Distributed Alpha-Fair Throughput Aggregation in Multi-RAT Wireless Networks

Ehsan Aryafar\(^1\) and Alireza Keshavarz-Haddad\(^2\)
\(^1\)Portland State University, Department of Computer Science, Portland, OR, USA
\(^2\)Shiraz University, School of Electrical and Computer Engineering, Shiraz, Iran

Abstract—Today’s wireless devices can be simultaneously connected to multiple communication networks based on different radio access technologies (RATs) such as WiFi, 3G, and LTE. Simultaneous aggregation of each client’s traffic across multiple such RATs or the corresponding base stations (BSs) can substantially improve the quality of experience for each client and improve the overall network utilization and efficiency.

Our goal in this paper is to design distributed resource allocation algorithms that can be independently executed by each BS and achieve alpha-fairness among the clients. In particular, we derive a simple water filling based solution and study its theoretical aspects such as convergence, optimality, evolution, and convergence time. We also characterize its performance through a comprehensive multi-RAT simulator, and show that it achieves superior performance to alternative distributed solutions due to its fast and low-overhead operation.

I. INTRODUCTION

The explosive growth in the number of wireless devices and emerging bandwidth intensive applications (e.g., augmented reality) has led the wireless industry to develop 5G New Radio (NR) technologies for different parts of the wireless spectrum, including sub-1 GHz, 1-6 GHz, and mmWave bands [1]. These new wireless technologies will augment existing heterogeneous cellular networks (HetNets), which already leverage a wide variety of RATs (e.g., 3G, LTE, and WiFi) to serve their clients. In parallel, client device (e.g., smartphone) manufacturers are also equipping their devices with an increasing number of RATs. Simultaneous aggregation of each client’s traffic across multiple such RATs can substantially enhance the client performance, e.g., boost its capacity, increase its resiliency when one RAT fails or becomes congested, and enhance its mobile operation (e.g., a mobile client can continue to receive traffic through LTE while switching its WiFi radio).

A. Related Work

To realize the performance gains associated with multi-RAT HetNets, several standardization organizations have been developing multi-RAT integration solutions. For example, IETF (Internet Engineering Task Force) Internet area working group is developing a multi-access management protocol to support multi-RAT combination on the core network side [2]. On a parallel front, the 3GPP has already standardized several PHY layer aggregation (e.g., LTE-U, LAA [3]) and MAC layer aggregation (e.g., LWA [4]) techniques. More recently, the 3GPP is standardizing different methods to support 5G NR access to unlicensed bands as well as mechanisms to support MAC layer aggregation across any combination of RATs (e.g., 3G, WiFi, LTE, and NR) [5]. These efforts in totality will support multi-RAT traffic aggregation at different network layers (including PHY, MAC, transport, and application layers) with diverse performance tradeoffs.

We consider MAC-level aggregation as it is generally more efficient than transport/application layer aggregation solutions due to the availability of instantaneous channel information at the MAC layer. It can also be applied to existing HetNets, whereas majority of consumer and network side wireless equipment do not yet support PHY layer aggregation.

In particular, our goal is to design a distributed resource allocation algorithm for each base station (BS)\(^1\) such that the total throughput achieved by each client across its RATs satisfies \(\alpha\)-fairness. Alpha-fairness \(\alpha\) is a unifying mathematical formulation to achieve fair throughput assignment. The degree of fairness is defined by a parameter \(\alpha \in [0, \infty)\), which controls the tradeoff between fairness and efficiency (i.e., total throughput maximization). Several special cases of \(\alpha\) correspond to well-known fairness metrics, e.g., \(\alpha \to \infty\) corresponds to max-min fair allocation (which may be considered as the most fair allocation), \(\alpha = 2\) corresponds to delay minimization, \(\alpha \to 1\) corresponds to proportional fairness, \(\alpha = \frac{1}{2}\) corresponds to harmonic fairness, and \(\alpha = 0\) corresponds to throughput maximization (without any fairness consideration). Alpha-fairness has been applied to a wide variety of networking problems, but this is the first paper to propose algorithms that achieve it in HetNets.

