A Resource Management System for Interference Mitigation in Enterprise OFDMA Femtocells

Mustafa Y. Arslan*, Jongwon Yoon†, Karthikeyan Sundaresan‡,
Srikanth V. Krishnamurthy*, Suman Banerjee†
*University of California Riverside, e-mail: {marslan, krish}@cs.ucr.edu
†University of Wisconsin Madison, e-mail: {yoonj, suman}@cs.wisc.edu
‡NEC Laboratories America, Inc., e-mail: karthiks@nec-labs.com

***Abstract*—To meet the capacity demands from ever-increasing mobile data usage, mobile network operators are moving towards smaller cell structures. These small cells, called femtocells, use sophisticated air interface technologies such as Orthogonal Frequency Division Multiple Access (OFDMA). While femtocells are expected to provide numerous benefits such as energy efficiency and better throughput, the interference resulting from their dense deployments prevents such benefits from being harnessed in practice. Thus, there is an evident need for a resource management solution to mitigate the interference that occurs between collocated femtocells. 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 and low overhead. We implement FERMI on a prototype four-cell WiMAX femtocell testbed and show that it yields significant gains over conventional approaches.**

***Index Terms*—WiMAX, femtocells, resource management**

I. 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]. Femtocells are installed indoors (e.g., enterprises, homes) and use the same spectrum and access technology as macrocells (traditional cell towers), while connecting to the core network through cable or DSL backhaul. The poor cellular signal problem indoors, experienced by many users today, can easily be overcome by femtocells. Moreover with femtocells, mobile devices (e.g., 4G-enabled smartphones) can save energy by transmitting to a nearby femtocell (rather than a distant cell tower) and enjoy high data rates. In addition, the small range of femtocells increases the cellular network capacity via increased spatial reuse. These advantages allow mobile broadband service providers to (i) improve coverage and service quality and (ii) offload traffic from macrocells to femtocells in a cost-effective manner.

To harness the aforementioned benefits in practice, one first needs to mitigate the interference that occurs in femtocell networks. Although previous studies (e.g., [2]) have proposed solutions to alleviate the interference between macrocells and femtocells, interference mitigation *between collocated femtocells* has not drawn considerable attention and thus forms the focus of this work. However, the design of a resource management solution for femtocells is complicated by the fact that the femtocells have to inter-operate with the cellular standards that they inherit from macrocells. There are several other key aspects that make the resource management problem both challenging and unique. We articulate these below.

Femtocells versus Macrocells: Typical femtocell deployments are significantly more dense compared to the well-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 between femtocells.

Femtocells versus WiFi: OFDMA dictates a synchronous medium access for femtocells, on a licensed spectrum. On the contrary, WiFi stations access the unlicensed spectrum in an asynchronous manner (i.e., random access via CSMA). In a typical WiFi network, interfering cells either operate on orthogonal channels or use carrier sensing to arbitrate medium access on the same channel. In an OFDMA femtocell network, there is no carrier sensing and multiple central frequencies (channels) are not available in the licensed spectrum; therefore, existing solutions for WiFi are not applicable. Interfering femtocells can either operate on orthogonal parts (called subchannels) of the spectrum, or directly project interference on the clients of each other. Further, in OFDMA, transmissions to different clients of a single cell are multiplexed in each frame. Since every client of a cell may not need spectral isolation (for purposes of interference mitigation), blindly operating adjacent cells on orthogonal 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 executed by jointly considering interference mitigation and leveraging spatial reuse opportunities. Since a WiFi access point transmits data to a single client at a time (using the entire channel width assigned to it), this challenge does not arise.

Our contributions in brief: We design and implement one of the first practical 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 the frequency domain (as opposed to the time domain). 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 that require just link adaptation (i.e., clients that can reuse the spectrum) 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 spectrum 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 FERMI on a 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 system implemented on a real OFDMA femtocell testbed. Comprehensive evaluations show that FERMI yields significant gains in system throughput over conventional approaches.

II. Background and Related Work

While broadband standards employing OFDMA (WiMAX, LTE) are relatively recent, related research has existed for quite some time [3]. There are studies that address problems pertaining to single cell (e.g., [4]) and multi-cell systems [2], [5], [6], [7], [8], [9], [10], [11]. The above studies on multi-cell systems have looked at the interference between macrocells and femtocells, for both downlink (e.g., [11]) and uplink communications (e.g., [10]). In summary, the solutions in these studies leverage the localized interference and the planned cell layouts of macrocells, and they are restricted to theoretical studies with simplifying assumptions that prevent their adoption in practice. On the other hand, femtocell deployments in practice are not planned and thus, do not benefit from localized interference.

For interference between macrocells and femtocells, we assume one of the models in [11]. In this model, interference is mitigated by partitioning the frame resources across macrocells and femtocells in a semi-static manner (based on macro and femto users' traffic). Thus, our focus in this work is specifically on interference *between* femtocells. While FERMI's goal of interference mitigation is similar to that of the above studies, it is the first study that achieves this with a standards-compatible implementation. We evaluate FERMI on real OFDMA hardware and show that it is viable for commercial femtocell deployments.

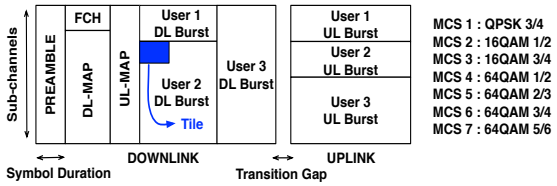


Fig. 1. Illustration of the WiMAX frame structure.

