FERMI: A FEmtocell Resource Management System for Interference Mitigation in OFDMA Networks

Mustafa Y. Arslan*‡ Jongwon Yoon†‡ Karthikeyan Sundaresan‡
Srikanth V. Krishnamurthy* Suman Banerjee†

*UC Riverside {marslan, krish}@cs.ucr.edu †U Wisconsin, Madison {yoonj, suman}@cs.wisc.edu ‡NEC Labs America Inc. {karthiks}@nec-labs.com

ABSTRACT

The demand for increased spectral efficiencies is driving the next generation broadband access networks towards deploying smaller cells (femtocells) and sophisticated air interface technologies (Orthogonal Frequency Division Multiple Access or OFDMA). The dense deployment of femtocells however, makes interference and hence resource management both critical and extremely challenging. In this paper, we design and implement one of the first resource management systems, FERMI, for OFDMA-based femtocell networks. As part of its design, FERMI (i) provides resource isolation in the frequency domain (as opposed to time) to leverage *power pooling* across cells to improve capacity; (ii) uses measurement-driven triggers to intelligently distinguish clients that require just link adaptation from those that require resource isolation; (iii) incorporates mechanisms that enable the joint scheduling of both types of clients in the same frame; and (iv) employs efficient, scalable algorithms to determine a fair resource allocation across the entire network with high utilization. We implement FERMI on a prototype four-cell WiMAX femtocell testbed and show that it yields significant gains over conventional approaches.

1. INTRODUCTION

The demand for higher data rates and increased spectral efficiencies is driving the next generation broadband access networks towards deploying smaller cell structures (called femtocells) with OFDMA [1]. They are installed in enterprises and homes, and operate using the same spectrum and technology as macrocells, while connecting to the core network through cable or DSL backhaul. In addition to the increased user throughput from short ranges, the smaller size of femtocells increases the system capacity by enabling spatial reuse. This allows broadband access service providers to (i) improve coverage and service quality, (ii) effectively balance load by offloading traffic from macrocell to femtocells, and (iii) reduce operational expenses and subscriber churn.

To retain the aforementioned benefits, femtocells have to inter-operate with and use the same access technology as macrocells. Hence, resource management solutions for femtocells cannot be designed from scratch. Although there are methods proposed to alleviate the macro-femto interference [2], interference mitigation between femtocells has not drawn much attention and thus forms the focus of this work.

There are several key aspects that make the resource management problem both challenging and unique in OFDMA femtocells. We articulate these aspects below.

Femtocells versus Macrocells: Femtocell deployments are significantly more dense compared to the planned deployments of macrocells. Hence, while interference is localized at cell edges in macrocells, it is less predictable and more pervasive across femtocells. This renders Fractional Frequency Reuse (FFR) solutions (proposed for macrocells) inadequate in mitigating interference in femtocells.

Femtocells versus WiFi: In femtocell networks, OFDMA uses a synchronous transmission access policy across cells, on a licensed spectrum. In contrast, with WiFi unlicensed spectrum is accessed asynchronously. This affects resource management (interference mitigation) in the two systems in a fundamental way. In a typical WiFi system, interfering cells are either tuned to operate on orthogonal channels or use carrier sensing to arbitrate medium access on the same channel. In an OFDMA femtocell system, there is no carrier sensing. Interfering cells can either operate on orthogonal parts of the spectrum, or directly project interference on the clients of each other. In OFDMA femtocells, transmissions to different clients are multiplexed in each frame. Since every client may not need spectral isolation, simply operating adjacent cells on separate parts of the spectrum comes at the cost of underutilization of the available capacity. In other words, resource isolation in OFDMA femtocells needs to be administrated with care. In a WiFi system, since an access point transmits data to a single client at a time (using the entire channel assigned to it), this challenge does not arise.

Our contributions in brief: We design and implement one of the first resource management systems, FERMI, for OFDMA-based femtocell networks. FERMI decouples resource management across the network from scheduling within each femtocell and addresses the former. This allows resource allocation across femtocells to be determined by a central controller (CC) at coarse time scales. Frame scheduling within each femtocell can then be executed independently on the allocated set of resources. The four key cornerstones of FERMI's resource management solution include:

- *Frequency Domain Isolation:* It isolates resources for clients in each femtocell, in frequency (as opposed to time). This allows for *power pooling* to jointly mitigate interference and increase system capacity (discussed later).
- *Client Categorization:* It employs proactive, measurement-driven triggers to intelligently distinguish clients in each femtocell that require just link adaptation from those that require resource isolation with an accuracy of over 90%.
- *Zoning:* It incorporates a frame structure that supports the graceful coexistence of clients that can reuse the spec-

trum and the clients that require resource isolation.
- *Resource Allocation and Assignment:* It employs novel algorithms to assign orthogonal sub-channels to interfering femtocells in a near-optimal fashion.

We have implemented a prototype of FERMI on an experimental four-cell WiMAX femtocell testbed. FERMI provides a complete resource management solution while being standards compatible; this enables its adoption on not only experimental platforms but also on commercial femtocell systems. To the best of our knowledge, we report the first resource management solution implemented on an actual OFDMA femtocell testbed. Comprehensive evaluations show FERMI's resource management to yield significant gains in system throughput over conventional approaches.

Organization: We describe background and related work in §2. Experiments that motivate FERMI's design are in §3. The building blocks of FERMI are described in §4. In §5, we describe the resource allocation algorithms that are part of FERMI. We evaluate FERMI in §6. We conclude in §7.

2. BACKGROUND AND RELATED WORK

In this section, we describe relevant related work. We then provide a brief background on WiMAX femtocell systems.

Macrocellular Systems:While broadband standards employing OFDMA (WiMAX, LTE) are relatively recent, related research has existed for quite some time [3]. There is research that addresses problems pertaining to single cell [4] and multi-cell [5] OFDMA systems. Several efforts have looked at the interference between macrocells and femtocells [2, 6] leveraging the *localized* interference coupled with *planned* cell layouts. However, the interference *among* femtocells remains in question since femtocells lack the features of localized interference and planned deployments of macrocells. There have been some recent works [7] that address interference among femtocells via distributed mechanisms but are restricted to theoretical studies. In contrast, we implement a centralized resource management system to mitigate interference among femtocells.

Spectrum Allocation: There has been research addressing resource allocation using graph coloring for WiFi [8, 9, 10]. The main objective in these studies is to allocate a minimum number of orthogonal contiguous channels to each interfering AP. Instead, our objective is to realize a weighted max-min fair allocation while utilizing as many sub-channels as possible. In addition, resource allocation is just one component of our work; we implement a novel resource management system with several enhancements tailored to OFDMA. There are also approaches that allocate spectrum chunks to contending entities [11, 12]. However, these studies rely on asynchronous random access and associated sensing capabilities. We address a more challenging problem in OFDMA synchronous access systems and satisfy requirements that are specific to OFDMA femtocells.

WiMAX Preliminaries: While our study applies to multi-cell OFDMA femto networks in general, our measurements are conducted on a WiMAX (802.16e [13]) femtocell testbed.

In WiMAX, OFDMA divides the spectrum into multiple tones (sub-carriers) and several sub-carriers are grouped to form a sub-channel. A WiMAX frame is a two-dimensional template that carries data to multiple mobile stations (MSs) across both time (symbols) and frequency (sub-channels). The combination of a symbol and a sub-channel constitutes a tile (the basic unit of resource allocation at the MAC). Data to users are allocated as rectangular bursts of tiles in a frame.

In OFDMA femtocells, frame transmissions are synchronized both between the BS[1] and MSs as well as across BSs (by virtue of synchronizing to the macro BS [14]). An example of a WiMAX TDD (time division duplexing) frame is shown in Fig. 1(a); the transmissions from the BS to a MS (downlink) and those from the MS to the BS (uplink) are separated in time. The frame consists of the preamble, control and data payload. While the preamble is used by the MS to lock on to the BS, the control consists of FCH (frame control header) and MAP. MAP conveys the location of the data burst for a MS in a frame and consists of both the downlink and uplink MAPs. A BS schedules the use of resources both on the downlink and the uplink. The DL-MAP indicates where each burst is placed in the frame, which MS it is intended for, and what modulation (MCS) decodes it. Similarly the UL-MAP indicates where the MS should place its data on the uplink frame. The uplink frame has dedicated sub-channels for HARQ which is used by the MSs to explicitly acknowledge (ACK/NACK) the reception of each burst.