Several research works have shown the performance gains associated with multi-RAT traffic aggregation, e.g., in the case of LTE-WiFi aggregation [7]. However, none of these works has considered fairness among the clients. In our recent work [8], we presented a water filling based algorithm that can achieve proportional fairness. This paper builds on the algorithm in [8] by showing that water filling-based algorithms can also enforce generic \(\alpha\)-fairness, presents the theoretical underpinning of the water filling solution (Theorem 1) and its connection to game theory, introduces new proof approaches to analyze algorithm convergence, optimality, and speed, and uses a comprehensive simulator (calibrated with 3GPP specifications) to demonstrate its practical performance.

B. Research Contributions

Our key contributions can be summarized as follows:

\(^1\)We use BS as a generic term that refers to NB in 3G, eNB in LTE, AP in WiFi, and gNB in 5G.
- **Algorithm Design**: We first study the best resource allocation policy for each BS (assuming that other BSs do not change their resources) to gain intuition for our problem. We show that if a BS wants to unilaterally maximize the system objective, it should allocate its resources according to an \( \alpha \)-dependent water filling operation [Theorem 1]. We use the learnings from this observation to design a distributed water filling algorithm (named “WFRA”) for generic HetNets.

- **Theoretical Analysis**: We study the theoretical aspects of WFRA. We show that as BSs autonomously execute it, the system converges to an equilibrium [Theorem 2]. We next characterize some of the useful properties of the equilibria [Lemma 1] and prove that any outcome of WFRA is globally optimal [Theorem 3]. We also show that while the vector of optimal throughput allocations across all the clients is unique, there could be infinitely many resource allocations that can realize them [Proposition 1]. We finally derive tight bounds on WFRA’s convergence time [Theorem 4] and show that while WFRA results in an optimal allocation for any finite value of \( \alpha \), the resulting outcome is not necessarily optimal as \( \alpha \to \infty \).

- **Performance Evaluation**: We use a comprehensive multi-RAT simulator to quantify WFRA’s performance. We show that our algorithm converges to the optimal solution within a few iterations. We also compare its performance against alternative distributed solutions and show that WFRA can get very close to the ideal fair outcome even in the presence of channel fluctuations and packet losses.

The paper is organized as follows. We formulate our problem and discuss the details of our algorithm in Section II. We discuss the theoretical aspects of WFRA in Section III. We present the results of our extensive simulations in Section IV. Finally, we conclude the paper in Section V. Due to page limitations, we (i) limit the discussion of related work, (ii) omit proofs of all lemmas, and (iii) omit comparisons against DComp, an alternative distributed algorithm based on well-known dual decomposition techniques. These details are provided in the technical report [9].

II. SYSTEM MODEL

We first present our network model and problem formulation. Next, we derive a theorem that provides us with necessary intuition to design a distributed resource allocation algorithm for each BS. Finally, we present the details of our algorithm.

**A. Network Model**

We consider a HetNet which is composed of a set of BSs \( M = \{1, ..., M\} \), and a set of clients \( N = \{1, ..., N\} \). Each BS has a limited transmission range and only serves the clients within its range. Each client has a specific number of RATs, and therefore has access to a subset of BSs\(^2\). We assume that clients split their traffic over their RATs and focus on the traffic splitting problem for each client. It is itself a challenging problem to associate each client’s RAT to one of the corresponding 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 association rule could be any load balancing algorithm (e.g., [10]), or based on the received signal strength, among others.

Similar to [10], [11], 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/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 (denote by \( r_i \)) would be equal to \( \sum_{j=1}^{M} \lambda_{i,j} R_{i,j} \).