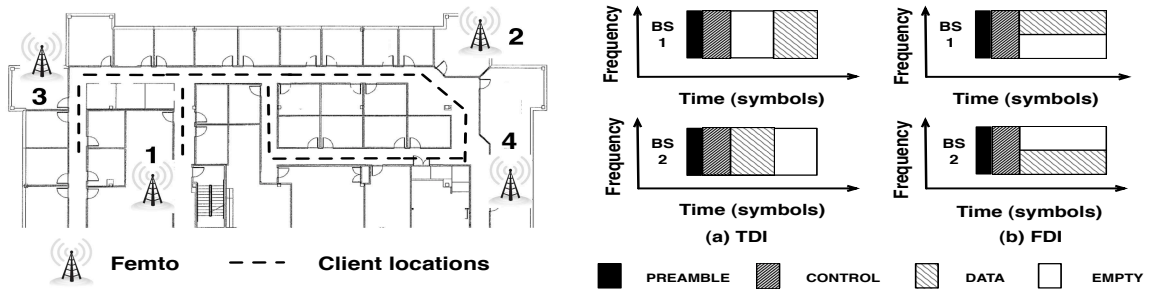
In addition to the studies in the cellular domain, there have been studies addressing resource allocation using graph coloring for WiFi systems (e.g., [12], [13], [14], [15]). The main objective in these studies is to allocate a minimum number of orthogonal contiguous channels to each interfering AP. Instead, our objective is to realize a weighted max-min fair allocation while utilizing as many sub-channels (fragments of the spectrum) as possible. In addition, resource allocation is just one component of our study; we implement a novel, complete resource management system with several enhancements specifically tailored to OFDMA. There have also been approaches that allocate spectrum fragments to contending stations (e.g. [16], [17]). 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.

A. WiMAX Preliminaries

While our study applies to OFDMA femtocells in general, our measurements are conducted on a WiMAX (802.16e [18]) femtocell testbed. WiMAX divides the spectrum into multiple sub-carriers and groups several sub-carriers to form a sub-channel. Specifically, we use the distributed grouping mode (PUSC [18]) in our implementation since it is mandatorily supported. In PUSC, the individual sub-carriers forming a sub-channel are distributed throughout the spectrum. This distribution (i.e., which sub-carriers are selected as part of a sub-channel) is subject to specific permutations. Thus, two different sub-channels at the MAC level map to two distinct sets of sub-carriers at the PHY level. In general, interference from the same source may be different on different sub-channels if frequency selectivity is taken into account. However, note that PUSC picks the subcarriers composing a sub-channel randomly from the spectrum. This averages the effect of frequency selectivity and interference on a given sub-channel, thereby giving a uniform effect across sub-channels.

The two-dimensional WiMAX frame (see Fig. 1) 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 resource at the MAC). Data to users are allocated as rectangular bursts of tiles in a frame. The BS [1] schedules the use of tiles both on the downlink and the uplink. Frames are synchronized in time both between the BS and MSs as well as across BSs (by virtue of synchronizing to the macro BS [19]). The frame consists of the preamble, control and data payload. While the preamble is used by the MSs to lock

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



(a) Testbed Deployment (b) Illustration of TDI and FDI

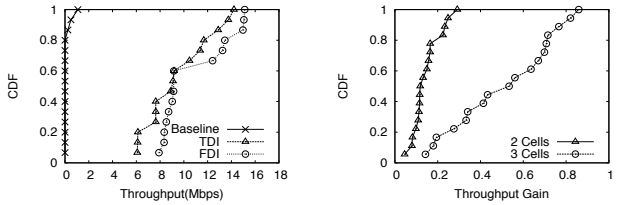
Fig. 2. The deployment of our testbed (a) and alternative resource isolation strategies (b).

on to a particular 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. The DL-MAP indicates where each burst is placed in the frame, which MS it is intended for, and what modulation level (MCS as shown in Fig. 1) decodes it. Similarly, the UL-MAP indicates where the MS should place its data on the uplink frame. The uplink frame also has dedicated sub-channels for HARQ, which is used by the MSs to explicitly acknowledge (ACK / NACK) the reception of each burst sent by the BS.

III. Design Aspects of FERMI

To derive the right design choices for interference mitigation, we conduct extensive measurements on our testbed. The testbed consists of four PicoChip [20] WiMAX femtocells (cells 1-4) deployed in an indoor enterprise environment at NEC Labs (Fig. 2(a)). The clients are commercial WiMAX USB dongles (with unmodifiable, proprietary source code [21]) plugged into laptops running Windows XP. All cells use 8.75 MHz bandwidth with the same carrier frequency of 2.59 GHz. For this frequency, we obtained an experimental license from FCC to transmit WiMAX signals on the air.

We consider downlink UDP traffic (generated by *iperf*) from the cells to the clients. The traffic rate saturates the available tiles in the frame. We call a triplet $\{cl, bs, int\}$ an interference *topology*, where cl is the location of the client whose throughput is being measured, bs is the cell that the client is associated to and int is the set of other BSs that interfere with the data reception of cl. Each measurement 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. 2(a)). Moving the clients provides a finer control on the interference magnitude received from other BSs, 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 other BSs. This is unlike in WiFi, where in dense deployments, an AP can preclude the transmissions of a nearby AP due to carrier sensing. In other words, in OFDMA where there is no carrier sensing, the locations of the clients (rather than the BSs) and whether they are subject to interference are important. Thus,



(a) Resource Isolation (b) TDI vs FDI

Fig. 3. Benefits of FDI over TDI.

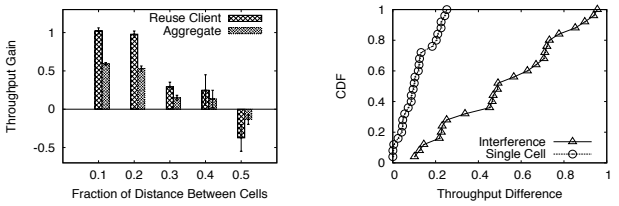
we believe that our setup captures a reasonable set of scenarios that could arise in practical femtocell deployments.

The baseline strategy for our measurements is one where a BS uses the entire spectrum, while performing an ideal link adaptation (MCS selection) for its clients. We want to decouple the effect of a particular choice of link adaptation algorithm from our measurements. With this in mind, we run the experiment for each data point over all MCS levels and record the one that delivers the highest throughput. Since we conduct the experiments in a slowly-varying indoor environment, we see that the throughputs between experimental runs do not exhibit significant variations for a given interference topology.