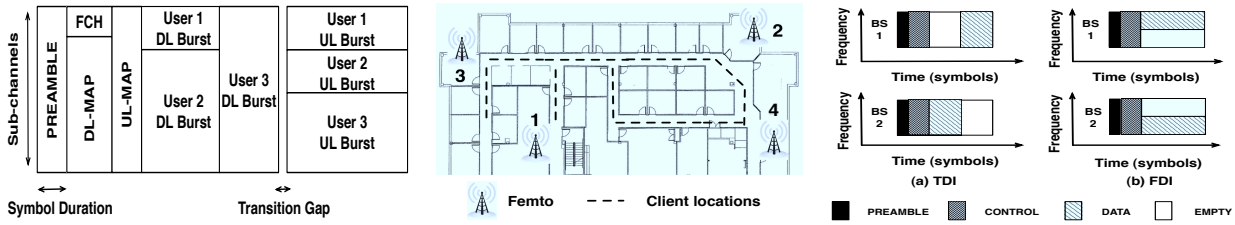
3. DESIGN ASPECTS OF FERMI

To derive the right design choices for interference mitigation, we perform extensive measurements on our femtocell testbed.

Experimental Setup: Our testbed consists of four femtocells (cells 1-4) deployed in an indoor enterprise environment (Fig. 1(b)). We use PicoChip's [15] femtocells that run 802.16e (WiMAX). Our clients are black boxes using commercial WiMAX cards from Accton [16]. The cells operate on a 8.75 MHz bandwidth with the same carrier frequency of 2.59 GHz. For this frequency, an experimental license has been obtained to transmit WiMAX signals on the air.

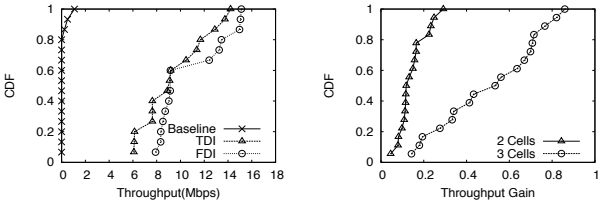
We consider downlink UDP traffic from the BSs to the clients generated by *iperf*. The traffic rate is set large enough to saturate the available resources. Each data point corresponds to an interference topology and is obtained by running an experiment for 7 minutes, measuring the throughput and averaging it over several such runs. We generate different interference topologies by varying the locations of the clients (along the path shown in Fig. 1(b)). Moving the clients provides a finer control on the inter-BS interference magnitude as opposed to changing the locations of the BSs. More importantly, note here that we only need to account for whether or not a client of a BS is interfered by another BS. This is unlike in WiFi, where in dense deployments, a WiFi AP can preclude the transmissions of a nearby AP due to carrier sensing. In other words, in an OFDMA setting, the locations of the clients (rather than the BSs) are important. Thus, we believe that our setup captures a reasonable set of

[1]We use the terms femtocell, BS, cell interchangeably.



(a) WiMAX Frame Structure (b) Testbed Deployment (c) Illustration of TDI and FDI

Figure 1: Illustrations for frame structure, deployment and resource isolation alternatives.



(a) Resource Isolation (b) TDI vs FDI

Figure 2: Benefits of FDI over TDI.

scenarios that could arise in practical deployments.

The baseline strategy for our measurements is one where a BS operates on its entire spectrum, while performing an ideal link adaptation (MCS selection) for its clients. For each of our data points, we run the experiment over all MCS levels and record the one that delivers the highest throughput.

Coping with Interference: There are two approaches to coping with interference in OFDMA. Switching to a lower MCS level via link adaptation (rate control) could suffice if the received signal quality is above the threshold required by the lower MCS level. With strong interference (typical in dense deployments), the received SINR could be even lower than that required for the lowest MCS operation. Isolating the resources utilized by interfering cells helps alleviate the effects, but it results in a reduced set of transmission resources in each cell. Clearly, the choice between link adaptation and resource isolation must be made depending on the nature of interference. In a two-dimensional WiMAX frame, resources can be isolated among BSs either in time (symbols) or in frequency (sub-channels) as depicted in Fig. 1(c). Time domain isolation (TDI) isolates resources in time by leaving empty (guard) symbols to prevent collisions; frequency domain isolation (FDI) allocates orthogonal sets of sub-channels to different BSs for their transmissions.

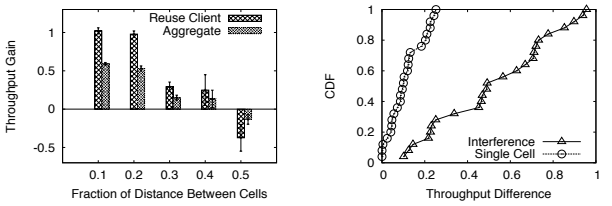
Our goal is to answer: *Does link adaptation suffice in coping with interference or is resource isolation needed? If needed, should resource isolation be in time or in frequency?* Towards this, we experiment with three strategies: (a) the baseline strategy where, BSs operate using all resources, (b) TDI and (c) FDI where, BSs operate using half of the (orthogonal) available set of symbols and sub-channels, respectively, in each frame (Fig. 1(c)). All strategies employ link adaptation via cycling through MCS levels. We first consider cells 1 and 2, and present the CDF (over the client locations) of the aggregate throughput in Fig. 2(a). We see that resource isolation provides significant gains over the baseline and that FDI outperforms TDI in aggregate throughput by about 20%. We repeat these experiments with cells 1, 2 and 3. Each cell operates on a third of the resources with TDI or FDI. In Fig. 2(b), we see that the median percentage throughput gain of FDI over TDI increases from about 17% for two cells to about 60% for three cells.

This interesting observation is due to what we refer to as *power pooling*, only possible with FDI. The energy transmitted by a BS is split over its constituent sub-channels in OFDMA. With a smaller subset of sub-channels, the average power per sub-channel increases, potentially allowing the cell to operate using a higher MCS. As more cells are activated, the number of (orthogonal) sub-channels available per cell decreases; this however, increases the average power and hence the throughput *per* sub-channel. Eventually, the higher per sub-channel throughput in each cell contributes to the higher network throughput capacity. We notice that the MCS supported by client 1 is indeed higher with FDI than TDI. The average MCS difference between FDI and TDI is 1 level in the 2 cell topology and almost 2 levels with 3 cells.

Accommodating Heterogeneous Clients: As discussed earlier, for clients in close proximity to their BS link adaptation may be sufficient to cope with interference. Invoking resource isolation for such clients will underutilize resources. Given that OFDMA multiplexes multiple client transmissions in a frame (to saturate resources), it becomes necessary to accommodate clients with heterogeneous requirements (link adaptation vs. resource isolation) in the same frame.

Towards this, we propose to use *zoning*, where an OFDMA two-dimensional frame is divided into two data transmission zones. The first zone operates on all sub-channels and is used to schedule clients that need just link adaptation (hereafter referred to as *reuse zone*). The second zone utilizes only a subset of sub-channels (determined by FDI) and here, clients that require resource isolation are scheduled (referred to as *resource isolation zone*). Link adaptation is also performed for clients in this zone.

We perform an experiment with two cells to understand the benefits of zoning. Cell 2 causes interference while cell 1 transmits data to its clients. Cell 1 schedules two clients: one by reusing all sub-channels (reuse client) and the other one by isolating resources (from cell 2). The reuse client is moved from the proximity of cell 1 towards cell 2; the other client is static. We compare the throughput that cell 1 achieves against a scheme where there is no reuse (both clients are scheduled by isolating resources). As one might expect, as long as the reuse client does not experience appreciable interference from cell 2, reusing sub-channels provides



(a) Zoning Gain (b) Calibration Measurements

Figure 3: Motivation for zoning (a) and calibrating measurements for categorization (b).

a throughput gain over the pure resource isolation scheme. We plot these throughput gains in Fig. 3(a) as a function of the reuse client's distance from cell 1. Interestingly, significant gains (at least 20%) from reusing sub-channels can be availed even when the client is at 40% of the distance between the interfering cells. Beyond this distance, the interference from cell 2 degrades throughput. We revisit zoning when we describe the algorithms in FERMI in Sec. 5.

