Max-Min Fair Resource Allocation in HetNets: Distributed Algorithms and Hybrid Architecture

Ehsan Aryafar
Portland State University
Portland, OR

Alireza Keshavarz-Haddad
Shiraz University
Shiraz, Iran

Carlee Joe-Wong
Carnegie Mellon University
Silicon Valley, CA

Mung Chiang
Princeton University
Princeton, NJ

***Abstract*—We study the resource allocation problem in RAN-level integrated HetNets. This emerging HetNets paradigm allows for dynamic traffic splitting across radio access technologies for each client, and then for aggregating the traffic inside the network to improve the overall resource utilization. We focus on the max-min fair service rate allocation across the clients, and study the properties of the optimal solution. Based on the analysis, we design a low complexity distributed algorithm that tries to achieve max-min fairness. We also design a hybrid network architecture that leverages opportunistic centralized network supervision to augment the distributed solution. We analyze the performance of our proposed algorithms and prove their convergence. We also derive conditions under which the outcome is optimal. When the conditions are not satisfied, we provide constant upper and lower bounds on the optimality gap. Finally, we study the convergence time of our distributed solution and show that leveraging appropriate policies in its design significantly reduces the convergence time.**

I. Introduction

Heterogeneous networks (HetNets) have recently emerged as the de-facto networking architecture to address the capacity and service demand of next generation 5G networks. The architecture comprises a hierarchy of 3GPP (Third Generation Partnership Project) LTE macro cells for ubiquitous coverage, and small cells of various sizes (*e.g.*, micro, pico, femto) and across different radio technologies (*e.g.*, LTE, WiFi, WiGig, 5G mmWave) to augment the capacity. To realize the gains associated with HetNets, consumer devices are also increasingly being equipped with multiple radio access technologies (RATs), and some are already capable of simultaneously utilizing various radio transceivers. This can potentially increase the device's performance, however, in practice the network wide optimization of resources would also require appropriate signaling and support from the network infrastructure.

To address the issue, the 3GPP has recently been very active in specifying interworking solutions across different RATs. The invention of Access Network Discovery and Selection Function (ANDSF) as part of the LTE Evolved Packet Core (EPC) network was a step in this direction [1]. ANDSF allows a client device to discover non-3GPP access networks (*e.g.*, WiFi, WiMAX) that can be used for data communication in lieu of 3GPP access networks (*e.g.*, HSPA, LTE). It also provides clients with traffic steering rules to control their connections. Currently, 3GPP is working on an emerging HetNets paradigm that allows for multi-RAT integration at the Radio Access Network (RAN) level [2]. Such a RAN-level integration allows for dynamic traffic splitting across RATs for each client, and then for aggregating the traffic inside the network to improve the overall resource utilization [4].

Designing appropriate architectures and algorithms for resource allocation in such RAN-level integrated HetNets is a highly challenging problem. Specifically, (i) a centralized solution can potentially find an optimal allocation and can be implemented by a server inside the core network. However, such an architecture would require real time signaling between BSs[1] and the central entity, and moreover would have a very high computational complexity. This makes a fully centralized solution infeasible, particularly in networks with *high dynamics*, *large size*, or *when low latency links between BSs and the central entity do not exist*; (ii) in HetNets, each client would be at different distances from different BSs. As a result, the PHY layer rate that a client sees would be different for different BSs, and the rates would be naturally different for different clients. This multi-rate property of HetNets makes it particularly challenging to design low complexity algorithms with performance guarantee even in a centralized architecture.

In this paper, we study the resource allocation problem in RAN-level integrated HetNets. We focus on the *max-min* service rate fairness as our performance metric. Service rate is a generic metric introduced in this paper that captures the QoS or performance achieved by each client, and can be equal to throughput, or weighted throughput of each client, among others. We analyze the properties of the optimal max-min fair solution in HetNets, and based on the learnings design appropriate architectures and algorithms that try to approximate that with low complexity. We next analyze some of the most important properties of our algorithms, including convergence to equilibria, optimality gap, and time to convergence. The key results are summarized as follows:

- We study the optimal resource allocation properties. We show that at optimality, clients and BSs can be divided into groups with a 1:1 mapping between the groups. Clients in the same group achieve the same service rate. Further, the optimal resource allocation matrix would have non-zero values only on specific diagonal blocks and the corresponding PHY rate matrix would be lower triangular [Theorem 1]. Based on the properties of the optimal outcome, we have designed a *distributed fair resource allocation* algorithm

[1]**We use BS as a generic term that denotes AP in WiFi, NB in 3G, etc.**

called DFRA. DFRA is executed autonomously and in parallel by each BS and has a guaranteed convergence property [Theorem 2]. However, we provide an example that shows the outcome of DFRA is not necessarily optimal.

- We design a hybrid computation architecture that leverages *opportunistic* centralized supervision to enhance the outcome of DFRA. Further, we design a *centralized resource allocation modification* algorithm called CRAM that has *tunable computational complexity* with guaranteed convergence. CRAM searches for cyclic shifts of clients' traffic across BSs and enhances the outcome of DFRA by moving clients' traffic to BSs with higher PHY rates. CRAM can operate with only partial network information. This, coupled with the tunable complexity property, makes the hybrid architecture design adaptable to network dynamics as well as the latency between BSs and the central entity.
- We quantify the distance between the outcome of DFRA and the unique optimal point. We show that in a particular class of multi-rate systems, the outcome of DFRA is optimal [Theorem 3]. When the conditions are not satisfied, we provide constant *upper* and *lower* bounds on the optimality gap [Theorem 4]. We also study the convergence time properties of DFRA and provide an upper bound that is exponential in the number of clients N [Theorem 5]. However, we show that leveraging appropriate policies in the design of DFRA would guarantee an $O(N \log N)$ bound on convergence time [Theorem 6].

This paper is organized as follows. We discuss the related work in Section II. We present our system model in Section III. In Section IV we discuss the properties of the optimal resource allocation and present the design of our algorithms. We provide bounds on *optimality gap* and *time to convergence* in Section V. We present the results of our simulations in Section VI. Finally, we conclude in Section VII. **All proofs are placed in the Appendix.**

II. Related Work

Multi-BS Communication. Recent works have built prototypes that allow a single client device to simultaneously communicate with multiple WiFi BSs and aggregate their bandwidths [3], [7] . This not only increases the client's throughput, but also results in significant improvement in reliability and stability of the connection due to the multi-BS diversity gain. These observations also hold in RAN-level integrated HetNets [4]. Other work [12] has used game theoretic tools to model selfish traffic splitting by each client in WLANs such that the total throughput is maximized. Similarly, MOTA [6] addresses the same objective of selfish utility maximization by each client, however, the client can also select its operator of choice. In contrast, the resource allocation problem in RAN-level integrated HetNets is primarily controlled and addressed by the network, potentially based on the feedback from the clients.

Fractional Client-BS Association. The problem of network controlled *integral association*, in which each client is connected to only a single BS, has been addressed in WLANs [5], [9], and a HetNet composed of a *single* WiMAX BS and multiple WiFi BSs [13]. The objective has been to achieve max-min fairness in [5], [13], and proportional fairness in [9]. Similar to our paper, all these works assume enough frequency planning such that different BSs do not interfere with each other. Since the problem is NP-Hard, all have designed centralized approximation algorithms based on the optimal *fractional association* outcome. With fractional association, a client can split its traffic across different BSs. This can potentially turn an NP-Hard integer programming problem into a linear program (LP) which can then be solved optimally with commercial tools, albeit with high computational complexity. Indeed, problem $\mathcal{P}_1$ addressed in this paper can be solved optimally based on the LP techniques of [10] with $O(N \times LP(N, M))$ complexity, where N is the number of clients, M is the number of BSs, and $LP(N, M)$ is the complexity of LP. However, these centralized solutions have high complexity and require real time signaling between BSs and the central entity, which make them infeasible particularly in large, dynamic networks, or when low latency links between BSs and the central entity do not exist.

III. System Model

A. Network Model

We consider a heterogeneous wireless network deployment which is composed of a set of BSs $\mathbf{M} = \{1, ..., M\}$, and a set of clients $\mathbf{N} = \{1, ..., N\}$. Here, BS is simply a generic term to collectively represent NB in 3G, eNB in 4G, AP in WiFi, femtoBS in femto-cells, etc. Each BS has a limited transmission range and only serves clients within its range. Neighboring BSs may overlap in their coverage. Each client has a specific number of RATs, and therefore can have access to a subset of BSs. We assume that all clients split their traffic over multiple BSs and focus on the traffic splitting problem for each client. It is itself a challenging problem to determine, which BS to associate with among same technology BSs (*e.g.*, choosing the optimal WiFi BS if a client has a WiFi RAT). We assume there exists a rule to pre-determine client RAT-BS association. The pre-determination rule could be any load balancing association algorithm [13], [14], or based on the received signal strength, among others.

Similar to [5], [8], [9], [12], [13], we assume that the transmission in one BS does not interfere with an adjacent BS. This can be achieved by means of spectrum separation between BSs that belong to different access networks, and frequency reuse among same kind BSs. We consider a multi-rate system and use $R_{i,j}$ to denote the PHY rate of client i from BS j. Since each BS generally serves more than one client, clients of the same BS need to share resources such as time and frequency slots (*e.g.* in 3/4G) or transmission opportunities (*e.g.* in WiFi). The service rate experienced by client i from BS j thus depends on the load of the BS and will therefore be a fraction of $R_{i,j}$. We assume that each BS employs a TDMA throughput sharing model and let $\lambda_{i,j}$ denote the fraction of time allocated to client i by BS j. Hence, the throughput

achieved by client i from BS j is equal to $\lambda_{i,j}R_{i,j}$ and its total throughput across all its RATs would be $\sum_{j=1}^{M}\lambda_{i,j}R_{i,j}$.