A. Coping with Interference

There are two approaches to coping with interference in OFDMA. Switching to a lower MCS 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. In such cases, isolating the resources (tiles) utilized by interfering cells helps alleviate the effects, but it results in a reduced set of tiles 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, the tiles can be isolated among BSs either in time (symbols) or in the frequency (sub-channels) domain as depicted in Fig. 2(b). Time domain isolation (TDI) isolates tiles 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 alone suffice in coping with interference or is resource isolation needed? If needed, should resource isolation be performed in time or in frequency?* Towards this, we experiment with three strategies: (a) the baseline strategy where BSs use all tiles, (b) TDI



(a) Zoning Gain (b) Calibration Measurements

Fig. 4. Motivation for zoning (a) and measurements for categorization (b).

and (c) FDI where BSs use half of the (orthogonal) available set of symbols and sub-channels, respectively, in each frame (Fig. 2(b)). 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. 3(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 the experiment with cells 1, 2 and 3. Each cell now uses a third of the resources with TDI or FDI. In Fig. 3(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*, which is 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 use a higher MCS. As more cells are activated in an interference domain, 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.

B. Accommodating Heterogeneous Clients

As discussed earlier, for clients in close proximity to their BS, link adaptation alone may be sufficient to cope with interference. Resource isolation for such clients will underutilize the tiles in the frame. Given that OFDMA multiplexes data to multiple clients in a given frame (to fill the available tiles), it becomes necessary to accommodate clients with heterogeneous requirements (link adaptation vs. resource isolation) in the same frame. To achieve this, we propose to use *zoning*, where an OFDMA frame is divided into two data transmission zones [2]. The first zone uses all the sub-channels and schedules clients that need just link adaptation (hereafter referred to as the *reuse zone*). The second zone utilizes only a subset of the sub-channels (determined by FDI) and here, the clients that require resource isolation are scheduled (referred to as the *resource isolation zone*). Link adaptation is also performed for clients in this zone albeit only within the restricted subset of sub-channels.

We conduct 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 data for two clients: one by reusing all sub-channels (the 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 delivers (aggregate throughput of both clients) against a scheme where there is no reuse (i.e. 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 throughput gain over the pure resource isolation scheme. We plot this throughput gain in Fig. 4(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 starts degrading the throughput. We revisit zoning in detail in §V.

[2]A zone in OFDMA is a dedicated portion of the frame in which one or more bursts can be scheduled.

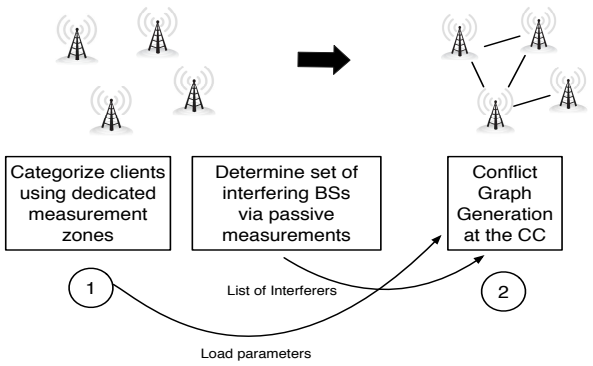


Fig. 5. The building blocks of FERMI.

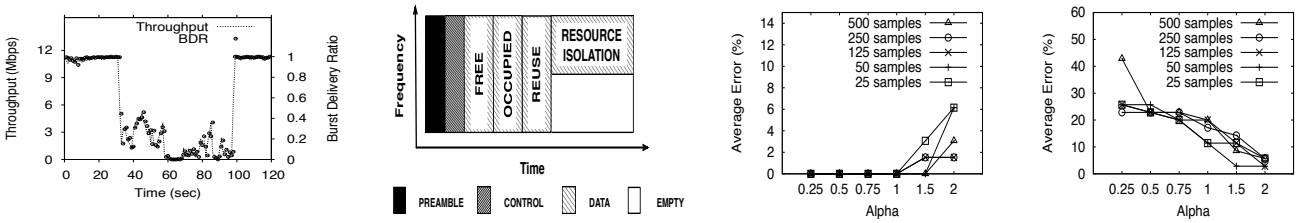
Although zoning holds promise, it only dictates how to accommodate heterogeneous clients; it does not provide a complete resource management solution. Several challenges remain in achieving this goal. Specifically, for each cell, we need to **(a)** determine the size (in symbols) of the reuse zone **(b)** 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.

IV. Building Blocks of FERMI

We depict the relationship between the blocks of FERMI in Fig. 5. In a nutshell, the *categorization* of clients allows each BS to determine how the frame should be divided into zones, from its perspective (block 1). Each BS then determines the set of BSs that cause interference on those of its clients that require resource isolation. This information, along with cell-specific load parameters, is then fed to the central controller (CC), which then constructs an interference map (block 2). Using the interference map (i.e., conflict graph), the CC computes the network wide sub-channel allocation and zoning parameters (details in §V). It disseminates this information back to the BSs, which use these operational parameters until the next resource allocation update.

A. Client Categorization at the BS

The first building block categorizes clients into two classes; the first needs just link adaptation (class *LA*) while the second



(a) BDR Estimate vs. Throughput (b) Measurement Zones (c) Class *ISO* Ground Truth (d) Class *LA* Ground Truth

Fig. 6. Client Categorization Components (a-b) and Accuracy Results (c-d).

needs resource isolation together with link adaptation (class *ISO*). To understand how clients are to be categorized as either class *LA* or class *ISO*, 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 sequentially allocate two spectral *parts* (of equal size) of the frame to the client. Since, the fading effects on the two sets of assigned sub-channels are likely to be different, the client will have different throughputs with the two allocations. *We notice however, that this difference is at most 25 % in more than 90 % of the considered client locations* (Fig. 4(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, the throughput difference is over 25 % (in many cases, significantly higher) in more than 80 % of the topologies.*

These results suggest that the throughput (per unit resource) difference between an *interference-free* allocation and an allocation with interference can be used to categorize a client as class *LA* or class *ISO*. If this difference is less than a threshold (referred to as α later), link adaptation suffices for this client. 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 number of tiles to the client in the settings with and without interference. If such a client is categorized as class *ISO* and allocated a smaller set of isolated resources, its throughput may in fact only be similar to what it would achieve by being a class *LA* 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 *LA* or class *ISO*. Thus as a design choice, FERMI takes a conservative approach and categorizes all of such clients as class *ISO*. 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 and NACKs on the uplink. We define Burst Delivery Ratio (BDR) to be the ratio of successfully delivered bursts to the total number of transmitted bursts by the BS. The BS can *estimate* BDR by taking the ratio of the number of ACKs received to the total number of feedbacks (ACKS + NACKS) received from the clients. 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. 6(a) plots a sample result showing that indeed the BS can very accurately *track* the client throughput using the BDR estimates. We find that, as per the WiMAX standard, the feedback channels on the uplink modulate data using robust QPSK modulation. This helps in reducing the probability of a feedback being received in error by the BS and makes the BDR estimate accurate. Similar notions of uplink feedback channels are also available in other OFDMA standards such as LTE.