Although zoning holds promise, it only dictates how to accommodate heterogeneous clients; it does not provide a complete resource management solution. Several challenges remain in achieving this goal. Specifically, for each BS, we need to **(a)** determine the size (in symbols) of the reuse zone **(b)** determine the subset of sub-channels allocated to the resource isolation zone, and **(c)** adapt both these zones to the dynamics of the network in a scalable manner. FERMI incorporates novel algorithms to address these challenges.

4. BUILDING BLOCKS OF FERMI

We depict the relationship between the blocks of FERMI in Fig. 4. In a nutshell, the *categorization* of clients allows the BS to determine how the frame should be divided into zones, from its perspective (block 1). The BS then determines the set of BSs that cause interference on those of its clients that require resource isolation (block 2). This information is then fed to the central controller (CC) along with cell-specific load parameters. The CC then constructs an interference map and computes the network wide sub-channel allocation and zoning parameters (details in Sec. 5). It disseminates this information back to the BSs, which use these operational parameters until the next resource allocation update. Note that the CC is similar to the notion of a self-organizing network (SON) server [14] maintained by the service provider. Next, we explain the client categorization and the interference map generation components.

Client Categorization at Femto BSs: The first building block categorizes clients into two classes; the first needs only link adaptation (class 1) while the second needs resource isolation (class 2). To understand how clients are to be categorized as either class 1 or class 2, we perform calibration experiments. We consider two cells each with a single client. We experiment over a large set of client locations to generate a plurality of scenarios. We first consider a cell in isolation (i.e., no interference). At each client location, we experiment by sequentially allocating two spectral *parts* (of equal size) of the frame to the client. Since, the fading effects on

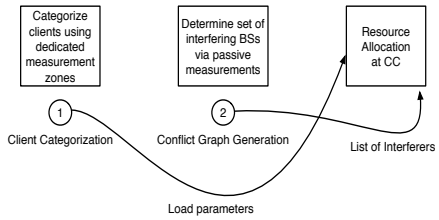


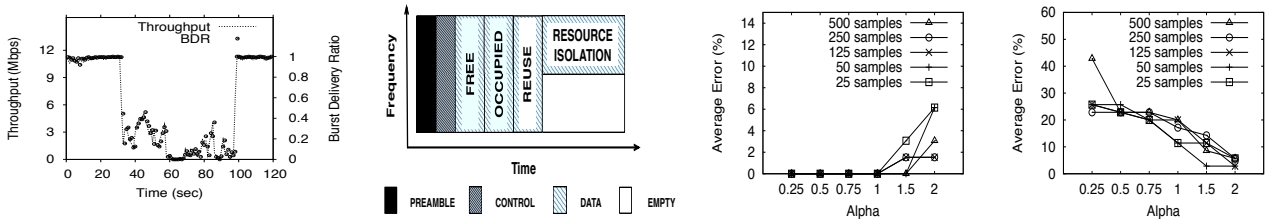
Figure 4: The building blocks of FERMI.

the two sets of assigned sub-channels are likely to be different, the client will receive different throughputs with the two different allocations. *We notice however, that the difference between the two allocations is at most 25 % in more than 90 % of the considered client locations* (Fig. 3(b)). We now repeat the experiment, but with interference. In one of the allocations (i.e. parts), the second cell projects interference on the client; in the other the operations are without interference (via resource isolation). *We observe that in this case, there is a throughput difference of over 25 % (in many cases, significantly higher) in more than 90 % of the topologies*.

These results suggest that the throughput (per unit resource) difference at a client between an interference-free allocation and an allocation with interference can be used to categorize it as class 1 or class 2. If this difference is less than a threshold (referred to as α later), link adaptation suffices for this client. However, if it is larger than the threshold, one cannot immediately determine if the client needs resource isolation. This is because the above experiments were done by allocating equal resources to the client in the settings with and without interference. If such a client is categorized as class 2 and allocated a smaller set of isolated resources (based on cell's load), the throughput it achieves may in fact only be similar to what it would achieve by being a class 1 client. Unfortunately, it is difficult to know the cell loads a priori and hence one cannot make a clear determination of whether to categorize these clients as class 1 or class 2. Thus, FERMI takes a conservative approach and categorizes all of such clients as class 2. We find that this helps accommodate fluctuations in the load and interference patterns.

Although a BS does not have access to the throughput at a client, it is informed about the reception of each burst via ACKs on the uplink. We define Burst Delivery Ratio (BDR) to be the ratio of successfully delivered bursts to the total number of transmitted bursts. The BS can *estimate* BDR by taking the ratio of the number of ACKs received to the total number of feedbacks received. Since the feedback itself might practically get lost on the uplink, this is an estimate of the actual BDR. We perform experiments to understand if the BDR estimate at the BS can provide an understanding of the throughput at the client. Fig. 5(a) shows that indeed the BS can very accurately *track* the client throughput using the BDR estimates.

To achieve categorization in practice, FERMI introduces two measurement zones in the frame as indicated in Fig. 5(b), namely the *occupied* and *free* zones. Every BS operates on all sub-channels in the *occupied* zone. Scheduling a client in this zone enables the BS to calculate the BDR in the presence of interference from other cells. Scheduling



(a) BDR Estimate vs. Throughput (b) Measurement Zones (c) Class 2 Ground Truth (d) Class 1 Ground Truth

Figure 5: Client Categorization Components (a-b) and Accuracy Results (c-d).

a client in the *free* zone to calculate the BDR without interference is slightly tricky. Given a set of interfering BSs, all BSs but for one must leave the *free* zone empty in any frame. Allowing only one of the interfering BSs to schedule its clients in the *free* zone, will enable it to measure the non-interference BDR at its clients. Hence, a random access mechanism with probability $\frac{\gamma}{n}$ is used to decide access to the *free zone*, where n is the number of interfering BSs and $\gamma \geq 1$ is a constant parameter set by the CC. Note that clients associate with BSs at different instants and hence it is unlikely that all interfering BSs will categorize their clients at the same time. Hence, γ is used to increase the access probability to the *free* zone. FERMI schedules regular data bursts in the measurement zones to calculate the BDR (a good throughput estimator), thereby keeping the process transparent to clients and retaining standards compatibility. While the *occupied* zone can be used as an extension to the reuse zone when categorization of the clients is completed, this is not possible for the *free* zone, whose utility is towards categorization in other cells. Here, the central controller that keeps track of client (dis)associations, triggers the use of the *free* zone (cast as a data zone) solely for the purpose of categorization in relevant parts of the network and disables it to minimize overhead once the procedure is complete.

The accuracy of client categorization is evaluated in Figs. 5(c) and 5(d). We again consider two cells; clients 1 and 2 belong to the two cells, respectively. We generate multiple topologies by varying the location of client 1 in the presence of interfering cell 2. First, the throughput of client 1 is measured for both zones (*free* and *occupied*) to identify the ground truth at each location; here leveraging our previous measurements, we conclude that if the throughput difference is less than 25%, client 1 is at a location where it only needs link adaptation. Otherwise, the particular scenario is deemed as one that needs resource isolation. After the ground truth is established, cell 1 collects BDR samples from both measurement zones to decide on the client category. The decision is made based on these samples: if the average *free* zone BDR is *at least* α% *higher* than the average *occupied* zone BDR, then the client category is class 2. Here, arbitrating the access to the *free* zone is a factor that reduces the accuracy of estimation. If two BSs schedule their clients in this zone at the same time, rather than getting a BDR sample without interference, they both could get a sample that indicates interference. Averaging the BDR over multiple samples is used to alleviate this.

The categorization accuracy when the ground truth is (a) resource isolation and (b) link adaptation is plotted in Figs. 5(c) and 5(d), respectively. In corroboration with our measurement based inference, it can be seen that increasing α beyond 0.25 decreases the accuracy of detecting resource isolation but conversely it increases the accuracy of detecting link adaptation. Further, while increasing the number of samples over which α is measured can help improve accuracy, the benefits are not significant. Hence, it pays to use fewer samples to categorize clients (towards reducing overhead). Thus, FERMI uses an α of 1 with 25 frame samples to obtain an accuracy greater than 90%.