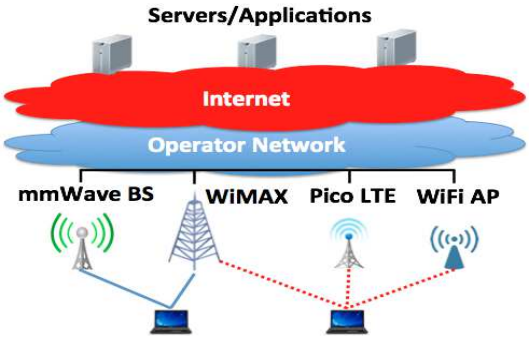


Fig. 1. **A HetNet with 4 access technologies. Each client is in the coverage area of a group of BSs and can aggregate its traffic across the corresponding BSs (RATs). The industry is expecting to add several new access technologies particularly in cm/mmWave bands (*e.g.*, 15GHz, 28GH, and 45GHz) for the next generation 5G networks, re-emphasizing the heterogeneity aspect of future generation wireless networks.**

B. Performance Metric

Before we can formally define our problem formulation, we need to choose a performance metric. We define h_i as the *service rate* of client i and choose it as our performance metric. All of our algorithms and analysis in this paper are generic and apply to *any* definition of h_i, as long as in the proposed definition h_i is an increasing function of $R_{i,j}$ and $\lambda_{i,j}$. Below are two example h_i definitions:

$$\begin{cases} h_i = \text{Total Throughput of Client i} = \sum_{j=1}^{M}\lambda_{i,j}R_{i,j} \\ h_i = \dfrac{\text{Total Throughput of Client i}}{\text{Client Specific Weight}} = \dfrac{\sum_{j=1}^{M}\lambda_{i,j}R_{i,j}}{\omega_i} \end{cases}$$

TABLE I
MAIN NOTATION

$\mathbf{N}$: Set of all clients in the network
N: Number of clients in the network
$\mathbf{M}$: Set of all BSs in the network
M: Number of BSs in the network
$R_{i,j}$: PHY rate of client i to BS j
$\mathbf{R}$: $[R_{i,j}]_{N\times M}$
R_{max}: maximum PHY rate across all clients and BSs
R_{min}: non-zero minimum PHY rate across all clients and BSs
$\lambda_{i,j}$: Fraction of time allocated to client i by BS j
$\mathbf{\Lambda}$: $[\lambda_{i,j}]_{N\times M}$
h_i: Service rate of client i
ω_i: Client i's weight

Definition 1. *Lexicographic Order:* Let $\mathbf{h} = (h_i, i \in \mathbf{N})$ denote the vector of service rates across all clients in the network, and $\bar{\mathbf{h}}$ be the allocation vector $\mathbf{h}$ sorted in nondecreasing order. An allocation vector $\bar{\mathbf{h}}_1$ is said to be lexicographically greater than another allocation vector $\bar{\mathbf{h}}_2$, denoted by $\bar{\mathbf{h}}_1 \succ \bar{\mathbf{h}}_2$, if the first non-zero component of $\bar{\mathbf{h}}_1 - \bar{\mathbf{h}}_2$ is positive. Consequently, an allocation vector $\bar{\mathbf{h}}_1$ is said to be lexicographically no less than another allocation vector $\bar{\mathbf{h}}_2$, denoted by $\bar{\mathbf{h}}_1 \succeq \bar{\mathbf{h}}_2$, if $\bar{\mathbf{h}}_1 - \bar{\mathbf{h}}_2 = 0$, or the first non-zero component of $\bar{\mathbf{h}}_1 - \bar{\mathbf{h}}_2$ is positive.

C. Problem Formulation

Our objective in this paper is to study lexicographic max-min fair service rates in HetNets. A service rate allocation is said to be *lexicographic max-min fair* if the corresponding service rate allocation vector is lexicographically no less than any other feasible service rate allocation vector. In the lexicographic max-min fair allocation, therefore, a service rate component can be increased only at the cost of decreasing a service rate component of equal or lower value. Mathematically, when h_i is defined as total throughput the problem can be formulated as

$$\begin{aligned} \mathcal{P}_1: \quad \text{lex}\max \quad & \bar{\mathbf{h}} = \text{non-decreasing order } (h_i, i \in \mathbf{N}) \\ s.t. \quad & h_i = \sum_{j=1}^{M}\lambda_{i,j}R_{i,j} \quad \forall i \in \mathbf{N} \\ & \sum_{i=1}^{N}\lambda_{i,j} \le 1 \qquad \forall j \in \mathbf{M} \\ \text{variables:} \quad & \lambda_{i,j} \ge 0 \qquad \forall i \in \mathbf{N}, j \in \mathbf{M} \end{aligned}$$

The feasible region in problem $\mathcal{P}_1$ is determined by linear constraints, and is therefore convex. Since all the $R_{i,j}$ and $\lambda_{i,j}$ values are non-negative and finite, it follows that the feasible region is also bounded, and therefore compact. The results in [10], [11] have proven that the lexicographic max-min fair allocation vector on any convex and compact set exists and is unique. However, finding the optimal solution could have a very high computational complexity. This makes it all the more challenging to design low complexity distributed algorithms with guaranteed performance.

IV. Algorithm Design

We first describe some useful properties of the optimal resource allocation that we leverage for designing our distributed algorithm. Next, we propose a distributed resource allocation algorithm that achieves guaranteed performance. Finally, we discuss the design of a hybrid computation architecture and show how a minimal amount of centralized network supervision can enhance the outcome.

A. Optimal Resource Allocation Properties

In order to obtain an intuition on the properties of the optimal resource allocation, consider an example wireless network with 2 BSs and 3 clients as depicted in Fig. 2. Here j_1 and j_2 represent the 2 BSs, while i_1, i_2, and i_3 represent the 3 clients. The first 2 clients (i_1 and i_2) have non-zero PHY rates only from j_1, whereas i_3 has non-zero PHY rates from both j_1 and j_2. Since i_1 and i_2 have non-zero PHY rates from only a single BS (j_1), therefore at optimality they must have non-zero λs from j_1. This is because if either λ is 0, the corresponding service rate h (and hence the lexicographic allocation vector $\bar{\mathbf{h}}$)

can be increased by BS j_1 balancing the two λs such that none is zero. Further, i_1 and i_2 must achieve the same service rate at optimality (*i.e.*, $h_1 = h_2$). This is because if for example h_2 is greater than h_1, then BS j_1 can equalize the service rates (and hence increase the lexicographic allocation $\bar{\mathbf{h}}$) by allocating to i_1 some of the resources that belong to i_2.

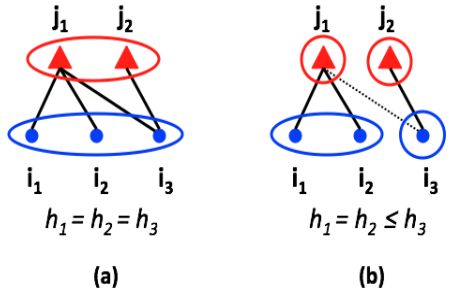


Fig. 2. **An example network with 2 BSs (j_1, j_2) and 3 clients (i_1, i_2, i_3). Non-zero $R_{i,j}$s are demonstrated through solid or dotted lines. Further, solid lines show non-zero $\lambda_{i,j}$s, whereas dotted lines show zero $\lambda_{i,j}$s. BS and client groups are shown by circles. Two possible outcomes exist in this topology at optimality: (a) a single client and a single BS group exist, and all clients achieve the same service rate, and (b) 2 BS and 2 client groups exist. Note that neither i_1 nor i_2 has a non-zero PHY rate $R_{i,j}$ to j_2.**

On the other hand, h_3 can never be less than h_1 or h_2 at optimality (note that as we just discussed, $h_1 = h_2$ at optimality). This is because i_3 has a non-zero PHY rate from j_1 and hence j_1 can equalize all the three service rates (thereby increasing the lexicographic allocation $\bar{\mathbf{h}}$) by allocating to i_3 some of the resources that belong to i_1 and i_2.

The above discussion shows that only two possible optimal outcomes exist for the topology depicted in Fig. 2, while the unique optimal solution depends on the specific PHY rates ($R_{i,j}$s). In the first possible outcome (Fig. 2(a)), the clients and BSs can be partitioned into 2 separate groups (in Fig. 2 each group is denoted as a circle). All the 3 clients achieve the same service rate, and the clients in this single group receive all their service from the corresponding BS group (composed of both j_1 and j_2).

In the second possible outcome (Fig. 2(b)), clients and BSs can be partitioned into 2 BS, and 2 client groups. Clients in the same group (*e.g.*, the first client group composed of both i_1 and i_2) achieve the same service rate, and each client group is only served by a corresponding BS group. Further, client groups can be ordered from low to high service rates in a manner that no client group would have a non-zero PHY rate to a BS that corresponds to a client group with a higher service rate (*e.g.*, i_1 and i_2 in Fig. 2(b) do not have a non-zero PHY rate to j_2).

Our first Theorem shows that these observations apply to the optimal (*i.e.*, lexicographic max-min fair) allocation in generic HetNets. Assume that each BS has a non-zero PHY rate to at least one client. Similarly, assume that each client has a non-zero PHY rate to at least one BS. Let $\{A_k\}$ and $\{U_k\}$ denote partitioning of BSs and clients into separate disjoint groups, respectively.

***Theorem** 1:* Let $\mathcal{B}$ be the lexicographic max-min fair allocation. Then, there exists a partitioning of BSs and clients into groups $\{A_k\}$ and $\{U_k\}$ with a 1:1[2] mapping between them such that: (i) all the clients in the same group achieve the same service rate, (ii) clients in the same group receive all their service only from BSs in the corresponding BS group, and (iii) client groups can be ordered from low to high service rates in a manner that no client group would have a non-zero PHY rate to a BS that corresponds to a client group with higher or equal service rate.