**B. Problem Formulation**

Our objective is to design a distributed resource allocation algorithm for each BS such that the vector of total throughput values across all the clients \( (r_1, r_2, ..., r_N) \) satisfies \( \alpha \)-fairness. Specifically, in this paper we derive a generic algorithm and analyze its performance assuming \( \alpha \) is a number between 0 and 1, or a number greater than 1. We discuss the theoretical aspects of our algorithm for \( \alpha = 0 \to 1 \), and \( \to \infty \) in Section III-D. Mathematically, our problem can be formulated as

\[
P_1 : \max f(\lambda) = \sum_{i=1}^{N} \omega_i U_\alpha(r_i) \\
\text{s.t.} \quad r_i = \sum_{j=1}^{M} \lambda_{i,j} R_{i,j} \quad \forall i \in N \\
U_\alpha(r_i) = \frac{r_i^{1-\alpha}}{1-\alpha} \quad \forall i \in N \\
\sum_{i=1}^{N} \lambda_{i,j} \leq 1 \quad \forall j \in M
\]

variables: \( \lambda_{i,j} \geq 0 \quad \forall i \in N, j \in M \)

input: \( \alpha \in (0, 1) \) or \( \alpha > 1 \)

Here, \( \omega_i \) is a positive number that denotes client \( i \)'s weight or priority. The second constraint captures the definition of

\( \alpha \)-fairness.

---

\(^2\)Model clients that can aggregate traffic across same technology BSs with multiple such RATs. For example, the LTE Dual Connectivity architecture allows an LTE client to connect to two eNBs that are on different frequencies, and simultaneously utilize the radio resources that belong to both of them. Thus, we model such a client with two LTE RATs.

\(^3\)In Section IV, we discuss how we can extend our model and algorithm to capture practical issues such as WiFi contention and channel dynamics.
\(\alpha\)-fairness and the third constraint ensures that the sum of time fractions at each BS is less than 1. The feasible region in problem \(P_1\) is 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. Since the objective function is also concave, it follows that the \(\alpha\)-fair allocation vector (i.e., \(r^{opt} = (r_1, \ldots, r_N)\)) exists and is unique.

Thus, problem \(P_1\) can be solved optimally in a centralized manner by using solvers such as CVX [12] in \(O(max(\mathcal{N}^3, \mathcal{N}^2M))\) computational complexity. However, a centralized solution requires information regarding the entire topology (i.e., \(R_{i,j}\)s and \(\lambda_{i,j}\)s between every BS and its clients) for every scheduling interval in order to solve the problem. This significantly increases the problem complexity (particularly as the network size increases) and also introduces significant communication overhead between every BS and the central entity. Further, HetNet operators use a wide variety of backhauling technologies with diverse delay and capacity characteristics between different BSs and the central entity. The backhauling constraints limit the amount and freshness of the information that can be passed between the BSs and the central entity. The backhauling delay (e.g., roundtrip latencies of 28 ms in cable and 62 ms in DSL [13]) also make the centralized solution not adaptable to system dynamics. These challenges in HetNets necessitate design of distributed resource allocation algorithms that can be independently executed by each BS.

C. Theoretical Underpinnings of the Water Filling Operation

There are two approaches to design a distributed resource allocation algorithm for problem \(P_1\) that can be autonomously executed by each BS. One approach, is to use dual decomposition [14] to derive a distributed algorithm\(^4\). The resulting algorithm converges to the optimal solution; however, it cannot be easily implemented in practice, and it incurs long convergence time and significant over-the-air (OTA) message passing overhead between every BS and its clients [9]. The second approach, is to study the optimal action by each BS in isolation to gain intuition for distributed algorithm design\(^5\). In particular, in our first theorem we show that if a single BS (e.g., \(j\)) wants to unilaterally maximize \(f(\lambda)\) (i.e., the objective function in \(P_1\)), then it must allocate its resources according to a water filling strategy. We use the intuition gained from this result to design a distributed resource allocation algorithm for HetNets in Section II-D. We show that the resulting algorithm still converges to the optimal outcome (Section III-B), but much faster and with far less OTA overhead than the dual decomposition based solution [9]. Our first theorem shows this water filling operation:

\[\textbf{Theorem 1} \quad \text{Let } \mathbf{n}_j = \{1, \ldots, n\} \text{ denote the set of clients with non-zero PHY rate to BS } j \text{ (i.e., } \forall i \in \mathbf{n}_j, R_{i,j} > 0). \text{ Let } r_i^j \text{ denote the total throughput of client } i \text{ from all BSs other than } j. \text{ Then, if BS } j \text{ wants to unilaterally maximize } f(\lambda) \text{ through allocation of its time resources (i.e., assuming no other BS changes its } \lambda_{i,j}\text{s), it must choose its } \lambda_{i,j}\text{s such that the following conditions are satisfied:}
\]

\[A \quad \forall i \in \mathbf{n}_j \mid \lambda_{i,j} > 0, \text{ then } r_i^j + \lambda_{i,j} R_{i,j} \geq \theta_j \]

\[B \quad \forall i' \in \mathbf{n}_j \mid \lambda_{i',j} = 0, \text{ then } \theta_j \leq r_{i'}^j \left(\frac{1}{\omega_{i',j}}\right) \]

\[C \quad \sum_{i=1}^{n} \lambda_{i,j} = 1, \quad \lambda_{i,j} > 0 \quad \forall i, j
\]

Here, \(\theta_j\) is the water fill level at BS \(j\). Condition A implies that every client that receives time resources from \(j\) should reach the same water fill level at the BS. Condition B implies that any client (e.g., \(i'\)) that does not get any time resources from \(j\), should have a higher \(r_{i'}^j \) than the water fill level at \(j\) (i.e., \(\theta_j\)). Fig. 2 shows this water filling operation.

\[\text{Proof: Let } \lambda_{i,j}^*, \lambda_{i,j}^*, \ldots, \lambda_{i,j}^* \text{ denote the optimal time fractions at BS } j \text{ that maximize } f(\lambda), \text{ assuming no other BS changes its time fractions. Now, if } \lambda_{i,j}^* > 0, \text{ then for any } i' \in \mathbf{n}_j \text{ we must have}
\]

\[f(\lambda) \mid \lambda_{i,j}^*, \ldots, \lambda_{i,j}^*, \lambda_{i',j}^* - \epsilon, \ldots, \lambda_{i,j}^* + \epsilon, \ldots \geq f(\lambda) \mid \lambda_{i,j}^*, \ldots, \lambda_{i,j}^*, -\epsilon, \ldots, \lambda_{i,j}^* + \epsilon, \ldots
\]

where \(\epsilon\) is a small positive number. Leveraging the definition of \(f(\lambda)\) from \(P_1\) (i.e., \(f(\lambda) = \sum_{i=1}^{N} \omega_i U(r_i)\)) in the above inequality and canceling the common terms we have

\[\omega_i U(r_i^j + \lambda_{i,j}^* R_{i,j}) + \omega_i U(r_{i'}^j + \lambda_{i',j}^* R_{i',j}) \geq \omega_i U(r_i^j + (\lambda_{i,j}^* - \epsilon) R_{i,j}) + \omega_i U(r_{i'}^j + (\lambda_{i',j}^* + \epsilon) R_{i',j})
\]

Therefore, if \(\epsilon \to 0\) we would have

\[\omega_i \frac{\partial U(r_i^j + \lambda R_{i,j})}{\partial \lambda} \mid_{\lambda = \lambda_{i,j}^*} \geq \omega_i \frac{\partial U(r_{i'}^j + \lambda R_{i',j})}{\partial \lambda} \mid_{\lambda = \lambda_{i',j}^*}
\]

---

\(\textbf{TABLE I}

\textbf{MAIN NOTATION}

\(M\) and \(N\): Set and number of all clients in the network

\(R_{i,j}^j\): PHY rate of client \(i\) to BS \(j\)

\(R_{max}^j (R_{min}^j)\): Maximum (non-zero minimum) \(R_{i,j}\) across all \(i, j\)

\(\lambda_{i,j}\): Fraction of time allocated to client \(i\) by BS \(j\)

\(r_{i,j}\): Total throughput of client \(i\) across all its RATs

\(\omega_i\): A positive number that represents client \(i\)’s weight or priority

\(\omega_{max} (\omega_{min})\): Maximum (non-zero minimum) \(\omega_i\) across all clients