Interference Map Collection at CC: The CC in FERMI generates an interference (conflict) map between BSs that not only incorporates point-to-point but also cumulative interference at clients. Note that interference is client dependent and since multiple clients are scheduled in tandem in each OFDMA frame, the interference patterns between BSs vary from one frame to another. This makes it impossible for any practical resource management scheme to gather schedule-dependent interference information, determine an allocation and disseminate it to the BSs for execution in every frame. Hence, the goal of the resource management scheme in FERMI is to allocate resources at a coarse time scale granularity (over hundreds of frames) by collecting *aggregate* interference statistics from each BS. This decouples resource allocation from frame scheduling in each BS, thereby allowing a conflict graph approach to adequately capture interference dependencies for our purpose.

In addition to client categorization, the measurement zones in FERMI also help in deciphering interference relations. If a BS causes interference to the clients of another BS so as to require resource isolation, then an edge is added between the two BSs in the conflict graph. Note that the interference relations need to be determined only for class 2 clients.

Measurements in the *occupied* zone are used as the basis to categorize a client as class 2. Note however, that *all* BSs operate in this zone and thus, the client experiences the cumulative interference from all interfering BSs. Adding an edge to each of these neighboring cells in the conflict graph would be overly conservative; some of them may only project weak levels of interference on the client. Hence, we need to determine the minimum set of interference edges that need to be added in the conflict graph to eliminate interference through resource isolation. Towards this, we use the following procedure following the initial categorization.

Consider a femtocell A and a class 2 client cl of A. cl *passively* measures the received power from neighboring BSs

(available during handover between BSs). If the power from a neighboring BS (B) exceeds a threshold, then B is added to cl's list of strong interferers. cl reports this list to A, which then consolidates it and reports the set of conflict edges (for each strong interferer) that must be added to the conflict graph to CC. CC uses this information for making the initial resource allocation decision. While this accounts for point-to-point interference, some clients may not see any individual strong interferer but the cumulative power from a subset of neighbors could be strong enough to require resource isolation. Such clients will continue to see interference after the initial resource allocation. These clients can be identified by comparing the BDR achieved on the assigned sub-channels with that seen in the *free* zone. We adopt an iterative approach to further refine the conflict graph to isolate such clients. To illustrate, let us consider one such client. We consider all the interfering cells for this client and add an edge in the conflict graph to the cell that causes the highest (in power) interference subject to a filtering based on the initial allocation. If the BDR for the client is sufficiently improved and is now within $\alpha\%$ of what is observed in the *free* zone, the process is complete. If not, the next strongest interfering BS is added to the conflict graph (again subject to filtering based on the previous allocation) and so on. We elaborate on this process in an anonymous tech. report [17].

Why Dedicated Measurements?: One could argue that using only the passive received power measurements from interfering BSs is an easier approach to categorize clients. Here, if a client receives a signal from an interfering BS that is higher than a threshold, it is categorized as class 2; otherwise, it is a class 1 client. However, for this method to work well in practice, a lot of calibration is needed to find accurate, often scenario dependent, threshold values. In addition, the received power does not necessarily give an indication of the throughput observed at the clients. To avoid these practical issues, FERMI relies on highly accurate direct measurements for client categorization, which allows it to have coarse thresholds for identification of strong interferers. Having categorized the clients and identified the interference dependencies between femtocells, we are now ready to present the resource allocation algorithms at the CC.

5. ALGORITHMS IN FERMI

The goal of resource management at CC is to determine for each femtocell (i) the size of the *reuse* zone and, (ii) the specific subset of sub-channels for operations in the *resource isolation* zone to obtain an efficient and fair allocation across femtocells. While the joint determination of parameters for both the zones is the optimal approach, this depends on throughput information that changes in each frame, thereby coupling resource allocation with per-frame scheduling decisions. Since, as discussed in Sec. 4, per-frame resource allocation is infeasible due to practical constraints, FERMI performs resource allocation at coarse time scales.

Each femtocell reports two parameters to the CC to facilitate resource allocation: (i) load (number of clients) in its *resource isolation* zone, and (ii) desired size (in time symbols) of its *reuse* zone. Alternative definitions for load can be adopted but number of clients is sufficient for our purposes (as in [9]). Note that a femtocell does not have the complete picture of interference dependencies across cells; it only has a localized view. Thus, it simply provides the load in its resource isolation zone and expects the CC to allocate resources proportional to its load. Each femtocell determines the desired size of its reuse zone based on the relative load in the two zones. Since class 2 clients will be scheduled immediately after the reuse zone (see Fig. 5(b)), if two interfering cells have different sizes for their reuse zones, then the cell with the larger reuse zone will cause interference to the class 2 clients of the other cell. Hence, an appropriate size for the reuse zone of each cell also needs to be determined by the CC based on the reported desired values. Next, we present the algorithm at the CC to determine the sub-channel allocation and assignment to each femtocell, followed by the selection of their reuse zone sizes.

5.1 Allocation and Assignment

The goal of the sub-channel allocation component in FERMI is to allocate and assign sub-channels to the resource isolation zone in each femtocell so as to maximize utilization of sub-channels in the network subject to a weighted max-min fairness model. The reasons for the choice of the weighted max-min fairness are two fold: (i) weights account for variations in load across different cells; and (ii) max-min allows for an almost even split of sub-channels betweens cells in a contention region, which in turn maximizes the benefits from power pooling (see Sec. 3). Thus, given the load for the resource isolation zone from each femtocell along with the conflict graph constructed, the CC's goal is to determine a weighted (load based) max-min allocation of sub-channels to femtocells (i.e. vertices in the graph).

THEOREM 1. *The sub-channel allocation and assignment problem in FERMI is NP-hard.*

We omit the proof due to space limitations. The interested reader can consult [17].

While the allocation problem in FERMI may seem similar to multi-coloring at the outset, this is not the case. In fact, multi-coloring can only provide an assignment of sub-channels for a specified allocation. However, in FERMI we are also interested in determining a weighted max-min allocation in addition to the assignment, which makes the problem much more challenging. Further, every contiguous set of sub-channels allocated to a cell is accompanied by an information element in the control part of the frame (MAP), describing parameters for its decoding at the clients. This constitutes overhead, which in turn increases with the number of discontiguous sets allocated to a cell. Therefore, our goal is to *reduce* overhead due to discontiguous allocations, while ensuring an efficient allocation. We present A^3, the allocation and assignment algorithm in FERMI that achieves the aforementioned objectives.

Overview of A^3: Any resource allocation algorithm attempts to allocate shared resources between entities in a contention

region subject to a desired fairness. Each contention region corresponds to a maximal clique in the conflict graph. However, a given femtocell may belong to multiple contention regions and its fair share could vary from one region to another. This makes it hard to obtain a fair allocation, for which it is necessary to identify all maximal cliques in the conflict graph. However, there are an exponential number of maximal cliques in general conflict graphs with no polynomial-time algorithms to enumerate them. Hence, we propose an alternate, novel approach to resource allocation in A^3, which is both polynomial-time as well as provides near-optimal fair allocations with minimal discontiguity (overhead). The three main steps in A^3 are as follows.

Algorithm 1 Allocation and Assignment Algorithm: A^3

1: **Triangulate:** A^3 first transforms the given conflict graph G into a chordal graph G' by adding a minimal set of virtual interference edges to $G = (V, E)$.
2: **Allocate and Assign:** A^3 computes a provably weighted max-min allocation on the chordal graph G'.
3: **Restore:** A^3 removes the virtual edges from G' and updates the allocation to the vertices carrying the virtual edges to account for under-utilization on the original graph G.