Fig. 3 shows the partitioning of BSs and clients at optimality. There exists a 1:1 mapping between the K BS groups (denoted as $\{A_k\}$) and the K client groups (denoted as $\{U_k\}$). Clients that belong to the same group (*e.g.* clients in U_k) achieve the same service rate. This service rate is defined as the group service rate and is denoted as γ_k. Further, clients in the same group have non-zero λs only from BSs in their corresponding BS group (shown through solid lines). These hint that autonomous service rate equalization by each BS across its clients can potentially provide a good distributed solution to our problem $\mathcal{P}_1$. We will turn this idea into an algorithm in Section IV-B.

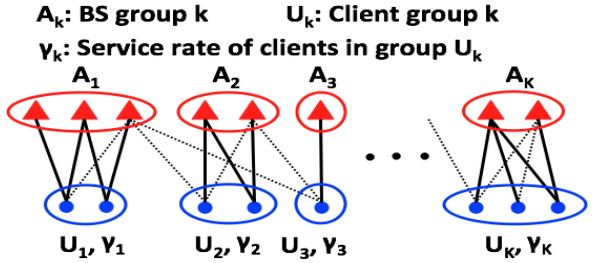


Fig. 3. **Partitioning of BSs and clients into separate groups at optimality. Non-zero $R_{i,j}$s are demonstrated through solid or dotted lines. Further, solid lines show non-zero $\lambda_{i,j}$s, whereas dotted lines show zero $\lambda_{i,j}$s. Clients in the same group (*e.g.*, U_k) achieve the same service rate. This service rate is defined as the group service rate and is denoted as γ_k. Here $\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_K$, hence no client in group U_k can have a non-zero λ (dotted line) to any BS in group $A_{k'}$ in which $k' > k$.**

In the example of Fig. 3 we have

$$\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_K$$

hence because of the third property in Theorem 1, none of the clients in group U_k can have a non-zero λ (dotted line) to any BS in a group $A_{k'}$ in which $k' > k$.

If we let Λ denote the $\lambda_{i,j}$ matrix (*i.e.*, $\Lambda = [\lambda_{i,j}]_{N \times M}$) and assume that $\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_K$, Theorem 1 and Fig. 3 imply that the optimal Λ can be written in the format of Fig. 4. In other words, if we arrange the client groups in non-decreasing service rate order, the resulting Λ matrix could have non-zero $\lambda_{i,j}$s only on the diagonal blocks that correspond to the client and BS groups.

[2]Here 1:1 means that for each client group there is a corresponding BS group with the properties described in Theorem 1. Note that the optimal partitioning is not necessarily unique. Further, not every partitioning with the properties of Theorem 1 is optimal.

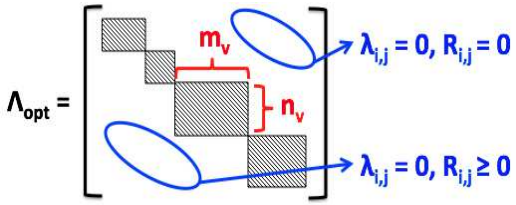


Fig. 4. **Optimal resource allocation matrix has non-zero values only on specific diagonal blocks. The corresponding PHY rate matrix is lower triangular.**

If we let $m_v = |A_v|$ and $n_v = |U_v|$ denote the number of BSs and clients in group v, then with abuse of notation, for the corresponding sub-matrix we have

$$\sum_{i=1}^{n_v} \lambda_{i,j} = 1$$

$$\sum_{j=1}^{m_v} \lambda_{i,j} R_{i,j} = \text{Client } i\text{'s Total Throughput}$$

Finally, we emphasize that while the optimal solution has the properties described in Theorem 1, and clients and BSs can be partitioned in the format depicted in Fig. 3, the reverse is not necessarily true. In other words, there may exist one or more sub-optimal partitionings that have the format depicted in Fig. 3 or have the properties described in Theorem 1. In the next section we provide one such example.

B. Distributed Fair Resource Allocation (DFRA)

We propose a distributed resource allocation algorithm that is implemented by each BS. Consider synchronized slotted time for now. In our design, each client i reports its current service rate h_i to all BSs with non-zero PHY rates (*i.e.*, $\forall j \; s.t. \; R_{i,j} > 0$). In return, each BS tries to lexicographically increase the service rate vector through local service rate equalization of its clients.

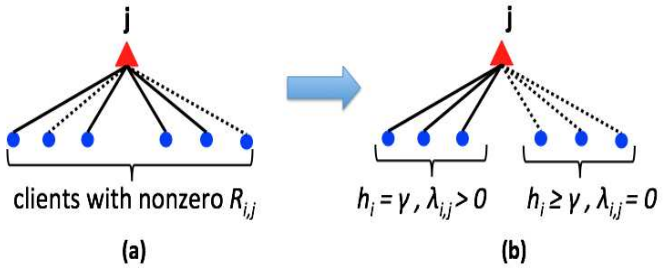


Fig. 5. **Local equalization by BS j. Upon execution, a group of clients achieve the same service rate with non-zero resources from j (*i.e.*, $h_i = \gamma$ and $\lambda_{i,j} > 0$). The rest of the clients would have equal or higher service rates with zero resources from j (*i.e.*, $h_i \geq \gamma$ and $\lambda_{i,j} = 0$). By taking this step, j locally increases the lexicographic order across its clients.**

Fig. 5 depicts the equalization procedure executed by BS j. Here n' clients have non-zero rates ($R_{i,j} > 0$) (Fig. 5(a)). Upon equalization, the BS takes some of the resources that belong to higher service rate clients and gives them to clients with lower service rates. At the end of this step, a group of clients achieve the same service rate (*i.e.*, $h_i = \gamma$) whereas the rest of the clients would have higher service rates ($h_i \geq \gamma$) with zero resources from BS j (*i.e.*, $\lambda_{i,j} = 0$). This outcome is depicted in Fig. 5(b).

Algorithm DFRA (Fig. 6) summarizes the resource allocation algorithm implemented by each BS j. DFRA equalizes service rates and increases $\bar{\mathbf{h}}$ by taking two steps: (i) clients are sorted based on their aggregate service rates from **other** BSs (Line 2), and (ii) **Equalization Step:** resources are given from high service rate clients to lower ones (Line 3). By taking this step, a BS increases the lexicographic allocation vector across its clients. Finally, we introduce the randomization parameter p_j in order to limit concurrent resource adaptation of a single client by multiple BSs (Lines 4-7).

Algorithm DFRA: Distributed Fair Resource Allocation
Input: h_i , $R_{i,j}$, and $\lambda_{i,j}$ $\forall$ client i for which $R_{i,j} > 0$
randomization parameter p_j
Output: Updated $\lambda_{i,j}$ and h_i
1. Let n' denote the number of clients *s.t.* $R_{i,j} > 0$
2. Sort clients based on their total achieved service rates from **other** BSs, *i.e.*,
$h'_1 \leq \ldots \leq h'_k \leq h'_{k+1} \leq \ldots \leq h'_{n'}$
3. Equalize service rates, *i.e.*, find k, γ, and $\lambda_{i,j}$ *s.t.*
$h_1 = h_2 = \ldots = h_k = \gamma \;\; | \;\; \lambda_{i,j} > 0$, $\sum_{i=1}^{k} \lambda_{i,j} = 1$
h_{k+1} , ... , $h_{n'} \geq \gamma$ *and* $\lambda_{i,j} = 0$
4. **if** $(rand < p_j^{mj+1})$ **then**
5. *Update* $\lambda_{i,j}$
6. **if** concurrent adaptation **then** increment m_j
7. **else** reset m_j to 0
8. **end**

Fig. 6. **Description of the resource allocation algorithm which is autonomously implemented by each BS j.**

The exact method for finding the appropriate resource fractions (*i.e.*, λs) for clients depends on the specific service rate definition. We next discuss the required calculations to implement Line 3 in Algorithm DRFA when h_i is defined as the weighted total throughput (*i.e.*, $h_i = \frac{\sum_{j=1}^{M} \lambda_{i,j} R_{i,j}}{\omega_i}$).

Let's assume $h'_1 \leq \ldots \leq h'_k \leq h'_{k+1} \leq \ldots \leq h'_{n'}$ is an ordering on clients based on their total received service rates from BSs other than j. Here n' denotes the number of clients at BS j such that $R_{i,j} > 0$. Next, we equalize service rates by finding resource fractions ($\lambda_{i,j}$s), client index $k \leq n'$, and client group service rate γ such that

$$h'_1 + \frac{\lambda_{1,j} R_{1,j}}{\omega_1} = h'_2 + \frac{\lambda_{2,j} R_{2,j}}{\omega_2} = \ldots = h'_k + \frac{\lambda_{k,j} R_{k,j}}{\omega_k} = \gamma \quad (1)$$

$$h'_k < \gamma \leq h'_{k+1} \quad (2)$$

$$\sum_{i=1}^{k} \lambda_{i,j} = 1 \qquad \lambda_{i,j} > 0 \quad (3)$$

Eq. (1) ensures that all k clients with lowest service rates achieve the same value (*e.g.*, γ). Eq. (2) ensures that the resulting operation increases the lexicographic allocation vector. Finally, Eq. (3) ensures that the rest of the clients' service rates would be higher than γ and that they would not receive any resources from BS j (*i.e.*, $\lambda_{i,j} = 0$).

