DEPLOYMENT, MANAGEMENT, AND ACCESS ACQUISITION OF SMALL-CELL BASED NETWORKS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of the Ohio State University

By

Zhixue Lu, M.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2014

Dissertation Committee:

Prasun Sinha, Advisor

Dong Xuan

Chunyi Peng

ABSTRACT

The increasing popularity of handheld devices, such as smartphones and tablets, has made the demand for high-data-rate wireless access more urgent than ever. Due to the scarcity of wireless spectrum and the limitation of physical size of handheld devices, *Small-Cell* based solutions are being widely adopted [82].

The concept of small cells encompasses WiFi access points, femtocells and microcells, etc. Small cells can be categorized as *Managed* and *Unmanaged*. While managed small cells allow access to all users, it is cost prohibitive for large-scale blanket deployment. On the other hand, unmanaged small cells are cost-efficient to expand, but they only provide service to dedicated users. Therefore, the issues of *budgeted deployment* and *resource allocation* for managed small cells and *access acquisition* of unmanaged small cells are critical and challenging to study. This dissertation studies those problems and makes the following contributions:

1. Sparse Deployment of Large Scale Managed Small Cell Networks. This dissertation first examines the deployment problem in large scale managed small cell networks. It presents a new metric, called *Contact Opportunity*, as a characterization of roadside WiFi networks. Contact opportunity measures the fraction of distance or time that a mobile user is in contact with some APs when moving on a certain trajectory. Our objective is to find a deployment that ensures a required

level of contact opportunity with the minimum cost. This is the first work that addresses the challenges in achieving a *sparse* wireless infrastructure that provides QoS assurance to mobile users *in the face of uncertainty*.

2. Resource Management in Managed Dense Small Cell Networks. Fast expansion of small cells makes the problem of resource management in urban dense networks challenging. To achieve both high throughput and fairness among users, the second part of this dissertation studies two dynamic resource allocation problems: 1) **Achieve** $max - min$ **fairness of throughput to mobile users in multiple collision domains.** We propose bounded centralized and distributed approximation algorithms for allocating resource in femtocell networks across multiple collision domains. 2) **Achieve QoE** $max-min$ **fairness to video streaming users in a single collision domain.** We extend the QoS (throughput) fairness metric to QoE fairness by solving the problem of bandwidth allocation in a single collision domain.

3. Incentive Mechanism Design for Access Acquisition of Unmanaged Small Cells. The last part of this dissertation considers the problem of utilizing unmanaged small cells for data offloading service. We propose a reverse auction scheme to incentivize the owners to make their services available to the service providers. This dissertation introduces the notions of *Perceived Valuation*, *Partial Truthfulness* and *Imprecision Loss*, which together characterize the *quality* of truthful auctions while considering imprecision in estimation of the true valuations. This is a first such notion. It then proposes EasyBid, a novel mechanism with heuristic algorithms which enable conducting truthful auctions in the presence of imprecise valuations.

This dissertation is lovingly dedicated to my family

ACKNOWLEDGMENTS

It is with immense gratitude that I acknowledge the guidance, support and encouragement of the great people around me, to only some of whom it is possible to give particular mention here. I am indebted to them for helping me get this far.

This dissertation would not have been possible without the advice, support and patience of my advisor, Prof. Prasun Sinha. As a great advisor, he always gave insightful guidance and inspiring comments to me and my work. His wisdom, acute sense, and unsurpassed knowledge always impressed me. I will always appreciate his generous and continuous support in the last nearly six years.

I consider it an honor to work with Prof. R. Srikant in the University of Illinois at Urbana-Champaign. We have worked together on one of my recent papers. It has always been rewarding to discuss research with Prof. Srikant.

I also owe a lot to my great teachers in both the CSE department and ISE department at OSU. Their patient instructions are invaluable in helping me to set up a solid background in both engineering and scientific fields.

I would like to thank my colleagues in Prof. Sinha's research group: Zizhan Zheng, Dong Li, Tarun Bansal, Wenjie Zhou, and Yousi Zheng. They are great and helpful colleagues and friends. We had great time in discussing ideas, conducting experiments, and preparing for submissions. I also wish to thank my friends Hong Sun, Enhua Tan, Rubao Li, Zhizhou Li, Wenjie Lin, Jin He, Bo Chen and many others at OSU. I enjoyed those wonderful time spent with them.

Finally, I would like to thank my family. I cannot find words to express my gratitude to my beloved parents, Bangxing Lu and Xiufen Zhai. They were always there when I was in need of warmth and comfort. Their support, encouragement, and constant love have sustained me throughout my life. I am thankful to my wife Shuai Chang and our amazing kids. I'd never be where I am today without their constant love and support in me. I am also thankful to my brothers, Zhijun Lu, Zhimin Lu and Bing Lu and my nephews and nieces, for their continuous support in the past years.

VITA

January 2, 1982	Born - Hebei, China
2005	B.S. Information Management, Peking University, China
2008	M.S. Computer Science, Peking University, China
2013	M.S. Department of Computer Science and Engineering, Ohio State University, Columbus, OH
2008-present	Ph.D. Department of Computer Science and Engineering, Ohio State University, Columbus, OH

PUBLICATIONS

Zizhan Zheng, Zhixue Lu, Prasun Sinha and Santosh Kumar. "Ensuring Predictable Contact Opportunity for Scalable Vehicular Internet Access On the Go". *IEEE/ACM Transactions on Networking (TON)*, To Appear

Zhixue Lu, Prasun Sinha and R. Srikant. "EasyBid: Enabling Cellular Offloading via Small Players". *Proc. of IEEE INFOCOM 2014*, Toronto, Canada, Apr 2014

Dong Li, Zhixue Lu, Tarun Bansal, Erik Schilling and Prasun Sinha. "ForeSight: Mapping Vehicles in Visual Domain and Electronic Domain". *Proc. of IEEE INFOCOM 2014*, Toronto, Canada, Apr 2014

Zhixue Lu, Tarun Bansal and Prasun Sinha. "Achieving User-Level Fairness in Open-Access Femtocell based Architecture". *IEEE Trans. on Mobile Computing (TMC)*, 12.10 (2013): 1943-1954

Dong Li (co-primary), Tarun Bansal (co-primary), Zhixue Lu (co-primary) and Prasun Sinha. "MARVEL: Multiple Antenna based Relative Vehicle Localizer". *Proc. of ACM MobiCom*, Istanbul, Turkey, Aug 2012

Dong Li (co-primary), Tarun Bansal (co-primary), Zhixue Lu (co-primary) and Prasun Sinha. "Demo Abstract: MARVEL: Multiple Antenna based Relative Vehicle Localizer". *Proc. of ACM MobiCom*, Istanbul, Turkey, Aug 2012

Zizhan Zheng, Zhixue Lu, Prasun Sinha and Santosh Kumar. "Maximizing the Contact Opportunity for Vehicular Internet Access ". *Proc. of IEEE INFOCOM*, San Diego, March 2010

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Specialization: Networking

Studies in	Wireless Network	Prof. Prasun Sinha
		Prof. Kannan Srinivasan
		Prof. Ness B. Shroff
		Prof. Anish Arora
	Theory and Algorithms	Prof. Tamal Krishna Dey
		Prof. Marc E. Posner
		Prof. Suvrajeet Sen
	Data Mining	Prof. Srinivasan Parthasarathy
		Dr. Krish Krishnasamy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

INTRODUCTION

1.1 Background

The increasing popularity of handheld devices, such as smartphones and tablets, has made the demand for high-data-rate wireless access more urgent than ever. According to [43], cellular data traffic has been doubling every year. Due to the explosive growth of cellular data traffic, increasing system capacity has become one of the most critical challenges for wireless service providers (WSPs).

New communication techniques such as MIMO, which can provide higher data rates, have been widely adopted in the next generation wireless networks. However, given the scarcity of wireless spectrum and the limitation of physical size of handheld devices, *Small Cell* based solutions have been considered most promising [82]. Small cells are low-powered radio access nodes that operate in licensed or unlicensed spectrum that have a range of 10 meters to 1 or 2 kilometers, compared to a mobile macrocell which might have a range of a few tens of kilometers [10]. Most small cells are designed to make efficient use of radio spectrum and support data offloading. Small cells encompass WiFi access points, femtocells and microcells, etc.

The attempt to deploy commercial WiFi hotspots dates back to 1990s [4]. Since then, tremendous effort has been made by WSPs to the deployment. WiFi hotspots have been rapidly mushrooming in every city. They either operate independently as

a competitive way of data access, or act as a complementary service and help offload the overburdened cellular networks [15]. Although large deployments of WLANs can be used to provide high data-rate services over large areas, the cost becomes prohibitive due to the sheer number of access-points (APs) required. In addition to the deployment cost, the maintenance and management complexity has led to abandonment or scaling back of several WLAN projects from San Francisco to Philadelphia [14].

Microcells, Femtocells and their counterparts (Picocells, Metrocells, etc.) are representatives of small cells on the licensed spectrum. A microcell is a cell in a mobile phone network served by a low power cellular base station (tower), covering a limited area such as a mall, a hotel, or a transportation hub [7]. Femtocell is a recent emerging concept. Femtocells are low-power, easy-to-install, indoor or outdoor cellular base stations that interact with the cellular backbone network via the broadband Internet connection. New characteristics, especially the plug-and-play, self-organizing and self-managing properties, make them especially easy to install. Microcells are traditionally deployed and managed by WSPs, while some Femtocells could be owned by the property owners, making some of them not accessible to other users.

1.2 Problems to Study

From the WSP's perspective, small cells can be categorized as *Managed* and *Unmanaged*. While managed small cells are more intriguing in that they allow access by all users (of the WSP's), it is cost prohibitive for large-scale blanket deployment. On the other hand, unmanaged small cells (individual or third-party owned WiFi hotspots and femtocells) are cost-efficient to expand, without requiring power supply, backhaul and real estate of WSPs. However, they only provide service to dedicated users while increasing the complexity of managing the wireless spectrum. Therefore, the issues

of *budgeted deployment* and *resource allocation* for managed small cells and *access acquisition* of unmanaged small cells are critical and challenging to study. Toward this, this dissertation studies the following problems:

- **Sparse Deployment of Large Scale Managed Small Cell Networks.** This dissertation first examines the deployment problem in large scale managed small cell networks. The objective is to provide throughput assurance to mobile users with minimum cost.

- **Resource Management in Managed Dense Small Cell Networks.** This dissertation then studies the resource allocation problem in dense small cell based networks in urban area. It considers two problems: 1) Achieve $max-min$ fairness of throughput to mobile users in multiple collision domains. 2) Achieve QoE $max - min$ fairness to video streaming users in a single collision domain.

- **Incentive Mechanism Design for Access Acquisition of Unmanaged Small Cells.** This dissertation finally proposes a truthful auction mechanism to incentivize small parties: the private and business owners of WiFi hotspots or femtocells to provide cellular data offloading service.

1.3 Contributions of Dissertation

Sparse Deployment of Large Scale Managed Small Cell Networks. To provide guaranteed performance to mobile users, we present a new metric, called *Contact Opportunity*, as a characterization of a roadside WiFi network. Informally, the contact opportunity for a given deployment measures the fraction of distance or time that a mobile user is in contact with some APs when moving through a certain trajectory. Such a metric is closely related to the quality of data service that a mobile user might experience while driving through the system. Our objective is to find a

deployment that ensures a required level of contact opportunity with the minimum cost. This is the first work that addresses the challenges in achieving a *sparse* wireless infrastructure that provides QoS assurance to mobile users *in the face of uncertainty*. Our contributions are three-fold:

- We present a metric, called Contact Opportunity, as a characterization of roadside WiFi deployment, which is closely related to the quality of data service that a mobile user might experience when driving through the network.
- We design an efficient deployment method that ensures a required level of contact opportunity at a minimum cost by utilizing submodular optimization techniques.
- We extend the concept of contact opportunity and the deployment techniques to average throughput by taking various dynamic elements into account, and propose algorithms for minimizing the worst-case cost and the expected cost, respectively.

Resource Management in Managed Dense Small Cell Networks. Fast expansion of small cells, especially the femtocells, makes the problem of resource management in urban dense networks challenging. To achieve both high throughput and fairness among users, dynamic resource allocation algorithms are studied. Specifically, we consider two objectives: 1) Achieve $max - min$ fairness of throughput to mobile users in multiple collision domains. 2) Achieve QoE $max - min$ fairness to video streaming users in a single collision domain.

1) Fairness of Throughput to Mobile Users in Multiple Contention Domains. Our study considers two models in the solution. The non-interfering model (NINT model) assigns power levels to each femtocell in such a way that the femtocells do not interfere with each other, allowing for independent scheduling of

users within each femtocell. This model requires low coordination as the femtocells can operate independently for scheduling transmissions to their users. The more general interfering model (INT model) allows the femtocells to interfere but the sub-channel assignment disallows interfering links to simultaneously transmit in the same time slot and the same sub-channel. Although the level of coordination needed is higher in this model, better performance can be expected as it is a generalization of the NINT model. As the femtocells and the macrocell can use the wired backbone for exchanging control messages, both models are feasible to implement in practice. The contributions are:

- Under the NINT model, we propose a $max\{\beta, 1/N\}$ bounded centralized approximation algorithm and a distributed solution for the *maxmin* throughput problem, where β is the fraction of users that are outside the coverage range of any femtocell, and N is the number of users.

- We show that throughput can be further improved by introducing the INT model, and reduce the problem to the *partition coloring problem* [70], for which approximation algorithms with *provable bounds* were not known thus far. We then develop both centralized algorithm and localized implementation, bounded by $O(\Delta \log N)$ where Δ is the maximum inter-partition degree and N is the number of users.

- We evaluate the performance of these solutions with extensive simulations and compare with two baseline approaches. While the solutions under the NINT model achieve $2x$ of the minimum throughput, the solutions under the INT model achieve up to $3x$ of the minimum and average throughput, compared with DRA+ algorithm [107].

2) Fairness of QoE to Video Streaming Users in Single Contention Domain. Due to the increasing popularity of video streaming services, we extend the QoS (throughput) fairness metric to QoE fairness by solving the problem of bandwidth allocation in a single collision domain. We show that accurate estimation of required bitrate to sustain the current video being played is critical to the bandwidth allocation algorithm at the APs. While an existing solution relies on Deep Packet Inspection (DPI) [33], which is both expensive and infeasible for encrypted traffic, we propose a novel solution which infers such information based on observed traffic patterns at the APs and feedback provided by a software running at the clients, without requiring to modify the players on clients. Our contributions include:

- We propose novel solutions for APs to infer the current bitrate of each video streaming client, without parsing the content of the packets or making changes to video players, which are expensive to implement and deploy.
- Based on the bitrate-inferring solution, we design and implement algorithms that efficiently probe the set of candidate bitrate of each video.
- We propose a bandwidth allocation algorithm that achieves QoE domain fairness by allocating the available bandwidth to clients based on the candidate bitrates of each video client.

Incentive Mechanism Design for Access Acquisition of Unmanaged Small Cells. In order to incentivize the businesses and individual owners of small cells to make their services available to the WSPs to help offload data, we propose a reverse auction scheme, wherein the WSP provides monetary incentives to small players while small players bid to provide services to a WSP. We first introduce the imprecise valuation problem, in which, small players are unable to estimate their true valuations due to the ambiguity of the value of data offloading service. To solve this problem, this

dissertation introduces the notions of *Perceived Valuation*, *Partial Truthfulness*, and *Imprecision Loss*, which together characterize the *quality* of a truthful auction while considering imprecision in estimation of the true valuation. For any given values of the above parameters, with the goal of maximizing the WSP's utility, this work develops EasyBid, a novel mechanism with heuristic algorithms which allow conducting truthful auctions, considering that the sellers only know their perceived valuations which may differ from their true valuations. We make the following contributions on this problem:

- This work introduces the notions of *Perceived Valuation*, *Partial Truthfulness*, and *Imprecision Loss*, which together characterize the *quality* of a truthful auction while considering imprecision in estimation of the true valuations. This is a first such notion.

- For any given values of the above parameters, with the goal of maximizing the WSP's utility, this work develops EasyBid, a novel mechanism with heuristic algorithms which enable conducting truthful auctions, considering that the sellers only know their perceived valuations which may differ from their true valuations.

- Through simulations, we show that the utility achieved by EasyBid with imprecise valuations is close to the optimal solution that assumes precise valuations, under reasonable partial truthfulness and imprecision loss constraints.

1.4 Organization of Dissertation

This dissertation is organized as follows: Chapter 2 studies the deployment problem in large scale managed small cell networks. Chapters 3 and 4 propose resource allocation algorithms for managed dense small cell networks, which achieve throughput fairness

in multiple collision domains, and QoE fairness in single collision domain, respectively. Chapter 5 proposes a novel auction scheme EasyBid, to incentivize unmanaged cell owners to provide data offloading service to WSPs. Chapter 6 summarizes the results of the dissertation, and discusses possible future work.

Chapter 2

ENSURING PREDICTABLE CONTACT OPPORTUNITY FOR SCALABLE MOBILE INTERNET ACCESS

WiFi hotspots have been rapidly mushrooming in every city to meet the ever-increasing demand of data. They either operate independently as a competitive way of data access, or act as a complementary service and help offload the overburdened 3G networks [15]. But, their primary target is static users. These networks fail to provide any assured level of service to a mobile user. Although large deployments of WLANs can be used to provide high data-rate services over large areas, the cost becomes prohibitive due to the sheer number of access-points (APs) required. For instance, to cover a 2km x 2km area in Mountain View, Google needed to deploy 400 access points [123] to barely provide coverage at the base data rate. In addition to the deployment cost, the maintenance and management complexity has led to abandonment or scaling back of several WLAN projects from San Francisco to Philadelphia [14].

New Wireless Wide-Area Networking (WWAN) technologies such as 3GPP LTE (Long Term Evolution) and mobile WiMAX are expected to provide *either* long range coverage *or* high data rates, but practical numbers are far from the promised levels. For example WiMAX is intended to support data rates as high as 75 Mbps per 20 MHz channel, or a range of 30 miles [45]. However, one of the first deployments of WiMAX in US is reported to provide a downlink bandwidth of 3 Mbps [14], which is only within a factor of 2 better than the current 3G networks. Note that these

resources will potentially be shared by a large number of active users within the respective sector of the antenna. Given the resistance from majority of users to pay high monthly fees for mobile data access, which is essential for supporting expensive new deployments, ubiquitous service from such new deployments could take several years, and possibly decades.

The two objectives – an economically scalable infrastructure and quality of service assurance – can be achieved by a carefully planned *sparse* deployment of WiFi APs at roadside. In this work, we study deployment techniques for providing roadside WiFi services. We envision a wireless service provider that implements a deployment using two types of APs, new APs that are deployed for serving mobile users exclusively, and existing APs that are incentivized for sharing their capacity between static and mobile users. It is likely that these existing APs are initially deployed for serving static users or users with limited mobility and are possibly owned by other service providers or end users, and therefore will give higher priority to their original, mostly static, users.

To provide guaranteed performance to mobile users, we present a new metric, called **Contact Opportunity**, as a characterization of a roadside WiFi network. Informally, the contact opportunity for a given deployment measures the fraction of distance or time that a mobile user is in contact with some APs when moving through a certain trajectory. Such a metric is closely related to the quality of data service that a mobile user might experience while driving through the system. Our objective is to find a deployment that ensures a required level of contact opportunity with the minimum cost. Since the problem is NP-hard, we have designed an efficient approximation solution by exploiting a diminishing return property in the objective function. We further show how to extend this concept and the deployment techniques to a more intuitive metric – the average throughput – by taking various dynamic

elements into account. In particular, we take an interval based approach to model the uncertainties associated with road traffic conditions and the time varying data traffic load of static users. The deployment algorithm is then extended to achieve a required level of average throughput under uncertainties, where we consider both a robust optimization approach that minimizes the cost in the worst-case scenario, and a two-stage stochastic optimization approach that minimizes the expected cost.

While focusing on WiFi deployment, our study also provides useful insights to the large deployment of other types of wireless networks, such as femotcells, for serving mobile users. Femotocells are small cellular base-stations initially designed to improve the indoor cellular coverage. But they are currently being extended to provide high data-rate coverage over short ranges to the outdoor environment as well [8, 9], and can potentially be utilized to support data-incentive services for mobile users. Our techniques can be applied to deploying new femotocell base-stations (FBSs) as well as acquiring service from existing FBSs that originally target at static users. One challenge to achieve a scalable infrastructure for serving mobile users using FBSs is to properly model t he dynamics of data traffic load associated with both femtocells and macrocells.

This is the first work that addresses the challenges in achieving a *sparse* wireless infrastructure that provides QoS assurance to mobile users *in the face of uncertainty*.

2.1 Related Work

The idea of Drive-thru Internet by connecting to existing roadside WiFi Access Points is introduced in [90]. Subsequently, evaluations in various controlled environments [90, 48, 86, 24] and in situ WiFi networks [29, 86, 24, 44] have been conducted, further confirming the feasibility of WiFi-based Vehicular Internet Access for non-interactive applications. In addition to WiFi, small cell architectures such as femtocells, which

were initially designed to improve the indoor cellular experience, are being extended to provide high data rate coverage over short ranges to the outdoor environment [8, 9].

In spite of these efforts, scalable solutions for the *deployment* and *management* of WiFi APs or femtocell base-stations to enable efficient vehicular Internet Access have not been fully understood so far. Instead, simple heuristics without performance guarantees are commonly adopted in most previous works. For instance, a simple non-uniform strategy that places more stationary nodes in the network core was considered in a recent work [25]. Previous work on Alpha Coverage [135, 136] initiated research on scalable deployment of road-side APs for providing guaranteed service to mobile vehicles. In Alpha Coverage, the objective is to bound the gap between two consecutive contacts while ignoring the quality of each contact. In contrast, the notion of Contact Opportunity introduced in this work provides a more accurate and practical measurement of service quality for mobile entities.

2.2 Contact Opportunity Optimization

Ideally, we would like to have a scalable deployment of APs that is able to serve mobile users on the go with guaranteed performance in terms of some intuitive metric such as average throughput. Such an objective is complicated by various uncertainties in the system, such as unpredictable traffic conditions, unknown moving patterns of mobile users, and the dynamics involved in the performance of APs. To this end, we use an incremental approach; we introduce a performance metric for roadside AP deployment that is closely related to average throughput while avoiding the uncertainties such that an efficient solution can be obtained. In Section 2.3, several extensions that consider more intuitive performance metrics and more practical system models are introduced.

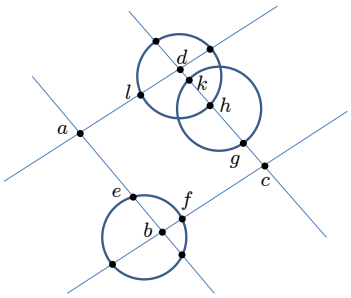


Figure 2.1: A road network with four roads (lines) and three candidate locations with coverage regions shown as disks. There are four road intersections, i.e., a, b, c, and d. The coverage disks partition the roads into subsegments such as ae, be, bf, cg, dl, etc.

2.2.1 System Model

We model a road network as a connected *geometric* graph, where vertices represent points where road centerline segments and road intersections meet, and edges represent road centerline segments connecting road intersections. For a curved road segment, we introduce artificial road intersections, so that each edge represents a straight line segment. Without loss of generality, the road network graph is assumed to be undirected. Let d_e denote the length of road segment e, and let E denote the set of road segments.

In this section, we focus on the deployment of new APs that serve mobile users exclusively. Extensions to acquiring service from existing APs and the coexistence of static and mobile users will be considered in Section 2.3. Let A denote a set of known candidate locations in the 2D region covering the road network where new APs can be deployed. Note that any points in the 2D area can be a candidate location, although for simplicity, we take the set of road intersections as candidate locations in our simulations. Associated with each candidate location $a \in A$, there is a fixed cost $w_a \in \mathbb{R}^+$ for installing an AP at a, and a coverage region C_a, which is a connected

region in the 2D space consisting of the set of points where the received SNR from an AP deployed at a is higher than a fixed threshold. The coverage regions together with road intersections partition the road network graph into smaller segments called *subsegments*. Figure 2.1 shows a road network with four roads (lines) and three candidate locations with coverage regions shown as disks, which partition the roads into subsegments such as ae, be, bf, cg, dl, etc. A subsegment may be covered by multiple coverage regions, such as hk, or not covered at all, such as al. Although the coverage regions are plotted as disks in Figure 2.1, our problem definitions and solutions are independent of the shape of a coverage region.

Let L denote the set of all the subsegments in the road network graph with respect to A. For each $l \in L$, let $d_l \in \mathbb{R}^+$ denote the length of the corresponding road centerline segment. Let $L_e \subseteq L$ denote the set of subsegments on edge $e \in E$. Let $L_S \subseteq L$ denote the set of subsegments covered by a deployment $S \subseteq A$, that is, $L_S = \{l \in L : l \subseteq \cup_{a \in S} C_a\}$.

A movement on a road network is modeled as a simple path on the corresponding graph. We assume that there is a set of movements, denoted as P, given as part of the input to the deployment decision maker. For instance, P could be a set of shortest (or fastest) paths or a set of most frequently traveled paths between a set of sources and destinations. Such information can be learned from a road network database [1] and historical traffic data [50]. The concrete definition of P is independent of our problem definitions and solutions, while the size of the set P impacts the computational complexity and performance guarantee of our solutions as discussed below. For each $p \in P$, let $E_p \subseteq E$ denote the set of edges on p, and $L_p \subseteq L$ the set of subsegments on p.

2.2.2 Problem Statement

We now define a performance metric for roadside deployment that does not require any information about the dynamics of the system. Given a deployment $S \subseteq A$, the **Contact Opportunity in Distance** of a path $p \in P$, denoted as η_p^d, is defined as the fraction of distance on p that is covered by some AP in S. Formally,

$$\eta_p^d(S) = \frac{\sum_{l \in L_p \cap L_S} d_l}{\sum_{l \in L_p} d_l}. \tag{2.1}$$

When a mobile user travels at a constant speed where each AP has the same data rate, and there is only one user in the system, contact opportunity in distance can be directly translated into average throughput that the user will experience. We show in Section 2.3 how to extend this concept by taking various dynamic elements into account. Our objective is to provide a required level of contact opportunity over all the movements in P at a minimum cost. Formally, let λ_p denote the required contact opportunity for path p, and $w(S)$ the cost of a deployment $S \subseteq A$, that is, $w(S) = \sum_{a \in S} w_a$, the first optimization problem that we consider is:

$$\min_{S \subseteq A} w(S), \text{ subject to } \eta_p^d(S) \geq \lambda_p, \forall p \in P. \tag{2.2}$$

For a given set of parameters λ_p, let $\lambda = \min_{p \in P} \lambda_p$ and $\tilde{\eta}_p^d(S) = \eta_p^d(S) \times \lambda / \lambda_p$. Then the constraint in (2) is equivalent to requiring that the minimum $\tilde{\eta}_p^d(S)$ among *all* the paths in P is at least λ. To simplify the notation, we will use η_p^d to denote $\tilde{\eta}_p^d$ in the rest of the chapter. Thus, the optimization problem to be solved is:

$$\begin{aligned} \textbf{P1:} \quad & \min_{S \subseteq A} \; w(S) \\ & \text{s.t.} \; \min_{p \in P} \eta_p^d(S) \geq \lambda \end{aligned} \tag{2.3}$$

We will also study a dual problem that maximizes the minimum contact opportunity among all the paths for a given budget B:

$$\textbf{P2:} \quad \max_{S \subseteq A} \min_{p \in P} \ \eta_p^d(S)$$
$$\text{s.t. } w(S) \leq B \tag{2.4}$$

Hardness of the problem: We note that both problems are NP-hard in general. To see this, consider a road network graph where each vertex is a candidate location for APs. Assume that the coverage region of an AP at vertex a can fully cover all the edges incident to a and only those edges, and the set of movements in P are paths consisting of single edges. Then a reduction from Vertex Cover to the decision version of our problems can be easily constructed. Since Vertex Cover is NP-complete even when restricted to 3-connected, cubic planar graphs [95], both **P1** and **P2** are NP-hard. Hence, it is not likely that optimal solutions to these problems can be obtained for most practical settings. Our approach is to design efficient approximation algorithms that can be implemented even in a large scale system, while ensuring a guaranteed performance.

2.2.3 Minimum Cost Contact Opportunity

In this section, we first present a simple greedy algorithm to **P1** and show that the algorithm achieves a guaranteed performance, by a reduction to the submodular set covering problem [124]. We then discuss strategies to accelerate the computation in our context.

A Greedy Algorithm: We first note that if we define

$$\eta^d(S, \lambda) = \sum_{p \in P} \min\{\eta_p^d(S), \lambda\}, \tag{2.5}$$

then a subset $S \subseteq A$ is a feasible solution to **P1** iff $\eta^d(S,\lambda) = \eta^d(A,\lambda) = \lambda|P|$. To see this, first note that S is feasible iff $\eta_p^d(S) \geq \lambda$ for all p by the problem definition, which is true iff $\eta^d(S,\lambda) = \lambda|P|$ by (2.5). Moreover, **P1** has a feasible solution iff A is feasible. Hence the statement holds. Based on this observation, the greedy algorithm for **P1** is sketched in Algorithm 1. The algorithm starts with an empty set and in each iteration picks a new candidate location that is most cost-effective, i.e., the location that maximizes the incremental difference (normalized by the weight). The procedure repeats until the required contact opportunity is achieved.

Algorithm 1: Minimum Cost Contact Opportunity

Input: A, P, λ

Output: A subset $S \subseteq A$

1 $S \leftarrow \emptyset$

2 **while** $\eta^d(S,\lambda) < \eta^d(A,\lambda)$ **do**

3 Find $a \in A \backslash S$ that maximizes $\frac{\eta^d(S\cup\{a\},\lambda) - \eta^d(S,\lambda)}{w_a}$

4 $S \leftarrow S \cup \{a\}$

Approximation Analysis: To prove an approximation factor to Algorithm 1, we first observe some structural properties of $\eta_p^d(S)$ and $\eta^d(S,\lambda)$. In particular, we note that the set function $\eta_p^d : 2^A \rightarrow [0,1]$ satisfies the following properties: (1) nondecreasing, i.e., $\eta_p^d(S) \leq \eta_p^d(T)$ whenever $S \subseteq T \subseteq A$; (2) normalized, i.e., $\eta_p^d(\emptyset) = 0$; and (3) *submodular*, i.e., for all $S \subseteq T \subseteq A$ and $a \in A \backslash T$, $\eta_p^d(S \cup \{a\}) - \eta_p^d(S) \geq \eta_p^d(T \cup \{a\}) - \eta_p^d(T)$. The last property is formally proved below, which essentially says that *adding a new AP to a small set helps more than adding it to a large set*. It captures our intuition that the total coverage that two

APs can provide to a mobile user is reduced if their communication regions overlap with each other.

Lemma 2.2.1. *$\eta_p^d(\cdot)$ is submodular.*

Proof. For any $S \subseteq T \subseteq A, a \in A \backslash T$, $\eta_p^d(S \cup \{a\}) - \eta_p^d(S) = \frac{\sum_{l \in L_p \cap (L_{\{a\}} \backslash L_S)} d_l}{\sum_{l \in L_p} d_l}$, and $\eta_p^d(T \cup \{a\}) - \eta_p^d(T) = \frac{\sum_{l \in L_p \cap (L_{\{a\}} \backslash L_T)} d_l}{\sum_{l \in L_p} d_l}$. Since $S \subseteq T$, $L_{\{a\}} \backslash L_S \supseteq L_{\{a\}} \backslash L_T$. Therefore, $\eta_p^d(S \cup \{a\}) - \eta_p^d(S) \geq \eta_p^d(T \cup \{a\}) - \eta_p^d(T)$. □

We then note that $\eta^d(\cdot, \lambda)$ for a given λ is also a monotone submodular function since (a) $\min\{\eta_p^d(S), \lambda\}$ as a set function over subsets of A is submodular when η_p^d is submodular [87] and (b) the sum of submodular functions is submodular.

It follows that **P1** is an instance of the submodular set covering problem [124, 68]. In the general form of the problem, we are given a submodular function $f(\cdot)$ defined on a set A, and a cost $w(a)$ for any $a \in A$, and a constant λ. The objective is to find a subset S to minimize $w(S)$ such that $f(S) \geq \lambda$. We then have the following performance guarantee:

Proposition 2.2.1. *Algorithm 1 finds a feasible solution, the cost of which never exceeds the optimal cost by more than a factor $O(1) + \log(\max_{a \in A} D_a)$, where $D_a = \sum_{p \in P} \sum_{l \in L_p \cap L_{\{a\}}} d_l$ denotes the total distance covered by a single AP $a \in A$ over all the paths*

Proof. A classical result in [124] is that when f is monotone submodular and has *integer* values, the greedy algorithm achieves an approximation factor of $O(1) + \log(\max_{a \in A} f(\{a\}))$ to the submodular set covering problem. To apply this result in our context, we rewrite the constraint (3) in **P1** as $\sum_{l \in L_p \cap L_S} d_l \leq \lambda \sum_{l \in L_p} d_l$ for each p. By taking a proper unit, we can assume all the distance values are integral. For a given λ, if we define $f(S) = \sum_p \min\{\sum_{l \in L_p \cap L_S} d_l, \lambda \sum_{l \in L_p} d_l\}$, our problem

becomes a submodular set covering problem with respect to f. Hence, we get an approximation factor of

$$\begin{aligned} O(1) + \log(\max_a \sum_p \min\{\sum_{l \in L_p \cap L_{\{a\}}} d_l, \lambda \sum_{l \in L_p} d_l\}) \\ \leq O(1) + \log(\max_a \sum_p \sum_{l \in L_p \cap L_{\{a\}}} d_l) \\ = O(1) + \log(\max_a D_a) \end{aligned}$$

□

The above procedure can be naturally extended to improving an existing deployment by adding new APs, by substituting all the evaluations of $\eta_p^d(S)$ with $\eta_p^d(S \cup A_0)$, where A_0 indicates the set of APs previously deployed.

Techniques to Accelerate the Computation: Algorithm 1 requires O($|A|$) iterations (line 2 to line 4) where each iteration involves $|A|$ evaluations of $\eta^d(\cdot, \lambda)$. Hence, in all it requires $O(|A|^2)$ evaluations of $\eta^d(\cdot, \lambda)$, where each evaluation involves computing $\eta_p^d(\cdot, \lambda)$ for each $p \in P$, which takes $O(|P||V||A|)$ time. Hence, the total complexity is $O(|P||V||A|^3)$, which is very time consuming for a large road network, and with large $|A|$ and $|P|$. Below we propose several techniques to accelerate the computation in our context.