\(\theta_j\): Water fill level at BS \(j\)
Clients that do not receive time resources would have a higher clients that receive time resources reach the same water fill level (Fig. 2. A single associated client can use all the BS’s resources. That a client RAT is at most associated to one BS. Hence, even finally, property (across all its RATs), whenever there is a change in that. Thus, the total algorithm complexity is from the sorting of the clients, which can be implemented single client by multiple BSs (Line 5). Throughput from all BSs other than is the algorithm: (i) clients are first sorted based on their total (e.g., ). Finding the water fill level and the corresponding s for each client associated with it; and (iii) a randomization parameter is introduced to limit concurrent resource adaptation of a single client by multiple BSs (Line 5).

The key computational complexity of the algorithm arises from the sorting of the clients, which can be implemented in complexity (denotes the number of clients associated to BS ). Finding the water fill level and the corresponding s can be achieved in complexity [8]. Thus, the total algorithm complexity is .

The only OTA overhead introduced by the algorithm is that each client announces to its associated BSs its total throughput (across all its RATs), whenever there is a change in that.

\[ \Rightarrow \omega_i R_{i,j} (r_i^\ast + \lambda_{i,j} R_{i,j}) \leq \omega_f R_{i,j} (r_i^\ast + \lambda_{i,j} R_{i,j}) \]

\[ \quad \Rightarrow r_i^\ast + \frac{\lambda_{i,j}^\ast R_{i,j}}{\omega_i R_{i,j}^\ast} \leq \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_f R_{i,j}^\ast} \]

\[ \text{If we also have that } \lambda_{i,j}^\ast > 0, \text{ then we would have} \]

\[ \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_i R_{i,j}^\ast} \leq \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_f R_{i,j}^\ast} \]

\[ \text{and hence, from Eqs. (1) and (2) we would have} \]

\[ \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_i R_{i,j}^\ast} = \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_f R_{i,j}^\ast} = \theta_j \]

On the other hand, if \( \lambda_{i,j}^\ast = 0 \), from Eq. (1) we would have

\[ \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_i R_{i,j}^\ast} = \theta_j \leq \frac{r_i^\ast + \lambda_{i,j}^\ast R_{i,j}}{\omega_f R_{i,j}^\ast} = \frac{r_i^\ast}{\omega_f R_{i,j}^\ast} \]  

Finally, property (C) is because BS \( j \) can always increase \( f(\lambda) \) by giving its unused time resources to its clients. Note that a client RAT is at most associated to one BS. Hence, even a single associated client can use all the BS’s resources.

Fig. 2. Water filling with three clients. BS \( j \) allocates its \( \lambda_{i,j} \) such that clients that receive time resources reach the same water fill level (\( \theta_j \)). Clients that do not receive time resources would have a higher \( r_i \) divided by \( \omega_i R_{i,j}^\ast \) than \( \theta_j \). Here, client 2 is one such example (\( \lambda_{2,j} = 0 \)).

D. Distributed Resource Allocation in HetNets

We extend the water filling observation in Theorem 1 to design an autonomous resource allocation algorithm in HetNets. Algorithm WFRA depicted in Fig. 3 summarizes the steps that are executed by each BS (e.g., \( j \)). There are three main steps in the algorithm: (i) clients are first sorted based on their total throughput from all BSs other than \( j \) (Line 3); (ii) the BS finds the water fill level and the corresponding \( \lambda_{i,j} \) for each client associated with it; and (iii) a randomization parameter is introduced to limit concurrent resource adaptation of a single client by multiple BSs (Line 5).

The key computational complexity of the algorithm arises from the sorting of the clients, which can be implemented in complexity (denotes the number of clients associated to BS \( j \)). Finding the water fill level and the corresponding \( \lambda_{i,j} \) can be achieved in complexity [8]. Thus, the total algorithm complexity is .

The only OTA overhead introduced by the algorithm is that each client announces to its associated BSs its total throughput (across all its RATs), whenever there is a change in that.