We can find the appropriate k, γ, and λs that solve Eqs. (1)-(3) with a simple set of calculations. First, we can easily find k by checking the following inequalities

$$\begin{cases} \frac{(h'_2-h'_1)\omega_1}{R_{1,j}} > 1 \Rightarrow k=1 & \textbf{else} \\ \frac{(h'_3-h'_1)\omega_1}{R_{1,j}} + \frac{(h'_3-h'_2)\omega_2}{R_{2,j}} > 1 \Rightarrow k=2 & \textbf{else} \\ \dots \\ \frac{(h'_{n'}-h'_1)\omega_1}{R_{1,j}} + \dots \frac{(h'_{n'}-h'_{n'-1})\omega_{n'-1}}{R_{n'-1,j}} > 1 \Rightarrow k=n'-1 & \textbf{else} \\ k=n' \end{cases}$$

In the above inequalities, at each step we verify if there exists a valid resource allocation that can increase and equalize the service rates. For example, in the first inequality we check if $h'_2 > \frac{R_{i,j}}{\omega_1} + h'_1$. If this is true, because of Eq. (1), this implies that client 2 would receive a higher service rate than client 1 even if BS j allocated all its resources to client 1. As a result k should be equal to one and all BS resources should be allocated to client 1. The above procedure is then repeated until the appropriate k is found.

Once k is found, we can combine Eqs. (1) and (3) and find γ by solving the following simple equation:

$$\sum_{i=1}^{k} \frac{(\gamma - h'_i)\omega_i}{R_{i,j}} = 1 \tag{4}$$

With known k and γ, $\lambda_{i,j}$s can be found from Eq. (1). We emphasize that while we presented the detailed discussion on how to implement Line 3 of DFRA with h defined as weighted total throughput, the algorithm is generic and applies to any definition of h_i, as long as h_i increases with $R_{i,j}$ and $\lambda_{i,j}$.

C. Convergence and Complexity of DRFA

We next discuss the convergence properties of DFRA. First, we present the formal definition of an equilibrium.

Definition 2. *Equilibrium:* A resource allocation outcome is an equilibrium if none of the BSs can lexicographically increase the service rate vector across its clients through unilateral change of its time resource fractions (λs).

Our next Theorem guarantees convergence of DFRA.

***Theorem** 2:* If BSs follow the distributed resource allocation algorithm described in Fig. 6, then the resource allocation converges to an equilibrium. Further, the resulting partitioning structure of BSs and clients has the three properties of the optimal partition that was described in Theorem 1.

While Theorem 2 guarantees convergence of DFRA to equilibria, the resulting equilibrium is not necessarily optimal. Fig. 7 shows such an example in a topology with 2 BSs and 2 clients, and PHY rate matrix $\mathbf{R}$. Here $\mathbf{\Lambda_{eq}}$ shows a resource allocation matrix at equilibrium, which could be an outcome of DFRA. The resulting total throughput by each client is 1.8. However, at optimality (optimal equilibrium) the total throughput of each client is 2.4. This can be achieved by the optimal resource allocation $\mathbf{\Lambda_{opt}}$ depicted in Fig. 7(d).

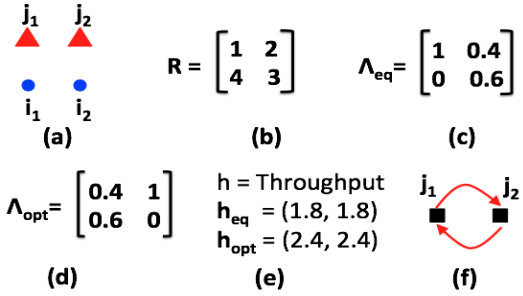


Fig. 7. **DFRA converges to an equilibrium, but the resulting equilibrium may not be optimal. Here, each client's throughput at equilibrium is 1.8. Per-client throughput at optimality (optimal equilibrium) is 2.4 (a-e). Construction of G and the corresponding loop in CRAM (f).**

DFRA's Computational Complexity and Message Passing Overhead. DFRA requires a certain number of steps to reach an equilibrium. In each step, the service rate vector improves lexicographically. During each step, a BS requires $O(n' \log(n'))$ complexity for sorting (n' is the average number of clients on a BS) and $O(n')$ complexity for service rate equalization. Thus, the overall complexity is $O(n' \log(n'))$.

After each equalization, DFRA updates clients on their updated resource fractions (λ s). This requires $O(m'n')$ (m' is the average number of BSs to which a client has a non-zero PHY rate) message passing overhead between clients and BSs. Finally, we emphasize that in networks where a wired backbone exists (*e.g.*, enterprise networks with an Ethernet backbone) each BS can broadcast service rate of clients affected by its equalization. This eliminates the required over the air message passing between clients and BSs.

D. Hybrid Computation Architecture and CRAM

In this section, we suggest a hybrid network architecture that uses a combination of both centralized and distributed computation to determine a fair resource allocation. We assume existence of a network controller (NC) that has information about the set of clients on each BS and their PHY rates.

In practical HetNet deployments such a central entity would have different communication links (with different delay, capacity, etc) to different BSs. This, coupled with the extent of network dynamics (*e.g.*, client mobility), poses a limit on how frequent resource allocation needs to be recalculated, and therefore, how much processing can be done in the NC.

Here we suggest a middle ground architecture that divides computation into two parts: distributed computation by each BS followed by *tunable* (in terms of computation time) and *opportunistic* centralized supervision that enhances the outcome of the distributed solution.

In order to understand how such a centralized supervision can enhance the outcome of DFRA, consider the equilibrium outcome in Fig. 7. Here each client has a higher time resource

fraction (λ) from the BS with lower PHY rate. For example for client i_1, $\lambda_{1,1} = 1$ ($R_{1,1} = 1$) and $\lambda_{1,2} = 0.4$ ($R_{1,2} = 2$). Similarly for client i_2, $\lambda_{2,1} = 0$ ($R_{2,1} = 4$) and $\lambda_{2,2} = 0.6$ ($R_{2,2} = 3$). Thus if time resources ($\lambda_{i,j}$) are shifted over links with larger PHY rates, both clients can achieve a higher service rate (throughput for the example in Fig. 7). For example, let ϵ denote the portion of time fraction shifted by each BS. Now if j_1 gives ϵ of i_1's resources to i_2 and j_2 gives ϵ of i_2's resources to i_1 we have

$$\lambda_{1,1} = \lambda_{1,1} - \epsilon \qquad \lambda_{2,1} = \lambda_{2,1} + \epsilon \tag{5}$$

$$\lambda_{1,2} = \lambda_{1,2} + \epsilon \qquad \lambda_{2,2} = \lambda_{2,2} - \epsilon \tag{6}$$

This is a superior allocation since $R_{2,1} > R_{2,2}$ and $R_{1,2} > R_{1,1}$. Now if we let $\epsilon = 0.4$, we reach the optimal allocation depicted in Fig. 7(d)

In general multi-BS HetNets we must look for a cyclic shift of time resources (λs) across BSs to enhance the resource allocation outcome. Fig. 8 depicts the appropriate shift in resources and the corresponding conditions. In Fig. 8(a), BS j_1 reduces the time resource allocated to i_2 by ϵ and gives that to i_1. BS j_2 increases the time allocated to i_2 by ϵ and reduces from i_3. The cyclic shift completes when the last BS (j_k) reduces i_1 by ϵ and increases i_k by ϵ. Note that at each step a client loses some time resource from some BS, but compensates that by receiving the same resource from another BS, albeit with higher PHY rate. This increases the client's service rate.

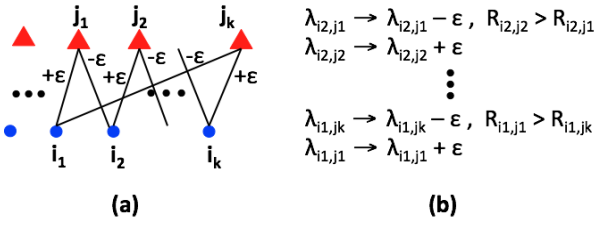


Fig. 8. **Resource allocation outcome can be enhanced by finding a cyclic shift of time resources across clients and BSs. In the above cycle, each client i reduces its time fraction resource ($\lambda_{i,j}$) by ϵ from BS j which has a lower PHY rate, but compensates that by receiving the same time resource from another BS j' with a higher PHY rate.**

Based on the above observations, we have designed an opportunistic *centralized resource allocation modification* algorithm called **CRAM** that searches for cyclic shifts of clients' traffic across BSs and enhances the outcome of the starting point (*e.g.*, outcome of DFRA) by moving clients' traffic to BSs with higher PHY rates.

Algorithm CRAM depicted in Fig. 9 summarizes the procedure. We construct a *directed traffic shifting graph* G by taking as input the PHY rate and resource allocation matrices of BSs and clients for which up-to-date information is available (Line 1). Each vertex in G corresponds to a BS $j \in \mathbf{M}$. There exists a directed edge between two vertices j and j' for each client i for which $R_{i,j} < R_{i,j'}$ and $\lambda_{i,j} > 0$[3]. Next, we find the maximum amount of time shifting that can be performed between each two BSs that have an edge between them (Line 2). Finally, we iterate over G for a maximum number of T iterations (Line 3). During each iteration, we try to find a cycle in G leveraging traditional cycle detection algorithms such as Depth First Search (DFS) (Line 4). Once a cycle is identified, we find the maximum amount of time shifting that can be performed across all the corresponding edges (Line 6), and modify the time resource fractions accordingly (Lines 9-10). At each iteration, at least one edge would be removed from G[4] (Line 11). The procedure continues until G becomes acyclic or the maximum number of iterations T is reached.