- First, we apply the accelerated greedy algorithm [83] to our problem, which significantly reduces the total number of evaluations of $\eta^d(\cdot, \lambda)$ needed through lazy evaluations. The submodularity of $\eta^d(\cdot, \lambda)$ implies that the incremental difference $\eta^d(S \cup \{a\}, \lambda) - \eta^d(S, \lambda)$ for any candidate location a is non-increasing in S. In the algorithm, a priority queue is used to maintain a set of incremental differences for all candidate locations. In each iteration, instead of checking all candidate locations as in the simple greedy algorithm, locations with higher

incremental differences up to this stage are first considered, which avoids a large number of evaluations. More details can be found in [83, 111].

- Second, we note that for any path $p \in P$, if p can be divided at certain road intersection into two sub-paths $p_1, p_2 \in P$ such that $\eta^d_{p_1}(S) \geq \lambda, \eta^d_{p_2}(S) \geq \lambda$, then $\eta^d_p(S) \geq \lambda$ as well. In this case, constraint (2.3) is automatically satisfied for p if they are satisfied for p_1 and p_2. Therefore, we can safely exclude p from P without loss of optimality. Now suppose P is composed of all the shortest paths of length at least α in the road network graph, and the maximum edge length in the graph is significantly less than α. Then it is likely that a shortest path of length greater than 2α can be divided into sub-paths of length between α and 2α. These longer paths can then be dropped to reduce the size of P.

- Third, we note that each candidate location only contributes to a small subset of $\hat{P}$, and therefore an incremental calculation is more efficient, where $\eta^d(S \cup \{a\}, \lambda)$ is obtained from $\eta^d(S, \lambda)$ by updating only η^d_p for those p covered by C_a.

We observe that these techniques improve the performance of our algorithm significantly in practice. For the 6×6 km^2 road network and a set of 10000 movements considered in our simulations (Section 2.4), the running time of Algorithm 1 to find a solution to **P1** is reduced from hours to a few seconds under the same machine configuration.

2.2.4 Contact Opportunity Maximization

After providing an approximation algorithm to Problem **P1**, we now propose a solution to **P2** by utilizing Algorithm 1 as a subroutine. The idea is to apply a binary search over $\lambda \in [0, 1]$. A similar approach has been applied in [68] to solve a submodular covering problem with a budget. The procedure is sketched in Algorithm 2.

Algorithm 2: Maximum Contact Opportunity

Input: A, P, B

Output: A subset $S \subseteq A$

1 $\lambda_1 \leftarrow \min_{p \in P} \eta_p^d(A)$; $\lambda_2 = 0$

2 **while** $\lambda_1 - \lambda_2 \geq \delta$ **do**

3 $\quad \lambda = (\lambda_1 + \lambda_2)/2$

4 $\quad S \leftarrow$ call Algorithm 1 with parameters A, P, λ

5 $\quad$ **if** $w(S) > B$ **then**

6 $\quad\quad \lambda_1 \leftarrow \lambda$

7 $\quad$ **else**

8 $\quad\quad \lambda_2 \leftarrow \lambda$

The algorithm maintains an upper bound and a lower bound for achievable λ, denoted as λ_1 and λ_2, respectively. Initially, $\lambda_1 = \min_{p \in P} \eta_p^d(A)$, the minimum contact opportunity that can be achieved when all the candidate locations are utilized to deploy APs, and $\lambda_2 = 0$. Algorithm 1 is then invoked with $\lambda = (\lambda_1 + \lambda_2)/2$ as the input. If the solution found surpasses the budget, the upper bound is decreased; otherwise, the lower bound is increased. The procedure continues until the difference between the upper and lower bounds is less than δ, where δ can be adjusted to trade accuracy with computational time. Note that to accelerate the computation, an extra condition can be added to the `while` loop of Algorithm 1 (line 2) so that whenever the current S maintained already violates the budget constraint, the above procedure can move on to a new λ.

For a given a budget B, the above binary search procedure always finds a feasible deployment. Moreover, the algorithm achieves a bi-criteria approximation as stated below.

Proposition 2.2.2. *Given a budget B, let $\lambda(B)$ and $\lambda^*(B)$ denote the contact opportunity achieved by Algorithm 2 and the optimal solution, respectively. Then we have $\lambda(B) \geq \lambda^*(B/\epsilon) - \delta$, where $\epsilon = O(1) + \ln(\max_{a \in A} D_a)$.*

Proof. In the binary search, if the value of λ is set to $\lambda^*(B/\epsilon)$, then Algorithm 1 with this λ as input will find a deployment S of cost at most B according to Proposition 2.2.1. Since S is feasible, the binary search won't miss this λ beyond the small gap defined by δ. □

2.3 From Contact Opportunity to Average Throughput

The concept of contact opportunity in distance discussed in Section 2.2 ignores several complexities involved in a real system and does not directly correspond to the quality of service for mobile users driving through the system. Therefore, we seek to design performance metrics that are more intuitive to mobile application designers and end users. In this section, we first extend the notion of contact opportunity to average throughput by modeling various uncertainties involved in the system (Section 2.3.1). We then study the deployment problem of achieving a required level of average throughput while minimizing the worst-case cost or the expected cost. To this end, we will consider a robust optimization approach in Section 2.3.2, and a two-stage stochastic optimization approach in Section 2.3.3, respectively.

2.3.1 Modeling Average Throughput Under Uncertainty

To obtain a meaningful definition of average throughput in our context, we start with modeling two key dynamic aspects in our system: road traffic conditions and the data traffic from static users.

First, it is clear that the average throughput that a mobile user can obtain depends on both its travel speed and the contact duration when it is associated with some APs. However, both the contact time and the travel time are not fixed due to the uncertainties of traffic conditions such as traffic jams, accidents and stop signs. Moreover, the traffic condition also affects the number of mobile users that are in the range of the same AP at the same time competing for the bandwidth of the AP. To model these uncertainties, we follow the interval based modeling approach from [67] and consider two key parameters in characterizing a traffic flow: speed and density [12].

Assumption 2.3.1. *The driving speed of a road segment $e \in E$, denoted as v_e, is within an interval $[v_e^1, v_e^2]$ for some constants $v_e^1 > 0$, $v_e^2 > 0$ and $v_e^1 \leq v_e^2$. Similarly, the traffic density on road segment e, i.e., the number of road-side WiFi service users on e per unit distance, denoted as h_e, is within an interval $[h_e^1, h_e^2]$ where $0 < h_e^1 \leq h_e^2$.*

Second, as stated before, we envision a deployment model where both new APs can be installed and existing APs targeting at static users can be acquired to share their service with mobile users at certain cost. With a slight abuse of notation, we again let A denote the (disjoint) union of both candidate locations for new APs and the locations of existing APs that can be utilized. Our objective is to guarantee the service quality to mobile users at the minimum cost without affecting static users. Since the data traffic of static users may vary over time, we again use the interval based approach.

Assumption 2.3.2. *The available date rate of an AP located at a for serving mobile users, denoted by r_a, is within an interval $[r_a^1, r_a^2]$, where $0 < r_a^1 \leq r_a^2$.*

The above intervals for modeling uncertainties are assumed to be given or can be learned from historical data. We define a *scenario* k to be an assignment of

values to the random variables defined above from the corresponding intervals. Let $v_e(k), h_e(k)$ and $r_a(k)$ denote the corresponding values of these variables in a scenario k. We define $v_l(k)$ and $h_l(k)$ as the speed and density for a subsegment l, derived from the corresponding values of the road segment that l belongs to. Let K denote the set of all possible scenarios. Note that K is an infinite set. We then state the first natural extension to the notion of contact opportunity in distance, where we replace distance with time. Formally, given a deployment $S \subseteq A$ and a scenario $k \in K$, we define the **Contact Opportunity in Time** of a path $p \in P$ as:

$$\eta_p^t(S,k) = \frac{\sum_{l \in L_p \cap L_S} d_l / v_l(k)}{\sum_{l \in L_p} d_l / v_l(k)}, \tag{2.6}$$

which captures the fraction of *time* that a mobile user is in contact with some AP when moving through p.

To move one step further and model average throughput, we need to make further assumptions regarding association control and scheduling in serving mobile traffic load. Consider a scenario k and a deployment S. Let $u_l(k)$ denote the expected number of mobile users on a subsegment l, which can be estimated as $u_l(k) = h_l(k)d_l$. We will focus on the steady state where the mobile users in the system are distributed according to above estimates. In theory, an optimal scheduling policy that maximizes the time average throughput over a movement can be derived by solving a maximum flow problem. However, due to its high complexity and the centralized nature, such a policy is not likely to be used for serving real time traffic. Instead, we focus on simple stateless and distributed strategies that are easily implementable. Our approach is to estimate the expected data rate that a mobile user on a subsegment l can obtain in the steady state, denoted as $r_l(k)$. Given the estimates, the **Average Throughput** for a mobile user moving through a path p, denoted as $\gamma_p(S,k)$, can be stated as:

$$\gamma_p(S,k) = \frac{\sum_{l \in L_p \cap L_S} [d_l / v_l(k)] r_l(k)}{\sum_{l \in L_p} d_l / v_l(k)}. \tag{2.7}$$

Below we outline one approach to estimate $r_l(k)$. We drop the index k to simplify the notation. We will consider the following simple association protocol as a case study.

Assumption 2.3.3. *Each mobile user picks an AP in its range at random to associate, and an AP serves all the users associated with it in an equal rate. A mobile user can associate with at most one AP at any time. The set of APs operate on orthogonal channels and do not interfere with each other.*

This protocol does not rely on any real-time information and can be easily implemented in practice. Consider a subsegment l that is within the coverage regions of multiple APs. By the random association assumption, a user in l has an equal chance to be served by any of these APs. Let n_l denote the number of APs that cover l, then u_l/n_l users are assigned to each of these APs. Now for any AP a, let L_a denote the set of subsegments in its range, and let $u_a = \sum_{l \in L_a} u_l/n_l$ denote the total number of users associated with a. All these users are served in an equal rate of r_a/u_a. For any user on segment l, its expected data rate can then be estimated as $r_l = \frac{\sum_{a \in S_l} r_a/u_a}{|S_l|}$. We observe that under this approach, average throughput reduces to contact opportunity in time when $r_a = 1$ and $u_a = 1$ for all $a \in A$.

Remark: The focus of this work is on optimal deployment of APs under a given association protocol. We have considered the above simple protocol as a case study, while our algorithms can be applied to more sophisticated association protocols. We note that, however, it is very challenging to implement optimal association and scheduling in practice, which requires real-time traffic data and the service history of each mobile user so that a higher priority can be given to users that have been underserved. What is more challenging is the joint optimization of deployment and association control. On the other hand, we envision that a service provider may actually prefer a simpler

protocol such as the one we considered to avoid the high cost of maintaining real-time data and service history.

As in Problem **P1**, our objective is to ensure the required average throughput at the minimum cost. Since the cost varies for different scenarios, we would like to minimize either the cost in the worst-case scenario, or the expected cost. We outline two approaches below.

2.3.2 Robust Optimization

We first study a robust optimization approach. Although there are infinitely many scenarios, we seek to find a deployment that performs well even in the worst case. To this end, we first present two problems to be studied, which extend **P1** and **P2**, respectively, with the objective of ensuring average throughput under a worst-case scenario. We then propose an efficient algorithm to identify a worst-case scenario for any given deployment, which is then utilized to derive our solutions to the robust optimization problems.

Problem Statement: Let w_a denote either the cost for installing a new AP at location a or the (one-time) cost to obtain service from an existing AP located at a, and again define $w(S) = \sum_{a \in S} w_a$. Our objective is to solve the following problem:

$$\textbf{P3:} \quad \min_{S \subseteq A} \; w(S)$$
$$\text{s.t.} \quad \min_{p \in P, k \in K} \gamma_p(S,k) \geq \lambda \tag{2.8}$$

where (2.8) ensures that for a mobile user moving through any path in P, an average throughput of λ is obtained under any scenario. We will also consider the dual problem:

$$\textbf{P4:} \quad \max_{S \subseteq A} \min_{p \in P, k \in K} \; \gamma_p(S,k)$$

$$\text{s.t. } w(S) \leq B \tag{2.9}$$

For a given deployment S, we define a scenario $k_S \in K$ to be a worst-case scenario if $\min_{p \in P} \gamma_p(S, k)$ is minimized at k_S among all the scenarios in K. Then the constraint (2.8) is equivalent to $\min_{p \in P} \gamma_p(S, k_S) \geq \lambda$. Note that there may exist multiple worst-case scenarios in general, since only the values associated with the road segments on and coverage regions touching the path with the minimum throughput matter.

Identifying a Worst-case Scenario: We now present an efficient algorithm to find a worst-case scenario, followed by our solutions to **P3** and **P4**. We first note that since r_l only appears in the numerator of (2.7), $\gamma_p(S, k)$ is minimized by taking the minimum possible value of r_l, that is, by setting $r_a = r_a^1$ and $h_l = h_l^2$ for all $a \in A$ and $l \in L$. We again usee r_l to denote this worst-case value when there is no confusion. We define $r_l = 0$ if $l \notin L_S$. It remains to determine the values of v_l. Our intuition is that a worst-case scenario is most likely to happen when the traffic condition is such that the travel speed is slow on road segments with poor data access while the speed is high on road segments with good data access. Since a constant travel speed is assumed for each road segment, we can rewrite (2.7) as follows, where we drop the index k to simplify the notation:

$$\gamma_p(S, \cdot) = \frac{\sum_{e \in E_p} r_e(S) d_e / v_e}{\sum_{e \in E_p} d_e / v_e}. \tag{2.10}$$

where $r_e(S) = (\sum_{l \in L_e \cap L_S} d_l r_l)/d_e$ indicates the average data rate over e under deployment S. The following proposition formalizes the above intuition (see the Appendix for a proof):

Proposition 2.3.1. *For any path p, there is an assignment of v_e for $e \in E_p$ that minimizes γ_p such that the following two conditions are satisfied: (1) $v_e = v_e^1$ or*

$v_e = v_e^2$, for all $e \in E_p$; (2) there is an element $e^* \in E_p$, such that $v_e = v_e^1$ if $r_e \leq r_{e^*}$, and $v_e = v_e^2$ if $r_e > r_{e^*}$.

The proposition states that there is a worst-case scenario for a path p, where the driving speed of every edge in p takes one of its boundary values, and moreover, the assignment satisfies a dichotomy condition according to their average data rate r_e. Based on this observation, an assignment of v_e that achieves a worst-case scenario can be easily found by a search over all the edges in path p to find the pivot e^* that minimizes γ_p (see our technical report [134] for a formal description). A worst-case scenario over all the paths can be then found by a search over P.

We remark that in the special case of contact opportunity in time, i.e., $r_l = 1$ for all $l \in L$, if we further allow the subsegments on the same edge to have different driving speeds, the worst-case scenario allows a simpler characterization as follows. Let $v_l \in [v_l^1, v_l^2]$ denote the possible speed on subsegment l. Then for a given deployment S, a worst-case scenario is obtained by setting $v_l = v_l^1$ if l is not covered by S, and $v_l = v_l^2$ otherwise. To see this, note that the contact opportunity in time can be written as $\eta_p^t(S, \cdot) = \frac{t_1}{t_2 + t_1}$ where t_1 denotes the travel time over the set of subsegments in p that are covered by S and t_2 denotes the travel time over the other subsegments in p. Hence η_p^t is minimized when t_1 is minimized and t_2 is maximized, which happens under the above scenario.

Solutions to Robust Optimization: We then propose solutions to **P3** and **P4**. First note that, if we consider a fixed worst-case scenario for each deployment S, denoted as k_S, $\gamma_p(S, k_S)$ can be viewed as a set function over A. Hence a natural first attempt to **P3** is to apply Algorithm 1 by replacing $\eta_p^d(S)$ with $\gamma_p(S, k_S)$. However, this approach does not provide a performance guarantee. The difficulty is that although for a given scenario k, $\gamma_p(S, k)$ is submodular by a similar argument

as in Lemma 2.2.1, $\gamma_p(S, k_S)$ is not submodular in general as stated in the following proposition. A detailed proof is provided in our online technical report [134].

Proposition 2.3.2. *$\gamma_p(S, k_S)$ as a set function over A is nondecreasing and normalized, but not submodular.*

In fact, it has been observed that the robust versions of many optimization problems are significantly more difficult than the original problems [67]. Although an efficient solution with guaranteed performance to the general problem remains open, we propose the following two approaches as first steps that work well in many practical cases.

Our **first approach** applies when for every $p \in P$, the set of candidate locations that cover p, denoted as A_p, has small cardinality. The key idea is to view the constraint (2.8) as requiring that an average throughput is guaranteed over all the paths *and* under all the scenarios. For any path p, to identify a worst-case scenario with respect to p, it suffices to only consider the worst-case scenarios with respect to subsets of A_p, namely, $\{k_S \in K : S \subseteq A_p\}$, since other APs do not affect the performance over p. Therefore, to identify a worst-case scenario overall all the paths, it suffices to consider $K' = \{k_S \in K : S \subseteq A_p \text{ for some } p \in P\}$. Note that the size of K' is $\sum_{p \in P} 2^{|A_p|}$, which is polynomial in $|A|$ and $|P|$ when $|A_p| = O(\log A)$ for any p. We then define $\gamma(S, \lambda) = \sum_{p \in P, k \in K'} \min\{\gamma_p(S, k), \lambda\}$, which is again submodular. Algorithm 1 can then be applied to **P3** by replacing $\eta^d(S, \lambda)$ with $\gamma(S, \lambda)$. This approach has polynomial time complexity when $|A_p| = O(\log A)$ for any p. Moreover, by a similar argument as in Proposition 2.2.1, it achieves an approximation factor $O(1) + \log(\max_{a \in A} \mathcal{R}_a)$, where $\mathcal{R}_a = \sum_{p \in P, k \in K'} \sum_{e \in E_p} r_e(\{a\}) d_e / v_e(k)$ indicates the total throughput contributed by a single AP $a \in A$ across all the paths and all the scenarios in K'.

Our **second approach** is to approximate the deployment dependent worst-case

scenario by a single fixed scenario that is independent of the deployment chosen. Let k_0 denote the "mean speed" scenario with $v_e(k_0) = (v_e^1 + v_e^2)/2, h_e(k_0) = h_e^2, \forall e \in E$, and $r_a(k_0) = r_a^1, \forall a \in A$. It turns out that, if v_e^2/v_e^1 is small for all $e \in E$, k_0 can be used as a good approximation of the worst-case scenario. More concretely, we have the following proposition for any deployment S:

Proposition 2.3.3. *If $v_e^2/v_e^1 \leq \beta$ for all $e \in E$, then $\gamma_p(S, k_S) \leq \gamma_p(S, k_0) \leq \beta\gamma_p(S, k_S)$ for any $p \in P$.*

A formal proof is given in the Appendix. The proposition implies that if v_e^2/v_e^1 is bounded above by a constant $\beta \geq 1$, then for any path p, the loss of average throughput by replacing the worst-case scenario with the "mean speed" scenario is bounded by β. In fact, the second inequality holds between k_0 and any other scenario, not necessarily the worst-case scenario. Based on this observation, we then design an algorithm to **P3** as sketched in Algorithm 3.

Algorithm 3: Robust Minimum Cost Contact Opportunity

Input: A, P, λ

Output: A subset $S \subseteq A$

$v_e \leftarrow (v_e^1 + v_e^2)/2, h_e \leftarrow h_e^1, \forall e \in E$; $r_a \leftarrow r_a^1, \forall a \in A$

$m \leftarrow (\beta - 1)/\tau$

for $i = 0$ *to* m **do**

  $\lambda_0 \leftarrow (1 + i\tau)\lambda$

  $S \leftarrow$ call Algorithm 1 with parameters A, P and λ_0, where $\eta_p^d(S)$ is replaced by $\eta_p^d(S, k_0)$

  if $\min_{p \in P} \gamma_p(S, k_S) \geq \lambda$ **then**

    break

The algorithm searches over $\lambda_0 = \lambda, (1+\tau)\lambda, ..., \beta\lambda$, and for each λ_0, Algorithm 1 is invoked with $\eta_p^d(S)$ replaced by $\gamma_p(S, k_0)$. The search repeats until a deployment that achieves an average throughput of at least λ in the worst-case scenario is found. Note that such a deployment always exists, since by setting $\lambda_0 = \beta\lambda$, the deployment found by Algorithm 1 achieves an average throughput $\beta\lambda$ under scenario k_0, which ensures an average throughput of λ in the worst-case scenario by Proposition 2.3.3. Furthermore, the algorithm achieves a bi-criteria approximation in the following sense: the cost of the solution found is no larger than $c^*(\beta\lambda)(O(1)+\log(\max_{a\in A} R_a))$, where $c^*(\beta\lambda)$ is the optimal cost for achieving an average throughput of $\beta\lambda$, and $R_a = \sum_{p\in P}\sum_{e\in E_p} r_e(\{a\}, k_0)d_e/v_e(k_0)$ indicates the total throughput contributed by a single AP $a \in A$ across all the paths under the scenario k_0.

Proposition 2.3.3 also leads to a simple solution to **P4**. The idea is to simply invoke Algorithm 2 for the "mean speed" scenario, that is, replacing $\eta_p^d(S)$ with $\gamma_p(S, k_0)$. This approach always gives a feasible solution, while the minimum average throughput across all the path achieved is at least $\frac{1}{\beta}\Big(\lambda^*(B/\epsilon') - \delta\Big)$, where $\epsilon' = O(1)+\log(\max_{a\in A} R_a)$, and $\lambda^*(B/\epsilon')$ is the optimal achievable value under the budget B/ϵ'. Note that, compared with the non-robust version (Proposition 2.2.2), an extra factor of $1/\beta$ is incurred.

2.3.3 Two-stage Stochastic Optimization

In contrast to robust optimization, our second approach to achieving an economical deployment under uncertainty focuses on minimizing the *expected* cost for ensuring a required level of average throughput, based on knowledge of the scenario distribution. In this section, we adopt the 2-stage stochastic approximation framework widely used in decision making under uncertainty [109], which has a natural interpretation in our context as discussed below. We propose an efficient approximation solution based on

the sample average approximation (SAA) method [110], combined with an extension of Algorithm 1.

We envision a setting where a deployment is created in two stages, which can be readily generalized to the multi-stage case. In the first stage, the service provider implements an initial deployment by installing new APs or contracting with existing AP owners at selected locations. This decision is based on the prediction of system dynamics, such as road traffic condition and data traffic load from static users, for a relatively long period of time, say one month or one year. In the second stage, after the more accurate or actual traffic condition is realized, the initial deployment is augmented by acquiring service from additional APs, if needed, which happens at a relatively short time scale, say one day or one hour. Due to the short lead time in the second stage, it is expected that APs obtained in the second stage are more costly than that acquired in the first stage. Let w_a^1 denote the (amortized) cost per unit of time for an AP $a \in A$ installed/leased in the first stage, and $w_a^2 > w_a^1$ the corresponding cost if it is acquired in the second stage. Let $w_1(S) = \sum_{a \in S} w_a^1$ and $w_2(S) = \sum_{a \in S} w_a^2$. Let $\mathcal{K}$ denote a random scenario with all the possible realizations in K. The two-stage optimization problem can be formulated as follows.

$$\begin{aligned} \textbf{P5:} \quad & \min_{S \subseteq A} \; w_1(S) + \mathbb{E}_{\mathcal{K}}(f_k(S)) \\ & \text{where } f_k(S) = \min_{S_k \in A \setminus S} w_2(S_k) \\ & \qquad \text{s.t.} \quad \min_{p \in P} \gamma_p(S \cup S_k, k) \geq \lambda \end{aligned} \tag{2.11}$$

where the objective is to minimize the summation of the first stage cost and the expected second stage cost, with the expectation taken over all possible scenarios. For any scenario k that is realized in the second stage, additional APs are deployed, if needed, with the objective of minimizing the second stage cost while ensuring a

required average throughput under k. In general, both w_a^2 and λ can depend on k. But we focus on the above problem for the sake of simplicity. A dual problem that maximizes the expected throughout subject to a budget on the total (two stage) cost can be similarly defined.

We emphasize that minimizing the expected cost is *different* from minimizing the cost of the expected scenario. The latter problem reduces to the single scenario case once the expected scenario is identified, and Algorithm 1 can be readily applied. On the other hand, minimizing the expected cost is significantly more difficult. It is known that some significantly simplified stochastic problems for minimizing the expected cost are #P-hard even though their deterministic counterparts are polynomial time solvable [103].

A fundamental challenge in **P5** is due to the large number of possible scenarios, even if we discretize the scenarios and ignore the correlation in traffic distribution on nearby roads or APs. As a first step to address the challenge, we apply the sample average approximation method to reduce the infinite scenario problem to a polynomial-scenario problem. That is, a polynomial number of scenarios, denoted as $\mathcal{N}$, are first sampled by treating the distribution of scenarios as a black box. We then solved the sampled problem by replacing the objective function of **P5** with $w_1(S) + \frac{1}{N}\sum_{k\in\mathcal{N}} f_k(S)$, where $N = |\mathcal{N}|$. It has been proved that for a large class of 2-stage stochastic linear programs, a polynomial number of samples is sufficient to ensure that an ρ-approximation solution to the sample-average problem is an $(\rho+\kappa)$-approximation solution to the original problem [110, 109] for some constant $kappa > 0$, where the polynomial bound on N depends on the input size, the maximum ratio between the second stage cost and the first stage cost, and $1/\kappa$. Although this bound cannot be directly applied to our problem, we expect that the SAA method

provides a good performance for a reasonable number of samples, which is confirmed in simulations.

We then proceed to solve the polynomial-scenario problem, where we need to determine the initial deployment and the augmentation for each scenario in $\mathcal{N}$. As inspired by the stochastic set cover problem considered in [94], we extend our definition of $\gamma_p(S,k)$ for the single-scenario case as follows. First, $N+1$ copies are created for each AP. Let a^k denote the k-th copy of $a \in A$, with index $k \geq 1$ corresponds to the k-th scenario in $\mathcal{N}$, and index 0 corresponds to the initial deployment. The cost of a^k, denoted as $\tilde{w}_{a^k}$, is defined as w_a^1 if $k=0$ and $\frac{1}{N}w_a^2$ if $k \geq 1$. Let $\mathcal{A}$ denote the set of all the copies of APs. Any subset $\mathcal{S} \subseteq \mathcal{A}$ then indicates a solution to the polynomial-scenario problem, with the initial deployment defined as $S_0 = \{a : a^0 \in \mathcal{S}\}$ and the augmentation in k-th scenario defined as $S_k = \{a : a^k \in \mathcal{S}\}$. The cost of a solution $\mathcal{S}$ is then defined as $\tilde{w}(\mathcal{S}) = \sum_{a^k \in \mathcal{S}} \tilde{w}_{a^k} = w_1(S_0) + \frac{1}{N}\sum_{k=1}^{N} w_2(S_k)$, which is the summation of the first-stage cost and the expected second stage cost respecting $\mathcal{S}$. For any scenario $k \in \mathcal{N}$, we define $\tilde{\gamma}_p(\mathcal{S},k) = \gamma_p(S_0 \cup S_k, k)$. The polynomial-scenario problem can then be refined as

$$\begin{aligned} \textbf{P6:} \quad & \min_{\mathcal{S}\subseteq\mathcal{A}} \ \tilde{w}(\mathcal{S}) \\ & \text{s.t.} \ \min_{p\in P, k\in\mathcal{N}} \tilde{\gamma}_p(\mathcal{S},k) \geq \lambda \end{aligned} \tag{2.12}$$

Observe that **P6** has a similar form to **P3**. Based on this, we then define $\tilde{\gamma}(\mathcal{S},\lambda) = \sum_{p\in P,k\in\mathcal{N}} \min(\tilde{\gamma}_p(\mathcal{S},k),\lambda)$, and observe that $\tilde{\gamma}(\mathcal{S},\lambda)$ is again monotone submodular. Hence, Algorithm 1 can be applied to **P6** and achieves an approximation factor of $O(1) + \log(\max_{a\in A}\mathcal{R}'_a)$, where $\mathcal{R}'_a = \sum_{p\in P,k\in\mathcal{N}}\sum_{e\in E_p} r_e(\{a\},k)d_e/v_e(k)$ indicates the total throughput contributed by a single AP $a \in A$ across all the paths and all the scenarios in $\mathcal{N}$. Compared with the single scenario case, the approximation factor is worsen by an $O(\log N)$ factor. Therefore, although a larger N improves

sampling accuracy, it also incurs a worse approximation factor when solving the sampled problem. An interesting open problem is then to identify an optimal N that balances the two effects and optimizes the overall performance.

The above solution has a complexity depending on N. We then consider a simple heuristic with a lower complexity. The idea is to find the initial deployment by simply applying Algorithm 1 to the "mean" scenario k^0, where $v_e(k^0) = (v_e^1 + v_e^2)/2, h_e(k^0) = (h_e^1 + h_e^2)/2, \forall e \in E$, and $r_a(k^0) = (r_a^1 + r_a^2)/2, \forall a \in A$. Note the difference between k^0 and k_0 considered before. Also note that k^0 is the *expected* scenario when v_e, h_e, and r_a are independently and uniformly distributed in the corresponding intervals. We will compare this heuristic and the SAA based approach in the simulations.

2.4 Simulations

In this section, we evaluate our roadside AP deployment algorithms via numerical results and ns3-based simulations [114], using real road networks retrieved from 2008 Tiger/Line shapefiles [1]. We compare our robust optimization algorithms with two baseline algorithms to study the worst-case cost for achieving a required level of average throughput under uncertainty, as well as the level of QoS guarantee that can be provided under a budget constraint. We further compare the SAA based algorithm with two heuristics to study the expected deployment cost under the two-stage setting. Simulation results for contact opportunity in distance can be found in our online technical report [134].

2.4.1 Numerical Results

To understand the performance of our algorithms in a relatively large scale and under various parameter settings, we first resort to numerical study.

Figure 2.2(left) shows the road network used in our study. The network has 1802

road intersections and 2377 road segments. We assume each road segment has two lanes in the opposite directions and ignore the width of lanes. The travel speed of each segment is in the interval [10 m/s, 20 m/s]. Each road intersection is a candidate location for deploying APs with a data rate in the interval [5 Mbps,10 Mbps]. The coverage region at each candidate location is modeled using a sector based approach from [99], where each region is composed of 4 sectors of 90° with radius randomly selected from [150 m, 250 m], as shown in Figure 2.2 (right). Except in the two-stage setting discussed in Section 2.4.1, each AP has a unit cost. The set of movements P consists of 10000 paths randomly sampled from all the shortest paths of length at least 2km connecting two road intersections. For Algorithm 2, the parameter δ used in the binary search is set to 0.0005, and for Algorithm 3, the parameter τ is set to 0.01.

To simulate the traffic density on each road segment, we generate a movement file with 1000 mobile users moving in the network for 24 hours. A restricted random waypoint mobility model is considered. A user starts at a randomly selected road intersection a, and randomly picks another road intersection b of distance at least 2km away from a, and moves to b by following the shortest path connecting the two intersections. After reaching b, the user immediately picks a new destination c of 2km away, and moves towards c, and so on. The travel speed on each road segment is sampled from the corresponding interval. We then estimate the user density on each segment from the movement file.

Robust Average Throughput Optimization

We compare our algorithms with the following two baseline algorithms, where $\hat{A} \subseteq A$ denotes the set of coverage regions that touch at least one path in P:

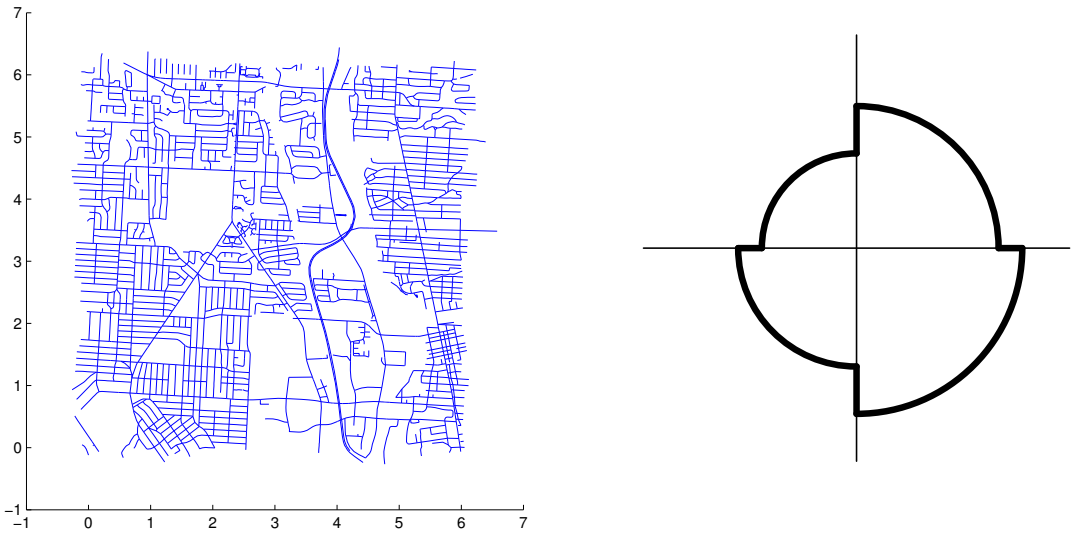


Figure 2.2: Left: A road network spanning an 6×6 km^2 region. Right: An instance of AP's coverage region with its boundary highlighted.

1. **Uniform random sampling** (Rand for short), which at each step randomly picks a new element from $\hat{A}$ until the required average throughput is obtained (for the minimum cost problem **P3**), or until the budget is reached (for the maximum coverage problem **P4**).

2. **Max-min distance sampling** [112] (Dist for short), which starts at a randomly selected location in $\hat{A}$, and at each step finds a new element from $\hat{A}$ that maximizes the minimum graph distance (in terms of shortest paths) from the elements already selected, until the required average throughput is obtained (for **P3**), or until the budget is reached (for **P4**).

Note that both algorithms involve randomness. In the simulation, each of them is repeated 100 times.