Algorithm WFRA: Water Filling Resource Allocation

**Input:** \( r_i, R_{i,j} \), and \( \lambda_{i,j} \) \( \forall \) client \( i \) for which \( R_{i,j} > 0 \)

**Output:** Updated \( \lambda_{i,j} \)

1. Let \( n \) denote the number of clients \( s.t. R_{i,j} > 0 \)
2. Find the throughput of each client \( i \) provided to it by all other BSs (i.e., \( r_i^\ast \))
3. Sort clients based on their throughput from other BSs divided by their weight times PHY rate, i.e.,
   \[ \frac{r_1'}{w_{1,R_{1,j}}} \leq \frac{r_k'}{w_{k,R_{k,j}}} \leq \frac{r_{k+1}'}{w_{k+1,R_{k+1,j}}} \leq \cdots \leq \frac{r_n'}{w_{n,R_{n,j}}} \]
4. Run water fill, i.e., find \( \theta_j \) and \( \lambda_{i,j} \) for each client
5. If (\( \text{rand} < p_j \)) then Update \( \lambda_{i,j} \)

**III. CONVERGENCE, OPTIMALITY, AND SPEED OF WFRA**

In this section, we investigate the convergence properties of WFRA. We first show that as BSs autonomously execute WFRA, the system converges to an equilibrium. Next, we prove that this equilibrium is optimal. Finally, we derive tight upper bounds on WFRA’s convergence time.

A. Convergence to an Equilibrium

We first present a formal definition of an equilibrium.

**Definition 1 Equilibrium:** The vector of time fractions across all the BSs and clients is an equilibrium outcome if none of the BSs can increase its water fill level through unilateral change of its time resource allocations.

Our next theorem guarantees the convergence of WFRA.

**Theorem 2** Let each BS autonomously execute WFRA. Then, the system converges to an equilibrium.

**Proof:** Consider \( f(\lambda) \) as defined in problem \( P_1 \). Let at each time slot a single BS execute the water filling algorithm. From Theorem 1, it follows that each time a BS makes a change to its resource allocations it increases \( f(\lambda) \). Since the number of clients, BSs, and all the PHY rates \( (R_{i,j}) \) are all finite numbers, \( f(\lambda) \) would be upper bounded. Since at each time slot \( f(\lambda) \) increases and it is a bounded function, the value of \( f(\lambda) \) must converge. Let \( r^t = (r_1^t, ..., r_N^t) \) denote the clients’ throughput sequence corresponding to the evolution of \( f \) over time. We will later show in Theorem 3 that any equilibrium outcome of WFRA converges to the same point. Hence, any converging subsequence of \( r \) converges to the same point. Now, leveraging the fact that \( f(.) \) is a continuous function, all \( r_i \)s are bounded, and \( r \) is compact, this results in the convergence of the \( r^t \) sequence to an equilibrium i.e., \( r^t \rightarrow r^eq = (r_1^eq, ..., r_N^eq) \). Since \( i \in \mathbb{N} \), \( r_i \) converges to an equilibrium \( r_i^eq \), and we allow only one of the BSs change its \( \lambda_{i,j} \)s, therefore the changes on \( \lambda_{i,j} \) must converge to zero, i.e., \( \lambda_{i,j} \rightarrow 0 \). This also results in the convergence of the water fill levels at each BS, i.e., \( \theta_j \rightarrow \theta_j^eq \).
B. Optimality and Other Properties of the Equilibria

In Theorem 2, we proved that as BSs autonomously execute WFRA, the system converges to an equilibrium. In this section, we first derive some useful properties of the equilibria and prove that the optimal solution ($r^{\text{opt}}$) is also an equilibrium. Next we prove that any equilibrium outcome ($r^{eq}$) is also optimal. Finally, we show that while $r^{eq}$ is unique, there could be infinitely many resource allocations (i.e., $\lambda_{i,j}$s) that can realize it. We start by analyzing the equilibria properties.

