and $Q_h(t)$ reach 0. Thus, the total data amount transferred to $Q_h(t)$ from $t = t_h^s + 1$ to the end of the service round is at most X_h^*. Therefore:

$$\begin{aligned} Q_p(t) &\leq 2V_h \rho_h^* + 2A^* + \mu^{i*} + X_h* \\ &= 5\frac{V_h \rho_h^*}{2} + 3A^* + \mu^{i*} = Q_h^* \quad \forall t \end{aligned} \tag{50}$$

We complete the proof for Theorem 3 here.

A.5. Proof of Lemma 2

We define the worst case delay bound for a request served by mobile hotspot h as T_h. From the analysis in Theorem 2 and Corollary 1, we can see that T_h is bounded by the longest duration of a service round because at the end of the round, $X_h(t) = Q_h(t) = 0$ (i.e., all data agreed to serve by h is completely transferred to destination). We consider the longest duration of a round includes at most two time periods: t_h^s and t_h^r, where t_h^s is the longest time h can still admit more requests, and t_h^r indicates the longest time h spends on delivering the data remaining in $X_h(t)$ and $Q_h(t)$ after t_h^s to destination. Therefore, we can say $T_h = t_h^s + t_h^r$. While t_h^s is already known by Corollary 1, we now have to determine t_h^r.

It is easy to see that t_h^r is bounded by a delay for transferring all data in $\bar{Q}_h^*$ (the worst case bound of $Q_h(t)$ right after t_h^s) and X_h^* to the Internet in a circumstance that h does not accept any more requests (i.e., right after t_h^s), and all virtual and real queues reach their maximum value. Clearly, $\bar{Q}_h^* = 2X_h^* + t_h^s \delta_h = 2V_h \rho_h^* + 2A^* + \mu^{i*}$ as shown in the proof of Theorem 3. We now go find t_h^r's bound.

Right after t_h^s, in the worst case, all queues reach their maximum backlog, and data is blocked from being sent to h due to Congestion Control at 1 because $Q_c(t) > X_c(t) + Y_c(t)$. Let t_h^+ be the duration of a period starting right after t_h^s to time slot t where

$Q_h(t) \leq X_h(t) + Y_h(t)$ for the first time. We can see that $t_h^+ = \frac{\bar{Q}_h^* - 2X_h^* - t_h^s\delta_h}{\mu_p^o + \delta_p} = \frac{\mu^{i*}}{\mu_h^o + \delta_h}$ from $Q_h(t) = \bar{Q}_h^* - \mu_h^o t_h^+$ and $X_h(t) + Y_h(t) = 2X_h^* + t_h^s\delta_h + t_h^+\delta_h$.

We denote t_h^- as a delay to deliver all remaining data after $t_h^s + t_h^+$ to the Internet, i.e., $t_h^r = t_h^+ + t_h^-$. We now determine t_h^- to complete calculating t_h^r. Right after $t_h^s + t_h^+$, queue backlogs $Q_h(t)$ and $X_h(t)$ are $\hat{Q}_h^* = \bar{Q}_h^* - t_h^+\mu_h^o$ and X_h^*, respectively. Note that at that instance, $Q_h(t) \leq X_h(t) + Y_h(t)$. We consider two cases:

(1) Case 1: $\delta_h \geq 3\mu_h^i - \mu_h^o$. There will be no 'stop' commands issued by mobile hotspot h according to Congestion Control procedure. It is because the increase of $X_h(t)$ and $Y_h(t)$, $\delta_h - 2\mu_h^i$, is larger or equal to the increase of $Q_h(t)$, $\mu_h^i - \mu_h^o$. There are two subcases: $\mu_h^i \geq \mu_h^o$ and $\mu_h^i < \mu_h^o$.
   — $\mu_h^i \geq \mu_h^o$: Bottleneck happens at h's data connection. Thus, it takes $t_h^- = \frac{\hat{Q}_h^* + X_h^*}{\mu_h^i}$ slots to send all data admitted to its virtual queue and stored in its real queue to destination.
   — $\mu_h^i < \mu_h^o$: Bottleneck is at incoming data to h. Let t be the time h spends on sending all data stored in its real queue to destination. We have $t\mu_h^o = \hat{Q}_h^* + t\mu_h^i$. Thus, $t = \frac{\hat{Q}_h^*}{\mu_h^o - \mu_h^i}$. The remaining data admitted in the virtual queue will be sent in $\frac{X_h^* - t\mu_h^i}{\mu_h^i}$. The total time would be $t_h^- = t + \frac{X_h^* - t\mu_h^i}{\mu_h^i} = \frac{X_h^*}{\mu_h^i}$.

(2) Case 2: $\delta_h < 3\mu_h^i - \mu_h^o$. There could be 'stop' commands issued from h's Congestion Control procedure. Let's define t_0 be time slot when $Q_p(t)$ reaches 0 for the first time after $t_h^s + t_h^+$ in a service round. Let $\hat{X}_h$ denote the amount of data transmitted to the real queue from after $t_h^s + t_h^+$ to t_0. We also examine two subcases:
   — $\mu_h^i \geq \mu_h^o$: We can see that $t_0 = \frac{\hat{Q}_h^* + \hat{X}_h}{\mu_h^o}$. After t_0, if all admitted data is sent to h, i.e., $\hat{X}_h = X_h^*$, it is obviously that $t_h^- = \frac{\hat{Q}_h^* + X_h^*}{\mu_h^o}$. Otherwise, the remaining admitted data is sent to the real queue and then delivered out of h. This would take $\frac{X_h^* - \hat{X}_h}{\mu_h^o}$ to transfer the remaining incoming data. The total time is thus $t_h^- = \frac{\hat{Q}_h^* + \hat{X}_h}{\mu_h^o} + \frac{X_h^* - \hat{X}_h}{\mu_h^o} = \frac{\hat{Q}_h^* + X_h^*}{\mu_h^o}$.
   — $\mu_h^i < \mu_h^o$: The total time to deliver all admitted data is $t_h^- = \frac{\hat{Q}_h^* + \hat{X}_h}{\mu_h^o} + \frac{X_h^* - \hat{X}_h}{\mu_h^i} \leq \frac{\hat{Q}_h^*}{\mu_h^o} + \frac{X_h^*}{\mu_h^i}$.

We complete our proof for the lemma.

A.6. Proof of Lemma 3

The one-step Lyapunov for mobile client c defined in (27) is expanded as follows.

$$\begin{aligned}\triangle(\phi_c(t)) &= \mathbb{E}\{L(\phi_c(t+1)) - L(\phi_c(t))|\phi_c(t)\} \\ &= \frac{1}{2}\mathbb{E}\{Z_c^2(t+1) - Z_c^2(t)|\phi_c(t)\} \\ &= \frac{1}{2}\mathbb{E}\{(\lambda_c(t) - m_c(t) - \xi_c)^2 \\ &\qquad - 2[\lambda_c(t) - m_c(t) - \xi_c]Z_c(t)|\phi_c(t)\}\end{aligned} \tag{51}$$

Let us define α_c as a constant:

$$\begin{aligned}\alpha_c(t) &\triangleq \frac{1}{2}(\lambda_c(t) - m_c(t) - \xi_c)^2 \\ &\leq \max(S, m^* + \xi_c)^2 \triangleq \alpha_c\end{aligned} \tag{52}$$

The drift (51) becomes:

$$\triangle(\phi_c(t)) \leq \alpha_c - \mathbb{E}\{[\lambda_c(t) - m_c(t) - \xi_c]Z_c(t)|\phi_c(t)\} \tag{53}$$