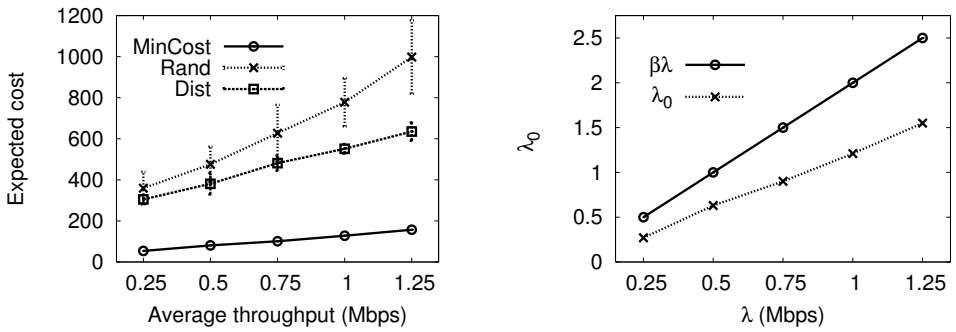
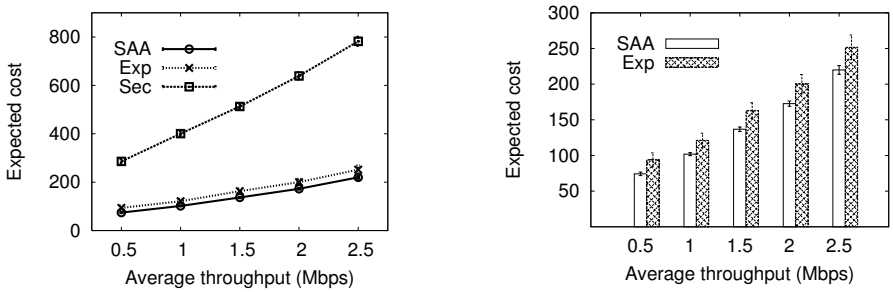


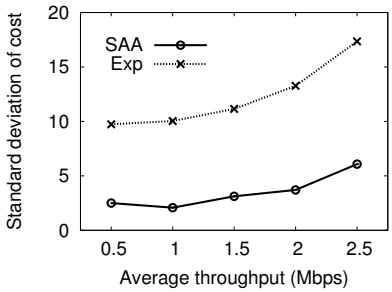
Figure 2.3: (a) Cost for achieving a required average throughput (across all the movements and all the scenarios). (b) Minimum λ_0 for getting a feasible solution in Algorithm 3.

We first study the performance of Algorithm 3 for **P3** for achieving robust average throughput under uncertainty (MinCost for short) . The performance of the "mean speed" scenario based algorithm for **P4** will be studied in ns-3 based simulations. Figure 2.3(a) shows the expected cost for achieving a required average throughput for a mobile user moving through any path in P and under any scenario, where the error bars again denote the standard deviations. We observe that our algorithm reduces the cost to less than 25% of the baseline cost, and random sampling again performs worst among the three algorithms. Figure 2.3(b) shows the minimum value of λ_0 in Algorithm 3 when the solution first becomes feasible (line 7 in the algorithm). As we shown in Section 2.3.2, such a λ_0 is upper bounded by $\beta\lambda$. Figure 2.3(b) verifies this result with $\beta = 20/10 = 2$. Moreover, it shows that λ_0 is actually bounded by 1.25λ in the simulation setting; hence, Algorithm 3 has a better performance than the theoretical bound.

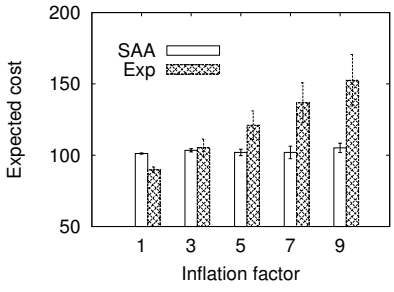


(a) Total Cost

(b) SAA vs Exp



(c) Standard Deviation



(d) Total Cost vs Inflation Factor

Figure 2.4: (a) Total cost for a required average throughput in two-stage deployment. (b) Total cost for SAA vs. Exp. (c) Standard deviation of total cost for SAA vs. Exp. (d) Total cost for a required average throughput of 1Mbps under various inflation factors.

Two-stage Stochastic Optimization

Finally, we study the performance of the SAA based algorithm (SAA for short) for minimizing the total cost for achieving a required average throughput in the two-stage setting. The scenario distribution is generated by assuming v_e, h_e, and r_a are independently and uniformly distributed in the corresponding intervals for all the road segments and all the APs. Each AP has a unit first stage cost, and a second cost determined by an inflation factor. We first generate 2000 samples of scenarios, and use 1000 of them as learning samples for the SAA based method, that is, the initial deployment is found for these samples using the polynomial-scenario extension

of Algorithm 1 presented in Section 2.3.3. Note that the sample size is relatively small compared with the network size and the number of movements considered. The rest 1000 samples are then used for testing, where in each scenario, the initial deployment is supplemented to meet the throughput requirement. This algorithm is compared with the following two heuristics:

1. **Expected scenario** (Exp for short), which is discussed in Section 2.3.3, where the initial deployment is found by directly applying Algorithm 1 to the "mean" scenario, which is then augmented for each of the 1000 testing samples using Algorithm 1.

2. **Second stage only** (Sec for short), which does not consider the first stage, and a new deployment is found for each testing sample using Algorithm 1.

We first consider a fixed inflation factor of 5 (hence each AP has a fixed second stage cost of 5). Figure 2.4(a) shows the total cost for achieving a required average throughput. The second stage only approach is clearly the worst among the three algorithms due to the high cost of the second stage. To see the performance of SAA and Exp clearly, their total cost and standard deviations are replotted in Figures 2.4(b) and (c), respectively. We observe that Exp performs 15% - 25% worse than SAA and suffers from a high standard deviation. Figure 2.4(d) further illustrates the performance of SAA and Exp for different inflation factors. We observe that SAA performs worse only when the inflation factor is close to 1. Actually, when the inflation factor is 1, that is, the two stages have the same cost, there is no benefit to have an initial deployment, and hence the second stage only approach is the best (not shown in the figure). For large inflation factors, SAA always performs better than Exp and the reduction in cost increases as the inflation factor becomes larger, which highlights the deficiency of using the "mean" scenario cost to estimate the expected cost. Moreover,

SAA has a stable performance under different inflation factors and a small standard deviation.

2.4.2 Ns-3 Simulations

We then conduct ns-3 based packet level simulations to further validate the performance of our algorithms. Our focus is on throughput maximization for a given budget under a randomly generated traffic scenario.

Simulation Setting

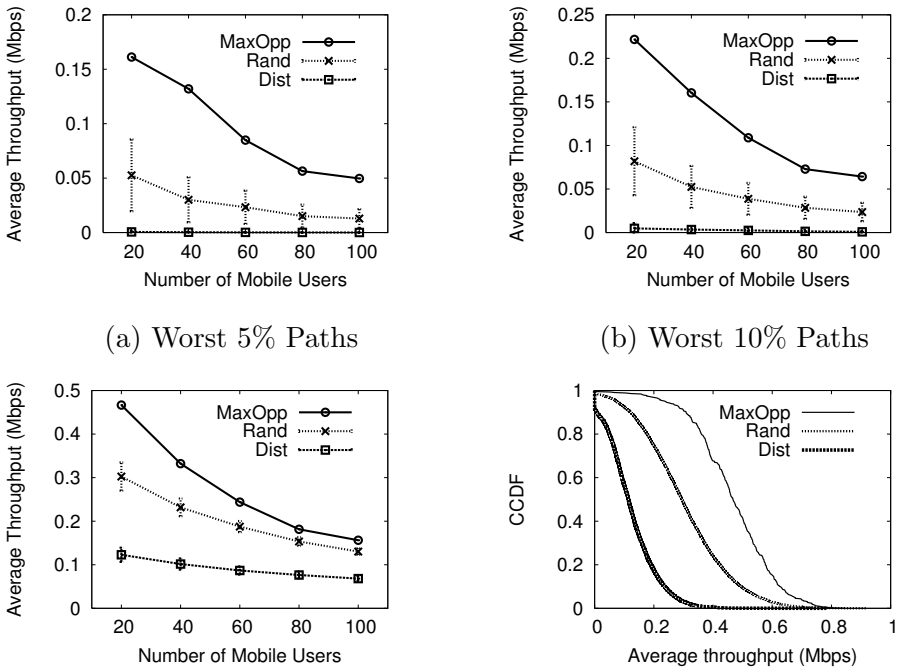
Due to the high overhead for simulating large scale mobility and data transmission in ns-3, we use a smaller road network (a 2km $\times$ 2km subregion in the same area as the large network with the same travel speed distribution). We fix the number of APs at 20 and vary the number of mobile users, denoted as K, between 20 and 100. For each K, we first generate a 24 hour movement file with K users as before. The movement file is then used to estimate the user density. We then run our "mean speed" scenario based algorithm for problem **P4** (MaxOpp for short) to generate a deployment, and the random sampling algorithm and the distance sampling algorithm to generate 20 deployments each.

In each simulation, 20 static nodes are set up as APs with their locations determined by a deployment file, and K mobile nodes are generated with their mobility determined by the movement file. The set of nodes are configured as follows. In the physical layer, we use the constant speed propagation delay model with the default speed (the speed of light), and the Friis propagation loss model [52]. We have extended the loss model to allow four different energy thresholds that match the communication ranges in the four directions as illustrated in Figure 2.2 (right). All the ranges are randomly sampled from the interval of [150m, 250m] as before. In

the MAC layer, 802.11g protocol is used with a constant data rate of 6 Mbps. Each AP has a different SSID, and APs that are close to each other are assigned different channels to avoid interference (ns-3 WiFi does not model cross-channel interference). Each mobile node is configured with multiple channels so that it can download data from any APs in range, but the association protocol ensures that a node is associated with at most one AP at any time. In the application layer, CBR traffics are generated from each AP to mobile users served by it. To reduce communication overhead, mobile nodes do not actively probe channels. They only wait for beacons from APs. Whenever a node encounters a new AP or is disassociated from an old AP, it chooses from the set of APs in range the one with the least number of users associated, where the tie is broken by giving higher priority to the newly encountered AP. An AP serves all the nodes associated with it in an equal data rate with the total rate bounded by 1 Mbps.

Simulation Result

In Figure 2.5(a), the average throughput for the bottom 5% of paths is plotted, where the average is taken over all these paths and over all the deployments. Figure 2.5(b) shows the similar results for the bottom 10% of paths. These figures illustrate the performance of the algorithms for less served paths. In both cases, our algorithms achieves more than 150% of higher performance. In addition to improving the worst-case performance, our algorithm also achieves significant higher throughput in the average sense as shown in Figure 2.5(c), where the average throughput over all the paths is plotted. Figure 2.5(d) plots the complementary cumulative distribution of throughput across all the paths for the 20 mobile user case. The figure shows that our algorithm not only achieves a better worst-case and average performance, but also dominates the baselines in the stochastic sense (roughly). From these figures,



(a) Worst 5% Paths

(b) Worst 10% Paths

(c) All Paths

(d) CCDF of All Paths

Figure 2.5: (a) Average throughput of the worst 5% paths vs. budget. (b) Average throughput of the worst 10% paths vs. budget. (c) Average throughput of all the paths vs. budget. (d) CCDF of average throughput across all the paths and deployments (20 mobile users).

we also observe that distance sampling performs worse than random sampling for throughput maximization, which is contrast to the case of cost minimization as we observed before. One explanation is that distance sampling distributes APs in a more uniform way and when the budget is low, it does not provide enough coverage to short movements.

2.5 Experimental Evaluation

We set up a small scale controlled experiment to better understand the performance of our approach. The experiment was carried out in a 180m $\times$ 120m parking lot located at the west campus of OSU and is free of potential interference from other WiFi networks. The experiment was usually carried out at night when the parking lot was empty. We artificially divided the parking lot area into a 6 by 4 grid and use it as a small road network. All the 24 intersections are treated as candidate locations for deploying APs.

A single mobile node carried by a car and 4 APs are used in the experiment. Each AP is a laptop equipped with an Orinoco 802.11b/g PC card and an external antenna mounted on a 1.7m high tripod so that the signal will not be blocked by the car in the test. The single mobile node is a laptop equipped with a Ubiquiti Networks SRC 802.11a/b/g PC card and two external antennas fixed at the two sides of the car. The transmission power of each AP is set to 6 dBm, which is tested to give an effective transmission distance of no more than 50 meters. Each node runs Ubuntu Linux with Linux 2.6.24 kernel and madwifi device driver for the 802.11 interface. The physical layer data rate of each node is fixed at 54Mbps.

A total of 5 random deployments are evaluated and compared with a deployment computed by Algorithm 2 for maximizing the contact opportunity in distance across the set of shortest paths between intersections of length at least 200m (there are 30 such paths in total), with a budget 4. The algorithm assumes that each AP has a unit cost and the coverage region of each AP is a disk of a radius 50m.

Because of the large volume of driving work and limited availability of that place, we picked 6 representative shortest paths that go through different parts and directions of the parking lot, and drove through each of them 3 times for each deployment. The moving speed is kept at about 10mph. When moving through a path, the mobile

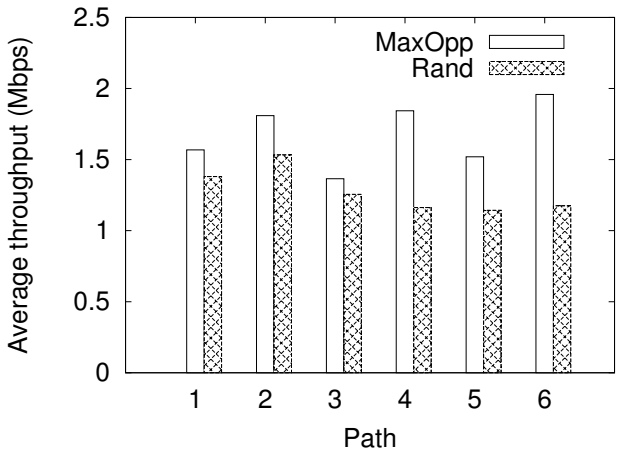


Figure 2.6: The average throughput of the 6 paths under evaluation, where Rand represents the average of 5 random deployments.

node attempts to associate with an AP with the strongest signal. Once associated, it downloads UDP packets from the AP until it is disconnected from that AP. The mobile node then finds another AP with the strongest signal to associate. Figure 2.6 shows the average throughput of each of the 6 paths. For random sampling, the average results across the five deployments are plotted. We observe that our solution achieves up to 66.7% higher throughput, and across all the 6 paths the average improvement in throughput is 26.4%. These results are promising and serve as a first step towards larger scale prototype deployment. Compared with simulations, the improvement is less significant mainly due to the following reasons. First, channel condition is not very stable in the outdoor environment as we observed, especially when we have to control the transmission radius to within a small range to make the small scale experiment meaningful. Second, both association and disassociation incur overhead in terms of packet loss and delay. Third, because the driving area is small

and there is only a single mobile user, even a random deployment has a high chance to cover the entire area with a reasonable data rate.

Chapter 3

ACHIEVING USER-LEVEL FAIRNESS IN OPEN-ACCESS FEMTOCELL BASED ARCHITECTURE

Cellular service providers usually set aside a fixed and arbitrarily chosen number of channels for use by its femtocell owners, and do not provide any guidelines for configuring the transmission power of arbitrarily deployed Femtocells. Such static solutions are neither scalable nor optimum for controlling the interference between multiple Femtocells, and between Femtocells and macro base-station (MBS). To achieve both high throughput and fairness among users, dynamic resource allocations have to be studied.

Our study considers two models in the solution. The non-interfering model (NINT model) assigns power levels to each Femtocell in such a way that the femtocells do not interfere with each other, allowing for independent scheduling of users within each femtocell. The more general interfering model (INT model) allows the femtocells to interfere but the sub-channel assignment disallows interfering links to simultaneously transmit in the same time slot and the same sub-channel.

Our study differs from the previous works on fairness in WLANs [26, 71, 74], which either address fairness by solely performing association control [26, 71], or by controlling the contention behavior of nodes in the IEEE 802.11 MAC layer [74].

3.1 Related Work

Due to its significance, resource allocation algorithms in OFDMA networks have been studied in many prior works. [39, 63] try to maximize the aggregate throughput, while [133, 132] aim to minimize power consumption. However, none of them address the fairness issue. Proportional fairness is considered in [102, 66, 88]. Resource allocations in those works are formulated as convex optimization problems, the objectives of which are to maximize sum of user rate. Proportional fairness is assured by imposing a set of constraints, and power assignment is either considered as a constraint, or is evenly allocated among all channels. Unlike those works, this work formulates the resource allocation problem using graph based approach.

Graph-based approaches such as [32] apply graph coloring technique to solve the fractional frequency assignment problem in OFDMA networks with homogeneous cell size, without considering heterogeneous cell size or fairness. [75] solves the subcarrier selection, transmission mode selection and relay selection problem for relay-assisted bidirectional OFDMA network, which is not suited for femtocell network. [130] develops optimal algorithms for resource allocation problem with user constraints. However, the objective of [130] is to maximize system throughput, which is essentially different from this work.

In the femtocell literature, femtocell solutions in the market are primarily UMTS and CDMA based, driven from a business perspective [47], that aim to improve indoor coverage using available backhaul (cable, DSL). However, as an emerging technology, the challenge of mitigating intra- and cross-tier interference is still critical in the current solutions [16, 31, 46]. Interference is usually addressed through power control [30, 37, 55]. [30] develops an uplink capacity analysis of a CDMA two-tier network. The authors show that interference avoidance can help achieve higher user capacity and avoid the design of protocols that require the mobile to sense

the spectrum during handoff. [37] discusses some key requirements for co-channel operation of femtocells such as auto-configuration and public access, and proposes a power control method that ensures a constant femtocell radius in the downlink and a low pre-definable uplink performance impact to the macrocells. A simulation of femtocells deployed in a residential scenario is studied in [55], which shows that the deployment of these femtocells would not pose a significant impact on the dropped call rate of mobile users. The uplink interference problem in co-channel deployed femtocell networks is studied in [129], which presents a trifecta of distributed algorithms, mainly focusing on protection of macrocell users.

OFDMA-based femtocells have been gaining increased attention recently. In OFDMA based femtocell solutions, intelligent sub-channel allocation is an alternative to power adjustment to mitigate interference while improving the system capacity. A coverage and interference analysis based on a realistic OFDMA macro/femtocell scenario is provided in [77], and some guidelines on how the spectrum allocation and interference mitigation problems can be approached are further discussed. [21] carries out experimental studies to characterize interference in OFDMA femtocell network. [119, 125] study the open and close access problem for OFDMA femtocells, and suggest to use limited access mode [119] or to adapt access mechanism based on average cellular user density. Energy efficiency problem was recently studied in [126, 73], which aims to achieve energy efficiency at femtocells [126], and maximize the lifetime of handsets [73]. Self-organizing frameworks are studied in [76, 17]. [76] proposes a two-phase self-organizing framework to minimize interference and maximize network capacity, while [76] assumes femtocells and macrocell work on the same channel, and applies a non-cooperative game approach to maximize weighted sum rate. Neither of them addressed the fairness issue.

Resource allocation was recently studied in [108, 20, 100, 54, 107, 22]. [108] proposes an adaptive resource scheduling algorithm for wireless relay OFDMA networks. [20] designs and implemens an uplink scheduler for OFDMA femtocells, without considering downlinks. [100] introduces an interference avoidance framework by letting femtocells utilize resource blocks occupied by far away mobile stations, without considering the fairness issue. [54] proposes a cluster-based resource allocation scheme, which first builds clusters, and then preforms optimal resource allocation for each cluster. However, power adjustment is not considered in [54]. In [107], the authors propose a dynamic resource allocation mechanism between macro and femtocells to achieve proportional fairness among users. [22] proposes a femtocell resource management system that divides one OFDMA frame into two zones – the reuse zone and the isolation zone, which also categorizes users into two groups, correspondingly. Users in reuse zone will be simultaneously active, and deal with interference through link adaptation, while users in the isolation zone are isolated via resource allocation (based on weighted max-min fairness). However, [107] and [22] only consider fixed power level and coarse resource allocation strategies (on per-femtocell basis) regardless of the possible variations of user density in different femtocells, all of which, in contrast, are considered in this work. Alternatively, this work could serve as a complementary work for the resource isolation part of [22], when power adjustment or user density is available in the system.

3.2 Problem Statement

3.2.1 Notations

Consider a single macrocell base station (MBS) deployed in a 2D region that contains M Femtocells $\mathcal{F} = \{f_1, \cdots f_M\}$. $\mathcal{U} = \{u_1 ... u_N\}$ is the set of N users in the system.

The location of user u_i is given by $L(u_i) = (x(u_i), y(u_i))$. The location of Femtocell f_j is given by $L(f_j) = (x(f_j), y(f_j))$. Each Femtocell can operate at a power level chosen from the set $\mathcal{P}$ indexed by $\{1 \cdots l\}$, where the selected power for index k is given by $p(k)$, and $p(k) < p(k')$ for $k < k'$. For any Femtocell f_i associated with power level $p(k)$, we define the transmission range of f_i at power level $p(k)$ as the range within which the received signal strength from f_i is higher than some threshold RSS_{tx}. Formally, let r_k^{tx} denote this range, and $rss(f_i, k, d)$ denote the received signal strength at distance d from f_i sending beacons at power level $p(k)$, then $r_k^{tx} = \max\{d : rss(f_i, k, d) \geq RSS_{tx}\}$. Similarly, we define the interference range of f_i at power level $p(k)$ as the range within which a receiver associated with another Femtocell will receive an interference level from f_i that is higher than some other threshold RSS_{int} ($RSS_{int} \leq RSS_{tx}$), i.e., $r_k^{int} = \max\{d : rss(f_i, k, d) \geq RSS_{int}\}$. We assume that the MBS when transmitting interferes with all other nodes in the system. Let $\mu(j, l)$ be the number of users within the transmission range of Femtocell j when operating at power level l. To simplify the presentation, we assume that the interference range of all Femtocells at their highest power levels are fully contained within the macrocell's boundary.

Let ρ_j be the power level selected by femtocell j. $\rho_j \in \mathcal{P} \cup \{0\}$, where $\rho_j = 0$ implies that Femtocell j is not active due to interference. The $N \times M$ matrix $\mathcal{B}$ represents the association of N users to the M femtocells. $\mathcal{B}_{ij} = 1$ if user i is associated with femtocell j, otherwise it is 0. Each user associates with at most one femtocell. Users not associated with any femtocell associate with the macrocell.

Consider an OFDMA frame as in Figure 3.1. Tiles are to be allocated in the system. Spatial reuse of the tiles is possible among Femtocells. A fraction of time is allocated for uplink and another fraction for downlink communication. For simplicity, the discussion focuses only on the downlink, but the same framework can be applied

to extend the solutions for uplink transmissions. Each downlink frame has t time-slots and c sub-channels. The tiles are further divided between the macrocell and the femtocells. A feasible allocation $\mathcal{A}$ is an assignment of a subset of links (Femtocell to user or MBS to user) to each tile, such that the links assigned to any given tile does not interfere with each other.

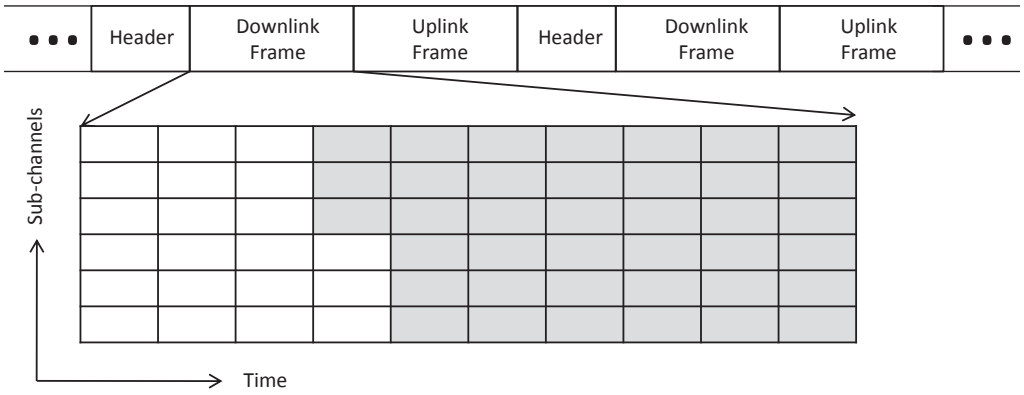


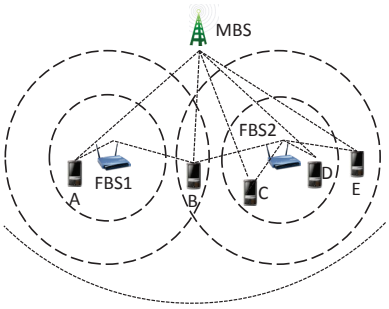
Figure 3.1: OFDMA Frame Structure [107]: Gray tiles are for Femtocells and the white for MBS. The header contains information on the allocation of the tiles.

3.2.2 Problem Statement

Our solution for resource allocation will determine multiple parameters: 1) the power level selected for each femtocell ($\vec{\rho}$); 2) the association of users to femtocells or the macro cell ($\mathcal{B}$); and, 3) assignment of tiles to the macrocell or multiple femtocells ($\mathcal{A}$). We focus on fair rate allocation among the users, and seek to compute the optimum *maxmin* rate assignment problem.

3.3 Resource Allocation with NonInterfering Femto-Cells

3.3.1 The NonInterfering Model



FBS1 power	FBS2 power	#Users (FBS1)	#Users (FBS2)	#Users (Macro)	Arg of Eq. 4	τ_m	τ_f	Throughput Vector (A,B,C,D,E)
Lo	Lo	1 (A)	2 (C,D)	2 (B,E)	3 – 2 = 1	1/2	1/2	1/2, 1/4, 1/4, 1/4, 1/4
Lo	Hi	1 (A)	4(B,C,D,E)	0	5 – 4 = 1	0	1	1, 1/4, 1/4, 1/4, 1/4
Hi	Lo	2 (A, B)	2 (C,D)	1 (E)	4 – 2 = 2	1/3	2/3	1/3, 1/3, 1/3, 1/3, 1/3
Hi	Hi	Not feasible in the non-interfering model						

Figure 3.2: Resource Allocation: The dotted lines show the various possible communication ranges. The τ_m that determines the optimum *maxmin* rate for the users is shown for each power combination. The (Hi,Lo) combination with a fractional allocation of 1/3 tiles for the MBS optimizes the objective.

Under this model, the solution will ensure that the femtocells are not interfering when operating on the same channel by adjusting their power levels, i.e., there are no overlapping femtocells in the final solution. Thus under this model, all femtocells will be simultaneously active in all the tiles allocated to femtocells, and inactive in all the tiles allocated to the macrocell. So, all users in the same femtocell will have equal rate allocation (by round robin of the tiles allocated for femtocells), and similarly, users that are served by the macrocell will have equal rate allocation. Let τ_f be the fraction of tiles allocated to the Femtocells, and $\tau_m = 1 - \tau_f$ be the fraction allocated

to the macrocell. Under the constraint of non-interfering femtocells in the solution, without loss of generality, we consider the following objective function [1]:

$$\mathbf{P1}: \max_{\mathcal{B},\vec{\rho},\tau_f} \min_{1\leq i\leq N} r_i \tag{3.1}$$

where r_i is the fraction of tiles allocated to user i from the downlink frame, which represents the effective data rate of user i.

Consider a user that is in range of an Femtocell in the final power allocation. If it associates with the MBS, then that slot will become busy for *all* Femtocells. However, if it associates with that Femtocell, then other Femtocells can also be active and reuse that slot (since all femtocells are non-overlapping in the solution). Thus associating with that Femtocell is the optimum decision. This implies that the association matrix $\mathcal{B}$ has been implicitly solved: for users that are within the range of some Femtocell, they will associate with the corresponding Femtocell, otherwise, associate with the MBS. So, $P1$ could be simplified to the following without loss of generality:

$$\max_{\vec{\rho},\tau_f} \min_{1\leq i\leq N} r_i \tag{3.2}$$

From this solution, the optimum association ($\mathcal{B}^*$) can be derived as follows. A user in range of an Femtocell operating at its computed power level will be associated with that Femtocell. All other users will be allocated to the MBS.

In the optimum solution, as the Femtocells will equally divide the τ_f among its users, the Femtocell associated with the maximum number of users (bottleneck Femtocell) will serve the lowest data rate. As the users served by the bottleneck Femtocell and the users served by the MBS occupy different time slots, the optimum allocation must provide equal rate to all such users. Recall that $\mu(j, \rho(j))$ is the number of users in the coverage range of Femtocell j operating at power level $\rho(j)$. Thus, the total

[1] Alternatively, our solutions can be applied to the weighted version of *maxmin* fairness problem

number of users served by the MBS ($N - \sum_{j=1}^{M} \mu(j, \rho(j))$) and the maximum number of users among the Femtocells ($\max_{j\in[1,M]}\{\mu(j, \rho(j))\}$) will together determine the allocation. Note that all these users will be served in different tiles and so, their minimum fractional rate will be $r_i = 1/(N - \sum_{j=1}^{M} \mu(j, \rho(j)) + \max_{j\in[1,M]} \mu(j, \rho(j)))$. So the optimization objective can be rewritten as:

$$\max_{\vec{\rho}}\{\frac{1}{N - \sum_{j=1}^{M} \mu(j, \rho(j)) + \max_{j\in[1,M]} \mu(j, \rho(j))}\} \tag{3.3}$$

$$= \max_{\vec{\rho}}\{\sum_{j=1}^{M} \mu(j, \rho(j)) - \max_{j\in[1,M]} \mu(j, \rho(j))\} \tag{3.4}$$

and τ_m^* and τ_f^* can be uniquely determined by

$$\tau_m^* = \frac{N - \sum_{j=1}^{M} \mu(j, \rho(j))}{N - \sum_{j=1}^{M} \mu(j, \rho(j)) + \max_{j\in[1,M]} \mu(j, \rho^*(j))} \tag{3.5}$$

$$\tau_f^* = 1 - \tau_m^* \tag{3.6}$$

After the simplifications, the resulting objective (Equation 3.4) has only one variable ($\vec{\rho}$) in the outer *max* operator, which makes it easier to design solutions.

An Example (Figure 3.2): Please be noticed that the transmission and interference ranges are shown as circular and identical in some examples of this work for simplicity of discussion. However, these assumptions are not required in our solutions. In Figure 3.2, each of the two femtocells has two power levels. The zero power level is not shown for simplicity as it leads to lower *maxmin* rate than the other combinations shown in the figure. For each power combination the optimum τ_m and the corresponding rates for all users are shown. The (Hi, Lo) combination of power levels for the two femtocells leads to the optimum *maxmin* rate vector $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$. Here, $\sum_{j=1}^{M} \mu(j, \rho(j)) = 4$ and $\max_{j\in[1,M]} \mu(j, \rho(j)) = 2$. The argument of (3.4) is $4 - 2 = 2$. By using the expression for τ_m^*, we get $\tau_m^* = 1/3$. The macrocell is serving 1 user (E) which gets a rate of 1/3. As $\tau_f^* = 2/3$, and both femtocells are serving two users each, all users are assigned a rate of 1/3.

Theorem 3.3.1. *The maxmin rate allocation problem P1 is NP-hard.*

Proof. We reduce the Maximum Independent Set problem for unit disks in a plane (MIS-DISKS), which is known to be NP-hard [35], to P1. In MIS-DISKS, the objective is to select a maximum subset of non-overlapping disks.

Given an instance of the MIS-DISKS problem with M disks, we construct an instance of P1. Each disk corresponds to a femtocell with its femto-BS situated at the center of that disk. Each femtocell has two power levels, zero and unit power level, where the latter corresponds to a unit transmission range and unit interference range. An additional femtocell f is added that does not overlap with any other femtocell (See Figure 3.3). A macro-BS is added with a large enough coverage range that includes the covered regions of all the femtocells.

Now consider a 2D lattice of points in the plane with a sufficiently high density (to be determined later). The lattice density will be chosen in such a way that the number of points within a unit disk is within a fixed range, say, $[K, K+\gamma]$, where $\gamma < \frac{K}{M}$. Each lattice point overlapping with any of the M femtocells will correspond to a user. In addition, $K+\gamma+1$ users are placed at any location within the range of femtocell f, thus making femtocell f the femtocell with the highest number of users.

If f is not in the optimum solution of the instance of P1, it can be added to increase the first term of expression Equation 4 with a lesser increase to the second term, leading to a resultant increase of the objective. So in the optimum solution to P1, f must be operating at unit power and the second term will have a value of $K+\gamma+1$.

Let S' be the set of disks corresponding to the femtocells other than f, that has a non-zero power allocation in the solution to P1. We claim that S' is a solution to the MIS problem. For the sake of contradition, let us assume that the optimum solution to the MIS problem, S, is such that $|S| > |S'|$. As the total number of users

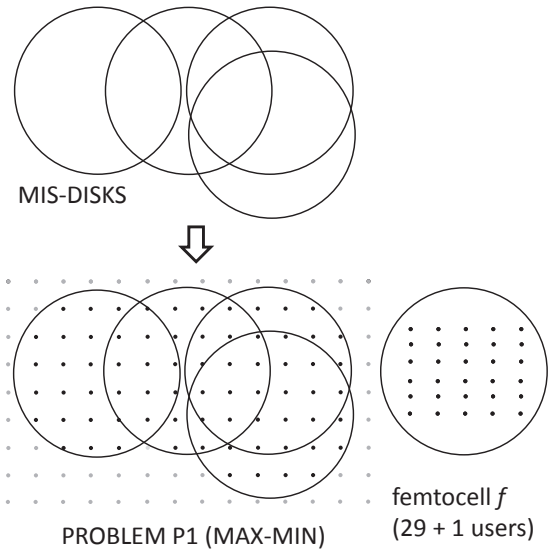


Figure 3.3: Reduction for NP hardness. Here the radius of the circle is 3 times d, and $\eta(3)$ is known to be 29 [53]. So the additional femtocell f has 29+1 = 30 users. The dark dots represent the users. The gray dots are the lattice points outside the disks that were not selected to represent users.

in range of the femtocells corresponding to S' is maximized, $K|S| \leq (K+\gamma)|S'|$. Therefore, $\gamma \geq \frac{|S|-|S'|}{|S'|}K \geq \frac{|S|-|S'|}{M}K \geq \frac{K}{M}$. But in our construction $\gamma < \frac{K}{M}$, which is a contradiction. Thus, $|S| \leq |S'|$, implying that S' is a solution to the MIS problem.