Algorithm CRAM: Centralized Resource Allocation Modification
Input: PHY rate matrix $[R_{i,j}]_{N\times M}$
Resource allocation matrix $[\lambda_{i,j}]_{N\times M}$
Number of iterations T
Output: Updated resource allocation matrix $[\lambda_{i,j}]_{N\times M}$
1. Construct a directed Graph $G = (V, E)$ with M vertices
Each vertex $j \in V$ corresponds to the BS $j \in \boldsymbol{M}$.
$\exists$ a directed edge $e \in E$ from j to j' (*i.e.*, $e = (j,j')$) for each *client* i for which $R_{i,j} < R_{i,j'}$ and $\lambda_{i,j} > 0$
2. Let $\varepsilon_{j,j'} = max_i \lambda_{i,j}$ and $i^{j,j'} = arg\ max_i\ \lambda_{i,j}$ be defined over those clients i for which $R_{i,j} < R_{i,j'}$ and $\lambda_{i,j} > 0$
3. **for** t = 1:T
4. Let $c = j'_1 \rightarrow j'_2 \rightarrow \ldots \rightarrow j'_k \rightarrow j'_1$ denote a directed cycle in G and let each e = (j',j'') ∈ c correspond to a client $i^{j',j''}$ as defined in Line 2
5. **if** c = {} **break**
6. Let $\varepsilon = min\ \{\varepsilon_{j'1,j'2}, \varepsilon_{j'2,j'3}, \ldots, \varepsilon_{j'k,j'1}\}$
7. ∀e = (j',j'') ∈ c **do**
8. let $i' = i^{j',j''}$
9. $\lambda_{i',j'} = \lambda_{i',j'} - \varepsilon$
10. $\lambda_{i',j''} = \lambda_{i',j''} + \varepsilon$
11. **if** $\lambda_{i',j'} = 0$ remove e from E
12. update $\varepsilon_{j',j''}$ and $i^{j',j''}$
13. **end**

Fig. 9. **Detailed description of CRAM.**

We have made CRAM's computational complexity controllable by making the number of times (T) that it looks for cycles in graph G tunable. Further, *CRAM can also operate with partial network information by performing resource modification to BSs and clients for which up-to-date information is available.* These properties make the hybrid architecture design adaptable to network dynamics, size, and the latency between BSs and the central entity.

CRAM enhances the outcome of DFRA by increasing the service rate across a group of clients while not affecting the rest of the clients. *Unlike DFRA, CRAM's outcome is not necessarily an equilibrium.* Hence, DFRA can further operate on the outcome of CRAM and lexicographically increase it until an equilibrium is reached. We will evaluate the joint operation

[3]Fig. 7(f) shows the graph G and the resulting cycle for the equilibrium allocation (Λ_{eq}) of the topology in Fig. 7(a).

[4]**Since the number of cycles in G is finite, CRAM converges. However, the resulting outcome may not be an** *equilibrium*.

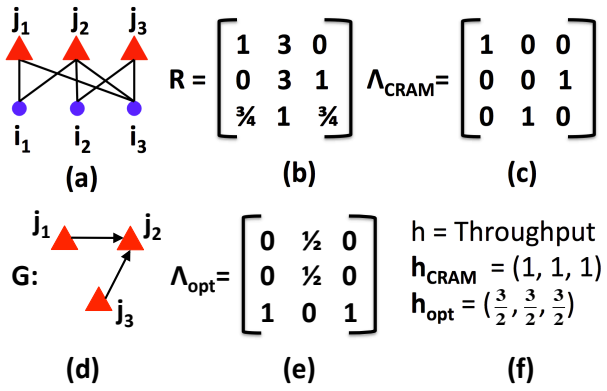


Fig. 10. **Outcome of CRAM is not necessarily optimal or an equilibrium. The example above shows that the outcome of CRAM is not necessarily optimal. Here we have an example topology with 3 BSs and 3 clients (a) with PHY rate matrix R (b). Resource allocation outcome of CRAM is denoted as Λ_{CRAM}(c). The resulting graph G is acyclic (d). The optimal allocation is denoted as Λ_{opt} (e). Per client service rate (throughput) outcome of CRAM and optimal are shown in (f).**

of DFRA and CRAM in Section VI. Finally, CRAM's outcome is not necessarily optimal. Fig. 10 shows one such example, in a topology with 3 BSs and 3 clients and PHY rate matrix $\mathbf{R}$. Here $\mathbf{\Lambda_{CRAM}}$ shows outcome of CRAM and $\mathbf{\Lambda_{opt}}$ shows the optimal allocation.

CRAM's Computational Complexity and Message Passing Overhead. The Network Controller (NC) constructs the traffic shifting graph G by performing $N \times \binom{M}{2}$ comparisons. Next, it uses Depth First Search (DFS) to find a directed cycle in G. Once a cycle is found, an edge is removed from G. This step is repeated for a controllable number of iterations T, or until all cycles are removed. Computational complexity of DFS is $O(M^2)$. Hence, the overall time computational complexity of CRAM is $O(TNM^4)$.

CRAM requires information about clients, their PHY rates, and BSs at a central entity (*e.g.* NC). Let m' denote the average number of BSs for each client. In CRAM, each client sends its PHY rate information to its BSs. This step requires $O(m'N)$ overhead. Further, each BS sends this information to the central entity with $O(M)$ overhead. Thus, the total message passing overhead is $O(m'N) + O(M)$.

V. Optimality and Speed

Beyond convergence, we perform theoretical analysis on the performance of DFRA. We first show that under some conditions the outcome of DFRA is optimal. When the conditions are not satisfied, we quantify the distance between the resulting equilibrium and the unique optimal point. We also investigate the convergence time properties of DFRA, and suggest a policy that when deployed would significantly reduce the convergence time.

A. Optimality Gap

We first consider a multi-rate system in which each client has the same PHY rate across different BSs, but this rate is different across clients (*i.e.*, $R_{i,j} = R_i \ \forall i \in \mathbf{N}, j \in \mathbf{M} \mid R_{i,j} \neq 0$). Our next Theorem proves that in such systems only a single equilibrium exists.

***Theorem** 3:* Let $R_{i,j} = R_i \ \forall \ i \in \mathbf{N}$, $j \in \mathbf{M} \mid R_{i,j} \neq 0$. Let service rate h_i of client i be defined as the weighted total throughput of client i (*i.e.*, $h_i = \frac{\sum_{j=1}^{M} \lambda_{i,j} R_{i,j}}{\omega_i}$). Then, there is a unique equilibrium in the system.

The following Corollary follows from the convergence property of DFRA:

Corollary 1: Algorithm DFRA reaches the optimal allocation when the following conditions are satisfied: (i) $R_{i,j} = R_i \ \forall \ i \in \mathbf{N}$, $j \in \mathbf{M} \mid R_{i,j} \neq 0$, and (ii) $h_i = \frac{\sum_{j=1}^{M} \lambda_{i,j} R_{i,j}}{\omega_i}$.

Note that while Theorem 3 presents the conditions that guarantee existence of a unique equilibrium, there maybe an infinite number of resource allocations that realize this unique optimal point. Fig. 11 provides one such example. In this example, $R_{i,j} = 1 \ \forall i \in \mathbf{N}, j \in \mathbf{M}$. When the resource allocation matrix $\mathbf{\Lambda_{opt}}$ in Fig. 11(c) is employed, each client's total throughput would be equal to 1. This can be realized for any value of $\alpha \in [0, 1]$.

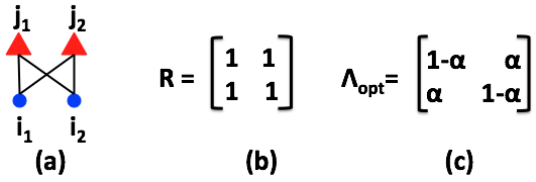


Fig. 11. **There exists a single equilibrium in the above example. Each client at equilibrium reaches a total throughput of 1. However, there could be an infinite number of resource allocations that realize this equilibrium. In the example above, any $\alpha \in [0, 1]$ reaches the optimal equilibrium.**

In generic HetNets, a single client is at different distances from different BSs. Hence, a single client would have heterogeneous rates across different BSs. We next provide bounds on optimality gap for such heterogeneous multi-rate systems.

***Theorem** 4:* Let the service rate of client i be defined as the weighted total throughput (*i.e.*, $h_i = \frac{\sum_{j=1}^{M} \lambda_{i,j} R_{i,j}}{\omega_i}$). Let h_i^{opt} and h_i^{eq} denote the service rate of i at optimality and an equilibrium, respectively. Let R_{max} and R_{min} denote the maximum and non-zero minimum PHY rates across all clients and BSs, respectively. Then the following bounds exist

$$\begin{cases} \frac{R_{min}}{R_{max}} \leq \frac{\min_i(h_i^{eq})}{\min_i(h_i^{opt})} \leq 1 & \text{Min Service Ratio} \\ \frac{R_{min}}{R_{max}} \leq \frac{\sum_{i=1}^{N} \omega_i h_i^{eq}}{\sum_{i=1}^{N} \omega_i h_i^{opt}} \leq \frac{R_{max}}{R_{min}} & \text{Weighted Avg. Service Ratio} \\ \frac{\omega_{min} R_{min}}{\omega_{max} R_{max}} \leq \frac{\sum_{i=1}^{N} h_i^{eq}}{\sum_{i=1}^{N} h_i^{opt}} \leq \frac{\omega_{max} R_{max}}{\omega_{min} R_{min}} & \text{Avg. Service Ratio} \end{cases}$$

B. Convergence Time

Before we can find a bound on the convergence time of DFRA, we need to define a discretization factor on service rates or time resource fractions. This technicality is due to the operation on real numbers.

Consider the example in Fig. 12. Here each non-zero PHY rate is equal to 1 and is shown through a solid line. The service rate (h_i) is defined as the total throughput. In the initial configuration, ($h_{i_1}, h_{i_2}, h_{i_3}$) is equal to (0,0,1). This is achieved by BS j_3 giving all its resources to i_3 and the rest of BSs not yet giving any resources to any of the clients. Now let's assume that during each time slot, a single BS executes DFRA. The BS implementing DFRA and the total throughput achieved by each client are depicted in Fig. 12(b). The system converges to (1,1,1) allocation in limit. However, as shown in Fig. 12(b) due to operation with real numbers the time to convergence is unbounded.