1: **INPUT:** $G' = (V, E')$ and load $\ell_i, \forall v_i \in V$
2: **Allocation:**
3: Un-allocated vertices $\mathcal{U} = V$, Allocated vertices $\mathcal{A} = \emptyset$
4: Determine all the maximal cliques $\mathcal{C} = \{C_1, \ldots, C_m\}$ in G' using perfect elimination ordering
5: Resource: $R_j = N$, Net load: $L_j = \sum_{i:v_i \in C_j} \ell_i, \forall C_j$
6: Determine tuples: $s_i = \max_{j:v_i \in C_j}\{L_j\}$, $t_i = \sum_j 1_{v_i \in C_j}, \forall v_i$
7: Determine initial allocation: $A_i = \min_{j:v_i \in C_j} \left\lfloor \frac{\ell_i R_j}{\sum_{k:v_k \in C_j} \ell_k} + 0.5 \right\rfloor, \forall v_i \in \mathcal{U}$
8: **while** $\mathcal{U} \neq \emptyset$ **do**
9: Pick un-allocated vertex with maximum lexicographic rank: $v_o = \arg\max_{i:v_i \in \mathcal{U}}(s_i, t_i)$
10: Allocate A_o sub-channels to v_o; $\mathcal{U} \leftarrow \mathcal{U} \backslash v_o$, $\mathcal{A} \leftarrow \mathcal{A} \cup v_o$
11: Update remaining resource: $R_j = R_j - A_o$, $\forall j : v_o \in C_j$
12: Remove v_o from cliques: $C_j \leftarrow C_j \backslash \{v_o\}, \forall j : v_o \in C_j$; Update $L_j \ \forall j$ and $(s_i, t_i) \ \forall v_i \in \mathcal{U}$
13: Update allocation: $A_i = \min_{j:v_i \in C_j} \left\lfloor \frac{\ell_i R_j}{\sum_{k:v_k \in C_j} \ell_k} + 0.5 \right\rfloor, \forall v_i$
14: **end while**

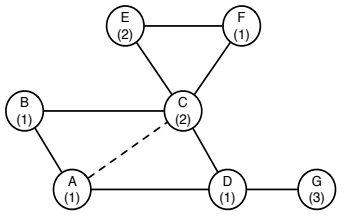
A *chordal* graph does not contain cycles of size four or more. Very efficient algorithms for important problems like maximum clique enumeration can be applied on chordal graphs [18]. The key idea in A^3 is to leverage the power of chordal graphs in obtaining a near-optimal allocation. In the interest of space, we do not present details of some aspects of the algorithm, whose solutions exist in literature; the reader can peruse the cited references. We now present details of the three steps in A^3 along with a running example in Fig. 6.

Triangulation: The process of adding edges to triangulate (chordalize) a graph is known as *fill-in*. Since adding edges to the conflict graph would result in a conservative allocation than is required, the goal is to add the *minimum* number of edges needed for triangulation. While this is a NP-hard problem in itself, A^3 employs a maximum cardinality search based algorithm [19] that is guaranteed to produce a *minimal* triangulation and runs in time $O(|V||E|))$. Fig. 6 depicts a fill-in edge between vertices A and C. As we shall subsequently see, the restoration (third) step in A^3 is used to alleviate the under-utilization introduced by triangulation.

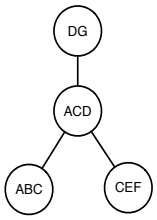
Allocation: A^3 uses the following algorithm to determine the weighted max-min allocation on the triangulated graph G'. Once the graph is triangulated, all its maximal cliques are listed in linear time ($O(|V|)$) by determining a perfect elimination ordering (PEO) [19]. A^3 determines the net load on each maximal clique (step 5) and for every un-allocated vertex (cell, v_i), it determines a tuple (s_i, t_i), where s_i indicates the highest load in the cliques that v_i belongs to and t_i is the number of cliques that it belongs to (step 6). A^3 then determines its weighted fair share in each of the maximal cliques it belongs to and determines its minimum (rounded) share amongst all its member cliques (step 7). It picks the vertex (v_o) with the highest lexicographic rank and allocates the computed share of sub-channels to it (vertex C is picked first with $s_c = 5$ and $t_c = 3$). v_o is then removed from the list of un-allocated vertices (steps 8-10). The allocated vertex is removed from its member cliques, and the clique load, resource and vertex tuples are correspondingly updated (steps 11,12). The weighted share for the remaining set of un-allocated vertices in each of the maximal cliques that v_o belongs to is updated based on the remaining resources in those cliques (step 13). The process is repeated untill all vertices receive allocation and runs in time $O(|V|^2)$.

Assignment: After the vertices get their weighted max-min allocation, the next step is to provide an actual assignment of sub-channels to satisfy the allocations. A^3 leverages clique trees for this purpose. A clique tree for a chordal graph G is a tree whose nodes are maximal cliques in G. Further, it satisfies some useful properties (as we show later).

A^3 generates a clique tree for the chordal graph G' (depicted in Fig. 6) in linear time by building on top of a PEO or by constructing a maximum spanning tree [18]. It picks an arbitrary node in the clique tree as its root and starts sub-channel assignment proceeding from the root to its leaves. At every level in the tree, it assigns sub-channels to un-assigned vertices in each of the nodes (maximal cliques) based on their allocation (vertex D is assigned first with sub-channels [1:5]). When assigning sub-channels to a vertex, a contiguous set of sub-channels that is disjoint with existing assignments to other vertices in the same clique is achieved. When contiguous assignment is not possible, assignment is made to minimize fragmentation (vertex B is assigned two fragments). Since a vertex may belong to multiple cliques,



Triangulation



Clique Tree

Vertex	Initial Allocation	Assignment	Restoration	Final Allocation	Benchmark
C	min(8, 10, 10) = 8	[12 : 19]	none	8	8
D	min(6, 5) = 5	[1 : 5]	N/A	5	5
G	15	[6 : 20]	N/A	15	15
E	8	[1 : 8]	N/A	8	8
A	min(7,6) = 6	[6 : 11]	[12:19]	14	10
F	4	[9 : 11] + [20]	N/A	4	4
B	6	[1 : 5] + [20]	N/A	6	10

[a : b] denotes the set of sub-channels from a to b (inclusive)

Figure 6: Illustration of A^3 algorithm for 20 sub-channels in the spectrum. The vertex loads are included in parentheses.

once its assignment is made, it is retained in all subsequent levels of the tree. We establish later that the above procedure that runs in $O(|V|)$ can yield a feasible assignment of sub-channels to satisfy the allocation.

Restoration: Fill-in edges could result in conservative (under-utilized) allocation of resources. While the triangulation in A^3 attempts to reduce the addition of such edges, we still need a final step to restore potential under-utilization. A^3 revisits vertices carrying fill-in edges and removes such edges one by one. When a fill-in edge is removed, the removal of a conflict may free up some sub-channels at each of the vertices carrying the edge. If so, the largest set of such sub-channels (that do not conflict with the assignment of neighbor vertices) are directly assigned to those vertices (for vertex A, sub-channels [12:19] are freed after the conflict removal with C and can be re-assigned to A). This can be done in $O(|V|)$.

To summarize, given the exponential number of cliques in the original graph, A^3 intelligently transforms the graph into a chordal graph with only a linear number of cliques and optimally solves the allocation and assignment problem. A^3 keeps the potential under-utilization due to virtual edges to a minimum with its triangulation and restoration components. Thus, it provides near-optimal performance for most of the topologies with a net running time of $O(|V||E|)$. We now establish two key properties of A^3.

PROPERTY 1. *A^3 produces a weighted max-min allocation on the modified graph G'.*

PROPERTY 2. *A^3 always produces a feasible assignment of sub-channels for its allocation.*

Proofs omitted due to space limitations. We encourage the reader to consult [17]. Based on these two properties, we have the following result.

THEOREM 2. *If G is chordal, then A^3 produces an optimal weighted max-min allocation.*

Through our comprehensive evaluations in Section 6, we show that over 70% of the topologies are chordal to begin with for which A^3 yields an optimal allocation. For the remaining topologies, A^3 sub-optimality is within 10%, indicating its near-optimal allocation.

Other possible comparative approaches: While greedy heuristics for multi-coloring do not address our allocation problem, to understand the merits of A^3, we propose and consider two extensions to such heuristics that also perform allocation and assignment (coloring).

The first heuristic is *progressive* (labeled *prog*); allocations and assignments are made in tandem one channel at a time. The vertex with the smallest weighted allocation ($\frac{\text{allocation}}{\text{load}} = \frac{A_i}{\ell_i}$) is chosen and assigned the smallest index channel that is available in its neighborhood. By assigning channels one at a time, this heuristic is able to achieve good fairness. However, its running time is $O(|V|^2N)$, where its dependence on N (number of sub-channels) makes it pseudo-polynomial, thereby affecting its scalability. Further, it results in a highly fragmented assignment of channels to vertices, which in turn increases the control overhead in frames.

Another heuristic that avoids the pseudo-polynomial complexity, is *interference degree* based (labeled *deg*). The share to every vertex is determined based on its weight and the remaining resources (after removing allocated vertices) in its interference neighborhood and is ($\frac{\ell_i(N-\sum_{j:(v_i,v_j)\in E, v_j\in\mathcal{A}})}{\sum_{j:(v_i,v_j)\in E, v_j\in\mathcal{U}}\ell_j}$). Then the vertex with the min. share is allocated as contiguous of a set of sub-channels as possible. This heuristic runs in $O(|V|^2)$ and also keeps the overhead low. However, its fairness is significantly worse.