Now we choose the appropriate value of lattice distance d ($d < 1$) such that the number of points within a unit disk is within the range $[K, K+\gamma]$. We say that a disk is a lattice-disk if its center coincides with a lattice point. If r is the ratio of the radius of the disk to the lattice distance, then using the Gauss' circle formula, the number of lattice points contained in it is represented as $\eta(r) = \pi r^2 + O(r)$ [53]. If the center of a unit disk is not aligned to a lattice point (Figure 3.4), then the number of lattice points will be within a range, $[K, K+\gamma]$. The nearest lattice point to any point on the plane is atmost at a distance of $\frac{d}{\sqrt{2}}$. So, centerd at that nearest lattice point, a

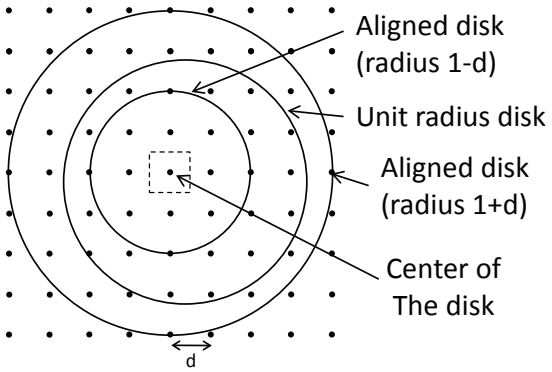


Figure 3.4: Gauss's Circle Problem for Non-Lattice-disks: Alligned disks are lattice-disks. The square represents the region closest to the point at the center of the square.

lattice-disk of radius $1-d$ is fully contained within the unit disk, and a lattice-disk of radius $1+d$ will fully contain the unit disk. So the minimum number of lattice points for a unit disk, K will be atleast $\eta(\frac{1-d}{d})$, i.e., $K \geq \eta(\frac{1-d}{d}) = \pi(\frac{1}{d}-1)^2 + O(\frac{1}{d})$. Similarly, $K+\gamma$ will be atmost $\eta(\frac{1+d}{d})$, i.e., $K+\gamma \leq \eta(\frac{1+d}{d}) = \pi(\frac{1}{d}+1)^2 + O(\frac{1}{d})$. Therefore, $\gamma \leq \pi(\frac{1}{d}+1)^2 - \pi(\frac{1}{d}-1)^2 + O(\frac{1}{d}) = 4\pi(\frac{1}{d}) + O(\frac{1}{d})$. As $\frac{1}{d}$ increases, K grows quadratically but γ grows linearly. So for a sufficiently high value of $\frac{1}{d}$ (depends on M and the constants in $O(.)$), K will exceed γM, or, $\gamma < \frac{K}{M}$.

K and γ will both be polynomials in M. So, the total number of users created in this reduction is polynomial and the reduction is polynomial time, thus completing the proof. □

3.3.2 Centralized Resource Allocation (NINT)

A central server periodically collects topology information from all the users, computes the optimal solution and informs the Femtocells and the users. A discussion of constructing conflict graphs is presented in section 3.6, and overhead messages

are described and counted in the simulation section. Observe that if the maximum weight Femtocell is known in an optimal solution, i.e., the $\max_{j\in[1,M]} \mu(j, \rho(j))$ term is known in formulation (3.4), we can then solve the problem by solving an instance of the MWIS (maximum weight independent set) problem. We will explore all possible values of that term to arrive at the optimum solution. Our approach is to first model the constraints using a conflict graph and then solve multiple instances of the resulting MWIS problems (Algorithm 4).

First we create the conflict graph for the Femtocells considering the various power levels. A node is created for each combination of power level and Femtocell ID. The *weight* of this node is counted by the number of users within the transmission range of the Femtocell at the chosen power level. Of course, some extra information needs to be stored for this node, such as Femtocell ID, location, power level, and the weight. If the transmission range of one node, i.e., the transmission range of this Femtocell at corresponding power level, overlaps with the interference range of another node, an edge is added between two such nodes. Observe that nodes corresponding to the same femtocell, but for different power levels, will form a clique. As the topology of the conflict graph depends only on power levels and locations of Femtocells, we only need to update the weight of each node at runtime, thus the overhead is relatively low. Figure 3.5 shows the conflict graph for Figure 3.2.

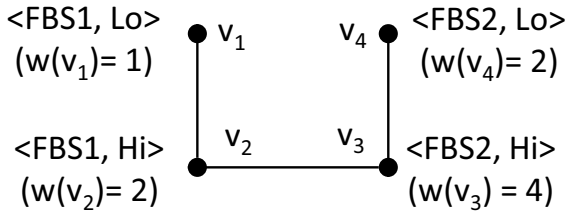


Figure 3.5: Conflict Graph for scenario in Figure 3.2

Note that by substituting the nubmer of users with the summation of normalized weight of users, this model can be easily modified to work for the *weighted maxmin* fairness problem.

The centralized resource allocation algorithm (Algorithm 4) uses the conflict graph to compute the power allocation ($\vec{\rho}$) for all Femtocells based on (3.4). The variables Q and $nmax$ are used to keep track of the independent set and the maximum weight of femtocells in that independent set, respectively (Lines 3-4). Set S is an enumeration of weights of all nodes (Line 5). Next, the independent set of nodes that maximize expression (3.4) is computed by trying all possible values of the second term in (4) (Lines 6-15). In each iteration, the variable s takes on values from the set S. Only the nodes with weight at most s is considered for constructing the induced subgraph G'. With a slight overuse of the term w, we use $w(I)$ to indicate the total weight of all nodes in the set I. For this induced subgraph the max weight independent set is computed, and stored if it is the best thus far. The optimum independent set is then used to compute the power allocation (Lines 16-22). Note that as shown in the previous section, using ρ^*, τ_f^* and $\mathcal{B}^*$ can be computed.

This algorithm gives the optimum solution to the *maxmin* problem if the MWIS can be exactly computed. However, MWIS is a NP-hard problem [115]. We can use a polynomial time approximation approach for the MWIS problem in line 10 . For example, a greedy algorithm that finds a maximal independent set can be used, which has a complexity of $O(M^2)$. Then from line $8, 9, 10$, the complexity of algorithm 4 can be denoted by $M * (M + M^2 + O(M^2))$, which is $O(M^3)$.

Regardless which algorithm we use for the MWIS problem in line 10, we can always achieve a bound of $max\{\beta, 1/N\}$, where β is the fraction of users (among all users) that are not covered by any femtocell, and N is the total number of users.

Theorem 3.3.2. *If βN users are outside the range of any femtocells, where $0 \leq \beta \leq$*

Algorithm 4: Centralized Resource Allocation (NINT)

input: conflict graph G

output: $\vec{\rho}$: power allocation vector for Femtocells

$Q \leftarrow \Phi$ // maximum independent set

$nmax \leftarrow 0$ // maximum weight of Femtocells from Q

$S \leftarrow$ set of possible values for the #users in a femtocell

foreach s *in* S **do**

    create an empty graph G'

    $V(G') \leftarrow \{v | v \in V(G) \text{ s.t. } w(v) \leq s\}$

    $E(G') \leftarrow$ edges induced by $V(G')$ in G

    $I \leftarrow$ max weight independent set of G'

    if $w(I) - s > w(Q) - nmax$ **then**

        $Q \leftarrow I$

        $nmax \leftarrow s$

for $j \in (1..M)$ **do**

    if $\exists q \in Q$ *s.t.* $id(q) = j$ **then**

        $\rho(j) \leftarrow level(q)$ //set this Femtocell the stored power level of node q.

    else

        $\rho(j) \leftarrow 0$

1, *and an approximation algorithm is used for the MWIS subproblem in Algorithm 4, then the lowest rate allocated by this algorithm will be at least* $max\{\beta, 1/N\}$ *times the lowest rate in the optimum rate allocation.*

Proof. Let the value of $\sum_{j=1}^{M} \mu(j, \rho(j))$ be A^{opt}, and the value of $\max_{j \in [1,M]} \mu(j, \rho(j))$ be B^{opt} in the optimum solution. Then the minimum rate allocated by the optimum solution to any user will be $r^{opt} = \frac{1}{N - A^{opt} + B^{opt}}$. Since algorithm 4 tries all possible

values of $\max_{j\in[1,M]} \mu(j,\rho(j))$. When it uses the value of B^{opt} as largest(s in Line 6 of Algorithm 4), let the computed value of $\sum_{j\in[1,M]} \mu(j,\rho(j))$ ($w(I)$ in Lines 10-11) be A. So, the minimum throughput computed by our algorithm is $r \geq \frac{1}{N-A+B^{opt}}$. So,

$$\frac{r}{r^{opt}} \geq \frac{N-A^{opt}+B^{opt}}{N-A+B^{opt}} > \frac{N-A^{opt}}{N-A} > \frac{N-A^{opt}}{N} \tag{3.7}$$

As $A^{opt} \leq (1-\beta)N$, we have,

$$\frac{r}{r^{opt}} > \frac{N-(1-\beta)N}{N} = \beta \tag{3.8}$$

On the other hand,

$$\frac{r}{r^{opt}} \geq \frac{N-A^{opt}+B^{opt}}{N-A+B^{opt}} \geq \frac{B^{opt}}{N-A+B^{opt}} \geq \frac{B^{opt}}{N} \geq \frac{1}{N} \tag{3.9}$$

Therefore, the algorithm is bounded by $max\{\beta, 1/N\}$. □

3.3.3 Distributed Approach (NINT)

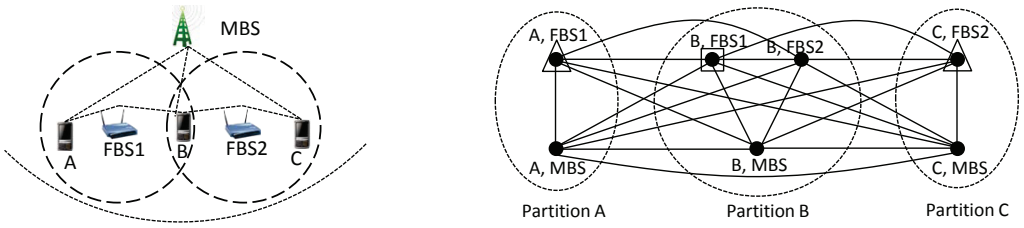
In the distributed algorithm (Algorithm 5) each Femtocell attempts to increase or decrease its power level and evaluate its impact within the local neighborhood to determine the best action to take. We assume the cost of exchanging messages among neighboring femtocells is negligible by using the broadband backbone. The impact is evaluated by the change to the argument of Formula (3.4). If some users have left the Femtocell, then the Femtocell attempts to increase its power level (Lines 3-11). It needs to obtain updated information on the number of users that can be supported if a higher power level is used by Femtocell j (Lines 5-7). Then the best power level is selected based on the argument to (3.4). If the number of users currently being served is the highest in the neighborhood then reducing it could possibly lead to increase in the argument to (3.4). For such a scenario, all power levels lower than the current one is explored in consultation with the interfering neighbors and the best power level is then selected (Lines 12-19).

Algorithm 5: Distributed Resource Allocation (NINT) at Femtocell f_j

input: power level $\rho(j)$; number of users $\mu(j, \rho(j))$; current $\rho(m)$ and $\mu(m, \rho(m))$ for each interfering neighbor m.
output: power level $\rho(j)$
if *some user(s) have left the femtocell* **then**
    foreach *power level k higher than the current* **do**
        foreach *interfering Femtocell* f_m **do**
            obtain $\mu(m, \rho(m))$ from f_m for highest feasible ρ if f_j switches to level k
        compute adjustment to arg of (3.4)
    select power level with max increase to arg (3.4)
if $\mu(j, \rho(j))$ *is the highest in the neighborhood* **then**
    foreach *power level k smaller than the current* **do**
        foreach *interfering Femtocell* f_m **do**
            obtain $\mu(m, \rho(m))$ from f_m for highest feasible ρ if f_j switches to level k
        compute adjustment to arg of (3.4)
    select power level with max increase to arg (3.4)

3.4 Resource Allocation with Interfering Femto-Cells

Allowing for interfering femtocells in the solution can lead to higher throughput. Consider the scenario of Figure 3.6(a), if we apply the NINT model, then in one of the optimum solutions, Femtocell1 will be selected and the power level for Femtocell2 will be set to 0. As a result the optimum value of τ_f is 2/3, and the corresponding rate vector is $\langle 1/3, 1/3, 1/3 \rangle$. However, if we allow for interference, then τ_f can be set to 1 (i.e, macrocell is not active), and Femtocell1 can transmit to A and Femtocell2 can transmit to C simultaneously for half of the tiles. In the remaining half, Femtocell1 can transmit to B, thus resulting in a rate vector $\langle 1/2, 1/2, 1/2 \rangle$.



(a) Topology

(b) LCG and Partition Coloring Problem

Figure 3.6: Operating Femtocells under interference (a) Two interfering femtocells (b) The link-conflict graph colored with two colors (triangle and square).

This kind of scheduling problem can be solved by constructing a conflict graph of links (we call it link-conflict graph or LCG to avoid confusion) and by performing a node coloring of this graph. However, our problem is quite different from prior work on scheduling flows in ad-hoc networks [79, 85] as it involves power assignment, and a macro base-station. Each node in the LCG represents a communication link. If an Femtocell can communicate with a user at more than one power level, the miminum power level allowing the communication is selected. So, for a given user, there will be multiple nodes corresponding to links of various Femtocells and the MBS. Such nodes corresponding to a single user will form one group, such that each group corresponds to exactly one user. The nodes in the LCG are thus partitioned into groups. All communication links that cannot be active simultaneously (due to interference or conflicting user occupancy), will be connected with edges in LCG. For example, nodes in the same group form a clique.

Constructing such kind of link conflict graph is outside the scope of this work. We recommend to use techniques from [18] which can do the job within milliseconds. This procedure can be even faster considering that some users will stay in the same

place within a short interval, thus only part of the graph needs to be updated on the fly.

Unlike the NINT model, the LCG based INT model allows a single Femtocell to transmit at different power levels to different users. Also, the allocation is more fine-grained as each tile is allocated to a specific user. Whereas in the NINT model, only the portions of tiles to macrocell and Femtocells(τ_m,τ_f) are determined. *Suppose we color the graph with the least number of colors such that at least one node in each group is colored.* We name this problem as *partition coloring problem*. The colors represent tiles when the corresponding subset of links will be activated. Minimizing the colors is equivalent to minimizing the number of tiles needed to transmit one unit of data to each user node. If we focus on the optimal solutions that have repeated schedules and serve one user with exactly one tile within a schedule cycle, then we have maximized the minimum throughput by solving the partition coloring problem.

Definition 3.4.1. Partition Coloring Problem: *Consider a graph $G = (V, E)$ with nodes partitioned into x groups $g_1 \cdots g_x$. Compute a color assignment that assigns a color to exactly one node in each group, such that nodes with the same color do not share an edge, and the number of colors is minimized.*

The partition coloring problem for the scenario in Figure 3.6(a) is shown in Figure 3.6(b). The triangles and the square represent two colors which correspond to two tiles in the optimum solution. By repeating this tile assignment for each pair of tiles, we can achieve the rate vector $\langle 1/2, 1/2, 1/2 \rangle$, for the three users. This also corresponds to the result discussed earlier in this section.

3.4.1 Centralized Algorithm (INT)

Algorithm 6 colors the partition graph by repeatedly picking up a maximal independent set and assigning the lowest color (or tile) to this set. After that, the partitions

of all the nodes in the sets are removed from the partition graph (Lines $5-11$). After the coloring, it tries to reuse some colors on some nodes following the *maxmin* metric (Line 12-19).

Observe that by controlling the elements of each independent set and the portion of colors assigned to each set, the *weighted maxmin* fairness can also be addressed by the INT model.

Algorithm 6: Centralized Partition Coloring (INT)

```
1  Input: Graph G(V, E) with N partitions denoted as V_1...V_N, ⋃_{i=1}^{N} V_i = V
2  Output: Colored V^C ⊆ V s.t. V^C ∩ V_i = 1 ∀i ∈ [1, N]
3  t ← 0
4  V^C ← Φ
5  while V is not an empty set do
6      t ← t + 1
7      Pick a maximal independent set V^M ⊆ V
8      Assign color t to all vertices v ∈ V^M.
9      Remove V_i from V, ∀V_i ∩ V^M ≠ Φ, and remove all edges that have at least one
       end point in V_i
10     V^C ← V^C ∪ V^M
11 for color i ← 1 to t do
12     Sort v ∈ V^C in increasing order by # colors of v
13     foreach v ∈ V^C do
14         if None of v and its neighbors has been colored by i then
15             Color v with i
```

Algorithm 6 takes at most $1 + \Delta$ colors to color G where Δ is the maximum

degree of nodes in V without considering intra-partition edges. The proof for this assertion is similar to the proof for the bound on any greedy algorithm for proper vertex coloring [69]. Further, as partition coloring is a generalization of proper vertex coloring, therefore, it is not possible to design a polynomial time algorithm that guarantees coloring V in less than $1 + \Delta$ colors [69].

3.4.2 Localized Implementation (INT)

In this subsection, we propose an *incremental, localized* implementation for the coloring assignment problem, which can lower down the system overhead and insertion time of a new user. We follow our previous assumption in the NINT model that the cost of exchanging messages among neighboring femtocells is negligible by using the broadband backbone.

Let us call the nodes in link-conflict graph G that represent the links between Femtocell f_j and its users as *the nodes of* f_j. We define the local link-conflict graph G_L^j of f_j as a subgraph of G, which only involves the *nodes of* f_j and other nodes (edges) that conflict with (incident on) these nodes. When a new user comes to f_j, it will show up as a new partition in G_L^j.

Initially, the centralized algorithm will be called. Let t be the returned number of colors. The localized implementation (Algorithm 7) works as follows. Whenever a user moves away from the transmission range of its Femtocell, it randomly selects a *proxy Femtocell* at its new location, which will help the user in securing a new time slot. Let f_j be the *proxy Femtocell*. If some color $i \in [1, t]$ is available, i.e., assigning this color will not cause conflict in G_L^j, it will assign the color and return (Line $3-4$). If not, the local adjustment among neighborhood will be triggered by f_j. To that end, f_j and its neighbors will first free all their assigned colors (Line 6), this results in some partitions (users) previously served by these Femtocells becoming uncolored.

Then the same technique as in the centralized algorithm will be explored (Line 8), i.e., for each color $i \in [1, t]$, find a maximum independent set in the local conflict graph that can be colored by i. Color i is assigned to this set and the local link-conflict graph is updated. Finally, if this algorithm is not able to color all partitions in the neighborhood, then the centralized algorithm is called (Line 12).

Algorithm 7: Localized Coloring (INT) at Proxy f_j

```
1  Input: Local Partition Graph of f_j, new user u
2  Output: A new schedule with all local partitions colored
3  if Some color i ∈ [1,t] can be assigned to the node corresponding to (f_j,u) in G_L^j
   then
4      Color it with i
5  else
6      f_j and its neighbors free all assigned colors, flag corresponding partitions in their
       partition graphs as uncolored
7      for color i from 1 to t do
8          Exploit the same techniques as in centralized algorithm to color all uncolored
           local partitions
9  if not all local partitions are colored then
10     Call the centralized algorithm
```

3.5 Simulations

We compare our solutions with two baseline algorithms and evaluate the minimum throughput, average throughput, and the impact to throughput due to factors such as femtocell density, the arrival rate and speed of users.

3.5.1 Simulation Settings

Our simulations are conducted using the open source LTE-EPC Network Simulator (LENA) [118] derived from the ns-3 project. LENA implements a spectrum framework based on the LTE spectrum model as described in 3GPP TS36.101 [116], which allows the use of different spectrum models for different types of cells. Specifically, it uses an outdoor propagation model for macrocells and an indoor propagation model for femtocells. A trace-based Jakes fading model based on 3GPP TS36.104 [117] was also included. The typical parameters for the fading model were varied depending on the user's speed for both the pedestrian and vehicular scenarios as specified in Annex B.2 of 3GPP TS36.104. A square region of $800m \times 800m$ is considered in the simulation. A macrocell of height $20m$ is placed at the center of this region with a transmission range of $600m$ that allows full coverage of the area. Femtocells are deployed indoor at randomly chosen locations. The size of each building is $10m \times 10m$, and femtocells are placed at the center of those buildings on the ground. For each femtocell, 3 power levels were available (p_0, p_1 and p_2). p_0 is set to 0, while p_1 and p_2 result in transmission radii of $50m$ and $100m$, and interference radii of $80m$ and $150m$, respectively.

The default value of downlink bandwidth in our simulation is 26, i.e., resource block (RB) size is 26. In the LENA simulator, resource block group (RBG) is the minimum unit of resource to allocate. Based on the specifications in 3GPP TS36.213 table 7.1.6.1-1, a downlink bandwidth of 26 results in $26/2 = 13$ RGBs in each subframe. The resource allocation algorithms are implemented in the MAC layer, and downlink RBGs are allocated to femtocells based on those algorithms. Mobile users arrive at this network at various arrival rates (with i.i.d. inter-arrival time) and speeds, and do a random walk for $60s$. Saturated UDP traffic over downlink is generated for every user in the simulation. During their connection time, users report

their topology information to base stations using an uplink channel. The duration of the simulation for every scenario was chosen to be 180 seconds.

To evaluate the performance of our solutions in different environments, we vary the values of femtocells, arrival rates and speeds of users to generate multiple scenarios. Number of femtocells is selected from $\{10, 20, 30, 40, 50,$ and $60\}$, arrival rates of users from $\{10, 20, 30, 40, 50, 60$ users/min$\}$. Speeds of users are also varied within $\{3.6\ km/h$ (pedestrian), $10, 20, 30, 40, 50, 60\ km/h$ (vehicular)$\}$. Unless otherwise specified, the default settings are, 30 Femtocells, 30 $users/min$ arrival rate and 3.6 km/h of moving speed.

We use the DRA+ algorithm proposed in [107] as our baseline algorithm, which schedules interfering neighboring femtocells via distributed hashing. As DRA+ does not consider power adjustment, we implement one instance of DRA+ for each power level:

DRA-P1 implements DRA+ algorithm on femtocells, assuming all femtocells work at power level p_1.

DRA-P2 implements DRA+ algorithm on femtocells, assuming all femtocells work at power level p_2.

3.5.2 Simulation Results

Lexicographic Throughput: Figure 3.7 shows the performance of the six algorithms evaluated for a single scenario with default settings. Users are sorted by the amount of data received during $60s$ of random walk. It shows that DRA-P1 performs better than DRA-P2. This is expected since the interfering neighbors are fewer when femtocells work at power level p_1 than at p_2. Our distributed solutions perform close to the centralized algorithms under both non-interfering and interfering models. *Compared to the baseline algorithms, while the non-interfering model achieves more*

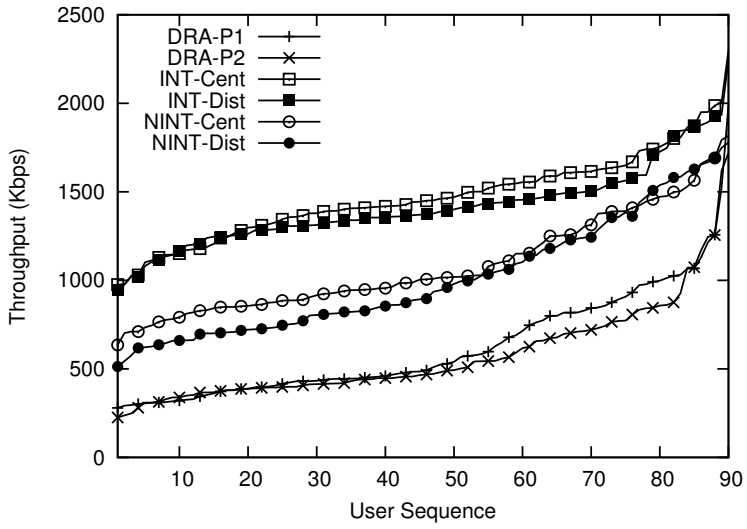


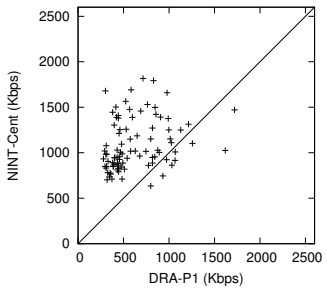
Figure 3.7: Users are sorted by their throughputs. One point is plotted on the line for every 3 users.

than $2x$ *of the minimum throughput, the interfering model achieves more than* $3x$ *of the minimum throughput.* Those improvements are expected because the DRA+ algorithm allocates resources on a per-femtocell basis (not per-user basis), without considering the power assignments of femtocells and densities of users, which are well exploited in our algorithms.

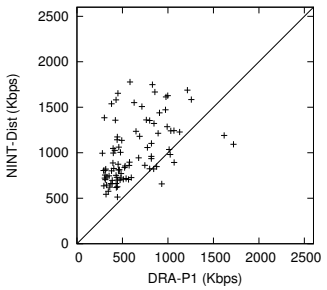
Algorithm Comparison: Figure 3.8 shows the comparison of pairs of algorithms using scatter plots of users, which leads to similar conclusions as above.

Impact on Throughput due to Various Factors: Next, we evaluate the impact on throughput by varying the number of femtocells, arrival rates and speeds of users. For this, we keep two factors as constant, and evaluate the impact of the third factor. Each data point shown is an average of 5 scenarios, where every scenario has a random placement of femtocells.

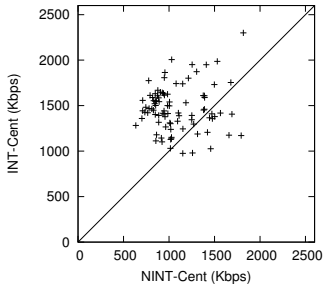
Figure 3.9 (a) shows *when the number of femtocells increases, the minimum*



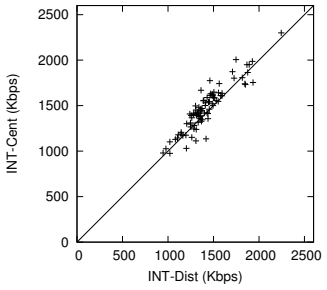
(a) NINT-Cent vs DRA-P1



(b) NINT-Dist vs DRA-P1



(c) INT-Cent vs NINT-Cent



(d) INT-Cent vs INT-Dist

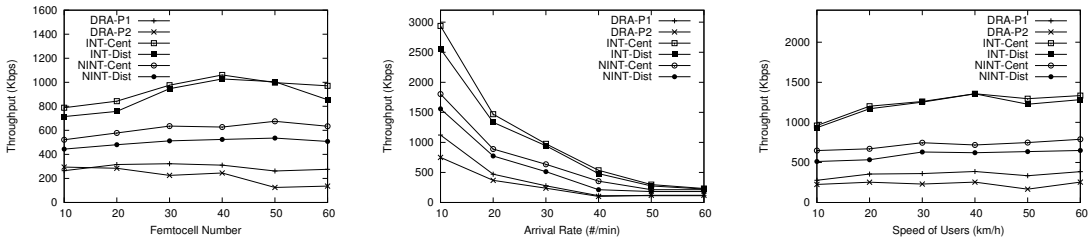
Figure 3.8: Scatter plots of throughputs of 90 users, one point denotes one user.

throughput among all users and scenarios tends to increase in our algorithms. Note that when the number of femtocells is more than 50, the minimum throughput in the INT model starts to decrease, due to the fact that although pair-wise interference between active links has been addressed in this model, *the accumulated interference becomes high enough at this point to affect the throughput. DRA algorithms, on the other hand, do not benefit as much when increasing the density of femtocells. DRA-P2 is better than DRA-P1 only when the number of femtocells is small (10 femtocells).* This can be explained by the fact that DRA algorithms do not perform power adjustment, and spectrum resource is divided equally among the neighboring femtocells. When the number of interfering neighbors is small in sparse deployments, DRA-P2

performs better due to its larger coverage area. However, when the number of interfering neighbors becomes large in dense deployments, the resource allocated to each femtocell starts to drop significantly, resulting in lower throughput to users in both algorithms. *Overall, the NINT and INT models can achieve up to* $2x$ *and* $3x$ *of the minimum throughput.* Average throughput in the same set of scenarios is shown in Figure 3.9 (d), which shows similar trends. Similarly, Figures 3.9 (b) and (e) show that *minimum and average throughputs drop when the arrival rate increases (more users in the system)*, and Figures 3.9 (c) and (f) show that *the minimum throughput increases slightly, and the average throughput decreases slightly when the speed of users is increased from* 10 km/h *to* 60 km/h. This is because when the speeds of users is increased, the chance that a user sees a femtocell in its lifetime increases. Thus the user that receives the lowest throughput has higher chances to increase its throughput. However, due to the Doppler effect, system-wide throughput gain is offset by the increased speed.

Approximation Ratio of INT Model: To understand the gap between our INT algorithm and its optimal solution, we then formulate the Partition Coloring problems as an Integer Linear Programming problem, and use lp_solve [78] to obtain the optimal solution. The partition graphs are re-constructed from the log files of our previous simulations, which guarantees that this evaluation is based on realistic settings. We show the resulting average number of colors by partitions for each algorithm. Note that the number of partitions (users) shown in Figure 3.10 are counted only for femtocell users, since any of the macrocell partitions (users) would have conflicting links with all other partitions (users) and will take one color in any algorithm.

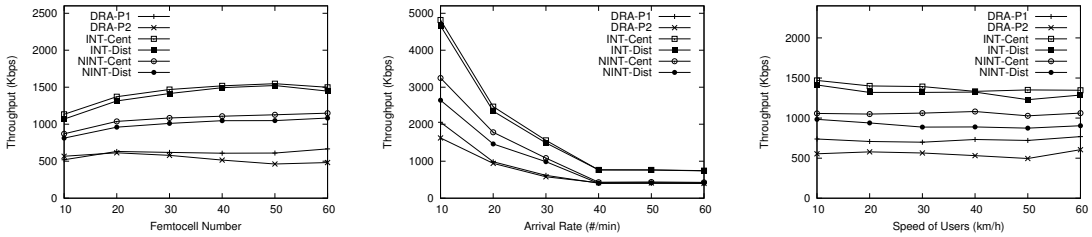
From Figure 3.10, it is clear that performance of our INT-Cent algorithm is very close to the optimal, even though the approximation ratio tends to decrease in larger graphs (i.e., more femtocell users).



(a) Min TP with variable femtocells

(b) Min TP with variable users

(c) Min TP with variable speeds



(d) AVG TP with variable femtocells

(e) AVG TP with variable users

(f) AVG TP with variable speeds

Figure 3.9: Variation of minimum/average throughput of all users due to various factors. (a,d) Minimum/Average throughput increases as #femtocells increases (arrival rate = $30/min$, speed = 3.6 km/h). (b,e) Minimum/Average throughput decreases as the arrival rate of users increases (#femtocells = 30, speed = 3.6 km/h). (c,f) Minimum/Average throughput increases/decreases as the speed of users increases (#femtocells = 30, arrival rate = $30/min$).

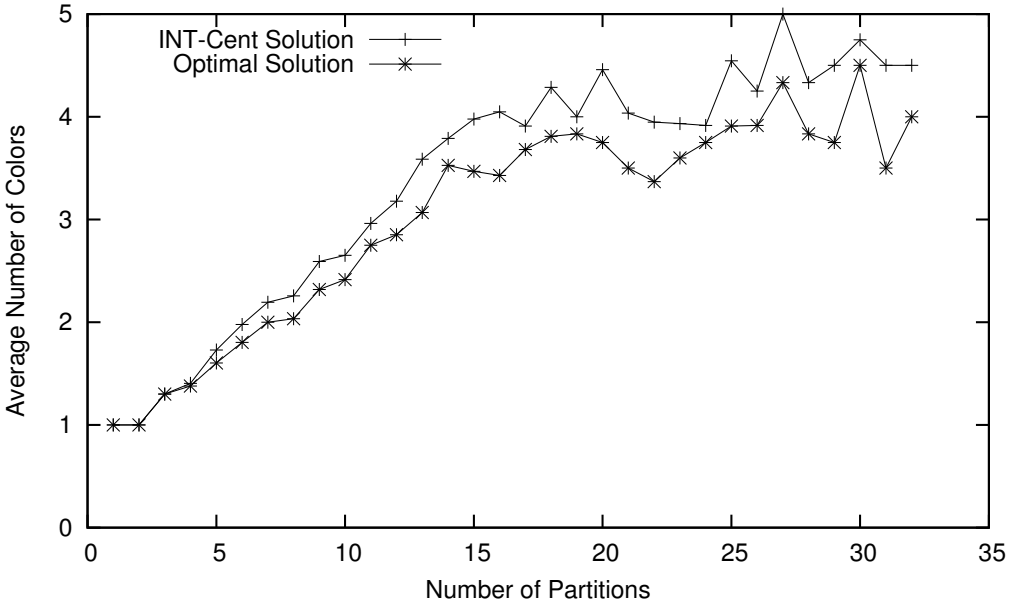


Figure 3.10: Average number of colors needed by INT-Cent algorithm and the optimal to color the same partition graph.

System Overhead: In simulations, in order to create the conflict graph, whenever a change in the topology occurs, users send their topology information to base stations through an uplink channel. Next, we evaluate the overhead of acquiring and sending this information. For the NINT model, we assume femtocells periodically send beacons at different power levels, and a node needs to send a message of 2 bytes for each beacon it receives, indicating which femtocell/powerlevel the beacon is from. For the INT model, we assume that a message of 2 bytes needs to be sent by a node for each pair of conflicting links it finds (assume there is some throughput test mechanism to identify conflicting links).

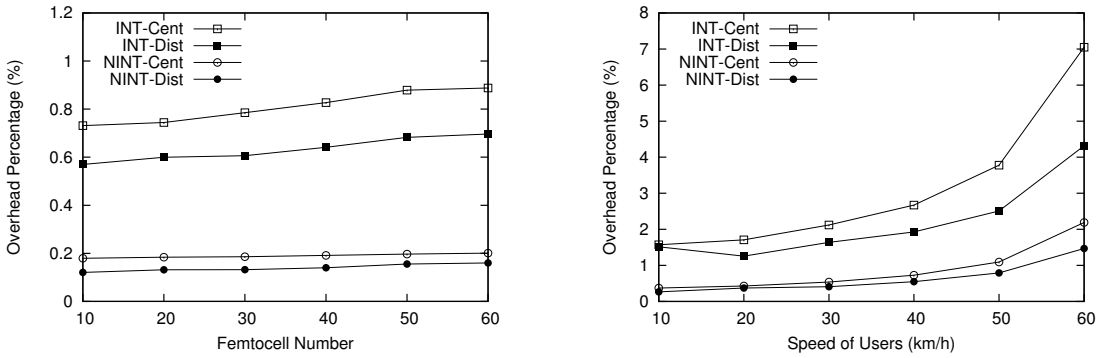
Figure 3.11 (a) shows the overhead imposed by those control packets as a percentage of all data packets with different numbers of femtocells in the network. Note that we did not count the coordination packets sent between neighboring Femtocells

in the distributed solutions, since their cost is relatively low. It shows that in all algorithms when users walk at 3.6 km/h, the percentage of overhead increases with increase in the number of femtocells, due to increase in the number of interference sources. While the INT-Cent algorithm has the highest overhead, *the distributed solutions can save upto* $1/3^{rd}$ *of system overhead from their centralized counterparts.* Recall from Figure 3.9 (d) that the total throughput of the network is also increasing when more femtocells are deployed, implying that the total number of overhead packets is increasing at a higher rate. Figure 3.11 (b) shows that given the same set of femtocells and users, when the speed of users is increased from 10 km/h to 60 km/h, the percentage of overhead increases at an even higher rate. This is caused by both the increasing chance of seeing interfering femtocells, and the result of fast fading channels. However, the savings of overhead in the distributed solutions compared with the centralized solutions are still substantial.