Having shown that BDR accurately represents throughput, we next describe our categorization solution. Before providing details, we sketch how throughput in an OFDMA system is computed. A rectangular burst consists of x symbols and y sub-channels which collectively form $t = x*y$ tiles. If a single burst is successfully received by the client, it delivers $t*b$ bits of information where b is the bits per tile that the current MCS encodes. Here, the MCS is typically chosen based on the SINR of the link. Thus, the throughput over j transmitted bursts can be computed as $t*b*j*BDR$.

To achieve categorization in practice, FERMI introduces two measurement zones in the frame as depicted in Fig. 6(b), namely the *occupied* and *free* zones. Every BS operates using 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 client in the *free* zone to calculate the BDR without interference is slightly more involved. 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 BDR without interference at its clients. Hence, a random access mechanism with probability $\frac{\gamma}{n}$ is emulated 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, thereby keeping the process transparent to clients and retaining standards compatibility [3]. While the *occupied* zone can be used as an extension to the reuse zone when categorization of

[3]If the user traffic is intermittent, FERMI can create dummy bursts and use them for client categorization. Since this process requires a few samples (shown later), we believe it will not cause a significant increase in cell load.

the clients is complete, this is not possible for the *free* zone, whose utility is towards categorization in other cells. Here, the CC 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. 6(c) and 6(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 calibration 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 performs the following set of actions.

1) For K samples, transmit to a client using the *occupied* zone and with the probability specified earlier, transmit to the same client using the *free* zone. During this period keep track of the BDR for each zone.
2) Calculate *occupied* throughput $T_{occ} = t*b_{occ}*K*BDR_{occ}$ where BDR_{occ} is the BDR of the *occupied* zone and b_{occ} is the number of bits that the sampled MCS encodes. Similarly, calculate $T_{free} = t*b_{free}*K*BDR_{free}$ (note that the number of tiles, t, is equal for the two zones).
3) If $T_{free} >= (1+\alpha)*T_{occ}$, then the client is of class *ISO*; otherwise belongs to class *LA*. Based on the calibration measurements, we set $\alpha = 0.25$.

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. We use the BDR average over multiple samples to alleviate such inaccuracy.

The categorization accuracy when the ground truth is (a) resource isolation and (b) link adaptation is shown in Figs. 6(c) and 6(d), respectively. In corroboration with our measurement-based inference, we see that increasing α beyond 0.25 decreases the accuracy of detecting resource isolation but 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 $\alpha = 1$ with 25 samples to obtain an accuracy greater than 90%.

B. Interference Map Generation

The CC in FERMI generates an interference (conflict) map that not only captures point-to-point but also cumulative interference experienced by the 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 (sent every 5ms in WiMAX). Hence, the goal of the resource management scheme in FERMI is to allocate resources at a coarser time scale (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 *ISO* clients. FERMI uses the measurements in the *occupied* zone as the basis to categorize a client as class *ISO*. 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 *ISO* 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 the CC. The 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. The client reports a list of neighbor BSs (not already reported) in decreasing order of received power as potential cumulative interferers to its BS. Now, FERMI needs to identify the smallest subset of these neighbors to whom adding a conflict edge will eliminate interference through resource isolation. To achieve this two filters are applied: (i) from the cumulative interference list sent by the client, the BS first removes those neighbors from the list to whom an edge was already added by one of its other clients; (ii) the pruned list is sent to the CC, where neighbors whose current resource allocation are orthogonal to that of the client's cell are further removed as they could not have caused interference. The CC then looks at the final pruned lists and adds a conflict edge to the neighbor that appears first in multiple lists. With the updated interference graph, resource allocation is determined once

again and disseminated to femtocells for the next resource allocation epoch. 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 current resource allocation) and so on. In addressing cumulative interference, conflict edges are added only one by one in each epoch by the CC. This is because most of the interference experienced by clients is strong in dense femtocell deployments. Hence, aggressively adding conflict edges to address cumulative interference will only result in under-utilization of resources. Thus, using both passive and active measurements at clients, FERMI accounts for both strong and cumulative interference in its resource allocation decisions in each epoch [4].

Why Dedicated Measurements?: One could argue that using only the passive received power measurements from interfering BSs may be 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 *ISO*; otherwise, it is a class *LA* 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.

V. Algorithms in FERMI

The goal of resource management at the 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 §IV, per-frame resource allocation is infeasible due to practical constraints, FERMI performs resource allocation at coarser 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 the number of clients is sufficient for our purposes (as in [13]). 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 *ISO* clients will be scheduled immediately after the reuse zone (see Fig. 6(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 *ISO* 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.

A. Allocation and Assignment

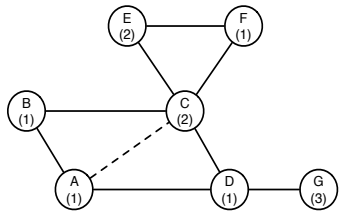
The goal of sub-channel allocation is to allocate and assign sub-channels to the resource isolation zone in each femtocell to maximize the utilization of sub-channels in the network subject to a weighted max-min fairness model. The reasons for the choice of the weighted max-min fairness are two-fold: (i) weights account for variations in load across different cells; and (ii) max-min allows for an almost even split of sub-channels between cells in a contention region, which in turn maximizes the benefits from power pooling (see §III). Thus, given the load for the resource isolation zone from each cell along with the conflict graph, 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).