By adopting a greedy approach, heuristics derived from multi-coloring either achieve low complexity and overhead at the cost of fairness but not both. A^3 however, deciphers interference dependencies with good accuracy to provide both near-optimal fairness and reduced complexity and overhead. Further, since the allocation and assignment is effected on the chordal graph G', dynamics in the form of arrival/departure of clients/cells (addition/deletion of edge conflicts) can be easily accommodated in a purely localized manner through incremental schemes [20]. This in turn allows A^3 to scale well to network dynamics unlike other heuristics.

Benchmarking: To understand how close A^3 is to the optimum, we need to obtain the weighted max-min allocation on the original graph G. This requires listing of all the maximal cliques, which are exponential in number. This is achieved in a brute-force manner (exponential complexity). Once all the maximal cliques are obtained on G, the allocation procedure of A^3 can be directly applied to obtain a weighted max-min allocation on G.

5.2 Zoning

We addressed the assignment of sub-channels to the resource isolation zone of each cell. Our next step is to determine the size of the reuse zone (in symbols) for each cell based on their desired sizes. There arise three challenges in determining the reuse zone size (s_r). (i) If two inter-

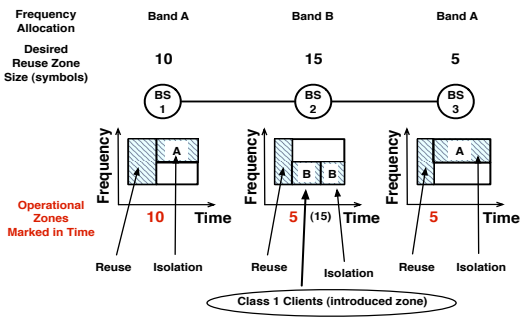


Figure 7: Illustration of zoning mechanism of FERMI.

fering cells use two different s_r's, the one with the larger s_r will cause interference to the class 2 clients of the other cell. Hence, a common reuse zone is required among interfering cells. (ii) Since allocation and zoning are meant to operate at coarse time scales (decoupled from per-frame scheduling), the common s_r among interfering cells cannot be determined based on throughput. Hence, the choice of the common s_r is restricted to either the minimum or maximum of the desired zone sizes of the neighboring cells. (iii) If each cell belongs to a single contention region (clique), choosing the common s_r is easy. However, since cells may belong to multiple cliques, this will result in a common s_r (minimum or maximum) propagate to the entire network. Cells with a desired zone size less than the common s_r may not have sufficient data for their class 1 clients to fill up to the s_r, while cells with a larger desired zone size will have to perform isolation (without reusing sub-channels). Either case results in under-utilization, which is exacerbated when a single common s_r is used in the network.

FERMI addresses the above challenge as follows (illustration in Fig. 7). For each cell, the CC determines the minimum of the advertised (desired) s_r's of all the cell's neighbors and uses that as its operational s_r (e.g. 10 symbols for BS1, 5 symbols for BS2). The cell schedules its class 1 clients in the reuse zone till the operational s_r (using all sub-channels). It continues to schedule class 1 clients in the second zone between its operational s_r and its desired s_r. However, these are scheduled only in the band allocated to the cell by A^3 (the scheduling of BS2 between the 5^{th} and the 15^{th} symbols). The class 2 clients are scheduled in the resource isolation zone (after the desired s_r) using the sub-channels (band) allocated by A^3.

Introducing a zone (marked on Fig. 7) that schedules class 1 clients between the operational and desired s_r's (using the band given by A^3), provides a graceful transition between the reuse and resource isolation zones. Since the chance for under-utilization is more when the operational s_r exceeds the desired s_r, the minimum of the desired s_r's in the neighborhood is used as the operational s_r for a cell. Further, since each cell computes its operational s_r only based on the desired s_r's of its neighbors and not their operational s_r's, propagation of a single common s_r in the network (and the resulting under-utilization) is avoided. As an example, this would correspond to every BS having the same s_r (i.e. global min.) of 5 symbols in Fig. 7. Using the minimum of the desired s_rs of neighbors (i.e. local min.) avoids this propagation for BS1 and allows it to have a s_r of 10 symbols. Hence, different regions of the network can have different s_r values, which increases the potential for sub-channel reuse. Further, cells that belong to multiple contention regions with different operational s_r's in the different cliques (e.g. BS2 in Fig. 7) will not see interference to their class 2 clients, since the operational s_r of all their cliques will be less than their desired s_r, while they schedule only class 1 clients in the region between their operational and desired s_r. As we shall show in our evaluations, zoning provides significant throughput gains as long as the s_r values in different cliques can be decoupled (i.e. a single globally minimum desired s_r does not propagate).

6. SYSTEM EVALUATION

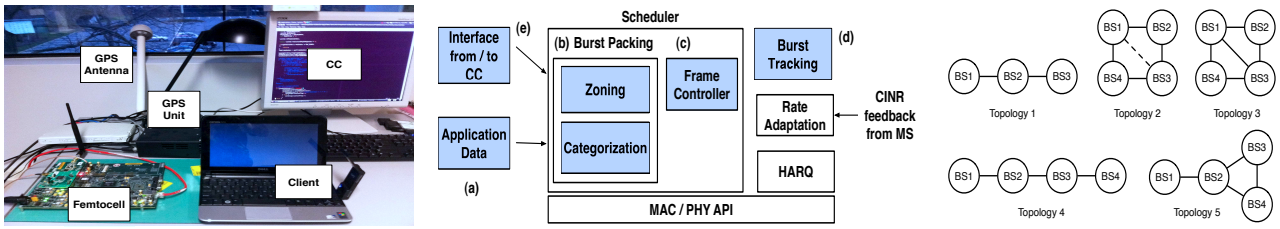
To evaluate FERMI, we both conduct experiments on our testbed as well as simulations. Simulations help evaluate the scalability and the relative performance of each algorithm with parameters that are not easy to adapt in practice.

6.1 Prototype Evaluations

Implementation Details: Fig. 8(a) shows a picture of our testbed equipment. Given that we do not have a macro BS at our disposal, we use external GPS modules to achieve synchronization among femtocells. The GPS modules are placed next to windows with cables providing a 1 pulse per second (pps) signal to each femtocell (antenna and cable depicted). The clients are USB dongles connected to laptops.

FERMI is implemented on the PicoChip platform which provides a *base reference design* implementation of the WiMAX standard. The reference design does not involve sophisticated scheduling routines and provides just a *working link* between the BS and the MS. Since the clients are off-the-shelf WiMAX MSs (with no possibility of modification), it is a challenge to realize a working implementation of various components such as categorization and zoning. Some of these challenges were to keep our implementation within the *boundaries* of the rigid WiMAX frame structure and to integrate commercial clients with our experimental testbed. We significantly extend (shown as colored components in Fig. 8(b)) the reference design to implement FERMI. Specifically, our implementation operates as follows.

(a) When data from higher layers is passed onto the MAC, we first route the data based on what MS it is intended for and whether that MS is already categorized (as in Sec. 4) or not. **(b)** If the MS is already categorized, its data is packed in the relevant zone of the frame that the MS needs (reuse vs. resource isolation). If not, its data is packed in the measurement (recall free and occupied) zones introduced for categorization. The burst packing component implements a rectangular alignment of the data of both MSs that have been categorized before as well as MSs that are being categorized. **(c)** After packing, the data is passed onto the frame controller which prepares the control payload before the frame is transmitted on the air. **(d)** Burst tracking component keeps an in-



(a) WiMAX Equipment (b) Implementation of FERMI (c) Topologies

Figure 8: WiMAX equipment (a), implementation of FERMI (b) and topologies for prototype evaluation (c).

formation tuple for measurement zones for the MSs that are being categorized. It tracks the ACK status of each measurement burst (populated as list of tuples). After enough BDR samples are collected, it decides on the client category and informs the burst packing component about the decision. **(e)** Interface with the CC leverages kernel sockets to communicate the load and conflict information to the CC and receives operational parameters for zoning and allocation (used by the burst packing component).