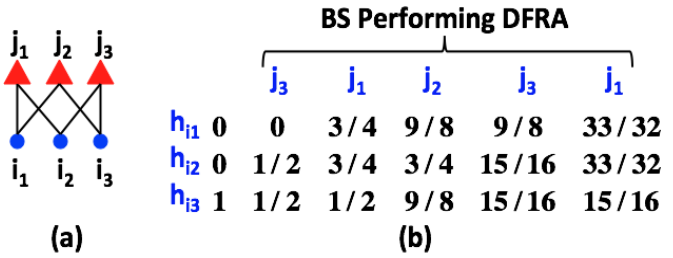


Fig. 12. **Discretization is required to bound the convergence time. The issue is shown here in a topology with 3 BSs and 3 clients. Service rate is defined as total throughput. Each non-zero $R_{i,j}$ is equal to 1 and is shown by a solid line (a). Initially, ($h_{i_1}, h_{i_2}, h_{i_3}$) is equal to (0,0,1). This is achieved by BS j_3 giving all its resources to i_3 and other BSs not yet giving any resources to any of the clients. At each step, a single BS implements DFRA (b). While the allocation converges in limit to (1,1,1), due to the operation with real numbers the time to convergence is unbounded.**

In practice, operations are of course in discretized levels, for example consider the following discretization policy:

Definition 3. *Policy 1:* During service rate equalization by a BS in DFRA, the local minimum service rate must increase by at least a multiplicative factor equal to $1+\eta$. Here $1+\eta$ is the service rate discretization factor.

Based on the above discretization level definition, we can derive the following bound on convergence time:

***Theorem** 5:* Let $h_{max,i}$ and $h_{min,i}$ denote the maximum and non-zero minimum service rates that can be allocated to client i, respectively. Now when Policy 1 is employed, the number of steps that it takes for DFRA to converge is upper bounded by

$$\prod_{i=1}^{N} \lceil \log_{1+\eta}(\frac{h_{max,i}}{h_{min,i}}) \rceil = O(\lceil \log_{1+\eta}(\frac{MN\omega_{max}R_{max}}{\omega_{min}R_{min}}) \rceil^N)$$

By employing appropriate policies that provide an order on when BSs execute DFRA, it is possible to significantly improve upon this bound. Consider the following policy:

Definition 4. *Policy 2:* A BS that serves the client with lowest service rate *across all clients whose service rates can be improved*, has a higher priority for equalization.

***Theorem** 6:* When Policies 1 and 2 are employed, the time to convergence is upper bounded by

$$\sum_{i=1}^{N} \lceil \log_{1+\eta}(\frac{h_{max,i}}{h_{min,i}}) \rceil = O(N \lceil \log_{1+\eta}(\frac{MN\omega_{max}R_{max}}{\omega_{min}R_{min}}) \rceil)$$

Note that the exponent N disappeared in the bound on convergence time in Theorem 6. Policy 2 can be easily implemented in DFRA by making the BS randomization parameter (p_j) inversely proportional to $\min_i h_i \; \forall i \mid R_{i,j} \neq 0$. Further, if a wired backbone exists, each BS that can enhance the service rate of its clients through equalization, can broadcast its min service rate to other BSs. Therefore BSs can distributedly determine an order (or adjust their p_js) based on other BSs' min service rates.

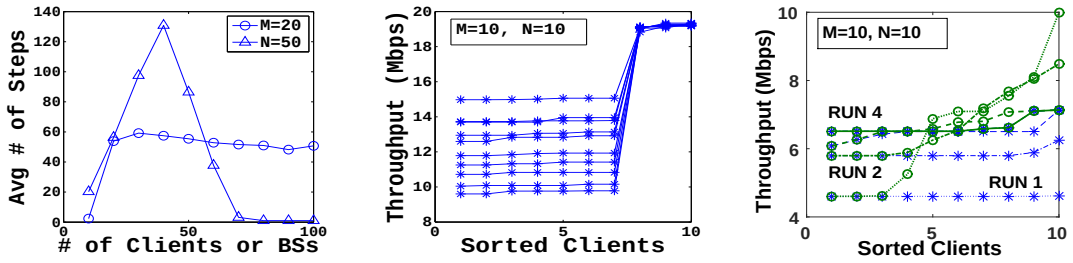
VI. Performance Evaluation

We conducted extensive simulations to study the performance of our proposed algorithms. Due to page limitations, we only present three of the most illuminating results.

Setup. We simulated a network composed of several clients and BSs. Half of the BSs use cellular technology and the other half use WiFi. Each client has access to 4 RATs, 2 cellular and 2 WiFi. The number of clients and BSs can each range from 10 to 100. We set the the discretization factor η equal to 0.02 and use total throughput as our service rate definition. During each simulation realization, clients' RATs are randomly associated to BSs and the corresponding PHY rate is randomly selected from the set $\{1, 2, 5.5, 11\}$ Mbps for WiFi and {5.2, 10.3, 25.5, 51} Mbps for cellular. Next, we run DFRA until convergence is reached. In the initial allocation, each BS equally divides the time resource fractions (λs) across its clients.

DFRA's Convergence Time. Fig. 13(a) depicts the impact of number of clients or BSs on the convergence time. Each data point is the average of 100 realizations. Two graphs are depicted in Fig. 13(a). In the graph with a circular marker, the x-axis denotes the number of clients, and the number of BSs (M) is fixed to 20. In the graph with a triangular marker, the x-axis denotes the number of BSs, and the number of clients (N) is fixed to 50. The results show that the maximum time to convergence happens when the number of clients and BSs are roughly equal. Beyond that, as the ratio between the number of clients and BSs (*i.e.*, $\frac{N}{M}$) decreases or increases, the time to convergence decreases and then rapidly stabilizes. The results show that even without employing Policy 2, DFRA rapidly converges.

DFRA's Throughput. Fig. 13(b) depicts the throughput of each client at equilibrium for ten different outcomes of DFRA. The number of BSs and clients are both set to 10. The clients are sorted based on their throughputs. Each realization starts from the same initial point. Two client groups are observed. Clients in each group roughly achieve the same throughput. The difference between clients' throughput in each group is due to discretization. The results show up to 50% difference in throughput *across the realizations* for the clients in the first group, whereas clients in the second group achieve the same



(a) (b) (c)

Fig. 13. **(a) Average number of steps to convergence in DFRA as a function of: (i) number of clients, with number of BSs (M) fixed to 20, and (ii) number of BSs, with number of clients (N) fixed to 50. (b) Sorted throughput of clients at equilibria for ten realizations of DFRA. Each realization starts from the same initial point. Two client groups exist at each equilibrium. The difference between clients' throughput in each group is due to discretization. There is 50% difference in throughput** *across the realizations* **for the clients in the first group. This motivates the potential for CRAM to look for cyclic shifts of clients' traffic in group 1, and ensure that the highest possible outcome is achieved. (c) Impact of CRAM on DFRA's outcome. Here, blue stars represent DFRA's outcomes whereas green circles denote CRAM's outcomes. Once DFRA reaches an equilibrium, CRAM can intervene and potentially enhance the throughput of some of the clients. In this example, DFRA and CRAM iterate on each others' outcomes for four times until an equilibrium is reached.**

throughput. This motivates the potential for CRAM to look for cyclic shifts of clients' traffic in group 1, and ensure that the highest possible outcome is achieved.

Impact of CRAM on DFRA's Outcome. Fig. 13(c) depicts the combined operation of CRAM and DFRA. Here we consider an example topology with 10 BSs and 10 clients. Four consecutive iterations/runs exist in this example realization. During each run, we first execute the DFRA and allow the system to converge to an equilibrium. Next, CRAM operates on the outcome of DFRA and enhances it by finding all cyclic shifts in clients' traffic. Since CRAM's outcome is not necessarily an equilibrium, DFRA can then operate on the outcome of CRAM and lexicographically increase it.

We plot the sorted clients' throughput for each outcome of DFRA and CRAM. In Fig. 13(c) blue stars represent DFRA's outcomes, whereas green circles denote CRAM's outcomes. In the first outcome of DFRA (RUN 1), a single equilibrium exists and all clients achieve 4.6 Mbps throughput. When CRAM operates on this outcome, three of the clients remain unchanged, however, the rest of the clients substantially increase their throughputs. Since this resulting outcome of CRAM is not an equilibrium, DFRA operates on this outcome of CRAM and lexicographically increases it. This joint operation of DFRA and CRAM happens 4 times. Each time DFRA and CRAM lexicographically increase each others' outcomes. The procedure continues until CRAM cannot enhance DFRA's outcome or vice versa. Fig. 13(c) shows that with 4 iterations, clients increase their throughput from an average of 4.6 Mbps to an average of 6.8 Mbps which is close to 50% increase in throughput. This shows that even a small amount of centralized network supervision can substantially enhance the performance.

VII. Conclusion

We addressed the problem of max-min fair service rate allocation in RAN-level integrated HetNets. We studied the properties of the optimal allocation, and accordingly designed a distributed low complexity algorithm with guaranteed performance. We provided bounds on the convergence time of the distributed solution, and showed that leveraging appropriate policies in its design can significantly reduce the convergence time. We also designed a hybrid computation architecture that enhances the outcome of the distributed solution, and showed that it can adapt to network dynamics as well as the delay between BSs and the central entity.

VIII. Acknowledgements

This research was supported in part by NSF grants CNS 1347234 and CNS 1527513. Authors would like to thank the anonymous reviewers for the useful comments.

References

[1] Access to the 3GPP evolved packet core (EPC) via non-3GPP access networks. In *3GPP technical specifications, 24.302*, 2013.

[2] Study on WLAN/3GPP radio interworking. In *3GPP technical report, 37.834*, 2013.

[3] H. S. an P. Gilbert, N. Banerjee, M. D. Corner, B. N. Levine, and L. Cox. Spider: improving mobile networking with concurrent WiFi connections. In *Proc. of ACM SIGCOMM*, 2011.

[4] S. Andreev, M. Gerasimenko, O. Galinina, Y. Koucheryavy, N. Himayat, S.-P. Yeh, and S. Talwar. Intelligent access network selection in converged multi-radio heterogeneous networks. *IEEE Transactions on Wireless Communications*, 21(6):86–96, Dec 2014.