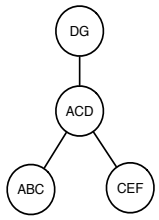
Theorem 1: The sub-channel allocation and assignment problem in FERMI is NP-hard.

Proof: Consider the simpler version of the problem, where we are interested only in the optimum objective (max-min) value and not in the specific allocation to all the cells. Hence, we are interested only in determining the largest number of sub-channels x such that all cells can at least be given an allocation of x. This in turn can be determined as follows. For a given integer i, replace each vertex (femtocell) in the conflict graph $G = (V, E)$ by a clique of size i and add edges between all vertices across the cliques if there existed an edge between the vertices corresponding to the cliques in the original graph G. Let the resulting graph be G_i. Now, determining the maximum i such that G_i is colorable with N colors (N being the # of sub-channels) yields the desired objective value. However, determining if G_i is colorable with N colors is the problem of multi-coloring G, where each vertex requires as assignment of i colors. Thus, the max-min value to our simpler problem is x if and only if G can be x-fold colored with N colors. Since multi-coloring is NP-hard and forms a special case of a simpler version of our problem, the hardness automatically carries over. ■

While the allocation problem 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 of sub-channels.

[4]We assume that the clients are static so that the interference graph does not change until the next epoch. FERMI in its current form cannot address high user mobility and we defer handling such cases to future work.





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)

Fig. 7. Illustration of A^3 algorithm for 20 sub-channels in the spectrum. The vertex loads are included in parentheses.

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 fair 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.

Overview of FERMI's resource allocation: 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 graph. However, there are an exponential number of maximal cliques in general graphs with no polynomial-time algorithms to enumerate them. Hence, we propose an alternate, novel approach to resource allocation in A^3 (outlined in Algorithm 1), which runs in polynomial-time and provides near-optimal fair allocation with minimal discontiguity (overhead).

Chordal Graphs: A *chordal* graph does not contain cycles of size four or more. Chordal graphs have significant applications in sparse matrix computations and have been extensively studied. Algorithms for important problems such as maximum clique enumeration can efficiently be applied on chordal graphs [22]. The key idea in A^3 is to leverage the power of chordal graphs in obtaining a near-optimal allocation. We now present details of the three steps in A^3 along with a running example in Fig. 7.

Triangulation: The process of adding edges to chordalize (triangulate) 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 [23] that is guaranteed to produce a *minimal* triangulation and runs in time $O(|V||E|)$), where V is the set of vertices and E is the set of edges in the graph. Fig. 7 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 the triangulation.

Algorithm 2 Weighted Max-min Fair Allocation Algorithm

1: **INPUT:** $G' = (V, E')$ and load ℓ_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**

Allocation: A^3 uses Algorithm 2 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) [23]. 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 a vertex's weighted fair share in each of the maximal cliques that 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 also removed from the cliques that it is a member of, and the clique loads, 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 until 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. 7) in linear time by building on top of a PEO or by constructing a maximum spanning tree [22]. 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, it picks a contiguous set of sub-channels that is disjoint with existing assignments to other vertices in the same clique. When contiguous assignment is not possible, A^3 makes the assignment to minimize fragmentation (e.g. vertex B is assigned two fragments). Since a vertex may belong to multiple maximal cliques, 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 (i.e. proper coloring of G') 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 re-visits vertices that carry 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'.

Proof: Before the proof, we recap the definition of max-min fairness, which needs to be slightly modified given that fractional channel allocations are not possible. An allocation vector is said to be *max-min* if it is not possible to increase the allocation (x) of an element without decreasing that of another element with an allocation of $x+1$ or lesser. The corresponding *weighted max-min* definition requires that an increase in the allocation (x) of an element with weight w_i is accompanied by a decrease in the allocation of another element j (with weight w_j) with an allocation of $\frac{w_j x}{w_i} + 1$ or lesser.

Note that a weighted max-min allocation on G', where the weights correspond to the load on the vertex, is equivalent to a max-min allocation on a transformed graph G^*, where each vertex v_i in G' is replaced by a clique with ℓ_i nodes (sub-vertices) in G^* and an edge between two vertices in G' translates to an edge between all sub-vertices of the two vertices in G^*. A clique C_m with m vertices in G' now translates to a clique with $\sum_{i:v_i \in C_m} \ell_i$ sub-vertices in G^*. Thus, to show A^3's allocation mechanism yields a weighted max-min allocation on G', it is sufficient to show that it yields a max-min allocation in G^* where each sub-vertex has unity load (weight). We will show this by contradiction.

Assume a given allocation is not max-min. Hence, consider two sub-vertices v_a, v_b in a maximal clique C_1 with respective allocations being x and $x+2$. For the allocation to not be max-min, we should be able to increase v_a's allocation either in isolation or by decreasing v_b's allocation. Since v_a's allocation was restricted to begin with, it must belong to a bigger clique than C_1, namely C_2. This is because, when all the loads are unity, the allocation mechanism picks vertices belonging to bigger cliques first. Now, if v_a's allocation can be increased by one sub-channel, then this must also not exceed the resource capacity of C_2. There are three possible cases.

(i) If v_a was allocated after other sub-vertices in C_2, then v_a's allocation would have already expanded to occupy the remaining capacity of C_2, which means that v_a's allocation cannot be increased further.

(ii) If v_a was not the last sub-vertex in C_2 to be allocated, then it might be that the sub-vertices that followed v_a were bottlenecked in other larger cliques and could only receive an allocation $< x$, thereby allowing v_a's allocation to be further increased to use up the remaining capacity in C_2. However, this is not possible since the sub-vertices that follow v_a in A^3 can only belong to equivalent or smaller cliques and will hence be able to receive an allocation $\geq x$, thereby using up C_2's capacity completely. Thus, v_a's allocation cannot be increased in this case as well.