3.6 Discussion and Future Work

- **Construct Conflict Graphs.** This work takes conflict graphs as the inputs of our algorithms. However, constructing such kind of conflict graphs sometimes could be nontrivial. Although conflict graphs are constructed based on the knowledge of users' locations in our simulations, this approach might be neither accurate nor feasible due to the lack of knowledge of users' locations. However, we can construct the conflict graphs without knowing users' locations.

  Constructing conflict graph for the NINT model is relatively simple. Recall that the conflict graph in NINT model is stable, i.e., it does not need to be reconstructed from time to time. For any Femtocell, it only needs to figure out which neighbor (at what power level) it interferes with, for once. This kind of requisite is very similar to the one in [107], in which every Femtocell needs



(a) Percentage of overhead with different number of femtocells

(b) Percentage of overhead with different speed of users

Figure 3.11: Percentage of overhead with different number of femtocells and mobile speeds (a) Percentage of overhead slowly increases as the network becomes denser (arrival rate = 30/min, speed = 3.6 km/h). (b) Percentage of overhead increases faster when increasing the speed of users (#femtocells = 30, arrival rate = 30/min).

to find out its set of interfering neighbors. To obtain such conflict graph, one way is to let each Femtocell send out beacons at different power levels, and look at the received signal strength of pilots from others. The second way is to construct the conflict graph based on users' reports. Whenever a user associated with some Femtocell encounters interference, it reports to the Femtocell. Then by testing the throughputs (also called bandwidth test) with and without the presence of another neighbor, this Femtocell can figure out if it interferes with the Femtocell (at the current power level) or not.

In NINT model, other than the conflict graph of femtocells, we also need to know the weight (# of users) of Femtocells at each power level. This weight info need to be updated from time to time. Similar to the above mentioned

method, by letting Femtocells send beacons at different power levels, users can report their received signal strengths from each Femtocell in vicinity. In this way, Femtocells know how many users are available to serve at each power level.

Unlike the NINT model, the link conflict graph in INT model is more complicated and needs to be updated on runtime. Most prior research on conflict graph construction uses bandwidth tests that tests a pair of links based on the observations of throughputs with presence and absence of simultaneous transmissions [89, 91]. This approach is also adopted in a latest femtocell resource allocation work [22]. In this work, we also assume that a bandwidth test framework is sufficient to construct the link conflict graph, and we evaluated the overhead of such approach in our simulation part.

Other than bandwidth test, another online approach is proposed in [18] which can do the job within milliseconds as claimed in the work. However, this approach requires to modify the air interface, which is usually prohibitive in cellular network. If this approach could be applied to femtocell network, the procedure of constructing link conflict graph might be even faster considering that some users will stay in the same place within a short interval, thus only part of the graph needs to be updated on the fly.

- **SINR based Interference Model.** We have so far only considered binary interference model. Alternatively, the SINR model can be considered. In the SINR model, let $SINR(i, l)$ be the SINR at user i in tile l, then it must be larger than a threshold γ for successful reception in tile l. In some sense, our binary approach is only an approximation of the underlying SINR based model.

Observe that $SINR(i, l)$ is dependent on the allocation of tile l on other femtocells, achieving *maxmin* resource allocation and maximal throughput under SINR model will be more interesting and challenging.

In order to solve this problem, we will explore a simplifying technique that limits the summation of noise only to neighboring femtocells by relaxing γ to $\gamma + \eta$, where η is appropriately chosen so that it bounds the maximum interference from all other non-neighboring femtocells. This construction will allow to focus on a limited number of neighboring femtocells for the purpose of scheduling and power assignment.

3.7 Conclusion

To address the *maxmin* and *weighted maxmin* problems in the context of resource allocation in femtocells, two models are considered in this work. The non-interfering model selects an independent set of femtocells, and determines the resource allocation factors based on this set. For the interfering model, the problem is transformed into the partition coloring problem. Algorithms with provable bounds are designed for both models. Improvements of up to 3x is observed for the minimum throughput for the interfering model over previous work.

Chapter 4

ACHIEVING QOE DOMAIN FAIRNESS THROUGH BITRATE INFERENCE AND BANDWIDTH ALLOCATION IN LOCAL AREA NETWORKS

Video traffic exceeds more than 50% of todays Internet traffic [98, 120], with Netflix and YouTube leading the list of most popular applications in desktop and mobile platforms [98]. The growth of video traffic, partially driven by the rapid proliferation of smartphones and tablets, makes the problem of content delivery more challenging. To improve user experience, new techniques such as DASH (Dynamic Adaptive streaming over HTTP), has been widely adopted in today's players.

In DASH, the server maintains multiple profiles of the same video, encoded in different bitrates and quality levels. The video object is partitioned in fragments, typically a few seconds long. A player can request different fragments at different bitrates, depending on the underlying network conditions [19]. One advantage of DASH over traditional customized video transport protocols is that it works with middleboxes such as NATs and firewalls. DASH has been implemented in most popular video players on different platforms, including Microsoft's Smooth Streaming, Adobe's OSMF player, Netflix player, and YouTube player, etc.

To provide the best quality of service to the end-users, most content providers implement proprietary bandwidth estimation and bitrate adaptation algorithms in

their players. However, when multiple clients share the same bottleneck link, the uncoordinated behaviors of players and the diversity of available bitrates of video files bring up many issues including *unfairness and inefficiency*[19, 65].

Two classes of solutions have been proposed for addressing these issues: *client-side adaptation* and *access point scheduling*. An example of the former class is [65], which develops a suite of techniques, including randomized download scheduler and stateful bitrate selection, etc., to guide the tradeoffs between fairness, efficiency and stability. Those client-side solutions require changes to the video players, which make them difficult to deploy in practice. In addition, such schemes are suboptimum due to lack of a global view. [33] proposes a scheduling framework at the access points(APs) for adaptive video delivery over cellular networks based on the set of candidate bitrates of each video file. Such information is critical to the scheduling algorithm of the AP. This will be discussed in detail in Section 4.2.1.

Existing access point scheduling frameworks assume that the bitrate information can be derived using techniques such as Deep Packet Inspection (DPI) [33]. DPI based techniques are in general difficult to implement, as the bitrate information of different video files (possibly from different providers) might be encoded in different formats, or transmitted in different fields of network packets. Moreover, some video files may be delivered over https (widely supported by YouTube, Facebook, Dropbox, etc.), and some video files may not even have such information (a private video uploaded to Dropbox).

In this work, we develop a set of novel techniques for achieving fair, and efficient video delivery to all clients without requiring any changes to the individual video players at the client-side. To acquire the candidate bitrates of the currently playing video files, we propose a novel solution which infers such information based on observed traffic patterns at the APs and feedback provided by a software running at

the clients. Note that the proposed software module at the clients accommodates all video players, and does not require modifications to any specific player.

4.1 Related Work

The problem of optimizing video delivery has been studied by a large number of prior works. [81] proposes an admission control and scheduling framework, which provides a uniform QoE to users based on a long-term dissatisfaction metric. [72, 28] propose adaptive video streaming solutions based on flexible coding schemes (H.264), in which, the transmitted bitrate is constantly adapted to the available network bandwidth, such that audio and video artifacts caused by packet loss are avoided. [101] presents the design of a mobile video-centric proxy cache located in the local cellular infrastructure. It uses a linear encoder to adapt the video bitrate based on the available bandwidth. [131] presents a cross-layer design of video transmission scheme, which jointly considers the application layer information and the wireless channel conditions. These aforementioned schemes are designed for non-DASH single-bitrate video flows, and they do not consider the bitrate adaptation at the client side.

DASH-based video delivery techniques have been widely adopted in today's players. [128] confirmed that DASH improves end-users' subjective perception greatly compared with fixed-rate streaming in terms of QoE. Authors of [19, 65] reported the issues of unfairness, inefficiency and instability in DASH video streaming, when multiple DASH clients share the same bottleneck link. A number of related works have been proposed to address those issues.

On the client side, [23] studies the bitrate selection problem in DASH players. It presents a quality-aware rate adaption scheme to maximize the client's QoE in terms of both continuity and fidelity (picture quality). [65] develops a suite of techniques, including randomized download scheduler and stateful bitrate selection, etc., to guide

the tradeoffs between fairness, efficiency and stability. Those client-side solutions require changes to the video players, which make them difficult to deploy in practice.

In the network infrastructure part, [93] proposes a wireless DASH proxy to enhance the QoE of wireless DASH. The authors propose to implement the rate adaptation logics at the edge between Internet and wireless cellular core networks, which is different from the conventional DASH that implement such logics either locally in the user equipments or remotely in the DASH server. [49] leverages the OpenFlow technology and proposes an OpenFlow-assisted QoE fairness framework to provide a control plane that orchestrates the bandwidth estimation of different DASH players. The proposed solution could be suboptimum due to the lack of a global view at local clients, and it requires modifications to all DASH players. [33] is the most relevant work to ours. The authors propose a scheduling framework at cellular basestations for adaptive video delivery over cellular networks based on the candidate bitrates of each video. The proposed solution relies on the DPI to acquire the bitrate information of the videos being played. However, acquiring bitrate information from DPI is unreliable or infeasible in the face of diverse packet structures and encryption. In contrast, this work proposes a solution that allows the AP to effectively infer the bitrate information.

4.2 System Model

4.2.1 Problem Description

Consider a 802.11 wireless access point that is associated with N clients which are streaming videos. We focus on the cases when the access link between the AP and clients is the bottleneck on the network traffic path, which are common in places where

the density of APs is high, such as airports, hotels, restaurants and apartments. This assumption is supported by prior works [105].

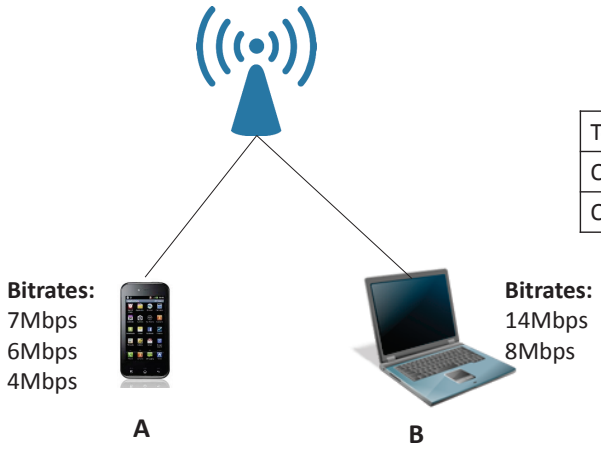
Given a certain amount of network resource to be shared by N video streaming clients, this work aims to design coarse-level resource allocation algorithms to achieve fairness in the QoE domain. Note that clients that are not streaming videos are not under the consideration of this work, and will not be allocated any resource by this framework. While a naive solution which equally divides the network resource across N clients may lead to quality of service (QoS) domain fairness, it is not necessarily the case for QoE domain fairness. Figure 4.1 shows one example that smart resource allocation based on the knowledge of bitrates can further improve the QoE domain fairness and the efficiency of the system.

4.2.2 Acquiring Bitrate Information

From the example shown in Figure 4.1, the bitrate information is critical to the scheduler for QoE domain fair resource allocation. Prior works rely on DPI to analyze the content of the packets, which is not feasible due to many reasons such as cost and encryption. Another possible solution is to let players report such information to the AP. However, due to the large diversity of players and platforms, it is not practical to require all players to provide such API.

To address this problem, we design a solution that allows the AP to effectively infer the bitrate information of clients. Our solution is incentivized by the periodical chunk-by-chunk download strategy of DASH players.

The periodical download strategy is fundamental to the adaptive bitrate selection mechanism, and has been widely adopted by DASH players. To verify this, we carried out experiments to test the download strategies of the official video players of some of the most popular content providers, including YouTube, Netflix, iTune, etc. We use a



Total Bandwidth	14Mbps
QoS Fair	A: 7Mbps, B: 7Mbps
QoE Fair	A: 6Mbps, B: 8Mbps

Figure 4.1: One AP serves two clients. Client A has three bitrates: {4 Mbps, 6 Mbps, 7 Mbps}. Client B has two bitrates: {8 Mbps, 14 Mbps}. The bottleneck air interface is 14 Mbps. In the QoS fair allocation, each client receives 7 Mbps. As a result, Client B's video is not watchable (frequent stalls). In the QoE domain fair allocation, client A receives 6 Mbps and B recieves 8 Mbps, and both clients can play the videos smoothly. Note that the bitrate information of each video is critical for the QoE solution.

laptop which connects to the Internet through an Ethernet cable to set up a wireless AP. The downlink traffic of the client devices was logged by the AP using Wireshark. Our result confirmed that the periodical download strategy has been adopted by almost all the tested video players, except for the YouTube HTML5 player. Figure 4.2 shows the observed download traffic patterns of the YouTube Flash Player, the YouTube IOS App and the Netflix Android App.

Our solution takes advantage of this phenomenon, and lets the AP infer the bitrate information based on it. The idea is that if the AP can detect the life cycle of an average period based on the client's downlink traffic pattern, then it can deduce its bitrate based on the total amount of data in the cycle and the length of the cycle. To get information about the candidate bitrates of the same client, the AP

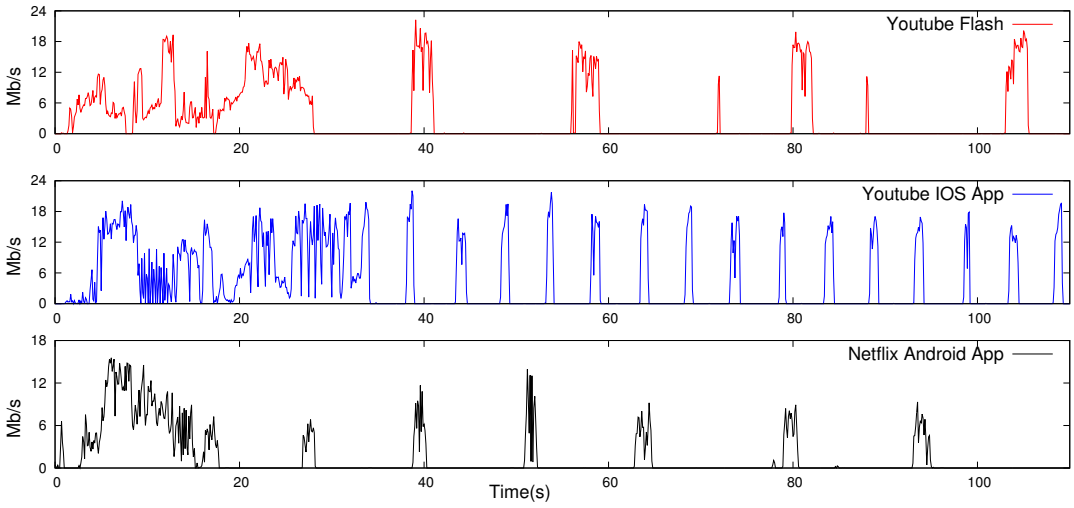


Figure 4.2: The observed periodical download traffic patterns of some video players: the YouTube Flash Player, the YouTube IOS App and the Netflix Android App. The players initially fill the buffer with a bulky download (about $20 - 40$ seconds in the figure), and then maintain the fullness of the buffer with periodical downloads.

can intentionally change the available bandwidth assigned to that client, which will force the client to adjust its bitrate. This novel solution does not require any change to the client. To extend our solution to the cases of legacy players, which do not exhibit periodical downloads, we also developed a client module. The client module is deployed as a system service on clients, and it can report the status of the video (play vs. stall) to the AP. More details about our bitrate inference solution is presented in Section 4.3. Note that, in this chapter, we assume that one video streaming client only has one video thread, thus the term "client" and "player" are used interchangeably throughout this work.

4.2.3 Objective and Challenges

Let P denote the total number of network resource units that is designated to all N video clients for each unit of time, and p_i denote the number of number resource units allocated to i. We use α_i to denote the physical transmission rate of i: $\alpha_i \times p_i$ is the maximum available throughput of i, and α_i depends on the signal-to-noise ratio (SNR) of i. For simplicity, we assume that α_i is not changing at the bitrate inference stage. However, the proposed solution can be easily extended to cases where α_i changes. Define r_i as the bitrate selected by player i using its internal rate selection algorithm based on the available throughput $\alpha_i \times p_i$. We use a function $\gamma_i(*)$ to represent this selection: $r_i = \gamma_i(\alpha_i \times p_i)$. In general, $r_i \leq \alpha_i \times p_i$.

This chapter aims to achieve the $max - min$ fairness of QoE among N clients by determining the values of $p_i, \forall i \in \{1..N\}$.

QoE is conventionally measured in terms of Mean Opinion Scores (MOS). To evaluate QoE online, existing works show that MOS can be mapped from a couple of objective metrics [40], including:

- Stall Ratio: The fraction of the session time (playing + freezing time) spent in buffering.
- Rate of Stall Events: The number of stall events over the session time.
- Bitrate: the bitrate level at which the video is being played.

According to [58, 40], stall ratio and rate of stall events are two dominant influential factors. Based on this, the QoE metric in this work is defined by a tuple $\langle s, b \rangle$, in which s denotes the primary sort key mapped from stall ratio and rate of stall events, and b denotes the secondary sort key mapped from bitrates.

Let client i's primary QoE metric $s_i = \mathcal{S}_i(r_i, \alpha_i \times p_i)$, in which $\mathcal{S}_i(*)$ is the QoE function that maps stall profiles to MOS for player i. Note that the stall profile of i

depend on its video bitrate (r_i), the throughput ($\alpha_i \times p_i$), and the streaming strategy of i. We assume $s_i = 0$ if $r_i \leq \alpha_i \times p_i$, i.e., no stall occurs if the throughput is sufficient for the selected bitrate. Similarly, let $b_i = \mathcal{B}_i(r_i)$, in which $\mathcal{B}_i(*)$ is the video quality function depending on the type of the video that i is currently playing and the bitrate of the video (r_i).

Studying the functions of $\mathcal{S}(*)$ and $\mathcal{B}(*)$ is outside the scope of this chapter. Many related works exist. We use the results of [84] and [49] in our experiments. For the ease of presentation, we assume that $\mathcal{S}(*)$ and $\mathcal{B}(*)$ are identical for all players, and it can be shown that the presented resource allocation algorithm also works in cases when $\mathcal{S}(*)$ and $\mathcal{B}(*)$ are different for different users.

Formally, the objective of this chapter is defined as follows:

$$\begin{aligned} \mathbf{P1}: \quad & \max_{p_i} \min_{1 \leq i \leq N} \langle s_i, b_i \rangle \\ s.t. \quad & \sum_{i=1}^{N} p_i \leq P \end{aligned} \tag{4.1}$$

where $\langle s_i, b_i \rangle$ is the QoE metric of client i, and tuples of different clients are ordered lexicographically.

To this end, we summarize the major challenges faced in this work:

- **Acquire Bitrate Information without Modifying Players.** Bitrate information is critical to resource allocation problems. The vast diversity of players makes it impractical to acquire bitrate information directly from the players by modifying them. Acquiring bitrate information without modifying the players is a challenge.

- **Accurate Event Identification.** This chapter proposes novel bitrate inference solutions based on events raised by players (download start, idle start, etc.). To guarantee the performance of the bitrate inference algorithm, we must be able to accurately identify such events, which is challenging.

- **Efficient Bitrate Probe.** Due to the delay in the client side bitrate adaptation as well as the AP's bitrate inference, discovering more candidate bitrates of the clients requires us to design efficient bitrate probe schedule, which is a challenge.

4.3 Inferring Bitrate Information

4.3.1 The Rationale of Bitrate Inference

DASH video players maintain a set of states (e.g., play, stall, download, idle, etc.). State transition is triggered when some criteria is met. For instance, when the buffer size is over some threshold (say τ_d), the player starts the periodical download to fill the buffer, and when the buffer size is larger than another threshold τ_i, the player stops the download and enters the idle stage.

This work proposes a bitrate inference solution by taking advantage of this common practice. The proposed solution detects the download start or idle start events of a player, and then calculates the player's current bitrate based on the traffic volume between those events. The idea is that, for a client i that periodically downloads the video, if the AP finds two consecutive *download start* events at time t_1 and t_2, with some idle time in between. The AP assumes that the video is playing smoothly during time $[t_1, t_2]$, since there is idle time. Another assumption is that the periodical download events started at t_1 and t_2 are due to the fact that the player's buffer size reduces to τ_d. Let $V_{(t_1,t_2)}$ denote the amount of data downloaded between $[t_1, t_2]$. *We conclude that $V_{(t_1,t_2)}$ has been fully consumed during $[t_1, t_2]$.* So the current bitrate of i can be estimated by $\frac{V_{(t_1,t_2)}}{t_2-t_1}$. *Note that our bitrate inference solution is independent of the values of τ_d and τ_i, i.e., it is a general solution for all DASH players.* By adjusting the allocated bandwidth to i, which in turn triggers i's bitrate adaptation mechanism, other candidate bitrates of i can be incrementally discovered.

4.3.2 Accommodate Continuous-Download Players

However, not all players exhibit periodical download behaviors. According to [56], there are non-DASH players that continuously download videos at the maximum possible speed, such as the YouTube HTML5 player. Through our experiments, we found that the YouTube HTML5 player, the Facebook flash player, and the Dropbox flash player are all continuous-download players. Figure 4.3 shows the downlink traffic observed by the AP with the YouTube HTML5 player.

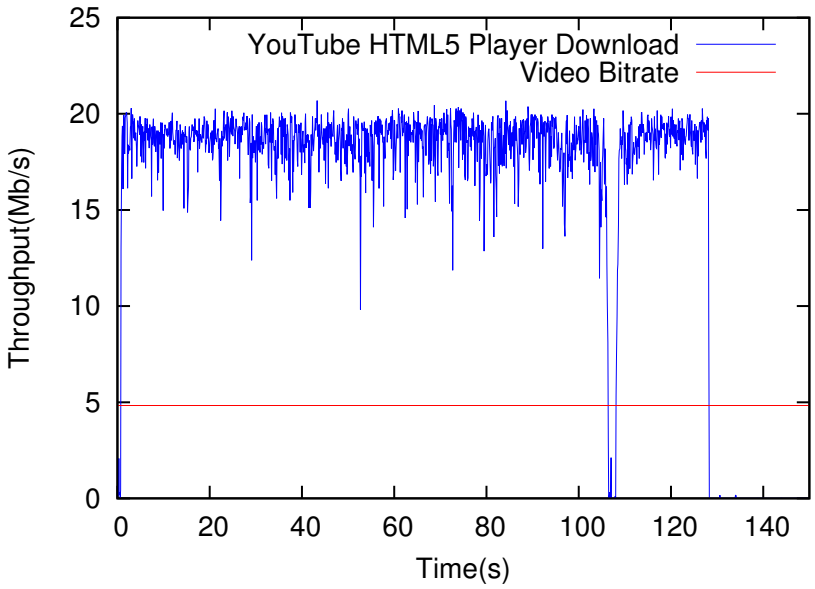


Figure 4.3: The YouTube HTML5 player continuously download the video content. A total bandwidth of 20 Mbps is allocated to the client. The client takes all the bandwidth until the whole video is downloaded at time $127s$, even though the video bitrate is only $4.9Mbps$.

Our solution relies on the *play* and *stall* events to accommodate the continuous-download players. Let τ_p be the threshold, above which the player starts to play the video, and τ_s be the threshold, below which the player stops playing the video (stall occurs). Suppose one stall occurs at time t_1, and then the player starts to play at t_2, and then another stall occurs at time t_3, assuming $t_1 < t_2 < t_3$. By applying the same idea as presented for periodical-download players, we can estimate the bitrate by $\frac{V_{(t_1,t_3)}}{(t_3-t_2)}$, in which $t_3 - t_2$ is the video play time.

Since the AP has no knowledge of *play* and *stall* events. We address this problem by deploying a software module on client devices. This module can detect stall and play events of the player and send it to the AP. We integrate this module as a system service, which means no modification to the players is needed. *Note that the stall-based solution can also be applied to the periodical-download players. This will help the AP infer the bitrate of the clients faster or with higher accuracy.*

4.3.3 System Architecture

Our solution consists of two modules: the *client module* and the *AP module*. The client module is installed on user devices. It reports the QoE related events and metrics, including the stall events, the current quality of the video, to the AP module. Since the performance of the bitrate inference algorithm at the AP could be affected by human interactions during its inference stage, the client module could also report such events to the AP. In this work, we assume no human interference during the bitrate inference stage, and leave this part for future implementation. The AP module records the downlink traffic volume and pattern of each client, and infers the current bitrate of each client based on the feedback of client modules. It also adjusts the bandwidth of each client to discover more bitrates information of the clients in the shortest time.

4.4 The Client Module

We implement the stall detection function on the client module by monitoring the output (video and audio) of client devices. For instance, the client module can sample the pixel values within the player window and report a stall event if all the pixel samples do not change for a certain time. One challenge of the proposed solution is that the player window might only occupy part of the screen, and we need to detect the part of the screen that is occupied by the player (called Effective Video Area in this chapter), and make sure pixel samples are taken within the effective video area. Another challenge is that the content of some videos could have very slow motions, which could affect the accuracy of the pixel-change based stall detection approach. Such issues need to be properly handled in the solution.

The client module is integrated in the system as a service, which can support all video players installed in the system. In doing this, no change is needed for any video player, which makes the solution easily applicable in real systems. The client module is activated by certain system events or under certain conditions. Such system event or condition could be, opening a video player application, creating a video-player related thread, visiting the domains of some known content providers, or when the network usage of some application is above certain threshold. The client module then starts to detect the effective video area and video play or stall events. Whenever a video play or stall event is detected, it reports such event to the AP, which then uses such information to infer the bitrate of the corresponding client and its quality of experience. We have implemented the client-server communication using socket programming. For simplicity, we ignore the delay in detecting such events, Instead, we use a small time offset at the AP to count in such delays.

4.4.1 Detecting Effective Video Area

To sample pixels of the video being played, the client module first finds the effective video area on the screen. For this, the client module maintains a matrix of integers, each of which represents the rate of changes of the corresponding pixel on the screen. The client module then seeks to find two horizontal lines and two vertical lines whose changing frequencies have the highest difference compared to their neighboring lines. This algorithm is elaborated in Algorithm 8. The idea is that the effective video area in general has higher rate of changing pixels compared to the non-video area. We found through experiments that the effective video area can be accurately detected within $2-3$ frames for most videos.

In Algorithm 8, *change* is used to save the change rate information for each pixel (Line 1), and $prePix$ and $curPix$ are used to save the screen of previous frame and current frame. By comparing the current frame against the previous frame, it finds if each pixel has changed and updates *change* accordingly (Lines 8-10). This was repeated for at least 1 second (Line 3) since we found our algorithm can accurately detect the effective video area within $2-3$ video frames. Lines 14-17 computes the relative changing rate of each horizontal and vertical line (saved to $hLines$ and $vLines$, by taking the difference of changing rates between the current line and its neighbor. Since we expect that the effective video area has larger changing rate than the non-video area, the borders of the video area should have the steepest difference of changing rates. So the largest two values of those two arrays are used to form a rectangular area, which is then returned by the algorithm (Lines 18-19).

Algorithm 8 is most accurate when the video is being played for whole or at least part of the time when the algorithm runs. In our implementation, we run this algorithm multiple times to guarantee the accuracy of the detection. The presented algorithm keeps track of all pixels ($W \times D$). The efficiency could be improved if we

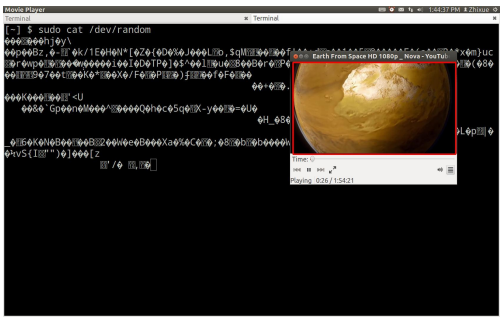
Algorithm 8: Detect Effective Video Area

Input: Pixel values of the screen over time

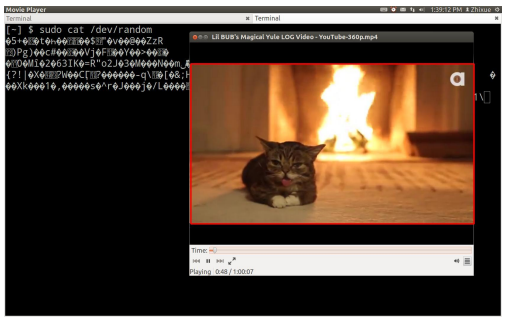
Output: Effective Video Area

```
1  change[W][H]: track screen pixel changes // W:width, H:height
2  initTime ← curTime()
3  while (curTime()−initTime > 1) do
4      curPix[W][H] ← capture current screen
5      if (prePix = null) then
6          prePix ← curPix;
7          continue // first screen capture, go to next step
8      for (i ← 0 to W) do
9          for (j ← 0 to H) do
10             change[i][j] ← change[i][j] + (prePix[i][j] = curPix[i][j])?0 : 1
11     prePix ← curPix
12 hLines[H] : count pixel changes of each horizontal line
13 vLines[W] : count pixel changes of each vertical line
14 for (i ← 1 to W) do
15     for (j ← 1 to H) do
16         vLines[i] ← vLines[i] + |change[i][j] − change[i − 1][j]|
17         hLines[j] ← hLines[j] + |change[i][j] − change[i][j − 1]|
18 Find (i1, i2, j1, j2), s.t. vLines[i1], vLines[i2] are largest in vLines[0..W]
   hLines[j1], hLines[j2] are largest in hLines[0..H]
19 return the rectangle formed by (i1, i2, j1, j2).
```

only track a lower density of horizontal and vertical lines, e.g., 1 line of pixels for every 10 lines.



(a) Medium-motion Video



(b) Low-motion Video

Figure 4.4: Detecting the effective video area of medium-motion videos and low-motion videos. Algorithm 8 can correctly detect the video area of both types of videos within $2-3$ frames with the presence of background noise. (a) A medium-motion video that is available at [3]. (b) A low-motion video that is available at [6].

In our experiment, Algorithm 8 was tested for different videos especially those with medium-motion or low-motion contents. Examples of medium-motion and low-motion videos are listed in [3] and [6]. The video player in the experiments occupies only part of the screen, and we use "cat /dev/random", to generate random output on the background screen to simulate the scenario that the user is doing some keyboard interactions and meanwhile watching videos. Figure 4.4 shows two sample results of the algorithm. The effective video areas are marked by red borders.

Through our experiments, we do find the performance of the algorithm is degraded on one type of video which only consists of a few static pictures. An example of such video can be found at [2]. As the fraction of such videos is insignificant, and the bitrate

of those videos are usually very low, they are not handled by the client module. The QoE of viewers of such videos can be guaranteed by reserving a minimum bandwidth for all players.

4.4.2 Detecting Stall Event

Stall information is needed for accounting the QoE levels of users and inferring the bitrate information of videos. Stall events can be detected by monitoring pixel changes in the effective video area. When all pixel samples are not changing for a certain duration of time, the client module decides that a stall is present. However, part or all of the effective video area might not be changing at certain time in some low-motion videos. In this section, we study the relationship between the number of pixel samples, sampling locations and the accuracy of stall detection.

Due to the fact that stall events of online videos caused by network congestion is not predictable, i.e., the ground truth of stall events are unknown, the experiments presented in this section were all offline. We use the default movie player of the Ubuntu system to play a selected video, and meanwhile use a script to generate the "space-pressed" event. Since the player responds to "space-pressed" events by switching between the "playing" and "paused" modes, stall events can be arbitrarily generated. The script also logs the timestamp of each occurrence of "space-pressed" event, which is then used to deduct the ground truth of stall events.

The length of each stall and play in the experiments are randomly selected from the range of $[2, 5]$, to simulate a scenario in which the available bandwidth is approximately half of the bitrate. The program samples the effective video area twice for every second. The number of sample pixels is varied within $\{1, 4, 9, 16, 25, 36\}$. Sample pixels are evenly distributed within the area so that their coordinates cut the video area into grids of the same size. For example, let W and H represent the width

and height of the effective video area. If only one pixel is to be sampled at every second, then it takes the pixel coordinated at $[\frac{W}{2}, \frac{H}{2}]$. Similarly, if 4 pixels are to be sampled, it takes the pixels coordinated at $[\frac{W}{3}, \frac{H}{3}], [\frac{2W}{3}, \frac{H}{3}], [\frac{W}{3}, \frac{2H}{3}], [\frac{2W}{3}, \frac{2H}{3}]$.

The experiment consists of a total of 12 videos from YouTube, with 9 of them labeled as "normal videos" with medium to low motions (examples are [3] and [6]), and 3 of them labeled as "partially static videos" which play static picture part of the time or in part of the video area. Those videos, are chosen to test the performance of our approach in worst cases (As mentioned earlier, videos that contain pure static pictures are not handled by the client module). Each video also has multiple levels of resolutions (typically $144p$, $240p$, $360p$, $720p$, $1080p$). The result shows that while the accuracy of stall detection is relatively independent of the resolution, it depends on the number of sample pixels.

Since the sampling frequency (twice per second) in the experiment is relative high compared to the stall duration ($2-5$ seconds), the stall detection module did not miss any stall event, e.g., the false negative is 0. However, false positives maybe present when the video is actually playing while all the sampled pixels are not changing. Figure 4.5 plots the number of reported stall events over the actual number of stall events. The result shows that while sampling 4 pixels suffices most videos, more pixels (16) are needed to handle partial static videos.