[5] Y. Bejerano, S.-J. Han, and L. Li. Fairness and load balancing in wireless LANs using association control. *IEEE/ACM Transactions on Networking*, 2007.

[6] S. Deb, K. Nagaraj, and V. Srinivasan. MOTA: engineering an operator agnostic mobile service. In *Proc. of ACM MOBICOM*, 2011.

[7] S. Kandula, K. C. J. Lin, T. Badirkhanli, and D. Katabi. FatVAP: aggregating AP backhaul capacity to maximize throughput. In *Proc. of NSDI*, 2008.

[8] A. Keshavarz-Haddad, E. Aryafar, M. Wang, and M. Chiang. HetNets selection by clients: convergence, efficiency, and practicality. *IEEE/ACM Transactions on Networking*, 2017.

[9] L. Li, M. Pal, and Y. Yang. Proportional fairness in multi-rate wireless LANs. In *Proc. of IEEE INFOCOM*, 2008.

[10] B. Radunovic and J. Y. Le Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 2007.

[11] S. Sarkar and L. Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. In *Proc. of IEEE INFOCOM*, 2000.

[12] S. Shakkottai, E. Altman, and A. Kumar. Multihoming of users to access points in WLANs: a population game perspective. *IEEE Journal on Selected Areas in Communication*, 25(6):1207–1215, 2009.

[13] W. Wang, X. Liu, J. Vicente, and P. Mohapatra. Integration gain of heterogeneous WiFi/WiMAX networks. *IEEE Transactions on Mobile Computing*, 2011.

[14] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews. User association for load balancing in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications*, 12(6):2706–2716, Jun 2013.

APPENDIX

Proof of Theorem 1: Consider client i with minimum service rate and a BS j for which $\lambda_{i,j} > 0$. Then as we discussed in the example of Fig. 2, for every other client i' that has a non-zero PHY rate to BS j (*i.e.*, $R_{i',j} \neq 0$) we have:

$$\begin{cases} \text{if } \ \lambda_{i',j} > 0 & \Rightarrow \quad h_i = h_{i'} \\ \text{if } \ \lambda_{i',j} = 0 & \Rightarrow \quad h_i \leq h_{i'} \end{cases}$$

Let M_i denote the set of BSs which provide non-zero resources (λs) to client i. Now initially set $U_1 = \{i\}$ and $A_1 = \{M_i\}$. Next, if we find any other client i' such that $h_{i'} = h_i$ and $\lambda_{i',j} > 0$ for at least one BS $j \in \{A_1\}$, we add the client (i') to U_1 and the corresponding BS set ($M_{i'}$) to A_1. We repeat this process until no other client or BS can be added to U_1 or A_1, respectively.

Next, we remove the clients in U_1 from the set of all clients ($\mathbf{N}$), and select the client i' with minimum service rate among the remaining clients. Similar to the previous step, we initially set $U_2 = \{i'\}$ and $A_2 = \{M_{i'}\}$ and keep on adding appropriate clients and BSs to U_2 and A_2, until no further client or BS can be added.

We repeat this procedure until all clients and BSs are grouped. The properties described in Theorem 1 follow from our group construction process.

Proof of Theorem 2: Part 1: At each step of DFRA, the service rate lexicographically increases for a subset of clients, while it does not change for others. This lexicographically increases the overall service rate vector across all the clients. As the number of BSs and clients is finite, the service rate vector cannot increase unboundedly and hence would converge to an equilibrium.

Part 2: Consider a resource allocation that results in an equilibrium. Consider client i with minimum service rate and a BS j for which $\lambda_{i,j} > 0$. Then, for every other client i' that has a non-zero PHY rate to BS j (*i.e.*, $R_{i',j} \neq 0$) we have:

$$\begin{cases} \text{if } \ \lambda_{i',j} > 0 & \Rightarrow \quad h_i = h_{i'} \\ \text{if } \ \lambda_{i',j} = 0 & \Rightarrow \quad h_i \leq h_{i'} \end{cases}$$

We can now follow the same steps as in the proof of Theorem 1 and construct BS and client groups that have the properties of Theorem 1.

Proof of Theorem 3: Before we can present and prove the theorem, we need the following lemma:

Lemma 1: Consider a resource allocation that results in an equilibrium. Assume that each client has the same PHY rate across different BSs, but that this rate is different across clients (*i.e.*, $R_{i,j} = R_i \ \forall i \in \mathbf{N}, j \in \mathbf{M} \mid R_{i,j} \neq 0$). Let U_l and A_l denote the 1:1 partitioning of clients and BSs into groups according to the properties described in Theorems 1 and 2. Let h_l be defined as the throughput achieved by clients in group l. Then, $h_l = \frac{|A_l|}{\sum_{k \in U_l} \frac{1}{R_k}}$.

Proof of Lemma 1: Let k denote a client in group U_l. Then, k's throughput is equal to $R_k \sigma_k$ in which $\sigma_k = \sum_{j \in A_l} \lambda_{k,j}$. This is because according to Theorems 1 and 2 the resources for clients in group U_l are only provided by BSs in group A_l.

Further, due to Theorem 2 all clients in the same group achieve the same throughput and hence we have

$$h_l = R_1 \sigma_1 = R_2 \sigma_2 = ... = R_{|U_l|} \sigma_{|U_l|} \tag{7}$$

Now since the resources for clients in group U_l are only provided by BSs in group A_l, we have

$$\sum_{k \in U_l} \sigma_k = |A_l| \tag{8}$$

By leveraging Eq. (7) in Eq. (8) we have

$$h_l \times (\frac{1}{R_1} + ... + \frac{1}{R_{U_l}}) = |A_l| \Rightarrow h_l = \frac{|A_l|}{\sum_{k \in U_l} \frac{1}{R_k}} \tag{9}$$

We are now ready to prove the Theorem:

Let $\{(U_1, A_1), ..., (U_k, A_k)\}$ and $\{(U'_1, A'_1), ..., (U'_{k'}, A'_{k'})\}$ denote the 1:1 partitioning of clients and BSs in the optimal partition and an arbitrary equilibrium, respectively. Assume that all client and BS groups are maximal sets. In other words, we merge client groups (and their corresponding BS groups) that have similar service rates. As a result, clients in different groups would have different service rates. Let γ_k and $\gamma'_{k'}$ denote the service rate of clients in groups k and k' of the optimal and the arbitrary equilibrium, respectively. Assume the following order on service rates

$$\gamma_1 < ... < \gamma_k \ \text{ and } \ \gamma'_1 < ... < \gamma'_{k'} \tag{10}$$

Let i be a client in client group U'_1. Let h'_i and h_i denote the throughput of client i in the arbitrary equilibrium and optimality, respectively. Then, from Lemma 1 we have

$$h'_i = R_i \sigma'_i = \frac{|A'_1|}{\sum_{k \in U'_1} \frac{1}{R_k}} \tag{11}$$

$$h_i = R_i \sigma_i = \frac{|A_1|}{\sum_{k \in U_1} \frac{1}{R_k}} \tag{12}$$

From optimality we have $h'_i \leq h_i$. Assume $h'_i < h_i$. Then from Eq. (11) we have

$$h_i = R_i \sigma_i > \frac{|A'_1|}{\sum_{k \in U'_1} \frac{1}{R_k}} \Rightarrow \sigma_i > \frac{\frac{1}{R_i}|A'_1|}{\sum_{k \in U'_1} \frac{1}{R_k}} \tag{13}$$

Since resources for clients in each group only come from BSs in the corresponding BS group, we have

$$\sum_{k \in U_1} \sigma_k = |A_1| \ \text{ and } \ \sum_{k \in U'_1} \sigma'_k = |A'_1| \tag{14}$$

On the other hand $\sum_{k \in U'_1} \sigma_k \leq |A'_1|$. This is because clients in group U'_1 have zero rates to BSs in groups $A'_2, ..., A'_{k'}$. As a result, sum of resources for such clients even in the optimal allocation cannot be more than $|A'_1|$. Further, if $\sum_{k \in U'_1} \sigma_k < |A'_1|$ then at least a single client would have a higher amount of total resources σ (and hence a higher service rate) in the arbitrary equilibrium allocation compared to the optimal allocation, which is a contradiction. Hence

$$\sum_{k \in U'_1} \sigma_k = |A'_1| \tag{15}$$

Eq. (15) provides a cut-set bound on h_i. In particular, from Eq. (15) it follows that if there exists a client $i \in U'_1$ which has the property of Eq. (13), there must exist another client $i' \in U'_1$ such that

$$\sigma_{i'} < \frac{\frac{1}{R_{i'}}|A'_1|}{\sum_{k \in U'_1} \frac{1}{R_k}} \tag{16}$$

Hence we have

$$h_{i'} = R_{i'}\sigma_{i'} < \frac{|A_1'|}{\sum_{k\in U_1'}\frac{1}{R_k}} = h'_{i'} = h'_i \tag{17}$$

Eq. (17) implies that client i''s service rate in the optimal allocation is less than what it achieves in the arbitrary allocation. Since i' has the minimum service rate in the arbitrary allocation, this is a contradiction. Hence, it follows that h'_i cannot be lower than h_i and we have

$$h_i = h'_i \quad \forall i \in U_1' \tag{18}$$

From Eq. (18) it follows that $U_1' \subseteq U_1$. Now if $U_1 \not\subseteq U_1'$, then the arbitrary equilibrium would be lexicographically greater than the optimal equilibrium. This is because U_1 and U_1' are maximal sets. As a result we also have that $U_1 \subseteq U_1'$, and hence $U_1 = U_1'$.

Now if we set aside clients and BSs in groups U_1/U_1' and A_1/A_1' from the set of clients and BSs, the same argument would apply to U_2/U_2' and A_2/A_2'. Hence, the maximal set groups are equivalent and achieve the same service rates. As a result a single equilibrium would exist.