(iii) Similar to v_b, let there by another sub-vertex v_c in C_2 that has an allocation of $x+2$ or more. In this case, v_a's allocation can be increased at the cost of v_b's and v_c's allocations in the two cliques. However, note that if v_c was allocated before v_a, then its allocation will be $\leq x$. Further, since v_a is bottlenecked in C_2, if v_c is allocated after v_a, then its allocation can at most be $x+1$. Thus, in either of these cases, v_a's increase will have to come at the cost of another sub-vertex with an allocation $\leq x+1$. ■

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

Proof: A^3 generates a clique tree for G' and starts assigning sub-channels to vertices in each of the nodes (cliques) in the clique tree starting from the root. A^3 can run into assignment problems if it encounters an un-assigned vertex v_i belonging to multiple cliques at the same level with conflicts such that it prevents a feasible assignment to v_i. This is where the *clique intersection property* of clique trees come into play.

The clique intersection property states that for every pair of distinct cliques C_1, C_2, the set of vertices in $C_1 \cap C_2$ will be contained in all the cliques on the path connecting C_1 and C_2 in the tree. Hence, if v_i appears at multiple cliques at a level in the tree, it must have appeared in isolation at some higher level in the tree. Given that all vertices, when encountered first, appear in a single clique at a level, a *feasible* assignment to the vertex is always possible since the allocations always satisfy the capacity in all cliques. Further, once a vertex is assigned sub-channels, its assignment carries over to other cliques containing it in subsequent levels. Thus, by using a clique tree for assignment, A^3 is able to ensure a feasible assignment of sub-channels given a weighted max-min allocation. ■

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.

Our evaluations in Section VI show that over 70% of the topologies are chordal to begin with, for which A^3 yields an optimal allocation. For other topologies, A^3's sub-optimality is $< 10\%$, indicating its near-optimal allocation capability.

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). These simpler heuristics do not need to operate on a complete list of maximal cliques as we describe next.

The first heuristic is *progressive* (labeled *prog*); here, the allocations and assignments are made in tandem one sub-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 indexed sub-channel available in its neighborhood. By assigning sub-channels one at a time, this heuristic achieves reasonable fairness. However, its running time is $O(|V|^2 N)$, where its dependence on N (number of sub-channels) makes it pseudo-polynomial, thereby affecting its scalability. It also results in a highly fragmented assignment of sub-channels, which in turn increases the control overhead in frames.

Another heuristic that can avoid 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 $(\frac{\ell_i(N - \sum_{j:(v_i,v_j)\in E, v_j \in \mathcal{A}} A_j)}{\sum_{j:(v_i,v_j)\in E, v_j \in \mathcal{U}} \ell_j})$. 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 as compared to *prog*.

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 conducted on the chordal graph G', dynamics in the form of arrival/departure of clients/cells (i.e. addition/deletion of conflict edges) can be easily accommodated in a purely localized manner through incremental schemes [24]. This in turn allows A^3 to scale well to network dynamics unlike other heuristics.

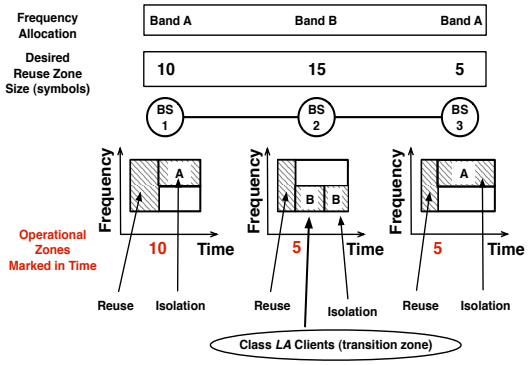


Fig. 8. Illustration of zoning mechanism of FERMI.

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 (not necessarily chordal). This requires listing all the maximal cliques, which are exponential in number. This can be done in a brute-force manner with exponential complexity. Once all the maximal cliques are listed, A^3 can be applied to obtain a weighted max-min allocation on G.

B. Zoning

After assigning 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 (referred to as s_r). (i) If two interfering cells use two different s_r's, the one with the larger s_r will cause interference to the class *ISO* 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 (min. or max.) 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 *LA* 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 propagates to the network.

FERMI addresses the above challenge as follows (illustration in Fig. 8). 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 *LA* clients in the reuse zone till the operational s_r (using all sub-channels). It continues to schedule class *LA* 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

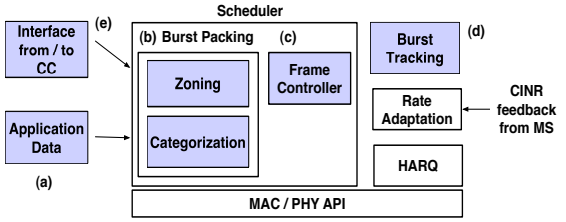


Fig. 9. Implementation details of FERMI.

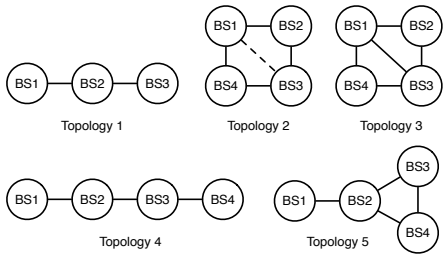


Fig. 10. Topologies used for prototype evaluation.

class *ISO* clients are scheduled in the resource isolation zone (after the desired s_r) using the band allocated by A^3.

Introducing a *transition* zone (marked on Fig. 8) that schedules class *LA* clients between the operational and desired s_rs (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, FERMI uses the minimum of the desired s_rs in the neighborhood as the operational s_r for a cell. Further, since each cell computes its operational s_r only based on the desired s_rs of its neighbors and not their operational s_rs, 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. 8. 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_rs in the different cliques (e.g. BS2 in Fig. 8) will not suffer from interference to their class *ISO* clients, since the operational s_r of all their cliques will be less than their desired s_r, while they schedule only class *LA* clients in the region between their operational and desired s_r. As we shall show in our evaluations, FERMI's zoning arrangement provides significant gains since the s_r values in different cliques can be decoupled (i.e. a single globally minimum desired s_r does not propagate).

VI. System Evaluation

A. Prototype Evaluations

Implementation Details: 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. 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 the reference design as follows (shown as shaded components in Fig. 9).