4.5 The AP Module

The AP module consists of the following functions: 1) **Traffic Accounting.** For each client IP, it builds the traffic volume profile which will be used for inferring video bitrates at the clients. 2) **Communication with Clients.** The AP module also communicates with different clients to record the play or stall events at each client, which will help infer the video bitrates. 3) **Download Event Detection.**

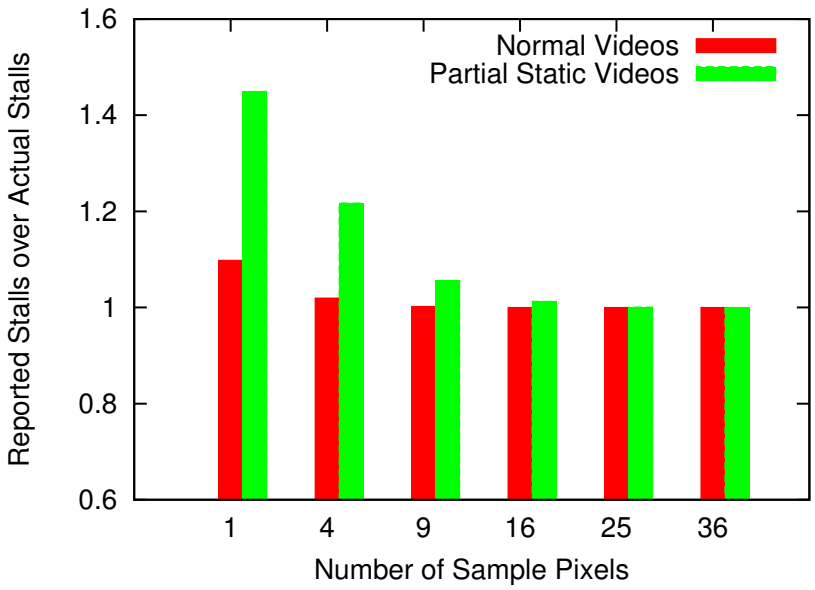


Figure 4.5: The ratio of the detected stall events over the actual stall events.

Most clients download videos periodically, and the server needs to identify the start or end of periodical downloads for each client, to infer the video bitrates of clients. 4) **Bitrate Probe Algorithm.** Note that the AP needs to adjust the bandwidth allocated to each client to trigger the bitrate adaptation mechanism at the clients in order to discover multiple bitrates of each client. This procedure is called bitrate probe. Efficient algorithms need to be designed to probe the available bitrates of all clients in a short time. 5) **Bandwidth Allocation Algorithm.** After the bitrate probe is finished, the AP has gathered the bitrate information for all clients. At this point, the AP performs effective resource allocation algorithm to achieve $max - min$ QoE fairness across all clients.

4.5.1 Traffic Accounting and Communication

For the ease of discussion, let us assume the $\langle IP, Port \rangle$ used by the video player of each client is known to the AP. Such information can be reported by the client modules. To achieve space efficiency, instead of saving all packets, the AP only saves a coarse statistics for each player. For this, the AP aggregates the amount of data transmitted to each player i by intervals of 10 ms, and saves the result (Interval Start Time, Amount of Data) in a TreeMap M_i. This allows the AP to quickly sum up the total amount of data transmitted to a given player during a given time interval.

The communication function on the AP listens on certain port to receive messages from the clients. The most common messages include the "Player IP and Port" message, the "Play Started" message, and the "Stall or Play Event Occured" message, etc. Specifically, there is no timestamp encoded in the message due to the lack of synchronization of clocks between the AP and clients. To find the time of occurance of the event, the AP estimates it with the local reception time minus a small offset, which is used to account for the event detection delay of the client module and the transmission delay.

4.5.2 Periodical Download Identification

This submodule aims to find the start time of each periodical download, which will be used to infer the bitrate of the client (Using the download finish events to infer bitrates is also possible, although this work uses the download start events.). Note that the traffic volume is not necessarily zero during the gap of two consecutive downloads (Figure 4.2). There might be some control packets between the player and the remote host. Based on our experiment, we found that the amount of traffic related to control messages is very small compared to the amount of periodical download traffic. Based on this, we present a Period Identification Algorithm (PIA), which

uses a threshold based method to differentiate the periodical download traffic from the occasional control traffic.

Let h_d and h_m be the thresholds to identify the existence of download activity, and distinguish control packets from video data download, respectively. We set $h_d = 1Kbps$ and $h_m = 100Kb$ in our experiment. For a given client i, let M_i be the traffic map of i. To check if a timestamp t is the start of a periodical download, the PIA algorithm (Algorithm 9) works as follows: PIA returns false if t is not recorded in the dataset (Line 1), or if the throughput in the current slot t is less than h_d (no download) or the throughput in the previous slot $t - 10ms$ is larger than h_d (download starts at previous slots, Line 3). PIA then sums up the total amount of data consecutively downloaded in this period (Lines 6-8), and decides if it is a video data download (Line 9).

Algorithm 9: Identify Periodical Download Start

Input: h_d, h_m, t, P

Output: If t is the start of a periodical download

1 **if** t *not in* $M_i.keys()$ **then**
2 | return false `// no record for the input`
3 **if** $M_i[t] < h_d$ *or* $M_i[t-10] >= h_d$ **then**
4 | return false `// the input is not the start time of a download`
5 $total \leftarrow 0$
6 **while** $M_i[t] > h_d$ **do**
7 | $total \leftarrow total + M_i[t]$ `// aggregate traffic starting from` t
8 | $t+ = 10$
9 return $(total > h_m)$

4.5.3 Probe More Bitrates

Based on the received stall events or detected download start events, the current bitrate of each client can be deducted based on the solution presented in Section 4.3.1. To find other candidate bitrates of the clients, the AP adjusts the bandwidth allocated to each player, to trigger the bitrate adaptation mechanism of the player. For this, the AP maintains an available bitrate set for each player. A bitrate is added to the set for a player if the AP finds a new bitrate of it.

However, two types of delays can affect the efficiency of the bitrate probe process. The first type of delay is the delay between the time when the AP adjusts the bandwidth and the time when the player adapts its bitrate. This delay depends on the proprietary adaptation algorithm of the player and its buffer size when the bandwidth adjustment occurs. To test the lengths of such delays in different players under different conditions, we assign a certain bandwidth to the player, and wait until it enters a stable stage (periodical download), and then decrease or increase the bandwidth by a certain percentage (50% and 200% in our test). Our test involves the flash player, IOS and Android Apps of both YouTube and Netflix. The result shows that such delay could vary from $30s$ to $90s$. Another type of delay is the bitrate inference delay between the time when the player adapts to its current bitrate and the time when the AP successfully infers the bitrate based on two (or more) consecutive events. This delay depends on the threshold values of the player, the bitrate of the video and the allocated bandwidth.

Due to the existence of large delays in bitrate detection, designing efficient probe algorithm is critical to ensure that the bitrate probe period is as short as possible for users. Recall that p_i denotes the fraction of bandwidth allocated to player i, and the maximum throughput i is given by $\alpha_i \times p_i$. And r_i is the selected bitrate of i based on p_i, denoted by $r_i = \gamma_i(\alpha_i \times p_i)$. To acquire adequate bitrate information in

a limited time, the probe sequence to player i (denoted by $p_i^1, p_i^2, ...p_i^m$) has to be well designed, i.e., $\gamma_i(\alpha_i \times p_i^1)$, $\gamma_i(\alpha_i \times p_i^2)$,...,$\gamma_i(\alpha_i \times p_i^m)$ should result in different bitrates.

Without the prior knowledge of bitrates of a given player i, finding such a probe sequence is challenging. Three possible solutions exist: random probe, which assigns the next probe p_i^{k+1} a random value after the k^{th} probe is finished; decreasing probe, which probes bitrates following a decreased order; and increasing probe, which probes bitrates following an increasing order.

This work uses the decreasing probe approach, since both random probe and increasing probe could be inefficient. For instance, assume the set of bitrates of i are $\{500Kbps, 1Mbps, 2Mbps, 5Mbps\}$, and player i uses the largest value of r_i without going over the allocated data rate $\alpha_i \times p_i$. Randomly assigning values of p_i could result in identical bitrates when those values are close (e.g., the AP would observe the same bitrate of $500Kbps$ if $600Kbps$ and $900Kbps$ are used in its probe). Similarly, increasing probe could also skip or produce identical bitrates as the AP has no knowledge of the next higher bitrate. On the other hand, the decreasing probe can use the current bitrate as a hint to the next lower bitrate. For example, assume the AP first probes at $p_i^1 = 10Mbps$ and finds $r_i^1 = 5Mbps$, it can use $5Mbps * \beta$ as the next probe value, and be assured that it did not skip any bitrate. The idea is that two neighboring bitrates in general are not very close (We found $\beta = 0.8$ a reasonable value after doing a statistics from 100 videos from YouTube).

Ideally, each client receives an equal bitrate: $\alpha_1 \times p_1 = \alpha_2 \times p_2 = ... = \alpha_N \times p_N$, and $\sum_{i=1}^{N} p_i = P$. We use this condition to find the initial probe value:

$$p_i^{init} = \frac{P}{(\frac{\alpha_i}{\alpha_1} + \frac{\alpha_i}{\alpha_2} + ... + \frac{\alpha_i}{\alpha_N})} \tag{4.2}$$

In this case, the players whose lowest bitrates are higher than the fair share will encounter stalls (which allow us to infer their lowest bitrates), while the other players will adapt appropriate bitrates based on the allocated bandwidth.

When multiple probes (for different players) are simultaneously executed, the AP might have to schedule those probes over time due to the limit of bandwidth. This requires the design of efficient probe scheduling algorithms, the objective of which is to acquire $r_i, \forall i \in \{1..N\}$ in the shortest time.

For a given set of players $\{1..N\}$ that are currently being probed, let l_i denote the delay from the time when p_i is allocated to i to the time when the AP finds r_i. Note that l_i includes both bitrate adaptation delay and bitrate detection delay. This work assumes that the average or distribution of l_i for a given player type is known to the AP. Such knowledge could be gained over time, and we leave this for future work. Instead, this work focuses on modeling and designing efficient probe scheduling algorithms,

We model this problem as a strip packing problem [60]: Given N items, each with width w_i and length l_i, and one container with fixed width and variable length, the problem is to pack all items with minimum length of the container (see Figure 4.6). This problem is known to be NP-hard. Only heuristics are known so far. And according to [60], the best solution is given by [27]. We implement this algorithm to schedule the bitrate detection module. The algorithm works by recursively picking up the item that best fits the current gap in unused bandwidth. Please refer to [27] for more details.

4.5.4 Resource Allocation

Recall that the objective of this work is to achieve the $max - min$ fairness of QoE, whose metric is defined by a tuple $\langle s, b \rangle$. s is the primary sort key based on stall ratio and rate of stall events, and b is the secondary sort key based on video quality, which is related to its bitrate.

Assume that the AP has gathered a set of bitrates $R_i = r_i^1, r_i^2, ..$ in the bitrate

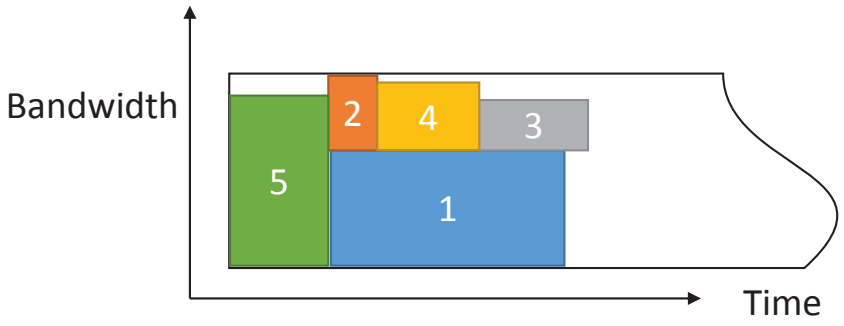


Figure 4.6: The Strip Packing Problem. Clients 1−5 have different bandwidth requirements and delays. The bandwidths are shown as widths and delays are shown as lengths. The problem is to pack all the items with minimum length of the strip.

inference stage for each player $i \in \{1..N\}$. Let r_i^{min} denote the minimum bitrate of player i in R_i. The resource allocation algorithm considers two cases: 1) The total resource P is not sufficient to support all players even at their lowest bitrates, i.e., $P < \sum_{i=1}^{N} \frac{r_i^{min}}{\alpha_i}$. In this case, the algorithm assigns physical network resource proportionally to their lowest bitrates (Line 3). 2) When $P \geq \sum_{i=1}^{N} \frac{r_i^{min}}{\alpha_i}$, the algorithm first assigns resource to guarantee the lowest bitrate for each player (Line 5), and then repeatedly increase the lowest bitrate among all N players (Lines 8-12), until no further increment is possible. This algorithm gives the optimal solution if the bitrate information of players are accurate and complete.

4.6 Experiments

To evaluate the performance of the proposed approach, we set up an AP hotspot using a Thinkpad laptop running Ubuntu 13.10. The laptop has dual network cards. The first (wired) network card acts as a bridge between the Internet and the second

Algorithm 10: Network Resource Allocation

Input: $P, R_i, \forall i \in \{1..N\}$

Output: p_i: i's share of P, $\forall i \in \{1..N\}$

if $P < \sum_{i=1}^{N} \frac{r_i^{min}}{\alpha_i}$ // $r_i^{min} := \min\{R_i\}$

then

  $p_i \leftarrow P \times \frac{r_i^{min}}{\alpha_i \times \sum_{j=1}^{N} \frac{r_j^{min}}{\alpha_j}}, \forall i$ // allocate resource proportionally

else

  $r_i \leftarrow r_i^{min}, \forall i$ // assign each player the lowest bitrate

  $P_{remain} \leftarrow P - \sum_{i=1}^{N} \frac{r_i}{\alpha_i}$

  while $P_{remain} > 0$ **do**

    find j such that $r_j = \min\{r_1..r_N\}$ and $r_j != \max\{R_j\}$

    $r_j^{+1} \leftarrow \min\{r \in R_j | r > r_j\}$ // next higher bitrate

    if $P_{remain} \geq \frac{r_j^{+1} - r_j}{\alpha_j}$ **then**

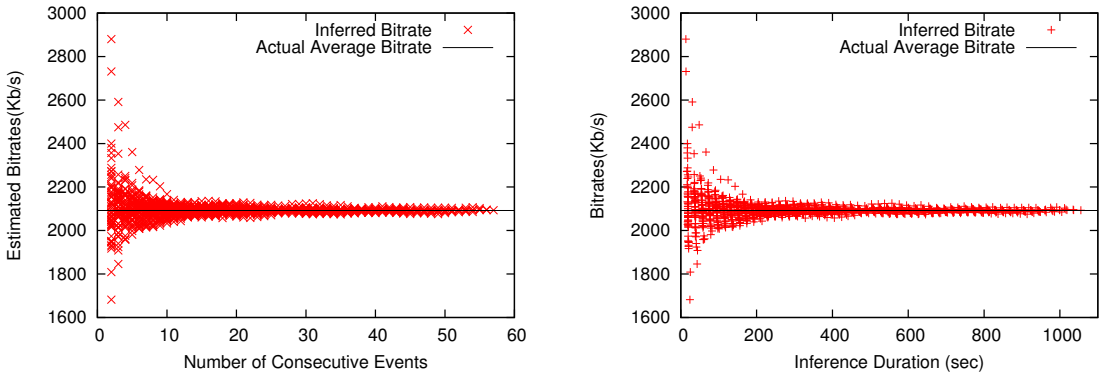
      $P_{remain} \leftarrow P_{remain} - \frac{r_j^{+1} - r_j}{\alpha_j}$

      $r_j \leftarrow r_j^{+1}$ // increase the bitrate of j

    else

      break

  $p_i \leftarrow \frac{r_i}{\alpha_i}, \forall i$



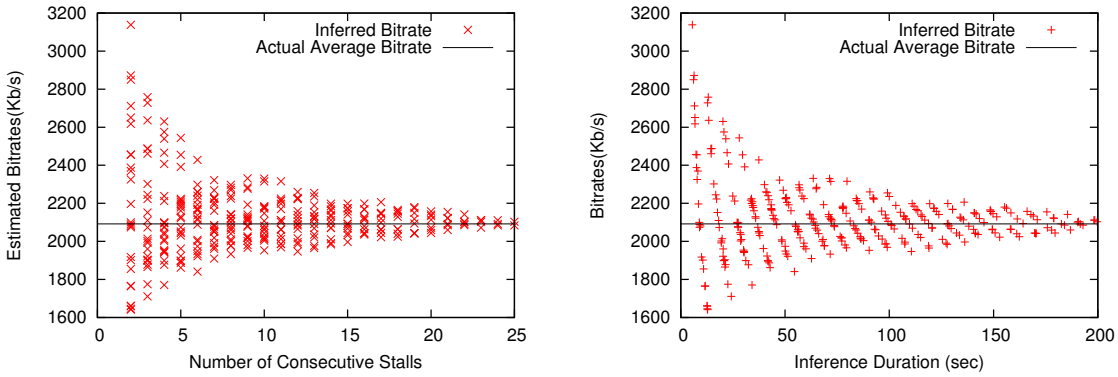
(a) Bitrate vs. Number of Download Events

(b) Bitrate vs. Time

Figure 4.7: The accuracy of bitrate inference with different number of periodical download start events and different inference time. Each point denotes an inferred bitrate corresponding to the number of events or inference time. (a) The inferred bitrates vs. the number of periodical download events. (b) The inferred bitrates vs. the length of the inference.

(wireless) network card, which serves as an AP hotspot. We use Dummynet to control the bandwidth allocated to each client, and use Wireshark to log the traffic volume. The client module was implemented on an HP desktop running Ubuntu 13.10 and a Samsung Galaxy S-IV smartphone running Android 4.3. The client and AP modules communicate through sockets.

The Accuracy of Bitrate Inference based on Periodical Download. To test the relationship between the accuracy of bitrate inference and number of consecutive events, we play a video on the client for a sufficiently long time, and then plot the inferred bitrates with variable number of consecutive events. Figure 4.7 shows the result based on the YouTube flash player. We found that the inferred bitrates are closer to the actual average bitrate with more consecutive events and longer inference time. Even with only two consecutive download events (i.e., one period, roughly 18 seconds), most of the inferred bitrate is still within 10% of the actual bitrate. We



(a) Bitrate vs. Number of Stall Events

(b) Bitrate vs. Time

Figure 4.8: The accuracy of bitrate inference with different number of stall events and different inference time. Each point denotes an inferred bitrate corresponding to the number of events or inference time. (a) The inferred bitrates vs. the number of stall events. (b) The inferred bitrates vs. the length of the inference.

consider this error is tolerable for the purpose of network resource allocation at a local AP. Note that to compare the inferred bitrate with the actual average bitrate in Figure 4.7, the packet header (overhead) was deducted from the accumulated traffic. In reality, we recommend to keep the overhead within the calculation. In this way, the result will be closer to the required throughput (with overhead considered), even though it is slightly higher compared to the original video bitrate.

The Accuracy of Bitrate Inference based on Stall Events. Figure 4.8 shows the inferred bitrates based on different number of consecutive stall events for the same video. The result shows that the stall-based bitrate inference has slightly higher error compared to the periodical download based inference approach, due to the delay in stall detection and the client-AP communication. Meanwhile, the time it takes to acquire more accurate bitrate information could be smaller, due to the

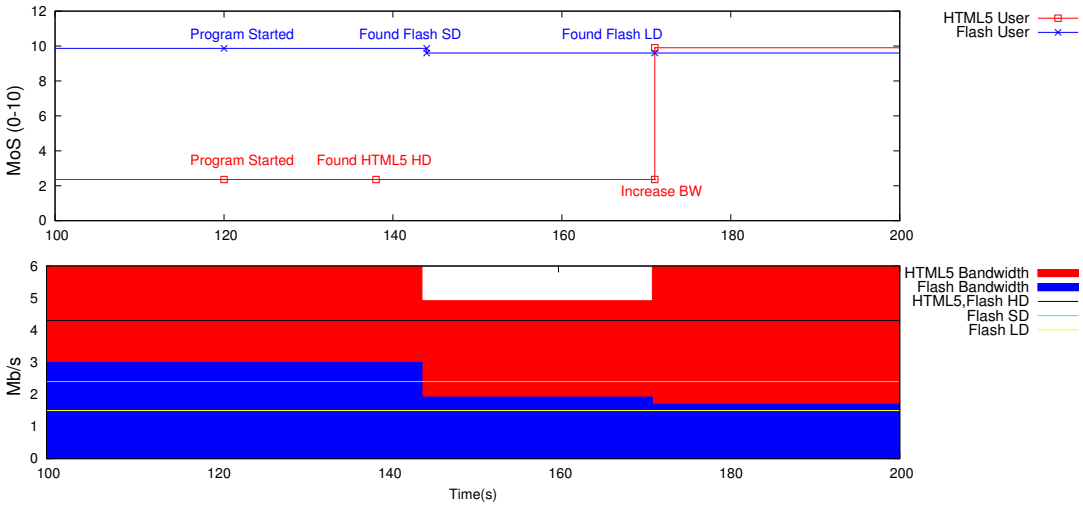


Figure 4.9: The proposed solution improves the QoE of clients. Two video clients share a bottleneck link of $6Mbps$. The HTML5 player is streaming a video at bitrate $4.3Mbps$, and the flash player is streaming a video with candidate bitrates $\{4.3Mbps, 2.4Mbps, 1.5Mbps\}$.

reason that stall events are more frequent than periodical download events in our experiment.

Improving QoE. This experiment tests how can the proposed solution improve the QoE of users. Two players are involved in the test: the first player is a DASH player (the YouTube Flash player) streaming a video with candidate bitrates of $\{4.3Mbps, 2.4Mbps, 1.5Mbps\}$. The second player is a non-DASH player (the YouTube HTML5 player) streaming the same video at $4.3Mbps$. The clients share a bottleneck link of $6Mbps$ (Figure 4.9), assuming that they have the same physical transmission rate. We use the function proposed in [84] as $\mathcal{S}(*)$ to map the stall profile to a MOS of $0-5$, and use the function proposed in [49] to map the video bitrate to a MOS of $0-5$. To get a combined value, only the first MOS is

used when stalls exist, while the summation of two MOSs are used when no stall exists. Initially, two clients are allocated the equal share of bandwidth $(0 - 120s)$. The HTML5 player has low score since stall occurs at the player. At time $120s$, we start the program, which gets the current bitrates of HTML5 player $(4.3Mbps)$ and flash player $(2.4Mbps)$ at time $137s$ and $144s$, respectively. Then the program decreases the bandwidth of the flash player to $(2.4 \times 0.8 = 1.92Mbps)$, and gets the next bitrate $(1.5Mbps)$ of the flash player at time $173s$. Finally, the program adjusts the bandwidth allocation to $4.3Mbps$ and $1.7Mbps$ for the HTML5 and flash player, respectively. As a result, the QoE of the user of HTML5 player is significantly improved at the cost of a slight decrease to the QoE of the user of the flash player.

Chapter 5

EASYBID: ENABLING CELLULAR OFFLOADING VIA SMALL PLAYERS

Deploying small cells to provide blanket coverage by WSPs alone is neither economically efficient nor practically feasible. Therefore, offloading through third-party owned small cells is an increasingly popular mechanism. In order to enable the small players, such as businesses and individual owners to make their services available to the bigger WSPs to help offload , a simple, practical and easy-to-use payment machinery needs to be devised.

Existing auction mechanisms usually assume that bidders can precisely estimate their true valuations, and they ignore the significant overhead to sellers incurred for obtaining a precise estimation. Such assumption is unrealistic in femtocell networks. To allow imprecise valuations, we introduce the novel concept of *perceived valuation*, which is a value that can be acquired by the seller at little or no cost.

We further propose two novel metrics: *partial truthfulness*, and *imprecision loss*, to measure the quality of a truthful auction that accepts perceived valuations. Based on this, we propose EasyBid, a new auction model that provides guarantees for truthfulness even when considering a system with imprecise valuations.

Finally, we design a dynamic programming based algorithm which aims to maximize the WSP's utility while satisfies any given constraints on partial truthfulness and imprecision loss. Through simulations, we show that the utility achieved by

EasyBid with imprecise valuations can be close to the optimal solution that assumes precise valuations.

5.1 Related Work

Efficient Auctions and *Optimal Auctions* are the main two types of auctions that aim to maximize the social welfare and the buyer's utility (in reverse auctions) , respectively. VCG auctions [121, 36, 51, 80] are among the most well-studied truthful efficient auctions. Unlike VCG auctions, [97] focuses on designing optimal auctions by a set of tools including posing a reserve price or charging an entry fee. One well-known result is that the optimal auction is simply the VCG auction with an optimal reserve price in simple environments (regular and i.i.d. distribution). Those works assume precise valuations. The topic of imprecise valuation has been covered in some economic works, including [113, 38, 57, 96]. In [113], the author brings out an intriguing phenomenon called Winner's Course, which says the winner will tend to overpay (i.e., receive negative utility) in common value forward auctions when bidders cannot precisely estimate the item. [38, 57, 96] focus on the procedure of valuation discovery, and strategy analysis of bidders. In contrast, this work focuses on handling imprecise valuations through mechanism design.

Existing auction works in the literature of wireless network can be roughly categorized as follows: wireless spectrum trade [138, 122, 137, 41, 64], cooperative communication [127, 139], and data offloading [34, 59, 42, 61, 92]. Wireless spectrum trade between primary and secondary owners has been studied with double auctions [138, 122], efficient VCG-based single auction [137, 41] and optimal single auction [64]. The problems studied in those works are different with this work. [127, 139] study the topic of auction design in cooperative communication. The objective is to maximize bandwidth by maximizing the efficiency of auction, which is different from

ours. Finally, efficient [34, 59, 92, 61] and optimal [42] auction based data offloading has recently gained a lot of interests. [34] focuses on the access control problem in femtocell networks. The authors propose a VCG-based reverse auction framework for fair and efficient access control. [92] proposes to use WiFi for cellular data offloading, and aims to maximize the system efficiency. [61] considers data offloading between multiple network operators and multiple femtocells by proposing a double auction framework aiming to maximize the social welfare. [59] proposes a VCG-based auction framework that aims to maximize efficiency. [42] is the closest work to ours. It proposes an auction framework that allows the WSP to leverage resources from third-party resource owners on demand. The problem of resource allocation between the WSP and third parties is formulated as a linear program, which aims to minimize the cost of the WSP. However, all those existing works simply assume precise valuations and ignore the problem of imprecise valuations, which, in contrast, is considered in this work.

5.2 Problem Formulation

5.2.1 Basic Settings

Consider a cellular network which consists of Macrocell Base Stations, third-party owned Femtocells (or WiFi hotspots), and Mobile Users. This network is geographically and chronologically divided into sub-networks to conduct separate auctions. This work focuses on one such sub-network that consists of M femtocells. The WSP and femtocell owners are the buyer and sellers of femtocell services, respectively, and the auction is transparent to mobile users.

Let V_f denote the true valuation of seller f $(1 \leq f \leq M)$ for each time unit of service on a single channel, which is a hidden value to seller f. We assume the true

valuations are non-negative, and there is a value V_{max}, such that, $V_f \in [0, V_{max}], \forall f$. V'_f denotes the *perceived valuation* of f, which is an approximate valuation that is possible for f to acquire at little or no cost. Assume $|V_f - V'_f| \leq \epsilon, \forall f$ for some constant ϵ, which denotes the estimation error of sellers. Similarly, we assume the perceived valuations are also non-negative, and bounded by V_{max}. Therefore, $V'_f \in [\max\{0, V_f - \epsilon\}, \min\{V_{max}, V_f + \epsilon\}]$. We introduce V'_f to address the following problems:

- f cannot precisely estimate V_f.
- V_f is a variable that varies within some range over the validity period of the auction, which can be defined by $V'_f \in [\max\{0, V_f - \epsilon\}, \min\{V_{max}, V_f + \epsilon\}]$.

To participate in the auction, f submits a bid denoted by B_f. *A truthful auction is redefined as one in which all sellers submit their perceived valuations as their bids, i.e.,* $B_f = V'_f, \forall f$.

Let G denote the average savings of the WSP for each unit of femtocell service, generated from the benefit of freed up cellular resources, reduced power consumption, etc. Since the WSP can arbitrarily divide the cellular network, e.g., by location, by time (weekday vs weekend, daytime vs nighttime, etc.), and conduct auctions separately, we assume G is stable and known to the WSP for a given sub-network.

We consider an online auction model in which a service request could be sent to any femtocell at any time, depending on the locations of mobile users, and the request needs to be immediately responded on arrival. Therefore, it assumes there is only one item (service) on transaction in one auction.

5.2.2 Motivation

Let us define the seller's (buyer's) *utility* as the difference of its received payment (saving) reduced by its true valuation (its payment). We illustrate the motivation and objective of EasyBid through the following examples.

One Femtocell, Precise Valuation: Consider a network with one femtocell f. Let $F_f(*)$ denote the cumulative distribution function (CDF) of V_f over $[0, V_{max}]$, and U_{WSP} denote the utility of the WSP. Assume $\epsilon = 0$ and a *reserve price* based optimal auction works in the following way:

- The WSP sets a reserve price x.
- f submits its bid B_f.
- x plays as a cutoff: f wins the auction and receives a payment of x if $B_f \leq x$.

It is obvious that the auction is *truthful* and *individually rational*, i.e., submitting V_f is a dominant strategy [104] for f, and f is guaranteed to not receive a negative utility. To find an optimal reserve price, note that $U_{WSP} = F_f(x) \times (G - x)$, in which, $F_f(x)$ is the probability that the auction is successful, and $G - x$ is the utility of the WSP if it is successful. U_{WSP} can be maximized based on $F_f(x)$. Take the uniform distribution for example, the optimal reserve price is $x = \min\{\frac{G}{2}, V_{max}\}$. Let $G = 14$ and $V_{max} = 10$, then $x = \$7$ and $U_{WSP} = \frac{7}{10} \times (14 - 7) = 4.9$.

One Femtocell, Imprecise Valuation: Now, assume $\epsilon > 0$, and V_f' may or may not be equal to V_f (a hidden value). The previous mechanism is now revised as follows:

- The WSP sets a reserve price x.
- f submits its bid B_f.
- $x - \epsilon$ is the cutoff: f wins the auction and receives x if $B_f \leq x - \epsilon$.

The cutoff is $x-\epsilon$ in this auction (compare to x in the precise valuation auction), such that when f submits V_f', its valuation V_f, upper bounded by $V_f' + \epsilon$, never exceeds the payment x to guarantee worst-case individual rationality (IR). Since $U_{WSP} = Pr(V_f' \leq x-\epsilon) \times (G-x)$, for ease of discussion, assume V_f' is also uniformly distributed over $[0, V_{max}]$. In this case, the optimal reserve price is $x = \min\{\frac{G+\epsilon}{2}, V_{max}\}$. When $G = 14$,$V_{max} = 10$ and $\epsilon = 2$, $x = \$8$, and $U_{WSP} = \frac{8-2}{10} \times (14-8) = 3.6$. Observe that:

1. U_{WSP} is less in this auction, because the auction fails when $V_f' > x - \epsilon = 6$, even though it might be the case that $V_f \leq 8$, i.e., to achieve worst-case IR, some potential transactions are rejected.

2. Submitting V_f' truthfully is a dominant strategy (DS) for f only if $V_f \in [0, 4]$ or $V_f \in [8, 12]$. Otherwise, if V_f is within $6 \pm \epsilon$, submitting V_f' is not necessarily the best choice: if $V_f' > 6$, f loses some potential utility.

3. If the loss happens, f loses 100% of its maximum possible utility. For example, assume $V_f = 5$ and $V_f' = 7$, then its maximum possible utility is $\$8 - 5 = 3$ assuming f precisely knows V_f. However, due to the imprecision issue, f actually receives 0 utility: a 100% loss. We call this percentage loss of utility the Imprecision Loss (IL, formally defined later). In an imprecise valuation auction system, sellers' IL needs to be accounted for, in order to incentivize sellers to participate.

One Femtocell, Imprecise Valuation, Multiple Reserve Prices: EasyBid can increase the utility of the WSP and reduce the IL of sellers by placing multiple reserve prices. One possible solution with two reserve prices proposed by EasyBid works as follows:

- The WSP sets two reserve prices: $\$8, \$10$.

- f submits its bid B_f.

- Sets a cutoff 4: if $B_f \in [0, 4)$, approve the transaction and pay f \$8; if $B_f \in [4, 10]$, approve the transaction with probability $\frac{2}{3}$ and pay f \$10 only if it is approved.

This approach guarantees worst-case individual rationality ($\$8 \geq 4 + \epsilon$) and has the following properties:

1. *Precision Compatible*: if f precisely knows V_f, bidding V_f truthfully is a dominant strategy for f. This will be clear after we present EasyBid in Section 5.3.

2. *Smaller IL*: Similarly, when V_f is within $\pm\epsilon$ of the cutoff value 4, f could lose part of its potential utility. For example, assume $V_f = 2$ and $V_f' = 4$, then the maximum possible utility f can get when knowing V_f is $(\$8 - V_f) = 6$ (simply let $B_f = V_f$). Without knowing V_f, f submits V_f', and its utility is $(\$10 - V_f) \times \frac{2}{3} = \frac{16}{3}$. The IL in this case is about 11%. Actually, the worst-case IL is 25% for any pair of V_f and V_f' (details omitted).

3. *Higher WSP Utility*: The rationale behind decreasing IL is that *if the overhead to f for acquiring a precise valuation (or strategizing on imprecise valuation) is larger than its IL in the auction, f is likely to accept the loss and submit V_f' truthfully.* Assume this is the case, then the utility of WSP is given by the summation of its expected utility from two possible outcomes: $Pr(V_f' < 4) \times 1 \times (G - 8) + Pr(4 \leq V_f' \leq 10) \times \frac{2}{3} \times (G - 10) = 4.0$.

Table 5.1 shows a comparison of the single reserve price solution and this solution.

Multiple Competing Sellers: Unlike the traditional auctions (e.g., VCG) that collect bids from all sellers and determine winners based on bids, EasyBid breaks

Table 5.1: Single vs Multiple Reserve Prices

Solutions	Single Reserve	Double Reserves
Worst-case IR?	Yes	Yes
WSP Utility	3.6	4.0
DS Range	$[0, 4]$,$[8, 10]$	$[0, 2)$,$[6, 10]$
Non-DS Range	$(4, 8)$	$[2, 6)$
Seller's IL	100%	25%
Seller's Preference	Low	High

the multi-seller auction into prioritized sequential one-seller auctions and pays the winner the corresponding reserve price. The priority of sellers can be determined in many ways, e.g., based on their historical service qualities, or the mobile user's received SINRs. For example, suppose sellers a and b can both serve a user who needs femtocell service, and a provides higher SINR than b. A one-seller auction is first conducted between the WSP and a. If a wins the auction, the WSP pays a a reserve price based on its bid; otherwise, the second auction is conducted between the WSP and b, and so on. Such a prioritized sequential auction model naturally suits the wireless resource auction in that: It reduces communication overhead between different parties: the WSP and the user only need to communicate with one seller at a time rather than with all sellers; and, the winner determination procedure, which depends on the priority of sellers, is more flexible compared to an alternate approach in which the winners are determined as an outcome of the bidding process.