A.7. Proof of Theorem 4

Following the definition of $Z_c(t)$, we have the following constraint at any time slot t:

$$Z_c(t+1) \geq Z_c(t) - \lambda_c(t) + m_c(t) + \xi_c \tag{54}$$

Let's denote T as the worst case delay c takes for getting the whole content admitted for transfer by mobile hotspots. At any time slot from t to $t+T+1$, using (54):

$$\begin{aligned}&Z_c(t+1) \geq Z_c(t) - \lambda_c(t) + m_c(t) + \xi_c \\ &Z_c(t+2) \geq Z_c(t+1) - \lambda_c(t+1) + m_c(t+1) + \xi_c \\ &... \\ &Z_c(t+T+1) \geq Z_c(t+T) - \lambda_c(t+T) + m_c(t+T) + \xi_c\end{aligned} \tag{55}$$

Sum up all the inequations above, we have:

$$Z_c(t+T+1) \geq Z_c(t) - \sum_{t}^{t+T}[\lambda_c(t) + m_c(t)] + T\xi_c \tag{56}$$

Note that $\sum_t^{t+T}[\lambda_c(t) + m_c(t)] = f_c$.

Let's define Z_c^* as a bound on $Z_c(t)$ at any time slot. We will show how to get Z_c^* later in this section. It follows that:

$$Z_c(t) - f_c + T\xi_c \leq Z_c^* \tag{57}$$

Therefore, $T \leq \frac{f_c + Z_c^*}{\xi_c} \leq \frac{F^* + Z_c^*}{\xi_c}$.

At $t+T+1$, all segments at c are admitted by mobile hotspots for transfer. Each segment admitted by a mobile hotspot h would take at most T_h to be transferred from the time it is admitted to the Internet. Thus, the worst case content delay to transfer the whole content to the Internet would be:

$$D_c = T + T_h^* \leq \frac{F^* + Z_c^*}{\xi_c} + T_h^* \tag{58}$$

where $T_h^* = \max_{h \in H_c(t) \forall t}(T_h)$, known as the maximum request delay among mobile hotspots serving c.

Now we show a bound on virtual queue $Z_c(t)$ at mobile client c. $Z_c(t)$ is bounded by:

$$Z_c(t) \leq \max[V_c\eta_c^* + \xi_c + m^*, F^*] = Z_c^* \tag{59}$$

Let's start from $t = 0$. At $t = 0$, $Z_c(t) = f_c \leq F^*$. So, (59) is satisfied. Let's consider $F^* \leq V_c\eta_c^* + \xi_c + m^*$. Thus we would prove at any $t > 0$, $Z_c(t) \leq V_c\eta_c^* + \xi_c + m^*$. We assume that at time slot $t > 0$, $Z_c(t)$ satisfies (59), and we will show that at $t+1$, $Z_c(t+1)$ also satisfies the same thing. An assumption is that at any time slot, there is at least one mobile hotspot willing to serve request from c and $\xi_c \leq S - m^*$. There are two cases happening at $t+1$.

(1) Case 1: Mobile client c does not select any mobile hotspot at $t+1$. The reason c does this is because according to Algorithm 2, $Z_c(t+1)\lambda_c(t) < V_c C_c(t)$ for all mobile hotspots in $H_c(t)$. Therefore, $Z_c(t+1) < V_c \frac{C_c(t)}{\lambda_c(t)} \leq V_c \eta_c^*$. At the end of time slot $t+1$, at most $\xi_c + m^*$ is added to $Z_c(t+1)$ in this case. So, at any instant in slot $t+1$, the virtual queue is bounded by $V_c\eta_c^* + \xi_c + m^*$.
(2) Case 2: Mobile client c selects a mobile hotspot to transfer the current segment at $t+1$. We thus have $Z_c(t+1) \leq Z_c(t) \leq V_c\eta_c^* + \xi_c + m^*$.