(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 §IV) 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)** The burst tracking component keeps an information tuple for the measurement zones for the MSs that are being categorized. It tracks the ACK status of each measurement burst. After enough BDR samples are collected, it decides on the client category and informs the burst packing component about the decision. **(e)** The interface with the CC leverages kernel sockets to communicate the load and conflict information to the CC via Ethernet and receives operational parameters for zoning and allocation (used by the burst packing component).

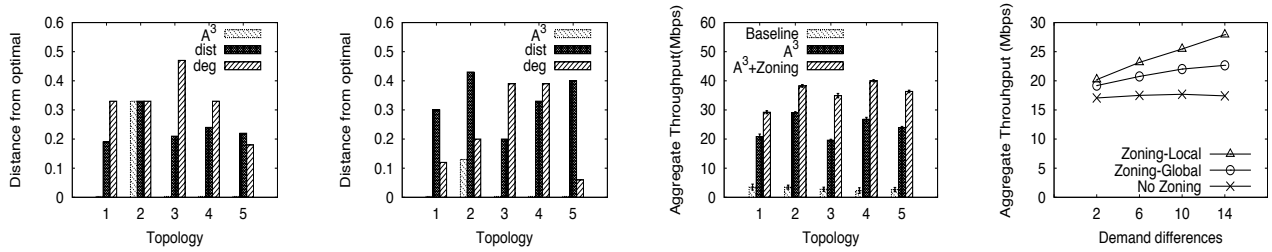
Experimental Evaluations: Next, we evaluate the performance of each algorithm using our testbed. We create five topologies as shown in Fig. 10. The dotted edge between BS1 and BS3 in Topology 2 is the $fill\text{-}in$ edge introduced by A^3 (other topologies are already chordal). In generating these topologies, we leverage our WiMAX testbed (see Fig. 2(a)) 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 $d = \sqrt{\sum_{i \in V}(t_i - s_i)^2} \ / \ \sqrt{\sum_{i \in V}(s_i)^2}$ [25] 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 testbed, the cells have $N = 30$ sub-channels available in the spectrum. Each BS has two clients (one class *LA*, one class *ISO*). When there is no zoning, we schedule both clients on 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 at each BS). Here, we introduce another 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).

	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 I

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.

Fig. 11. Fairness and Zoning Benefits of A^3.

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.

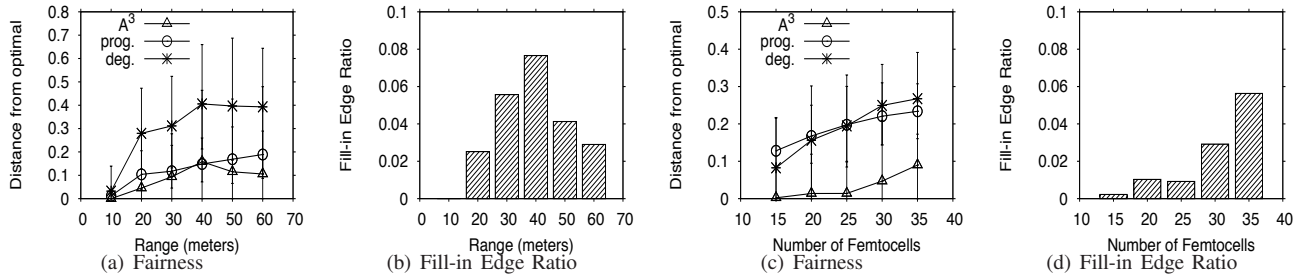
Table I summarizes the number of sub-channels allocated to each BS along with utilization and aggregate throughput measurements from the experiments. We observe that *dist* has the lowest utilization and therefore the lowest aggregate throughput. This is because it over-accounts for interference by just 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. 11(a) and 11(b) plot the fairness for equal load and variable load (the loads are listed next to each BS in parentheses in Table I), 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) helping it achieve the same utilization as the benchmark. In all other topologies, A^3 achieves the exact allocation as the benchmark (BM in Table I) since they are naturally chordal.

Zoning Benefits: We now present throughput measurements for A^3 - with and without zoning in Fig. 11(c). The baseline strategy is where each cell operates on all tiles 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 propagated to each contention region (global min.). For this experiment, we use topology 1 in Figure 10. We set equal reuse zone size demands for BS1 and BS2 (varied in each measurement) and a fixed demand of 4 symbols for BS3. Demand difference is defined as the difference between the common demands 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. 11(d). It is seen that both global min. and local min. zoning result in increasing throughput as the demand difference increases. For the global min., although the operational size is the same (4 symbols), the high demands of BS1 and BS2 allow them to schedule their class *LA* clients over a larger set of resources (recall the transition zone in §V). Note that since class *LA* clients are likely to support a higher MCS than class *ISO* clients, having a large demand contributes to throughput gains (as compared to scheduling class *ISO* 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.

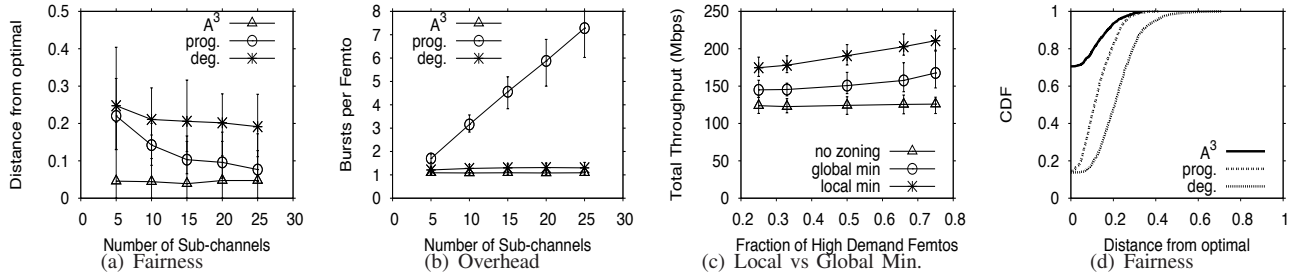
B. Evaluations with Simulations

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 [26]. This model captures 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 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 by randomly choosing grid locations for each cell. We then randomly generate a client location for each cell and determine the conflict graph. We measure the overhead of each algorithm as the number of contiguous sub-channel chunks allocated per cell. 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. Next, we present our simulation results. Each data point is an average over results from 100 randomly generated topologies.