5.2.3 Objective

EasyBid considers a network of M femtocells with imprecise valuations distributed over $[0, V_{max}]$, the CDF of which is denoted by $F(*)$. *We define the Partial Truthfulness Factor (PT Factor) of an auction system as the least probability (worst-case) that submitting one's perceived valuation is a dominant strategy.* In the previous example, the PT can be calculated based on the DS range: $\frac{(2-0)+(10-6)}{10} = 0.6$. Let $U_f(V_f)$ and $U_f(V'_f)$ denote the utility of seller f when it bids V_f (assume it knows) and V'_f, respectively. *The Imprecision Loss (IL) of seller f, given by $\frac{U_f(V_f)-U_f(V'_f)}{U_f(V_f)}$, is the worst-case fractional loss of utility when f submits V'_f.* This work assumes that certain requirements over PT and IL have to be met, for the WSP to compete with its opponents and incentivize femtocell owners. Given this, EasyBid seeks to find a multi-reserve-price based solution for the WSP to maximize its utility.

For a solution with N reserve prices, EasyBid computes three vectors: $\vec{S}, \vec{R}, \vec{P}$. Vector $\vec{S} = \{S_i, i = 1..N\}$ divides $[0, V_{max}]$ into N segments with lengths $S_1..S_N$, and $\sum_{i=1}^{N} S_i = V_{max}$. This work uses $\sum_1^N \vec{S} = \sum_{i=1}^{N} S_i$ for short. If the valuation V_f of some seller f satisfies $\sum_1^{i-1} \vec{S} \leq V_f < \sum_1^i \vec{S}$, we say seller f is in segment $i(i \in \{1, ..N\})$, denoted by $f \in S_i$. (With a slight abuse of notation, we also use S_i to denote segment i where it is unambiguous). Finally, let $f \in S_N$ if $V_f = V_{max}$. Each S_i is also associated with an approval ratio $R_i(0 \leq R_i \leq 1)$ and a payment $P_i(P_i \geq 0)$. For sellers in S_i, R_i denotes the probability of serving an incoming mobile user, while P_i is the amount of payment made to the sellers if the auction succeeds. $\vec{R} = \{R_i, i = 1..N\}, \vec{P} = \{P_i, i = 1..N\}$ are the vectors of R_i and P_i with length N, respectively. Note that $\vec{S}$ is used to describe the "cutoff" in the auction. The solution in the previous example can be denoted by: $\vec{S} = \{4, 6\}, \vec{R} = \{1, \frac{2}{3}\}, \vec{P} = \{8, 10\}$. Formally, for any given ϵ, α, β, let $\mathbb{R}_f \triangleq [\max\{0, V_f - \epsilon\}, \min\{V_{max}, V_f + \epsilon\}]$, then the

problem can be defined as follows,

$$\begin{aligned} \max_{N,\vec{S},\vec{R},\vec{P}} \quad & \textstyle\sum_{i=1}^{N} d_i \times R_i \times (G - P_i) \quad \text{s.}t. \\ IR: \quad & U_f(V_f') \geq 0,\ \forall f \in \{1,..M\}, \forall V_f' \in \mathbb{R}_f \\ PT: \quad & \frac{\left|\{f|U_f(V_f') \geq U_f(b), \forall V_f' \in \mathbb{R}_f, \forall b \in [0, V_{max}]\}\right|}{M} \geq \alpha \\ IL: \quad & \frac{U_f(V_f) - U_f(V_f')}{U_f(V_f)} \leq \beta,\ \forall f \in \{1,..M\}, \forall V_f' \in \mathbb{R}_f \end{aligned} \tag{5.1}$$

in which, d_i is the fraction of sellers that are within S_i, given by $F(\sum_0^i \vec{S}) - F(\sum_0^{i-1} \vec{S})$. For any single demand, assume that it could take place at any femtocell with equal opportunity, then d_i is the probability that it takes place at some femtocell in segment i. The objective function can be interpreted as the expected utility of WSP from a single demand. The first constraint guarantees worst-case IR for any f that submits its perceived valuation. The second constraint guarantees that submitting perceived valuations is truthful (has no IL) for at least α fraction of sellers. For the remainders, the third constraint guarantees that the maximum IL is no more than β.

5.3 The Framework of EasyBid

For ease of understanding, in this section we present the EasyBid framework *while assuming sellers can precisely estimate their true valuations,* i.e., $\epsilon = 0$ and $V_f = V_f', \forall f$. Given any N, we are to derive constraints over $\vec{S}, \vec{R}, \vec{P}$ to achieve truthfulness and individual rationality. We will show how this framework is applied to address Problem (5.1) in Section 5.4.

5.3.1 Constraints over $\vec{S}, \vec{R}, \vec{P}$

For any femtocell $f \in S_i$ with true valuation V_f, its expected utility from an average demand is given by,

$$U_f(V_f) = R_i \times (P_i - V_f) \tag{5.2}$$

in which, R_i is the probability that f serves incoming demands, and P_i is the payment it receives if it provides service. To achieve individual rationality, P_i has to be at least as large as the maximum value in segment S_i. So,

$$P_i \geq \sum_1^i \vec{S}, \forall i \in \{1, ..N\} \tag{5.3}$$

Note that, by manipulating its bid B_f, f could change the segment number to which it belongs, i.e., it could claim $f \in S_j (j \neq i)$ instead of $f \in S_i$, and get a different set of approval ratio and payment. To guarantee truthfulness, appropriate constraints need to be satisfied, outlined as follows:

Lemma 5.3.1. *$\vec{R}$ is a non-increasing sequence in truthful auctions.*

Proof. It holds trivially when $N = 1$. For $N \geq 2$, assume there is some increasing subsequence in $\vec{R}$, and suppose it occurs at i: let $R_i < R_{i+1}$ denote the occurence of increasing approval ratios, where $1 \leq i \leq N-1$. Let $f_1 \in S_i$ denote a femtocell whose true valuation $V_{f1} = \sum_1^{i-1} \vec{S}$ happens to be the minimum in segment i, and $f_2 \in S_{i+1}$ denote a femtocell whose true valuation $V_{f2} = \sum_1^i \vec{S}$ is the minimum in segment $i+1$. In truthful auction, there should be no incentive for f_1 to claim $f_1 \in S_{i+1}$. Based on the utility function in Equation (5.2), the utility f_1 receives when truthfully claiming $f_1 \in S_i$ should be at least as large as when claiming $f_1 \in S_{i+1}$: $R_i \times (P_i - V_{f1}) \geq R_{i+1} \times (P_{i+1} - V_{f1})$. Given that $V_{f1} = \sum_1^{i-1} \vec{S}$, we get $R_i \times (P_i - \sum_1^{i-1} \vec{S}) \geq R_{i+1} \times (P_{i+1} - \sum_1^{i-1} \vec{S})$. Similarly, there is no incentive for f_2 to claim $f_2 \in S_i$, $R_{i+1} \times (P_{i+1} - V_{f2}) \geq R_i \times (P_i - V_{f2})$. Given $V_{f2} = \sum_1^i \vec{S}$, we get $R_{i+1} \times (P_{i+1} - \sum_1^i \vec{S}) \geq R_i \times (P_i - \sum_1^i \vec{S})$. By adding the previous two inequalities and simplifying, we get $R_i \times S_i \geq R_{i+1} \times S_i$, which means $R_i \geq R_{i+1}$. This contradicts with our assumption that $R_i < R_{i+1}$. □

Lemma 5.3.2. *The payment vector $\vec{P}$ is a non-decreasing sequence in truthful auctions.*

Proof. Consider two segments S_i and S_j, for which $i < j$. Since $R_i \geq R_j$, then if $P_i > P_j$, all sellers in segment j would get higher utility if they lie and claim they are in segment i. Therefore, $\vec{P}$ is a non-decreasing sequence. □

Now we use lemma 5.3.1 and 5.3.2 to derive the following constraints to achieve truthfulness. Consider the first segment, and suppose seller $f \in S_1$. It is clear that f will not get higher utility by submitting a bid that is within the same segment as its true valuation, as doing that will not change its approval ratio or payment. To *prevent it from submitting a higher bid that belongs to the second segment*, $\forall V_f \in S_1$, the following constraint has to be satisfied:

$$R_1 \times (P_1 - V_f) \geq R_2 \times (P_2 - V_f), \forall V_f \in S_1 \tag{5.4}$$

$$\Leftrightarrow \quad \frac{R_2}{R_1} \leq \frac{P_1 - V_f}{P_2 - V_f}, \forall V_f \in S_1 \tag{5.5}$$

$$\Leftrightarrow \quad \frac{R_2}{R_1} \leq \frac{P_1 - S_1}{P_2 - S_1} \tag{5.6}$$

The left hand side of (5.4) is the utility of bidding its true valuation, while the right hand side is the utility of submitting a higher bid that is within the second segment. Note that since $P_1/P_2 \leq 1$ and $0 \leq V_f < S_1$, the right hand side of (5.5) achieves the minimum value when V_f tends to S_1, and that is how we get (5.6).

Similarly, to prevent f submitting a bid that is in the third or other following segments, it requires:

$$R_1 \times (P_1 - V_f) \geq R_3 \times (P_3 - V_f), \forall V_f \in S_1$$

$$\cdots \quad \cdots$$

$$R_1 \times (P_1 - V_f) \geq R_N \times (P_N - V_f), \forall V_f \in S_1$$

To sum up, we have:

$$\frac{R_2}{R_1} \leq \frac{P_1 - S_1}{P_2 - S_1}, \frac{R_3}{R_1} \leq \frac{P_1 - S_1}{P_3 - S_1}, \ldots, \frac{R_N}{R_1} \leq \frac{P_1 - S_1}{P_N - S_1} \tag{5.7}$$

Now consider sellers in other segments. Suppose $f \in S_2$, i.e., $\sum_1^1 \vec{S} \leq V_f < \sum_1^2 \vec{S}$. To prevent f placing a higher bid,

$$\begin{array}{ll} \frac{R_3}{R_2} & \leq \frac{P_2 - \sum_1^2 \vec{S}}{P_3 - \sum_1^2 \vec{S}} \\ \frac{R_4}{R_2} & \leq \frac{P_2 - \sum_1^2 \vec{S}}{P_4 - \sum_1^2 \vec{S}} \\ \ldots & \ldots \\ \frac{R_N}{R_2} & \leq \frac{P_2 - \sum_1^2 \vec{S}}{P_N - \sum_1^2 \vec{S}} \end{array} \tag{5.8}$$

Then by listing similar constraints for all other segments, eventually for segment S_{N-1}, we obtain:

$$\frac{R_N}{R_{N-1}} \leq \frac{P_{N-1} - \sum_1^{N-1} \vec{S}}{P_N - \sum_1^{N-1} \vec{S}} \tag{5.9}$$

Note that, if both of the first constraint of (5.7) and (5.8) are satisfied, then by doing a multiplication, we have:

$$\frac{R_3}{R_1} \leq \frac{(P_1 - S_1)(P_2 - S_1 - S_2)}{(P_2 - S_1)(P_3 - S_1 - S_2)} \leq \frac{(P_1 - S_1)}{(P_3 - S_1 - S_2)} \leq \frac{P_1 - S_1}{P_3 - S_1}$$

which means the second constraint of (5.7) is redundant. Similarly, it is trivial that the i^{th} constraint of (5.7) is redundant if both the first constraint of (5.7) and the $(i-1)^{th}$ constraint of (5.8) are satisfied. In the same fashion, we can find that only those first constraints in each constraint group of (5.7),(5.8), and up to the segment S_{N-1} shown in (5.9) are tight (details are omitted). Finally, we get the following constraints from those groups after omitting the redundants:

$$\frac{R_{i+1}}{R_i} \leq \frac{P_i - \sum_1^i \vec{S}}{P_{i+1} - \sum_1^i \vec{S}}, 1 \leq i \leq N-1 \tag{5.10}$$

Note that (5.10) lists all the constraints to prevent sellers placing higher bids than their true valuations. To *prevent sellers from receiving higher utilities by placing lower*

bids that fall into lower segments, we list constraints for each segment in a similar way. With the same set of techniques, we get the following results:

$$\frac{R_{i+1}}{R_i} \geq \frac{P_i - \sum_1^i \vec{S}}{P_{i+1} - \sum_1^i \vec{S}}, 1 \leq i \leq N-1 \tag{5.11}$$

By putting (5.10) and (5.11) together, we conclude that:

$$\frac{R_{i+1}}{R_i} = \frac{P_i - \sum_1^i \vec{S}}{P_{i+1} - \sum_1^i \vec{S}}, 1 \leq i \leq N-1 \tag{5.12}$$

Note that Equation (5.12) and Lemma 5.3.1 imply the two well-known properties of Dominant Strategy Equilibrium: *payment identity* and *monotonicity* [62].

5.3.2 Long Term Truthfulness

A solution ($\{\vec{S}, \vec{R}, \vec{P}\}$) based on the previous discussion guarantees truthfulness of sellers for a single arriving demand. The solution does not need to be repeatedly calculated for every femtocell at every transaction. Instead, it can be applied to all femtocells and for a long term. To guarantee their long-term truthfulness, the following requirements need to be satisfied: 1) The arrivals of demands at any seller cannot be controlled by the seller itself. This can be easily satisfied if the WSP controls the arrivals , or if let users select femtocells (while on the go) based on their own preferences(e.g., signal strength). 2) The approval ratios are respected. Note that the approval ratio at a femtocell might not be fulfilled if a demand arrives while the resources are depleted. For this, one possible solution is to amortize the seller with a future demand. We leave this for future study, and instead assume that there are sufficient channels available at all femtocells. With increasing number of femtocells, we believe that the number of channels will typically not be limited due to increased channel reuse. We also study in the simulation section how the truthfulness gets affected when this assumption does not hold.

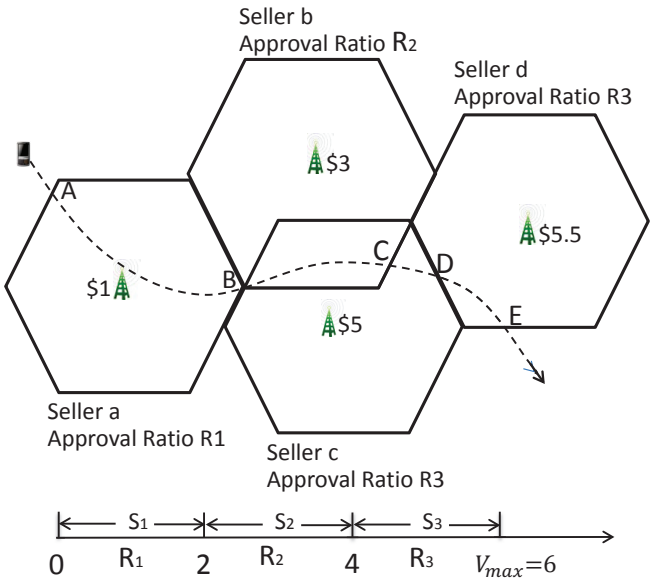


Figure 5.1: A user travels across a 4-seller femtocell network. Suppose $V_a = 1, V_b = 3, V_c = 5, V_d = 5.5$ and $V_{max} = 6$. For $N = 3$, one simple solution is $\{S_1 = S_2 = S_3 = 2\}$,$\{R_1 = 1, R_2 = \frac{1}{2}, R_3 = \frac{1}{4}\}$,$\{P_1 = 4, P_2 = 6, P_3 = 8\}$. Since $a \in S_1, b \in S_2, c, d \in S_3$, seller a uses R_1 as its approval ratio, seller b uses R_2, and sellers c and d use R_3.

Theorem 5.3.1. *Assume the arrivals of demands are independent of B_f for any given f, and sellers know their precise valuations. Given sufficient resources at local femtocells, a solution that follows lemma 5.3.1, lemma 5.3.2 and constraints* (5.3) *and* (5.12) *is truthful and individually rational.*

Proof. Section 5.3.1 shows that no seller can achieve higher utility by lying for a single demand. The independence of arrivals and sufficient resources assure that the seller cannot achieve higher long-term utility by manipulating its bid. Therefore, EasyBid is truthful. □

5.3.3 Implement EasyBid For Data Offloading

The EasyBid based online auction model can be implemented in real systems as follows: A central server computes a solution consisting of $\vec{S}, \vec{R}, \vec{P}$. Local femtocells then submit their bids to the central server. For any femtocell f, the server returns one corresponding approval ratio and payment to f, based on B_f and vector $\vec{S}$. When a mobile user sends a demand of service to f, f serves this user with probability R_i, and receives a payment P_i if f actually provides the service. After being admitted, the mobile user continues to receive service from f until it moves out of its range.

Consider the scenario shown in Figure 5.1 in which a user moves across a network of 4 femtocells. Upon reaching point A, the user sends a demand to a. a serves this user with probability R_1. If the user is approved, then it continues to receive service from a, while a receives a payment of P_1. Otherwise, it reaches B without receiving femtocell service. At point B, this user sends a new demand to seller c (c's signal strength is higher than b). Seller c approves with a probability of R_3. If the user is denied, then it can send a new demand to seller b. Otherwise, the user gets served until point D, at which point, a new demand is sent to Seller d. *Note that the expected utility of a for a given demand is* $1 \times (4-1) = 3$*, while it is* $1/2 \times (6-1) = 5/2$ *if a lies to* S_2*, and* $1/4 \times (8-1) = 7/4$ *if a lies to* S_3*. Therefore, a has no incentive to lie.* Similar arguments also hold for other sellers.

5.4 EasyBid: Deal with Imprecise Valuations

5.4.1 Understand the Constraints

When $\epsilon > 0$, to understand how the constraints in Problem (5.1) affect the solution $\{\vec{S}, \vec{R}, \vec{P}\}$,consider a naive solution that pays V_{max} to all sellers with approval ratio 1, regardless of the bids. This solution satisfies all three constraints: worst-case IR,

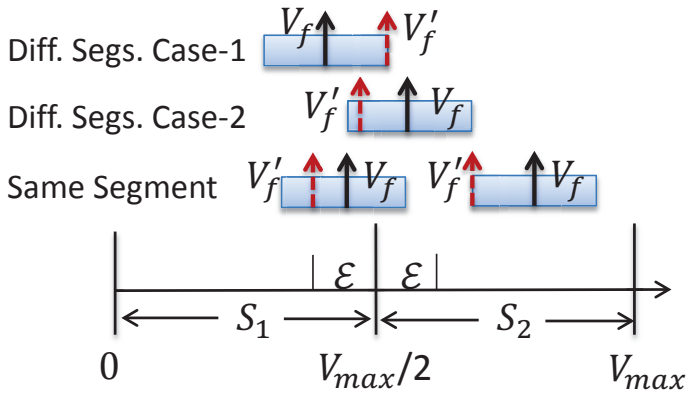


Figure 5.2: Divide the range into 2 segments. If $V_f \in \frac{V_{max}}{2} \pm \epsilon$, V_f and V_f' may or may not be in the same segment.

$\alpha - PT$ (it achieves full truthfulness) and $\beta - IL$ (0 loss if imprecise). However, it is non-optimal to the WSP due to the unique and high payment. By dividing the range into $N = 2$ segments, say $[0, V_{max}/2)$ and $[V_{max}/2, V_{max}]$, two different payments and approval ratios can be assigned based on Equation (5.12). The WSP could save on payment because P_1 (the payment to sellers in S_1) could be smaller than the naive solution. The consequence of this solution is that sellers in $[\frac{V_{max}}{2} - \epsilon, \frac{V_{max}}{2} + \epsilon]$ are not guaranteed of truthfulness (see Figure 5.2). If either $\{V_f \in S_1, V_f' \in S_2\}$, or $\{V_f \in S_2, V_f' \in S_1\}$ happens, an IL occurs. The IL of those two conditions are $\frac{R_1 \times (P_1 - V_f) - R_2 \times (P_2 - V_f)}{R_1 \times (P_1 - V_f)}$ and $\frac{R_2 \times (P_2 - V_f) - R_1 \times (P_1 - V_f)}{R_2 \times (P_2 - V_f)}$. Based on the constraints, this solution should satisfy: $1) P_1 \geq S_1 + \epsilon, P_2 \geq V_{max}$. $2) F(\frac{V_{max}}{2} + \epsilon) - F(\frac{V_{max}}{2} - \epsilon) \leq 1 - \alpha$. 3)The IL, given above, is less than or equal to β for all sellers. To conclude, *if seller f's true valuation V_f is at least ϵ far from the boundary of any segment (except 0 and V_{max}), then f is guaranteed of PT. Otherwise, it is not.* Intuitively, for the WSP, a smaller N results in higher average payment and higher α value, while a larger N can decrease the average payment as well as the α value.

5.4.2 Algorithms

Without loss of generality, let $[0, 1]$ denote the normalized range of $[0, V_{max}]$, and assume other values are normalized accordingly. Our solution considers a discrete version of problem (5.1), in which, the boundaries of segments can only be placed on a finite number of candidate locations. Selecting a subset of boundaries from n given candidate boundaries with constraints of α for PT and β for IL is a a combinatorial problem: each boundary introduces different percent of non-PT sellers and gives different increases in the utility, depending on the locations of other chosen boundaries (which affect the maximal IR of the new segment) and the payments to the new introduced segment. While holding the belief that this problem is NP-hard, we leave it for future work.

We design a dynamic programming based heuristic algorithm (Algorithm 11). The algorithm discretizes the original problem from two perspectives: the range $[0, 1]$ with an integer n and the non-PT budget $(1 - \alpha)$ with an integer m, wherein n and m are two integers that represent the number of discrete values considered. For example, if $n = 10$, then only multiples of $1/10 = 0.1$ are considered as candidate locations of boundaries. If $1 - \alpha = 0.2$ and $m = 2$, then only multiples of $0.2/2 = 0.1$ are considered as legal budgets. The algorithm takes n steps to finish (step size $\frac{1}{n}$). At each step, it finds and saves a set of m solutions for range $[x, 1]$, with x started from 1 and decreased to 0 finally. Those m solutions denote the best solutions for $[x, 1]$ with m possible non-PT budget values. At the i^{th} step (so, $x = 1 - \frac{i}{n}$), the algorithm finds the best solution for $[x, 1]$ with non-PT budget w ($w = \{1..m\} \times \frac{1}{m}$) by exhausting all possible lengths of its first segment. Since the length of the first segment (S_1) could be $\frac{1}{n}, \frac{2}{n} \cdots ..\frac{i}{n}$, so there are a total of i solutions exhausted. To construct the solution, take $S_1 = \frac{2}{n}$ for example, now that the first segment is $[x, x + \frac{2}{n}]$, the fraction of non-PT users in the first segment can be found based on $F(x)$. Let $NPT[x, y)$

Algorithm 11: Solve Utility Maximization Problem

input: α, β, ϵ, range $[0,1], F(*)$

output: $\{\vec{S}, \vec{R}, \vec{P}, U\}$// U:`utility`

for $(x \leftarrow 1; x \geq 0; x \leftarrow x - 1/n)$ **do**

  for $(w \leftarrow 0; w \leq 1-\alpha; w \leftarrow w + 1/m)$ **do**

    for $(y \leftarrow x + 1/n; y \leq 1; y \leftarrow y + 1/n)$ **do**

      if $(NPT([x,y)) \leq w)$//*use alg. 12* **then**

        if $(y = 1)$ **then**

          $e' \leftarrow \{1-x, 1, 1, (1-F(x)) \times (G-1)\}$

          if $(max[x][w] = \phi || e' > max[x][w])$ **then**

            $max[x][w] \leftarrow e'$

          continue

        $e \leftarrow max[y][\lfloor w - NPT([x,y)) \rfloor]$

        if $(e = \phi)$ **then** continue

        $S_y \leftarrow e.\vec{S}[0]$ // $1st$ `seg in` e

        $P_y \leftarrow e.\vec{P}[0]$ // $1st$ `pmt in` e

        $S_x \leftarrow y - x$ // `length of seg` $[x,y)$

        for $(P_x \leftarrow y + \epsilon; P_x \leq P_y; P_x \leftarrow P_x + 1/n)$ **do**

          $e' \leftarrow \{[S_x, e.\vec{S}], [1, \frac{P_x - y}{P_y - y} * e.\vec{R}], [P_x, e.\vec{P}]\}$

          if *!checkβIL*$([x,1], e', \epsilon)$*//alg.13* **then**

            continue

          $e'.U \leftarrow (F(y) - F(x)) \times (G - P_x) + \frac{P_x - y}{P_y - y} \times e.U$

          if $(max[x][w] = \phi \;||\; e' > max[x][w])$ **then**

            $max[x][w] \leftarrow e'$

return $(max[0][1-\alpha].U > 0)?max[0][1-\alpha] : \{V_{max}, 0, 0, 0\}$

denote this value. Since we are computing a solution with no more than w budget, the budget left for $[x + \frac{2}{n}, 1]$ is $w - NPT[x, y)$. Then it constructs the solution under consideration by expanding the previously saved solution of range $[x + \frac{2}{n}, 1]$, budget $w - NPT[x, y)$.

In Algorithm 11, $max[y][w]$ is used to save the optimal solution of range $[y, 1]$ with non-PT budget w ($0 \leq w \leq 1 - \alpha$). Lines 3-4 iterate for x and w, respectively. Line 5 tries to locate the first segment in $[x, 1]$ by locating its upper boundary y. Once the first segment $[x, y)$ is given, the portion of non-PT sellers in this segment is calculated (Line 6) using algorithm 12. and it has to be less than the current budget w. If $y = 1$ (Line 7), which means there is only one segment within $[x, 1]$, we construct the solution e' (Line 8), and save it if it is a better solution (Lines 9-10). Otherwise, it finds the saved solution for segment $[y, 1]$ with the remaining budget $\lfloor w - PTS(x, y) \rfloor$ in Line 12. (The $\lfloor\ \rfloor$ operation rounds the value down to multiples of $1/m$.) Line 15 skips current solution if it is not feasible. Line 17 iterates the payment P_x for the segment $[x, y)$, and Line 18 builds the new solution of $[x, 1]$ using the solution of $[y, 1]$. Lines 19-20 check if this solution satisfies $\beta - IL$ constraint using Algorithm 13. Lines 21-23 calculate the utility of current solution based on the utility of $[y, 1]$, and substitute the best solution if it is better than the saved solution in $max[x][w]$. Finally, the best solution for $[0, 1]$ is returned (Line 24). Note that the three constraints in Problem (5.1) are addressed in Lines 17, 4, 19, respectively.

Specifically, In Lines 18 and 21, a factor of $\frac{P_x - y}{P_y - y}$ is multiplied to $e.\vec{R}$ and $e.U$. This is because the utility of $[y, 1]$ was previously calculated based on the assumption that the the approval ratio of its first segment is 1. When another segment $[x, y)$ is added ahead of it, its new approval ratio is now given by $\frac{P_x - y}{P_y - y}$ (see Equation 5.12), in which y equals to the summed length of all segments between $[0, y)$. This factor needs to be multiplied to all other approval ratios in $e.\vec{R}$, which depend on the first one. Due

to this, a better solution ($>$) in Lines 9 and 22 is defined in the way that takes this factor into account: given two solutions of range $[x, 1]$, $e1$ and $e2$, let $e1.\vec{P}[0]$ and $e2.\vec{P}[0]$ denote the first payment element in their payment vectors, $e1$ is better than $e2$ when,

$$\begin{cases} \frac{e1.U}{e1.\vec{P}[0]-x} > \frac{e2.U}{e2.\vec{P}[0]-x} & \text{if } x > 0 \\ e1.U > e2.U & \text{if } x = 0 \end{cases}$$

Algorithm 12: Calculate Non-PT In [x,y)

1 **input:** $[x, y), \epsilon, F(*)$

2 **output:** Fraction of Non-PT Sellers

3 $\alpha' \leftarrow 0$`// the PT fraction`

4 $dsStart \leftarrow (x = 0)?x : x + \epsilon$

5 $dsEnd \leftarrow (y = 1)?y : y - \epsilon$

6 **if** *(*$dsEnd \geq dsStart$*)* **then**

7 $\quad \alpha' \leftarrow F(dsEnd) - F(dsStart)$

8 Return $F(y) - F(x) - \alpha'$

Algorithm 12 calculates the portion of non-PT sellers for a given segment $[x, y)$. The idea is that if V_f is at least ϵ far from the boundary (when the boundary is not 0 or 1), then f is counted as PT, otherwise, it is non-PT. Lines 4-5 require the minimum true valuation of a PT seller has to be at least ϵ larger than its lower boundary (except 0), and at least ϵ smaller than its upper boundary (except 1). The fraction of sellers that satisfy this constraint is calculated in Line 7. Line 8 returns the portion of non-PT as the complement of PT.

Algorithm 13 checks if a given solution for range $[x, 1]$ meets $\beta - IL$ requirement. For seller f, it can be shown that the maximum loss happens when the imprecision is maximized, i.e., $|V_f - V_f'| = \epsilon$. In Algorithm 13, for each segment S_i, it finds the minimum V_f that could have a corresponding V_f' in S_i (Line 7), and the maximum V_f that could have its V_f' in S_i (Line 13). The losses of those two cases are calculated (Line 10 and 16), and both of them have to be less than β (Lines 11-12, Lines 17-18). Based on Lines 5 and 7, the complexity of Algorithm 13 is $O(nlogn)$ (binary search on Line 7). For this, the complexity of Algorithm 11 is $O(mn^4logn)$.

5.5 Simulation

5.5.1 Simulation Settings

Our simulation considers a region of $1000m \times 1000m$, consisting of M femtocells (40 by default, varied from $10 - 100$). Each femtocell is placed at the center of a $20m \times 20m$ building, the position of which is randomly selected. Femtocells are by default assigned sufficient subchannels, while the case of limited subchannels will also be evaluated. We use the LTE module in ns-3 [114] to simulate the wireless communication. The transmission and interference radii are $100m$ and $250m$. For simplicity and ease of understanding, the simulation assumes the true valuations (in \$/second/subchannel) of femtocells are uniformly distributed within $[0, 1]$ (other distributions are also evaluated), and the perceived valuations are randomly generated within $\pm\epsilon$ of the true valuations (also $\in [0, 1]$).

Mobile users arrive at this network with certain arrival rate, and perform a random walk after that. Their speeds are randomly generated within $0 - 2m/s$. The arrival rate is by default $5usrs/min$, and varied from $1 - 10usrs/min$ (based on the speed profile, the average number of users within the region at any time is about $20 - 200$,

Algorithm 13: Check For $\beta - IL$ Requirment

input: $[x, 1], \{\vec{S}, \vec{R}, \vec{P}\}, \epsilon$

output: True or False

$\vec{L} \leftarrow$ find the sequence of lower boundaries based on x and $\vec{S}$

$\vec{U} \leftarrow$ find the sequence of upper boundaries based on x and $\vec{S}$

for $i \leftarrow 1$ *to* $\vec{S}.length$ **do**

    //Assume the index of $\vec{L}, \vec{U}$ starts at 1.

    $V_f \leftarrow \vec{L}[i] - \epsilon$, $j \leftarrow$ the segment number of V_f

    $j \leftarrow (j < 1)?1 : j$

    if $(j < i)$ **then**

        $loss = \frac{\vec{R}[j]*(\vec{P}[j]-V_f)-\vec{R}[i]*(\vec{P}[i]-V_f)}{\vec{R}[j]*(\vec{P}[j]-V_f)}$

        if $(loss > \beta)$ **then**

            return False

    $V_f \leftarrow \vec{U}[i] + \epsilon$, $k \leftarrow$ the segment number of V_f

    $k \leftarrow (k > \vec{S}.length)?\vec{S}.length : k$

    if $(k > i)$ **then**

        $loss = \frac{\vec{R}[k]*(\vec{P}[k]-V_f)-\vec{R}[i]*(\vec{P}[i]-V_f)}{\vec{R}[k]*(\vec{P}[k]-V_f)}$

        if $(loss > \beta)$ **then**

            return False

return True

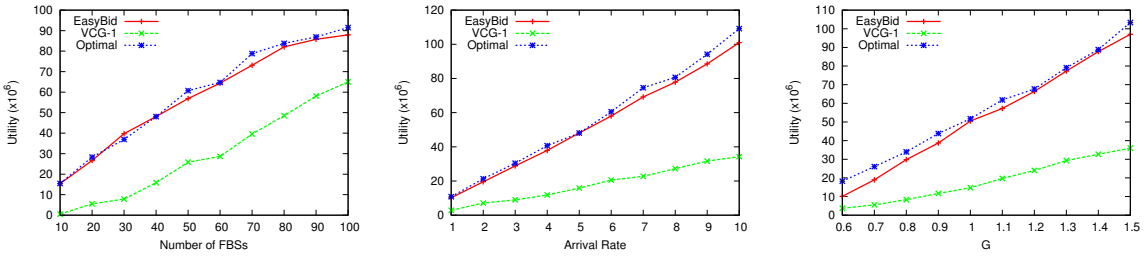
correspondingly). Users send requests to femtocells if they come across new femtocells and are not receiving services from other femtocells. The requested data rate of each user is randomly chosen from 3 categories: $\{5mbps, 500kbps, 50kbps\}$, which represent heavy, intermediate and light network usages, respectively.

The default value of G is 1 by default (varied in some setups). The discrete parameter n and m in Algorithm 11 are 500 and 100, respectively. The auction was conducted for one week in each setup. To evaluate the performance of EasyBid, we first compare EasyBid with VCG and the optimal auction assuming precise valuations, and then assume imprecise valuations and evaluate EasyBid alone by varying different factors $(\alpha, \beta, \epsilon)$. ($VCG$ is not evaluated for imprecise valuations since it loses truthfulness and worst-case individual rationality: Consider two sellers: a and b. $V_a = \$5$ and $V'_a = \$3$, while $V_b = \$2$ and $V'_b = \$4$. If they both submit their perceived valuations truthfully, a wins since $3 < 4$. The payment to a is the secondary bid \$4. As a result, a actually receives $4 - 5 = -1$ utility, and a could increase its utility to 0 by lying: submitting any value larger than 4.)