If $F^* > V_c\eta_c^* + \xi_c + m^*$, it is easy to see that there is no time slot when $Z_c(t) > F^*$. We have $Z_c(0) \leq F^*$. If $Z_c(0) > V_c\eta_c^* + \xi_c + m^*$, mobile client c selects one mobile hotspot to transfer data because $Z_c(0)\lambda_c(0) > V_c\eta_c^*\lambda_c(0) = V_c C_c(0)\eta_c^* \frac{\lambda_c(0)}{C_c(0)} \geq V_c C_c(0)\eta_c^* \frac{1}{\eta_c^*} = V_c C_c(0)$. This leads to $Z_c(1) \leq Z_c(0) \leq F^*$ because mobile client will send data out. Similarly, for any t where $Z_c(t) > V_c\eta_c^* + \xi_c + m^*$, $Z_c(t+1) \leq Z_c(t) \leq F^*$. For any t where $Z_c(t) \leq V_c\eta_c^* + \xi_c + m^*$, we follow the same proof presented above to show $Z_c(t+1) \leq V_c\eta_c^* + \xi_c + m^* < F^*$.

A.8. Proof of Theorem 5

We start the proof by looking at the one-step Lyapunov drift (60) for mobile client d:

$$\begin{aligned} \triangle(\phi_c(t)) &= \mathbb{E}\{L(\phi_c(t+1)) - L(\phi_c(t))\} \\ &\leq \alpha_c - \mathbb{E}\{[\lambda_c - m_c(t) - \xi_c]Z_c(t)|\phi_c(t)\} \end{aligned} \tag{60}$$

Let's define $\bar{C}_c^*$ as the minimum long term cost of the mobile hotspot selection problem achieved by some stationary randomized algorithm. We have the following constraint:

$$\begin{aligned} &\mathbb{E}\{L(\phi_c(t+1)) - L(\phi_c(t))\} + V_c\mathbb{E}\{C_c(t)|\phi_c(t)\} \\ &\leq \alpha_c - \mathbb{E}\{[\lambda_c(t) - m_c(t) - \xi_c]Z_c(t)|\phi_c(t)\} + V\bar{C}_c^* \end{aligned} \tag{61}$$

Let's define $\mathbb{E}\{\lambda_c(t) - m_c(t) - \xi_c\} = \epsilon_c$. Taking expectations of the above inequality and applying the law of iterated expectations with the distribution of $\phi_c(t)$ result in:

$$\begin{aligned} &\mathbb{E}\{L(\phi_c(t+1) - L(\phi_c(t))\} + V_c\mathbb{E}\{C_c(t)\} \\ &\leq \alpha_c - \epsilon_c\mathbb{E}\{Z_c(t)\} + V_c C_c^* \end{aligned} \tag{62}$$

Sum all time slots t from 0 to $T-1$, and then divide by V_cT:

$$\begin{aligned} &\frac{1}{V_cT}\mathbb{E}\{L(\phi_c(T)) - L(\phi_c(0))\} + \frac{1}{T}\sum_{\tau=0}^{T-1}\mathbb{E}\{C_c(\tau)\} \\ &\leq \frac{\alpha_c}{V_c} - \frac{\epsilon_c}{V_cT}\sum_{\tau=0}^{T-1}\mathbb{E}\{Z_c(\tau)\} + C_c^* \end{aligned} \tag{63}$$

Let's rearrange the terms, and use the facts that $L(\phi_c(T)) \geq 0$, $L(\phi_c(0)) = f_c \leq F^*$, $\epsilon_c \geq 0$ by selecting $\xi_c \leq \mathbb{E}\{\lambda_c(t) - m_c(t)\}$, and $Z_c(\tau) > 0$, we have:

$$\frac{1}{T}\sum_{\tau=0}^{T-1}\mathbb{E}\{C_c(\tau)\} \leq \frac{\alpha_c}{V_c} + \frac{F^*}{V_cT} + C_c^* \tag{64}$$

As T goes to ∞, the above inequality becomes:

$$\bar{C}_c = \lim_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1}\mathbb{E}\{C_c(\tau)\} \leq C_c^* + \frac{\alpha_c}{V_c} \tag{65}$$

This confirms our theorem's correctness.