Proof of Theorem 4: We refer to the non-optimal arbitrary equilibrium as a sub-optimal solution. Let i_1 denote the client with minimum service rate in the sub-optimal solution, *i.e.*, $i_1 \in U_1'$.

First Bound (Minimum Service Ratio): From Theorem 2 we know that all resources of clients in U_1' are provided by BSs in A_1'. Hence we have

$$\begin{aligned} h'_{i_1} = \frac{\sum_{j\in A_1'}\lambda'_{i_1,j}R_{i_1,j}}{\omega_{i_1}} \overset{\text{(I)}}{=} \frac{\sum_{i\in U_1'}\sum_{j\in A_1'}\lambda'_{i,j}R_{i,j}}{\sum_{i\in U_1'}\omega_i} \\ \geq \frac{|A_1'|R_{\min}}{\sum_{i\in U_1'}\omega_i} \end{aligned} \tag{19}$$

Further we have

$$\min_i(h_i) \overset{\text{(II)}}{\leq} \frac{\sum_{i\in U_1'}\sum_{j\in A_1'}\lambda_{i,j}R_{i,j}}{\sum_{i\in U_1'}} \leq \frac{|A_1'|R_{\max}}{\sum_{i\in U_1'}\omega_i} \tag{20}$$

Equation in (I) is due to the following. Let $i_1, ..., i_s \in U_1'$. Then $h_{i_1} = ... = h_{i_s}$, *i.e.*,

$$\frac{\sum_{j\in A_1'}\lambda'_{i_1,j}R_{i_1,j}}{\omega_{i_1}} = ... = \frac{\sum_{j\in A_1'}\lambda'_{i_s,j}R_{i_s,j}}{\omega_{i_s}} \tag{21}$$

$$= \frac{\sum_{i\in U_1'}\sum_{j\in A_1'}\lambda'_{i,j}R_{i,j}}{\sum_{i\in U_1'}\omega_i} \tag{22}$$

Eq. (22) follows from linear algebra. In particular $\frac{x_1}{y_1} = \frac{x_2}{y_2} = ... = \frac{x_n}{y_n}$ is also equal to $\frac{\sum_i x_i}{\sum_i y_i}$.

Equation in (II) is due to the following. Let $i_1, ..., i_s \in U_1'$. Then

$$\begin{aligned} \min_i(h_i) \leq h_{i_1} &= \frac{\sum_{j\in A_1'}\lambda_{i_1,j}R_{i_1,j}}{\omega_{i_1}} \\ &\leq h_{i_2} = \frac{\sum_{j\in A_1'}\lambda_{i_2,j}R_{i_2,j}}{\omega_{i_2}} \\ &... \\ &\leq h_{i_s} = \frac{\sum_{j\in A_1'}\lambda_{i_s,j}R_{i_s,j}}{\omega_{i_s}} \end{aligned} \tag{23}$$

$$\Rightarrow \min_i(h_i) \leq \frac{\sum_{i\in U_1'}\sum_{j\in A_1'}\lambda_{i,j}R_{i,j}}{\sum_{i\in U_1'}\omega_i} \tag{24}$$

Eq. (24) follows from linear algebra. In particular, if $\frac{x_i}{y_i}$ $i = 1, ..., k$ be a set of non-negative real numbers then

$$\min_i \frac{x_i}{y_i} \leq \frac{\sum_i x_i}{\sum_i y_i} \tag{25}$$

From Eqs (19) and (20) we have

$$\frac{h'_{i_1}}{\min_i(h_i)} = \frac{\min_i(h'_i)}{\min_i(h_i)} \geq \frac{\frac{|A_1'|R_{\min}}{\sum_{i\in U_1'}\omega_i}}{\frac{|A_1'|R_{\max}}{\sum_{i\in U_1'}\omega_i}} = \frac{R_{\min}}{R_{\max}} \tag{26}$$

Second Bound (Weighted Average Service Ratio):
For the optimal allocation we have

$$\sum_{i=1}^{N}\omega_i h_i = \sum_{i=1}^{N}\sum_{j=1}^{M}\lambda_{i,j}R_{i,j} = \sum_{j=1}^{M}\sum_{i=1}^{N}\lambda_{i,j}R_{i,j} \tag{27}$$

Further, for both the optimal and sub-optimal allocation we have $\sum_{i=1}^{N}\lambda_{i,j} = 1\ \forall j \in \mathbf{M}$. Therefore

$$\begin{aligned} R_{\min}M = R_{\min}\sum_{j=1}^{M}\sum_{i=1}^{N}\lambda_{i,j} \leq \sum_{j=1}^{M}\sum_{i=1}^{N}\lambda_{i,j}R_{i,j} \leq \\ R_{\max}\sum_{j=1}^{M}\sum_{i=1}^{N}\lambda_{i,j} = R_{\max}M \end{aligned} \tag{28}$$

The same bound can also be found for $\sum_{i=1}^{N}\omega_i h'_i$, therefore

$$\frac{R_{\min}}{R_{\max}} \leq \frac{\sum_{i=1}^{N}\omega_i h'_i}{\sum_{i=1}^{N}\omega_i h_i} \leq \frac{R_{\max}}{R_{\min}} \tag{29}$$

Third Bound (Average Service Ratio): We follow the same steps as in the second step. In particular,

$$\sum_{i=1}^{N}h_i = \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\lambda_{i,j}R_{i,j}}{\omega_i} \Rightarrow \tag{30}$$

$$\begin{aligned} \frac{R_{\min}}{\omega_{\max}}M = \frac{R_{\min}}{\omega_{\max}}\sum_{i=1}^{N}\sum_{j=1}^{M}\lambda_{i,j} \leq \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\lambda_{i,j}R_{i,j}}{\omega_i} \leq \\ \frac{R_{\max}}{\omega_{\min}}\sum_{i=1}^{N}\sum_{j=1}^{M}\lambda_{i,j} = \frac{R_{\max}}{\omega_{\min}}M \end{aligned} \tag{31}$$

We can derive a similar bound for $\sum_{i=1}^{N}h'_i$ and divide the two to reach the third bound.

Proof of Theorem 5: We assume that each client has a non-zero PHY rate to at least one BS and that the minimum service rate that is allocated to each client ($h_{\min,i}$) is at least equal to $\frac{R_{\min}}{N\omega_{max}}$.

We discretize $\log_{1+\eta}\frac{h_i}{h_{\min,i}}$ to integer levels. In particular, we define discretization value of client i (g_i) as follows

$$g_i = \begin{cases} 0 & \textbf{if } \frac{h_i}{h_{\min,i}} < 1+\eta \\ \lfloor \log_{1+\eta}\frac{h_i}{h_{\min,i}} \rfloor & \textbf{else} \end{cases}$$

If at some point i has the local minimum service rate, then because of Policy 1 its corresponding discrete value would increase by at least one after equalization by one of the BSs to which i has a non-zero PHY rate.

Now let $\mathbf{g}$ be the vector of sorted g_i values, *i.e.*, with abuse of notation $\mathbf{g} = (g_1, ..., g_N)$ in which $g_1 \leq g_2 \leq ... \leq g_N$. At each equalization step, $\mathbf{g}$ lexicographically increases. Further, each g_i can

take an integer value that belongs to [0, $\lfloor \log_{1+\eta} \frac{h_{\max,i}}{h_{\min,i}} \rfloor$]. Thus the total number of distinct values that $\mathbf{g}$ can take is upper bounded by

$$(1 + \lfloor \log_{1+\eta} \frac{h_{\max,1}}{h_{\min,1}} \rfloor) \times ... \times (1 + \lfloor \log_{1+\eta} \frac{h_{\max,N}}{h_{\min,N}} \rfloor) \quad (32)$$

In the worst case, $h_{\max,i} = \frac{MR_{\max}}{\omega_{\min}}$ which results in $O(\lfloor \log_{1+\eta}(\frac{MN\omega_{max}R_{max}}{\omega_{min}R_{min}}) \rfloor^N)$ bound on convergence time.

Note: if we define the discretization vector $\mathbf{g}$ over BSs instead of clients, where g_i is defined as the minimum service rate at BS j, then we can follow the above steps and find an $O(\lfloor \log_{1+\eta} \frac{MNR_{\max}\omega_{\max}}{R_{\min}\omega_{\min}} \rfloor)$ bound on convergence time. This is a better bound when $M < N$.

Proof of Theorem 6: Our proof is based on the observation that when policies 1 and 2 are employed, a client with minimum service rate increases its service rate in a manner that none of the previous min service rate clients decrease their value beyond a certain threshold. As a result, client and BS groups (*i.e.*, U_k and A_k) would be automatically constructed from lowest to highest service rates (*i.e.*, first U_1, A_1, next U_2, A_2, ...).

In particular, let i be the client with *minimum service rate*. After equalization, client i's service rate increases from h_i to at least $(1 + \eta)h_i$. Further, during future equalizations on other clients, client i's service rate would never drop below $(1 + \eta)h_i$. To see this, assume that client i' becomes the minimum service rate client at a future step (with current service rate $h_{i'}$) and that equalization on i''s service rate causes i's service rate to drop below $(1+\eta)h_i$. This can happen only if i''s service rate after equalization (*i.e.*, $h_{i'}$(new)) is less than $(1 + \eta)h_i$. Thus we would have

$$(1 + \eta)h_{i'} \leq h_{i'}(\text{new}) < (1 + \eta)h_i \Rightarrow h_{i'} < h_i \quad (33)$$

But this is a contradiction because h_i had the minimum service rate in the past and local equalization equalization always lexicographically increases the service rate vector (Theorem 2). Hence, none of the clients can get a future service rate that drops below h_i. Thus if we discretize all the service rates similar to the proof of Theorem 6, then the number of times that client i would have the minimum service rate and would be improved by equalization, would be at most equal to $1 + \lfloor \log_{1+\eta}(\frac{h_{\max,i}}{h_{\min,i}}) \rfloor$. The upper bound is then obtained by considering all the clients.