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

Fig. 12. Effect of Range (a-b) and Effect of Number of Femtos (c-d). A^3's performance is mainly affected by the fill-in edge ratio.



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

Fig. 13. Effect of Number of Sub-channels (a-b), Effect of Zoning (c) and Overall Fairness (d).

Effect of Cell Range: Fig. 12(a) plots the effect of cell 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 cell is assigned decreases (resources are shared among more cells). Recalling the fairness formula, a given difference in allocations (between the benchmark and the heuristics) becomes more pronounced with a smaller 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. 12(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 make the graph chordal. However for further ranges, increased connectivity turns in favor of A^3 since the cycles happen rarely and the fill-in edge ratio decreases again.

Effect of Number of Cells: We now fix a number of sub-channels (5) and a range (20 m.) and vary the number of cells. From Fig. 12(c), we see that A^3 has better fairness than other heuristics 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 cells due to an increased likelihood of cycles; the trend again follows that of the fill-in edge ratio (plotted in Fig. 12(d)). For *deg* and *prog*, the distance also increases with the number of cells because of a reduced number of sub-channels per cell (s_i), similar to the effect of cell range.

Effect of Number of Sub-channels: We now simulate 30 cells with a fixed range of 10 m. and vary the number of sub-channels. Fig. 13(a) shows that A^3 has a constant distance from the optimal. Since A^3's performance is mainly affected by the fill-in edge ratio, the number of sub-channels does not show a significant effect. In addition, the distance for *prog* and *deg* decreases with an increased number of sub-channels due to the increase in number of sub-channels per cell (s_i). This makes the differences in allocations (between the heuristics and the benchmark) less pronounced as compared to when there are fewer sub-channels. Fig. 13(b) shows that the overheads for A^3 and *deg* are very close to 1 and do not change with the number of sub-channels. This shows that both strategies can assign a single contiguous set of sub-channels to the cells. However, *prog* tends to have an increasing overhead trend. Since *prog* assigns a fragmented set of sub-channels, the overhead increases with an increasing number of sub-channels that can be assigned to a cell.

Effect of Zoning: In simulations with zoning, each cell has two clients: one that requires resource isolation (class *ISO*) and one that requires just link adaptation (class *LA*). We simulate 40 cells with range 10 m, and a frame structure having 30 sub-channels and 30 symbols. Among the cell population, we have three different types of reuse zone demands: i) high-demand cells that randomly demand a reuse zone size between 15 and 20 symbols ii) moderate demand cells that generate a demand value between 10 and 15 symbols and iii) low-demand cells with a generated demand between 5 and 10 symbols. We experiment by varying the fraction of the high-demand cells. Fig. 13(c) shows the total throughput achieved for each zoning strategy. It is seen that as the fraction of high-demand cells increases, the throughputs for both zoning strategies increase. However, the gain of local min. over global min. is more with a higher fraction of high-demand cells. This is a natural artifact of vertices converging to a higher local demand as opposed to the global minimum value which is typically 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 (i.e. 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 simulations for the three algorithms considered, in Fig. 13(d). We mainly use the results with range 10 m. and 20 m., as these represent a more realistic deployment (given the entire area is 100x100 meters). The results 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 reaches the exact same allocation as the benchmark 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 A^3.

VII. Conclusions

In this paper, we design and implement FERMI, one of the first resource management systems for OFDMA femtocell networks. Resource management in femtocells has a unique set of practical challenges that - to the best of our knowledge - have not been addressed to date. FERMI provides a complete resource management solution with several unique features. It uses measurement-driven triggers to classify clients into two categories, those that need resource isolation and those that do not. It provides a frame structure that supports the graceful coexistence of clients from both categories. For purposes of interference mitigation, it allocates orthogonal sub-channels of the OFDMA spectrum with high utilization and low overhead. We implement FERMI on our WiMAX testbed and show via both experiments and simulations that its performance is superior to other conventional methods.

References

[1] R. V. 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 *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 *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] Y. Sun, R. P. Jover, and X. Wang, "Uplink Interference Mitigation for OFDMA Femtocell Networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, Feb 2012.

[7] H. Jo, C. Mun, J. Moon, and J. Yook, "Interference Mitigation Using Uplink Power Control for Two-Tier Femtocell Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, Oct 2009.

[8] V. Chandrasekhar and J. G. Andrews, "Uplink Capacity and Interference Avoidance for Two-Tier Femtocell Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, Jul 2009.

[9] Y. Kim, S. Lee, and D. Hong, "Performance Analysis of Two-Tier Femtocell Networks with Outage Constraints," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, Sep 2010.

[10] J. Yun and K. Shin, "CTRL: A Self-Organizing Femtocell Management Architecture for Co-Channel Deployment," in *ACM MobiCom*, Sept 2010.

[11] K. Sundaresan and S. Rangarajan, "Efficient Resource Management in OFDMA Femto Cells," in *ACM MobiHoc*, May 2009.

[12] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 3, July 2005.

[13] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-Aware Spectrum Distribution in Wireless LANs," in *IEEE ICNP*, 2008.

[14] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre, "MDG: Measurement-Driven Guidelines for 802.11 WLAN Design," in *ACM MobiCom*, 2007.

[15] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "Client-driven Channel Management for Wireless LANs," in *IEEE Infocom*, 2006.

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

[17] 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.

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

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

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

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

[22] J. R. S. Blair and B. W. Peyton, "An Introduction to Chordal Graphs and Clique Trees," http://www.ornl.gov/info/reports/1992/3445603686740.pdf.

[23] A. Berry, J. R. S. Blair, P. Heggernes, and B. W. Peyton, "Maximum Cardinality Search for Computing Minimal Triangulations of Graphs," in *Journal Algorithmica*, vol. 39, no. 4, May 2004.

[24] 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.

[25] R. Jain, A. Durresi, and G. Babic, "Throughput Fairness Index: An Explanation," in *ATM Forum Document Number: ATM Forum / 990045*, February 1999.

[26] S. Yeh and S. Talwar, "Multi-tier Simulation Methodology IEEE C802.16ppc-10/0039r1," 2010.