Experimental Evaluations: We evaluate the performance of each algorithm using our testbed. We create five topologies as shown in Fig. 8(c). The dotted edge between BS1 and BS3 in Topology 2 is the $fill\text{-}in$ edge introduced by A^3 (other topologies are already chordal). In generating these topologies, we leverage our WiMAX testbed (see Fig. 1(b)) by changing the client locations for each BS. We measure the fairness of each algorithm relative to the optimal allocation (benchmark) as the normalized distance to the benchmark $d = \sqrt{\sum_{i \in V}(t_i - s_i)^2} \ / \ \sqrt{\sum_{i \in V}(s_i)^2}$ [21] where t_i and s_i denote the number of sub-channels assigned to vertex i by an algorithm and the benchmark, respectively.

Throughput and Fairness: In our WiMAX deployment, the BSs have 30 sub-channels available in the spectrum. Each BS has two clients (one class 1, one class 2). When there is no zoning employed, we schedule both clients in the same set of sub-channels allocated to the BS. For scenarios with zoning, the specific zoning strategy determines the size of the reuse zone and the resource isolation zone.We perform experiments for each topology with the allocation determined by each algorithm (assuming equal load in each BS). Here, we introduce a heuristic (labeled *dist*) that decides the share of a vertex based on its weight and the resources in the neighborhood (without removing the allocated vertices). The share of a vertex i becomes $\frac{\ell_i N}{\sum_{j:(v_i,v_j)\in E} \ell_j}$. It mimics a distributed degree-based allocation and helps us understand the importance of having a centralized approach.

Table 1 summarizes the number of sub-channels allocated to each BS along with utilization and aggregate throughput measurements from the experiments. We observe that *dist* has the lowest utilization and therefore the lowest aggregate throughput. This is because it over-accounts for interference by just considering the vertex degrees in allocation. *deg* inherently penalizes vertices with high degree and allocates more resources to the others; it slightly outperforms A^3 in utilization and throughput (albeit at the cost of fairness). Fig. 9(a) and 9(b) plot the fairness for equal load and variable load (listed next to each BS in parentheses in Table 1), respectively. It is seen that A^3 consistently outperforms the other algorithms except topology 2 (equal load case) where it requires a fill-in edge. However, the restoration step of A^3 can account for under-utilization (due to the fill-in edge) achieving the same utilization as the benchmark. In all other topologies, A^3 achieves the exact allocation as the benchmark (BM in Table 1) since they are naturally chordal.

Zoning Benefits: We present two measurements for A^3 - with and without zoning in Fig. 9(c). The baseline strategy is where all femto BSs operate on all available resources with link adaptation. We observe that even without zoning, A^3 has significant gains over the baseline. The gains are further pronounced when zoning is employed, giving A^3 a throughput increase of 50% on average.

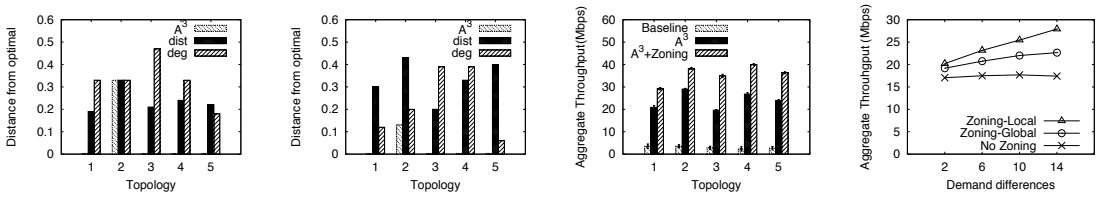
Next, we quantify the benefits of decoupling reuse zone demands in the network (local min.) against having a single reuse demand propagate to each contention region (global min). For this experiment, we use topology 1 in Figure 8(c). We set equal reuse zone demands for BS1 and BS 2 (varied in each measurement) and a fixed demand of 4 symbols for BS3. Demand difference is defined as the difference between the common demand of BS1 and BS2 and the demand of BS3 (4 symbols). As we vary the demand difference from 2 to 14, we measure the aggregate throughput and present it in Fig. 9(d). It is seen that both global min. and local min. zoning have increasing throughput as the demand difference increases. For the global min. zoning, although the operational size is the same (4), the high demand of BS1 and BS2 allow them to schedule their class 1 clients over a larger set of resources (recall the transition zone in Sec. 5). Note that since class 1 clients are likely to support a higher MCS than class 2 clients, having a large demand contributes to throughput gains (as compared to scheduling class 2 clients in the transition zone). For local min. zoning, the operational size for BS1 is significantly higher as compared to the global min. resulting in an increasing throughput gain over the global min. strategy. This shows FERMI'S benefits from decoupling desired reuse zone sizes between different contention regions in the network.

6.2 Evaluations with Simulations

System Model and Metrics: We implement a simulator to evaluate FERMI in comparison to its alternatives. The simulator incorporates a channel model proposed by the IEEE 802.16 Broadband Wireless Access Working Group for femtocell simulation methodologies [22]. This model captures

	Topology 1				Topology 2				Topology 3				Topology 4				Topology 5			
Algorithm	A^3	**dist**	**deg**	**BM**	A^3	**dist**	**deg**	**BM**	A^3	**dist**	**deg**	**BM**	A^3	**dist**	**deg**	**BM**	A^3	**dist**	**deg**	**BM**
BS1 (1)	15	15	20	15	20	10	20	15	10	7	7	10	15	15	20	15	20	15	23	20
BS2 (2)	15	10	10	15	10	10	10	15	10	10	16	10	15	10	10	15	10	7	7	10
BS3 (3)	15	15	20	15	20	10	20	15	10	7	7	10	15	10	10	15	10	10	11	10
BS4 (2)	-	-	-	-	10	10	10	15	10	10	16	10	15	15	20	15	10	10	12	10
Utilization	45	40	50	45	60	40	60	60	40	34	46	40	60	50	60	60	50	42	53	50
Throughput (Mbps)	20.87	18.36	21.80	-	29.04	19.73	27.86	-	19.61	15.87	20.76	-	26.79	22.72	27.08	-	23.95	19.94	25.06	-

Table 1: Throughput and utilization of each algorithm along with individual allocations (for equal load) for the BS.



(a) Fairness (equal load) (b) Fairness (variable load) (c) Throughput (d) Global vs Local Min.

Figure 9: Fairness and Zoning Benefits of A^3.

wireless effects such as log-distance path loss, shadow fading and penetration loss, typical of indoor deployments. The SNR from the model is mapped to a MCS using a rate table from our real WiMAX testbed to compute throughput.

The simulation area is a 7x7 grid where the distance between each grid point is 12 meters. In addition, the width and height of this area is 100 meters. We simulate a deployment in this area by randomly choosing grid locations for each femtocell. We then randomly generate a client location for each femtocell and determine the conflict graph. We measure the overhead of each algorithm as the number of contiguous sub-channel chunks allocated per femtocell. In addition to overhead, we define the *fill-in edge ratio* to be the ratio of number of fill-in edges to the edges that are already present in the conflict graph. If the conflict graph is chordal, the fill-in edge ratio is 0.

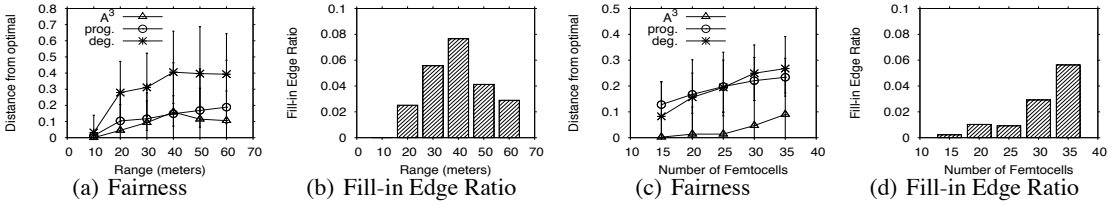
Simulation results: Next, we present our simulation results. Each measurement point is an average over results from 100 randomly generated topologies.