5.5.2 Simulation Results

Precise Valuations: Utilities of the WSP with Variable Femtocell Density, User Density, and Average Saving: We first assume the valuations are precise, and compare EasyBid with two auction schemes: 1) VCG1: the VCG auction with a reserve price 1. 2) Optimal: the VCG auction with reserve price 0.5, which is the optimal auction in our setup [97].

Figure 5.3 (a),(b) show that the WSP benefits from the increased density of femtocells and users in all three models. This is because higher density of femtocells results in larger coverage area and higher competition among femtocells, which can reduce



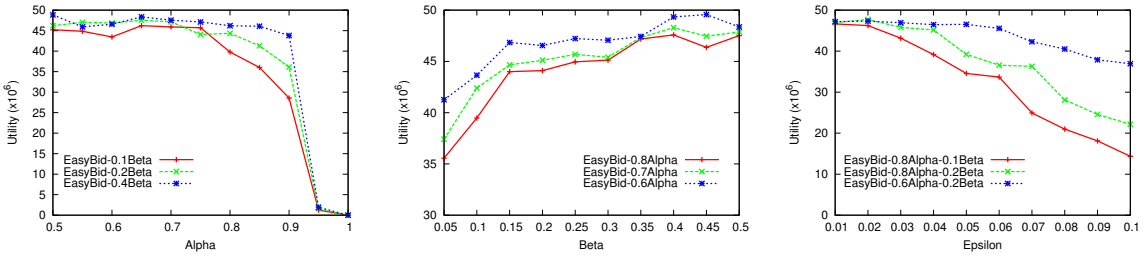
(a) Vary Femtocell Density (b) Vary User Density (c) Vary Average Saving G

Figure 5.3: The utility of the WSP under different femtocell density, user density and average cost, assuming precise valuations. EasyBid performs closely to the optimal solution when assuming precise valuations. (a) Vary the number of femtocells, the arrival rate is 5, $G = 1$. (b) Vary the arrival rate, $M = 40$ femtocells and $G = 1$. (c) Vary the average saving G, $M = 40$ and arrival rate is 5.

the average payment. (Note that EasyBid can also take advantage of the competition by polishing the approval ratios and payments.) Meanwhile, higher user density increases the chances of data offloading, thus it also benefits the WSP. The value of G unsurprisingly affects WSP as shown in Figure 5.3 (c). Overall, EasyBid performs closely to the optimal when assuming precise valuations, in spite of the smallgap between them, which might be caused by the sub-optimality of the algorithm.

Imprecise Valuations: Utilities of the WSP with Variable α, β, ϵ: When $\epsilon > 0$, we study how the values of α, β and ϵ affect the utility of the WSP, with the default setting of $M = 40$, arrival rate $= 5$ and $G = 1$.

Figure 5.4 (a) shows that increasing the value of α can decrease the utility of the WSP. However, given the same value of α, increasing the value of β can increase the utility of the WSP. This can be explained by Equation (5.1): a larger α means a tighter constraint, while a larger β corresponds to a looser constraint. Note that



(a) Increasing α Decreases Utility (b) Increasing β Increases Utility (c) Increasing ϵ Decreases Utility

Figure 5.4: The utility of the WSP under variables α, β, ϵ using$M = 40$ femtocells, 5 arrival rate and $G = 1$: assume imprecise valuations. (a) Vary α, with $\beta = \{0.1, 0.2, 0.4\}$ and $\epsilon = 0.04$. (b) Vary β, with $\alpha = \{0.6, 0.7, 0.8\}$ and $\epsilon = 0.04$. (c) Vary ϵ with different α, β.

the utilities in all three conditions are substantially decreased at $\alpha = 0.95$. This is because the simulation uses $\epsilon = 0.04$, and the algorithm can hardly find a multi-segment solution that guarantees more than 95% of sellers being PT. Note that under the same setup ($M = 40$, arrival rate $= 5$, $G = 1$), the utility was 48.5 in the optimal solution assuming precise valuations. Therefore, *we conclude that EasyBid can handle imprecise valuations (given reasonable α, β, ϵ) without significantly compromising the utility of the WSP.* The result of Figure 5.4 (b) is consistent with above discussion. Figure 5.4 (c) shows that the performance of EasyBid is relatively sensitive to the value of ϵ, but its affect could be mitigated by relaxing α and β.

Imprecise Valuations: Limited Subchannels: This simulation evaluates how limiting subchannels affects the truthfulness of the system. It uses 50 femtocells and $5 usrs/min$ arrival rate. Assuming $\alpha = 0.8, \beta = 0.1, \epsilon = 0.04$, we assign different number of subchannels to the femtocells. The $\{25, ..125\}$ subchannels in Figure 5.5

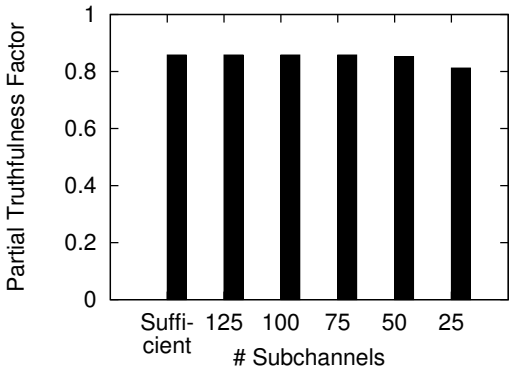


Figure 5.5: Limited Subchannels

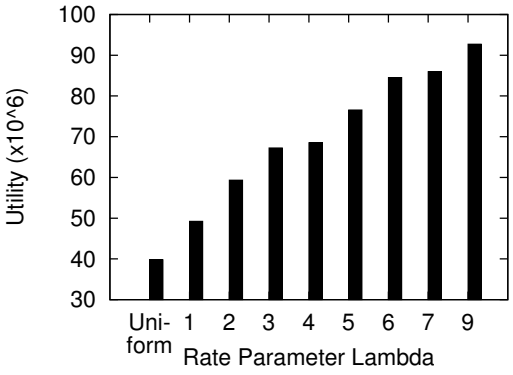


Figure 5.6: Non-uniform distribution

correspond to $5Mhz - 25Mhz$ in LTE network. For each setup of subchannels, same simulations are repeated such that, for each run, one out of 50 sellers lies to one of the N segment while others being truthful, i.e., $50 * N$ simulations are repeated for one channel profile. The percent of sellers that received their maximum utility when being truthful is calculated and shown in Figure 5.5.

Note that the PT constraint $\alpha = 0.8$ is a worst case bound. That is why the percentage of sellers being PT in the figure is actually larger than 0.8 when assuming sufficient subchannels. If the number of subchannels is limited, the percent of truthfulness is not affected until a very small number ($50, 25$ subchannels), for which the overall truthfulness is still above 0.8. The result shows that the α constraint is not violated even with a limited number of subchannels.

Imprecise Valuations: Non-uniform Distribution: This simulation uses the exponential distribution function within the interval $[0, 1]$ to generate the true valuations of 40 sellers, such that the distribution of true valuations are biased (to 0) instead of being uniform. Figure 5.6 shows that the utility of the WSP in general is

much higher in this biased distribution, as there are more low-valuation sellers that can accept lower payments.

5.6 Conclusion and Discussion

This chapter proposes EasyBid, a multiple reserve price based auction mechanism that considers imprecise valuations. Heuristic algorithms that aim to maximize the utility of the WSP under given constraints were presented. For that, we assume the knowledge of $F(x)$ is a prior. However, if the WSP has no such prior knowledge, possible solutions include: 1) If the WSP knows the distribution of perceived valuations, $F'(x)$, the same set of algorithms can also be applied on this function. 2). If $F'(x)$ is also unknown, the WSP could approximate $F'(x)$ based on the statistics of the bids it collected. The issue is that since WSP has claimed $\alpha - PT$ before the auction, the statistic result might not accurately represent $F'(x)$, i.e., the factor α has to be considered in its accuracy.

We outline the following directions that we are currently working on to make EasyBid more applicable in a variety of scenarios. 1) We are investigating the amortized arrival method to relax the sufficient-resource assumption. address the problem that the approval ratio cannot be fulfilled if there is no resource available at the time when a demand arrives. 2) We are looking to solve the optimal constraint parameters α and β by integrating them into the objective function. , such that the optimal parameters can be directly solved.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

This dissertation studies the problems of budgeted deployment and resource allocation in managed small cell networks and access acquisition of unmanaged small cell networks. To provide guaranteed throughput to users in large scale sparse networks, we present a new metric, called *Contact Opportunity*, as a characterization of a roadside WiFi network. We then propose an efficient deployment method that ensures a required level of contact opportunity at a minimum cost by utilizing submodular optimization techniques. This is the first work that addresses the challenges in achieving a *sparse* wireless infrastructure that provides QoS assurance to mobile users *in the face of uncertainty*. Aiming to solve the resource management problem in high-density networks, we propose solutions that achieve $max - min$ fairness of throughput across multiple collision domains. Two models are considered: the NINT model and the INT model. Algorithms with provable bounds are designed for both models. We further extend the QoS (throughput) fairness metric to QoE fairness by solving the problem of bandwidth allocation in a single collision domain. Finally, this dissertation designs an auction type incentive mechanism for the WSP to utilize the unmanaged small cells. To solve the imprecise valuation problem, it develops EasyBid, a novel mechanism with heuristic algorithms which allow conducting truthful

auctions, considering that the sellers only know their perceived valuations which may differ from their true valuations.

6.2 Future Work

6.2.1 Part-time Small Cells

Small-cell based data offloading has been regarded as a mobile network necessity. But there are many hurdles for large scale deployment, including power supply, backhaul, and site acquisition. Overcoming such deployment hurdles remains an open problem. One possible solution is to utilize solar power and cognitive radios as power supply and wireless backhaul. In this way, the deployment cost of small-cell networks can be significantly reduced.

According to [5], most small cells are running at low power (2W or less), which makes the use of renewable and harvested energy sources strong contenders for powering small cells outdoors. However, due to the limited energy harvesting rate of the solar panel, the host small cells might encounter power outage for certain durations, and network service might only be available at limited times in the day. The problem of optimally utilizing part-time available small cells is critical to study. To eliminate the cost of designated backhauls, some WSPs are using wireless backhaul (e.g.,cognitive radios, satellite) for femtocell deployments [13, 11]. However, given the uncertainties in the availability of both the power and backhaul, maximizing the utilization of this type of small cells under uncertainties is an interesting and challenging problem.

6.2.2 Sharing Small-cell Networks Across Multiple WSPs

Different WSPs usually have mutual agreements to authorize and provide cellular service to the clients of others. Certain deployment cost can be avoided if small cell infrastructures are also involved in such agreements. On the other hand, the idea of sharing radio access networks has been proposed[140], to reduce the cost of cell deployment and maximize the usage and efficiency of cellular networks. In a shared radio access network, one physical base station is shared among multiple WSPs, and the bandwidth is dynamically sliced based on the requests of WSPs. Those infrastructure-sharing solutions provide opportunities to WSPs to expand their small-cell networks at lower costs. However, cell maintenance and management, priority assignment (in terms of conflicts exist among WSPs), and pricing remain open problems in the literature.

6.2.3 New Incentive or Business Models for Unmanaged Small Cells

Incentive mechanisms are critical to enabling access to unmanaged small cells. Truthful auctions have been considered in this dissertation. However, one issue of the auction model is that it does not take the Internet service provider into account. Note that the successful operation of an unmanaged small cell actually involves three parties: the owner, who is usually the owner of the site; the Internet service provider, who provides backhaul to the cell; and the WSP, who owns the spectrum and wants to access the small cell. This problem might be modeled by a multiplayer game model [106]. We are looking to design realistic and efficient business models to solve this problem.

BIBLIOGRAPHY

[1] U.S. Census Bureau-TIGER/Line http://www.census.gov/geo/www/tiger.

[2] CSE 2111 Lecture-Data Validation, Worksheet Protection and Macros. Website. http://www.youtube.com/watch?v=y9rk4zxln0c.

[3] Earth From Space. Website. http://www.youtube.com/watch?v=38peWm76l-U.

[4] Hotspot (WiFi). http://en.wikipedia.org/wiki/Hotspot_%28Wi-Fi%29.

[5] How Small Cell Solar Power Implementation Increases 3G and LTE Performance in All Areas. www.digikey.com/us/en/techzone/energy-harvesting/resources/articles/how-solar-power-can-be-implemented-in-small-cells.html.

[6] Lil BUB's Magical Yule LOG Video. Website. http://www.youtube.com/watch?v=ZuHZSbPJhaY.

[7] Micocell. http://en.wikipedia.org/wiki/Microcell.

[8] Neighborhood Small Cells. http://www.qualcomm.com/research/projects/smallcells.

[9] Outdoor Rural Small Cells. http://www.ubiquisys.com/small-cells-outdoor-rural.

[10] Small Cell. http://en.wikipedia.org/wiki/Small_cell.

[11] The TFA Project. http://tfa.rice.edu/index.html.

[12] Traffic flow. http://en.wikipedia.org/wiki/Traffic_flow.

[13] Wireless Backhaul Solutions for Small Cells. http://www.ceragon.com/files/Ceragon%20-%20Small%20Cell%20-%20Application%20Note.pdf.

[14] Taking Wireless to the Max. *Business Week (businessweek.com/go/techmaven)*, pages 101–102, Nov. 2008.

[15] Wi-fi rides to wireless networks' rescue. `http://news.cnet.com/8301-30686_3-10451819-266.html`, 2010.

[16] 3GPP. 3G Home Node B (HNB) Study Item Technical Report (Release 8). *TR 25.820 V8.2.0(2008-9)*, 2008.

[17] A. Agustin, J. Vidal, and O. Munoz. Interference Pricing for Self-organisation in OFDMA Femtocell Networks. *in European Workshop on Broadband Femtocell Networks - Future Network and Mobile Summit (FuNeMS)*, 2011.

[18] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online Estimation of RF Interference. *Proc. of the 2008 ACM CoNEXT Conference*, 2008.

[19] S. Akhshabi, A. C. Begen, and C. Dovrolis. An Experimental Evaluation of Rate-adaptation Algorithms in Adaptive Streaming over Http. In *Proc. of the Second Annual ACM Conference on Multimedia systems*, pages 157–168. ACM, 2011.

[20] M. Arslan, K. Sundaresan, S. Krishnamurthy, and S. Rangarajan. Design and Implementation of an Integrated Beamformer and Uplink Scheduler for OFDMA Femtocells. *Proc. of ACM MOBIHOC*, 2012.

[21] M. Arslan, J. Yoon, K. Sundaresan, and S. Banerjee. Experimental Characterization of Interference in OFDMA Femtocell Networks. *Proc. of the IEEE INFOCOM*, 2012.

[22] M. Arslan, J. Yoon, K. Sundaresan, S. Krishnamurthy, and S. Banerjee. FERMI: A Femtocell Resource Management System for Interference Mitigation in OFDMA Networks. *Proc. of the ACM MOBICOM*, 2011.

[23] C. Bailey and X.-H. Peng. A Quality Driven Adaptation Scheme for DASH Streaming. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 308–313. IEEE, 2013.

[24] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan. Interactive WiFi Connectivity For Moving Vehicles. In *Proc. of ACM SIGCOMM*, Sept. 2008.

[25] N. Banerjee, M. Corner, D. Towsley, and B. Levine. Relays, Base Stations and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proc. of ACM MOBICOM*, Sept. 2008.

[26] Y. Bejerano, S. Han, and L. Li. Fairness and Load Balancing in Wireless LANs using Association Control. *Proc. of the ACM MOBICOM*, 2004.

[27] E. K. Burke, G. Kendall, and G. Whitwell. A New Placement Heuristic for the Orthogonal Stock-cutting Problem. *Operations Research*, 52(4):655–671, 2004.

[28] M. Burza, J. Kang, and P. v. d. Stok. Adaptive Streaming of MPEG-based Audio/Video Content over Wireless Networks. *Journal of Multimedia*, 2(2):17–27, 2007.

[29] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *Proc. of ACM MOBICOM*, Sept. 2006.

[30] V. Chandrasekhar and J. Andrews. Uplink Capacity and Interference Avoidance for Two-tier Femtocell Networks. *Global Telecommunications Conference, 2007*, pages 3322–3326, 2007.

[31] V. Chandrasekhar and J. Andrews. Femtocell Networks: A Survey. *IEEE Communication Magazine*, 46(9):59–67, 2008.

[32] R. Chang, Z. Tao, J. Zhang, and J. Kuo. A Graph Approach to Dynamic Fractional Frequency Reuse (FFR) in Multi-Cell OFDMA Networks. *in IEEE International Conference on Communications (ICC)*, 2009.

[33] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang. A Scheduling Framework for Adaptive Video Delivery over Cellular Networks. In *Proc. of the 19th Annual International Conference on Mobile computing & Networking*, pages 389–400. ACM, 2013.

[34] Y. Chen, J. Zhang, Q. Zhang, and J. Jia. A Reverse Auction Framework for Access Permission Transaction to Promote Hybrid Access in Femtocell Network . *IEEE INFOCOM mini-symposium*, 2012.

[35] B. Clark, C. Colbourn, and D. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86:165–177, 1990.

[36] E. H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11:17–33, 1971.

[37] H. Claussen. Performance of Macro- and Co-channel Femtocells in a Hierarchical Cell Structure. *in IEEE PIMRC*, Sep. 2007.

[38] O. Compte and P.Jehiel. The Wait-and-See Option in Ascending Price Auctions. *Jnl. of Euro. Economic Association*, 2(2-3), 2004.

[39] W. David, S. Hui, and V. Lau. Cross-Layer Design for OFDMA Wireless Systems with Heterogeneous Delay Requirements. *in IEEE Trans. Wireless Comm.*, 6(8):2872–2880, 2007.

[40] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the Impact of Video Quality on User Engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.

[41] M. Dong, G. Sun, X. Wang, and Q. Zhang. Combinatorial Auction with Time-Frequency Flexibility in Cognitive Radio Networks. *Proc. of the IEEE INFOCOM*, 2012.

[42] W. Dong, S. Rallapalli, R. Jana, L. Qiu, K. K. Ramakrishnan, L. V. Razoumov, Y. Zhang, and T. W. Cho. iDEAL: Incentivized Dynamic Cellular Offloading via Auctions. *Proc. of the IEEE INFOCOM*, 2013.

[43] Ericsson. Cellular Data Traffic Keeps Doubling Every Year. `http://arstechnica.com/business/2013/02/cellular-data-traffic-keeps-doubling-every-year`.

[44] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: A WiFi-Based Vehicular Content Delivery Network. In *Proc. of ACM MOBICOM*, Sept. 2008.

[45] W. FAQ. http://www.wimax.com/education/faq/, 2008.

[46] Femtoforum. Interference Management in UMTS Femtocells. http://www.femto forum.org/, 2008.

[47] Femtoforum. Femtocell Market Status Issue 1. http://www.femto forum.org/, 2009.

[48] R. Gass, J. Scott, and C. Diot. Measurements of In-Motion 802.11 Networking. In *Proc. of WMCSA*, Apr. 2006.

[49] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QoE Fairness using Openflow-assisted Adaptive Video Streaming. In *Proc. of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, pages 15–20. ACM, 2013.

[50] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag. Adaptive Fastest Path Computation on a Road Network: a Traffic Mining Approach. In *Proc. of VLDB*, Sept. 2007.

[51] T. Groves. Incentives in Teams. *Econometrica*, 1973.

[52] H. T. Friis. A Note on a Simple Transmission Formula. *Proc. of the Institue of Radio Engineers*, 34(5):254–256, 1946.

[53] G. Hardy. *Ramanujan: Twelve Lectures on Subjects Suggested by His Life and Work*. AMS Chelsea Publishing, New York, 3 edition, 1999.

[54] A. Hatoum, N. Aitsaadi, R. Langar, R. Boutaba, and G. Pujolle. FCRA: Femtocell Cluster-based Resource Allocation Scheme for OFDMA Networks. *IEEE ICC 2011*, 2011.

[55] L. Ho and H. Claussen. Effects of User-deployed, Co-channel Femtocells on the Call Drop Probability in a Residential Scenario. *in IEEE PIMRC*, September 2007.

[56] M. A. Hoque, M. Siekkinen, and J. K. Nurminen. Using Crowd-sourced Viewing Statistics to Save Energy in Wireless Video Streaming. In *Proceedings of the 19th Annual International Conference on Mobile computing & networking*, pages 377–388. ACM, 2013.

[57] T. Hossain. Learning by Bidding. *working paper, Deptt of Economics, Hong Kong University of Science and Technology, Hong Kong*, 2004.

[58] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau. Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience. In *Data Traffic Monitoring and Analysis*, pages 264–301. Springer, 2013.

[59] S. Hua, X. Zhuo, and S. S. Panwar. A Truthful Auction based Incentive Framework for Femtocell Access. *Proc. of WCNC*, 2013.

[60] S. Imahori, M. Yagiura, and H. Nagamochi. Practical Algorithms for Two-dimensional Packing. *Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC Computer & Information Science Series*, 13, 2007.

[61] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas. An Iterative Double Auction Mechanism for Mobile Data Offloading. *IEEE WiOpt*, 2013.

[62] J. Hartline. Lectures on Approximation and Mechanism Design. `users.eecs.northwestern.edu/~hartline/amd.pdf`.

[63] J. Jang and K. Lee. Transmit Power Adaptation for Multiuser OFDM Systems. *in IEEE J. Selected Areas in Comm*, 21(2):171–178, 2003.

[64] J. Jia, Q. Zhang, Q. Zhang, and M. Liu. Revenue Generation for Truthful Spectrum Auction in Dynamic Spectrum Access. *Proc. of ACM MOBIHOC*, 2009.

[65] J. Jiang, V. Sekar, and H. Zhang. Improving Fairness, Efficiency, and Stability in Http-based Adaptive Video Streaming with Festive. In *Proc. of the 8th International Conference on Emerging Networking Experiments and Technologies*, pages 97–108. ACM, 2012.

[66] M. Kaneko, P. Popovski, and J. Dahl. Proportional Fairness in Multi-Carrier System with Multi-Slot Frames: Upper Bound and User Multiplexing Algorithms. *in IEEE Transactions on Wireless Communications*, 7(1):22–26, 2005.

[67] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications*, volume 14. Springer, 1997.

[68] A. Krause, B. McMahan, C. Guestrin, and A. Gupta. Robust Submodular Observation Selection. *Journal of Machine Learning Research (JMLR)*, 9:2761–2801, 2008.

[69] M. Kubale. *Graph Colorings*. American Mathematical Society, 2004.

[70] G. Li and R. Simha. The Partition Coloring Problem and its Application to Wavelength Routing and Assignment. *Proc. of the First Workshop on Optical Networks*, 2000.

[71] L. Li, M. Pal, and Y. Yang. Proportional Fairness in Multi-rate Wireless LANs. *Proc. of the IEEE INFOCOM*, 2008.

[72] G. Liebl, T. Schierl, T. Wiegand, and T. Stockhammer. Advanced Wireless Multiuser Video Streaming using the Scalable Video Coding Extensions of H.264/MPEG4-AVC. 2006.

[73] J. Liu, Q. Chen, and H. D. Sherali. Algorithm Design for Femtocell Base Station Placement in Commercial Building Environments. *Proc. of the IEEE INFOCOM*, 2012.

[74] J. Liu and C. Lin. Achieving Efficiency Channel Utilization and Weighted Fairness in IEEE 802.11 WLANs with a p-persistent Enhanced DCF. *MSN*, 2007.

[75] Y. Liu, M. Tao, B. Li, and H. Shen. Optimization Framework and Graph-Based Approach for Relay-Assisted Bidirectional OFDMA Cellular Networks. *in IEEE Transactions on Wireless Communications*, 9(11):3490–3500, 2010.

[76] D. Lopez-perez, A. Ladanyi, A. Juttner, and J. Zhang. OFDMA Femtocells: A Self-Organizing Approach for Frequency Assignment. *in IEEE PIMRC*, 2009.

[77] D. Lopez-perez, A. Valcarce, G. D. L. Roche, and J. Zhang. OFDMA Femtocells: A Roadmap on Interference Avoidance. *IEEE Communications Magazine*, 47(9):41–48, September 2009.

[78] lp_solve. lp_solve 5.2.2.0. http://lpsolve.sourceforge.net/5.5/.

[79] H. Luo, S. Lu, and V. Bharghavan. A New Model for Packet Scheduling in Multihop Wireless networks. In *Proc. of ACM MOBICOM*, Aug. 2000.

[80] J. MacKie-Mason and H. Varian. Generalized Vickrey Auctions. *working paper*, 1994.

[81] R. Mahindra, R. Kokku, H. Zhang, and S. Rangarajan. MESA: Farsighted Flow Management for Video Delivery in Broadband Wireless Networks. In *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pages 1–10. IEEE, 2011.

[82] Matt Grob. Qualcomm on Ensuring Optimal User Experience. `http://www.commnexus.org/programs/event_20121029.php`.

[83] Michel Minoux. Accelerated Greedy Algorithms for Maximizing Submodular Set Functions. *Optimization Techniques, LNCS*, 7:234–243, 1978.

[84] R. K. Mok, E. W. Chan, and R. K. Chang. Measuring the Quality of Experience of HTTP Video Streaming. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 485–492. IEEE, 2011.

[85] N. Vaidya and P. Bahl and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *Proc. of ACM MOBICOM*, Aug. 2000.

[86] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-giel, and S. R. Das. MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access. In *Proc. of MOBISYS*, June 2007.

[87] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1988.

[88] T.-D. Nguyen and Y. Han. A Proportional Fairness Algorithm with QoS Provision in Downlink OFDMA Systems. *in IEEE Comm. Letters*, 10(6):760–762, 2006.

[89] D. Niculescu. Interference Map for 802.11 Networks. In *IMC*, 2007.

[90] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for "Automobile" Users. In *Proc. of INFOCOM*, Mar. 2004.

[91] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *IMC*, 2005.

[92] S. Paris, F. MARTIGNON, I. Filippini, and L. Chen. A Bandwidth Trading Marketplace for Mobile Data Offloading. *Proc. of the IEEE INFOCOM (Mini-conference)*, 2013.

[93] W. Pu, Z. Zou, and C. W. Chen. Video Adaptation Proxy for Wireless Dynamic Adaptive Streaming over Http. In *Packet Video Workshop (PV), 2012 19th International*, pages 65–70. IEEE, 2012.

[94] R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *Proc. of IPCO*, 2004.

[95] R. Uehara. NP-complete Problems on a 3-connected Cubic Planar Graph and their Applications. Technical Report TWCU-M-0004, Tokyo Woman's Christian University, 1996.

[96] E. Rasmusen. Strategic Implications of Uncertainty over One's Own Private Value in Auctions. *Advances in Theoretical Economics*, 6(1), 2006.

[97] R.B. Myerson. Optimal Auction Design. *MATH. OPER. RES*, 6(1):58–73, 1981.

[98] T. H. Reporter. Video Accounts for 53 Percent of Internet Traffic. Website. `http://www.hollywoodreporter.com/news/video-accounts-53-percent-internet-655203`.

[99] J. Robinson, R. Swaminathan, and E. W. Knightly. Assessment of Urban-Scale Wireless Networks with a Small Number of Measurements. In *Proc. of IEEE MOBICOM*, 2008.

[100] M. E. Sahin, I. Guvenc, M. R. Jeong, and H. Arslan. Handling CCI and ICI in OFDMA Femtocell Networks Through Frequency Scheduling. *IEEE Trans. Consumer Electronics*, 55(4):1936–1944, 2009.

[101] S.-H. Shen and A. Akella. An Information-aware QoE-centric Mobile Video Cache. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 401–412. ACM, 2013.

[102] Z. Shen, J. Andrews, and B. Evans. Adaptive Resource Allocation in Multiuser OFDM Systems with Proportional Rate Constraints. *in IEEE Transactions on Wireless Communications*, 4(6):2726–2737, 2005.

[103] D. B. Shmoys and C. Swamy. An Approximation Scheme for Stochastic Linear Programming and its Application to Stochastic Integer Programs. *Journal of the ACM*, 53:973–1012, 2006.

[104] Shor, Mikhael. Dictionary of Game Theory Terms. `http://www.gametheory.net/dictionary/DominantStrategy.html`.

[105] I. Society. Bandwidth Management: Internet Society Technology Roundtable Series (2012). Website. `http://www.internetsociety.org/sites/default/files/BWroundtable_report-1.0.pdf`.

[106] N. R. Sturtevant. *Multi-player games: Algorithms and approaches*. PhD thesis, Citeseer, 2003.

[107] K. Sundaresan and S. Rangarajan. Efficient Resource Management in OFDMA Femto Cells. *in Proc. of ACM MOBIHOC*, May 2009.

[108] K. Sundaresan and S. Rangarajan. Adaptive Resource Scheduling in Wireless OFDMA Relay Network. *Proc. of the IEEE INFOCOM*, 2012.

[109] C. Swamy and D. B. Shmoys. Algorithms Column: Approximation Algorithms for 2-Stage Stochastic Optimization Problems. *ACM SIGACT News*, 37(1):33–46, 2006.

[110] C. Swamy and D. B. Shmoys. Sampling-based Approximation Algorithms for Multi-stage Stochastic Optimization. *SIAM Journal on Computing*, 41(4):975–1004, 2012.

[111] T. G. Robertazzi and S. C. Schwartz. An accelerated sequential algorithm for producing D-optimal designs. *SIAM Journal of Scientific and Statistical Computing*, 10(2):341–358, 1989.

[112] S.-H. Teng. Mutually repellant sampling. *Minmax and its Applications, Ding-Zu Du and Panos M. Pardalos ed., Kluwer Academic Publishers*, pages 129–140, 1995.

[113] R. Thaler. Anomalies: The Winner's Curse. *Journal of Economic Perspectives*, 2(1):191–202, 1988.

[114] The NS-3 Network Simulator. `http://www.nsnam.org` (June 3, 2013).

[115] L. Trevisan. Inapproximability of Combinatorial Optimization Problems. *Technical Report TR04-065, Electronic Colloquium on Computational Complexity*, 2004.

[116] G. TS36.101. E-UTRA User Equipment Radio Transmission and Reception. `http://www.3gpp.org/ftp/Specs/html-info/36-series.htm`.

[117] G. TS36.104. E-UTRA Base Station Radio Transmission and Reception. `http://www.3gpp.org/ftp/Specs/html-info/36-series.htm`.

[118] Ubiquisys and CTTC. LTE-EPC Network Simulator. `http://iptechwiki.cttc.es/LTE-EPC_Network_Simulator_(LENA)`.

[119] A. Valcarce, D. Lopez-perez, G. D. L. Roche, and J. Zhang. Limited Access to OFDMA Femtocells. *in IEEE PIMRC*, 2009.

[120] Verizon. Video accounts for 50Website. `http://gigaom.com/2013/04/10/verizon-video-accounts-for-50-of-mobile-network-traffic-and-its-only-growing`.

[121] W. Vickrey. Counter-speculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[122] S. Wang, P. Xu, X. Xu, S. Tang, and X. Li. TODA: Truthful Online Double Auction for Spectrum Allocation in Wireless Networks. *DySPAN*, 2010.

[123] G. WiFi. Google's Mountain View WiFi Network. http://wifi.google.com/.

[124] L. A. Wolsey. An Analysis of the Greedy Algorithm for the Submodular Set Covering Problem. *Combinatorica*, 2(4):385–393, 1982.

[125] P. Xia, V. Chandrasekhar, and J. G. Andrews. Femtocell Access Control in the TDMA/OFDMA Uplink. *in Proc. of the IEEE Global Telecommunications Conference*, 2010.

[126] R. Xie, F. R. Yu, and H. Ji. Energy-Efficient Spectrum Sharing and Power Allocation in Cognitive Radio Femtocell Networks. *Proc. of the IEEE INFOCOM*, 2012.

[127] D. Yang, X. Fang, and G. Xue. Truthful Auction for Cooperative Communications. *Proc. of ACM MOBIHOC*, 2011.

[128] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, and Y. Dacheng. A Study on Quality of Experience for Adaptive Streaming Service. In *Communications Workshops (ICC), 2013 IEEE International Conference on*, pages 682–686. IEEE, 2013.

[129] J.-H. Yun and K. G. Shin. CTRL: A Self-organizing Femtocell Management Architecture for Co-channel Deployment. *Proc. of ACM MOBICOM*, 2010.

[130] A. Zaki and A. O. Fapojuwo. Optimal and Efficient Graph-Based Resource Allocation Algorithms for Multiservice Frame-Based OFDMA Networks. *in IEEE International Conference on Communications (ICC)*, 2009.

[131] H. Zhang, Y. Zheng, M. A. Khojastepour, and S. Rangarajan. Cross-layer Optimization for Streaming Scalable Video over Fading Wireless Networks. *Selected Areas in Communications, IEEE Journal on*, 28(3):344–353, 2010.

[132] Y. Zhang. A Multi-Server Scheduling Framework for Resource Allocation in Wireless Multi-Carrier Networks. *in IEEE Trans. Wireless Comm.*, 6(11):3884–3891, 2007.

[133] Y. Zhang and K. Letaief. Cross-Layer Adaptive Resource Management for Wireless Packet Networks with OFDM Signaling. *in IEEE Trans. Wireless Comm.*, 5(11):3244–3254, 2006.

[134] Z. Zheng, Z. Lu, P. Sinha, and S. Kumar. Ensuring Predictable Contact Opportunity for Scalable Vehicular Internet Access On the Go. Technical Report, available online at `http://arxiv.org/abs/1401.0781`, 2014.

[135] Z. Zheng, P. Sinha, and S. Kumar. Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access. Technical Report, Department of Computer Science and Engineering, Ohio State University, Aug. 2008.

[136] Z. Zheng, P. Sinha, and S. Kumar. Sparse WiFi Deployment for Vehicular Internet Access with Bounded Interconnection Gap. *IEEE/ACM Transactions on Networking*, 20(3):956–969, 2012.

[137] X. Zhou, S. Gandi, S. Suri, and H. Zheng. eBay in the Sky: Strategy-Proof Wireless Spectrum Auctions. *Proc. of the ACM MOBICOM*, 2008.

[138] X. Zhou and H. Zheng. TRUST: A General Framework for Truthful Double Spectrum Auctions. *Proc. of the IEEE INFOCOM*, April 2009.

[139] Y. Zhu, B. Li, and Z. Li. Truthful Spectrum Auction Design for Secondary Networks. *Proc. of the IEEE INFOCOM*, 2012.

[140] Z. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, H. Franke, and S. Sarangi. Virtual base station pool: towards a wireless network cloud for radio access networks. In *Proceedings of the 8th ACM International Conference on Computing Frontiers*, page 34. ACM, 2011.