Effect of Femto Range: We do not present throughput and utilization results here since all algorithms (including the benchmark) do not exhibit much difference ([17]). Fig. 10(a) plots the effect of range on fairness. We observe that both heuristics (*prog* and *deg*) consistently deviate more from the benchmark as range increases. With increasing range, the number of sub-channels that a femto is assigned decreases (resources are shared among more femtos). Recalling the fairness formula, a given difference in allocations (between the benchmark and the heuristics) becomes more pronounced with less number of sub-channels assigned by the benchmark (s_i). Interestingly, A^3 exhibits an improvement in fairness after a particular range (while maintaining less than 0.15 distance from the benchmark). We find that the fill-in edge ratio is the main factor that affects A^3's fairness (plotted in Fig. 10(b)). For small ranges, the graph contains some isolated vertices (very few cycles) and A^3 does not introduce fill-in edges. As range increases, cycles start to form and A^3 adds fill-in edges to maintain chordality. However for further ranges, increased connectivity turns in favor of A^3 since the cycles happen rarely and fill-in edge ratio decreases again.

Effect of Number of Femtos: We fix a number of sub-channels (5) and a range (20 m.) and vary the number of femtocells. From Fig. 10(c) we see that A^3 consistently outperforms the other heuristics in terms of fairness and is within 0.1 distance of the benchmark due to the rare need for fill-in edges. The distance increases with the number of femtos due to increased likelihood of cycles; the trend again follows that of the fill-in edge ratio (plotted in Fig. 10(d)). For *deg* and *prog*, the distance also increases with the number of femtos because of a reduced number of sub-channels per femto (s_i).

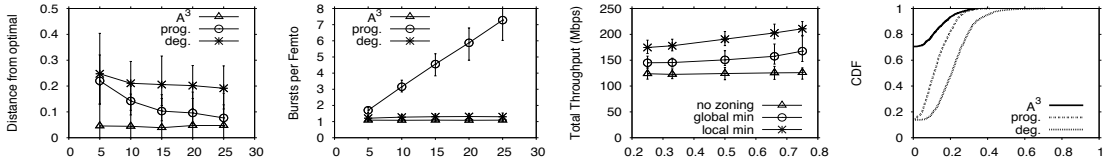
Effect of Number of Sub-channels: We simulate 30 femtocells with 10 m. range. In Fig. 11(a), it is seen that A^3 exhibits a constant distance from the optimal. Since A^3's performance is mainly influenced by fill-in edge ratio, the number of sub-channels does not have a significant effect on A^3's fairness. It is also seen that the distance for *prog* and *deg* decreases with increased number of sub-channels due to the increase in number of sub-channels per femto (s_i). This makes the differences in allocations (between the heuristic and the benchmark) less pronounced as compared to when there are fewer sub-channels. Fig. 11(b) shows the effect of number of sub-channels on overhead. The overhead for A^3 and *deg* is very close to 1 and does not change with number of sub-channels. This shows that they can assign a single contiguous set of sub-channels to the femtocells. However, *prog* tends to have an increasing overhead. Since *prog* allocates a fragmented set of sub-channels, the overhead increases with increasing number of sub-channels per femto.

Effect of Zoning: Each femtocell has two clients: one that requires resource isolation (class 2) and one that requires just link adaptation (class 1). We simulate 40 femtocells with range 10 m, 30 sub-channels and 30 symbols in the frame. We have three different types of reuse zone demands: i) high-demand femtos that randomly generate a reuse demand between 15 and 20 symbols ii) moderate demand femtos that generate a demand between 10 and 15 symbols and iii) low-demand femtos with generated demand between 5 and 10 symbols. We experiment by varying the fraction of the high-demand femtocells. Fig. 11(c) shows the total throughput achieved for each zoning strategy. It is seen that as the fraction of high-demand femtos increases, the throughput for both zoning strategies increases. However, the gain



(a) Fairness (b) Fill-in Edge Ratio (c) Fairness (d) Fill-in Edge Ratio

Figure 10: Effect of Range (a-b) and Effect of Number of Femtos (c-d).



(a) Fairness (b) Overhead (c) Local vs Global Min. (d) Fairness

Figure 11: Effect of Number of Sub-channels (a-b), Effect of Zoning (c) and Overall Fairness (d).

of local min. over global min. is more with higher fraction of high-demand femtocells. This is a natural artifact of vertices converging to a higher local demand value as opposed to the global minimum demand which is the same on average (generated by the low-demand vertices). The results reinforce FERMI's benefits of decoupling the reuse zone demands in different contention regions of the network (preventing a single demand from propagating).

Overall Fairness: Finally, we present the CDF of the distance from the optimal as a cumulative set of all previously described experiments for the three algorithms considered, in Fig. 11(d). We use the results with range 10 m. and 20 m., as these represent a more realistic deployment (given the entire area is 100x100 meters). These provide an understanding of how fair a given algorithm is in practical deployments with a large set of variables (# femtos, # sub-channels, zoning etc.). It is seen that A^3 is able to reach the exact benchmark allocation in about 70% of the topologies which is far superior to the performance of the other heuristics. *deg* has the worst performance and reaches the benchmark allocation in only 10% of the topologies. *prog* does better than *deg* but still significantly underperforms compared to A^3.

7. CONCLUSIONS

In this paper, we design and implement FERMI, one of the first resource management systems for OFDMA femtocell networks. Resource management in femtocells offers a set of unique practical challenges (posed by the requirement for standards compatibility) that to the best of our knowledge had not been addressed before. Our approach is based on a set of design decisions derived with extensive experiments on a four-cell WiMAX femtocell testbed. It is composed of two functional modules. The first module uses coarse-level measurements to classify clients into two categories, those that need resource isolation and those that do not. The second module assigns OFDMA sub-channels to the different femto-cells in a near-optimal fashion. We implement our approach on our testbed to show its superiority compared to conventional approaches. We also perform simulations to showcase its scalability and efficacy in larger scale settings.

8. REFERENCES

[1] R. Van Nee and R. Prasad, "OFDM for Wireless Multimedia Communications," *Artech House*, 2000.

[2] D. Lopez-Perez, G. Roche, A. Valcarce, A. Juttner, and J. Zhang, "Interference Avoidance and Dynamic Frequency Planning for WiMAX Femtocells Networks," in *Proc. of IEEE ICCS*, 2008.

[3] 3GPP, "Technical specification group radio access networks; 3G home NodeB study item technical report (release 8)," *TR 25.820 V1.0.0 (2007-11)*, Nov 2007.

[4] S. Kittipiyakul and T. Javidi, "Subcarrier Allocation in OFDMA Systems: Beyond water-filling," in *Proc. of Signals, Systems, and Computers*, 2004.

[5] T. Quek, Z. Lei, and S. Sun, "Adaptive Interference Coordination in Multi-cell OFDMA Systems," in *IEEE PIMRC*, 2009.

[6] J. Yun and S. Kang, "CTRL: A Self-Organizing Femtocell Management Architecture for Co-Channel Deployment," in *ACM MOBICOM*, Sept 2010.

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

[8] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," in *ACM SIGMOBILE Mobile Computing and Communications Review*, July 2005, vol. 9.

[9] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-aware spectrum distribution in wireless lans," in *IEEE ICNP*, 2008.

[10] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "Client-driven channel management for wireless lans," in *IEEE Infocom*, 2006.

[11] L. Yang, W. Hou, L. Cao, B. Zhao, and H. Zheng, "Supporting Demanding Wireless Applications with Frequency-agile Radios," in *USENIX NSDI*, 2010.

[12] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained Channel Access in Wireless LAN," in *ACM SIGCOMM*, Aug. 2010.

[13] IEEE 802.16e 2005 Part 16, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems ," *IEEE 802.16e standard*.

[14] WMF-T33-118-R016v01, "Femtocells Core Specification," .

[15] PicoChip, "http://www.picochip.com," .

[16] Accton, "http://www.accton.com," .

[17] Anonym. Tech. Doc., "http://dl.dropbox.com/u/1348085/tech.pdf," .

[18] J. R. S. Blair and B. W. Peyton, "An introduction to chordal graphs and clique trees," in *http://www.ornl.gov/info/reports/1992/3445603686740.pdf*.

[19] A. Berry, J. R. S. Blair, P. Heggernes, and B. W. Peyton, "Maximum cardinality search for computing minimal triangulations of graphs," in *Journal Algorithmica*, May 2004, vol. 39.

[20] A. Berry, P. Heggernes, and Y. Villanger, "A vertex incremental approch for dynamically maintaining chordal graphs," in *Algorithms and Computation, 14th Int. Symp. (ISAAC)*, December 2003.

[21] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," in *ATM Forum Document Number: ATM Forum / 990045*, February 1999.

[22] S. Yeh and S. Talwar, "Multi-tier simulation methodology ieee c802.16ppc-10/0039r1," 2010.