**Lemma 1** Consider an equilibrium outcome of WFRA. Let $r_{i}^{eq}$, $\theta_{i}^{eq}$, and $\lambda_{i,j}^{eq}$, denote the corresponding throughput of client $i$, water fill level at BS $j$, and the fraction of time allocated to client $i$ by BS $j$, respectively. Then

1. $\omega_{i}r_{i}^{eq} \leq \frac{1}{(\theta_{i}^{eq})^{\alpha}} \forall i \in \mathbb{N}, j \in \mathbb{M}$
2. $\sum_{i=1}^{N} \lambda_{i,j}^{eq} = 1 \forall j \in \mathbb{M}$
3. $\sum_{i=1}^{N} \omega_{i}r_{i}^{eq} \cdot (1-\alpha) = \sum_{j=1}^{M} \frac{1}{(\theta_{i}^{eq})^{\alpha}}$
4. The optimal outcome is also an equilibrium. Hence, we can re-write properties 1-3 by replacing all $r_{i}^{eq}$, $\theta_{i}^{eq}$, and $\lambda_{i,j}^{eq}$ with $r_{i}^{opt}$, $\theta_{i}^{opt}$, and $\lambda_{i,j}^{opt}$, respectively.

**Proof:** We use the properties in Theorem 1 as well as contradiction methods to derive all the properties [9].

In our next theorem, we prove that any equilibrium is the unique optimal solution to problem $P_1$.

**Theorem 3** Let $r^{eq} = (r_{1}^{eq}, \ldots, r_{N}^{eq})$ be an equilibrium and $r^{opt} = (r_{1}^{opt}, \ldots, r_{N}^{opt})$ be an optimal solution to problem $P_1$. Then, $r^{eq} = r^{opt}$.

**Proof:** Leveraging Lemma 1 we can write:

\[
\sum_{i} \frac{\omega_{i}r_{i}^{eq}}{(r_{i}^{eq})^{\alpha}} = \sum_{i,j} \omega_{i}r_{i}^{eq}\lambda_{i,j}^{eq} \leq \sum_{i,j} \frac{\lambda_{i,j}^{eq}}{(\theta_{i}^{eq})^{\alpha}} = \sum_{j} \frac{1}{(\theta_{i}^{eq})^{\alpha}} \sum_{i} \omega_{i}(r_{i}^{eq})^{1-\alpha}
\]

By taking similar steps to Eq. (5) and replacing all equilibrium variables with optimal ones and vice versa, we have

\[
\sum_{i} \frac{\omega_{i}r_{i}^{opt}}{(r_{i}^{opt})^{\alpha}} = \sum_{i,j} \omega_{i}r_{i}^{opt}\lambda_{i,j}^{opt} \leq \sum_{i,j} \frac{\lambda_{i,j}^{opt}}{(\theta_{i}^{opt})^{\alpha}} = \sum_{j} \frac{1}{(\theta_{i}^{opt})^{\alpha}} \sum_{i} \omega_{i}(r_{i}^{opt})^{1-\alpha}
\]

By multiplying the right sides and left sides of Eqs. (5) and (6), and subtracting the left side from the right side we have

\[
\sum_{i} \omega_{i}(r_{i}^{opt})^{1-\alpha} - \sum_{i} \omega_{i}(r_{i}^{eq})^{1-\alpha} = \sum_{i} \omega_{i}r_{i}^{opt} \sum_{i} \frac{\omega_{i}r_{i}^{opt}}{(r_{i}^{opt})^{\alpha}} \geq 0 \implies \sum_{k,l} \omega_{k}\omega_{l}
\]

Finally, while $r^{eq}$ is unique, we prove that there could be infinitely many resource allocations ($\lambda_{i,j}$s) to realize it.

**Proposition 1** Consider problem $P_1$ and let $r^{opt}$ denote the unique optimal outcome. Then, there could be infinitely many possible resource allocations to realize $r^{opt}$.

**Proof:** Our proof is based on an example. Consider a HetNet topology with two BSs and two clients. Let $\omega_{i} = 1 \forall i$ and $R_{i,j} = 1 \forall i,j$. Then, $\lambda_{1,1} = \lambda_{2,2} = \beta$ and $\lambda_{1,2} = \lambda_{2,1} = 1 - \beta$ results in the optimal allocation (i.e., $r_1 = r_2 = 1$) for any value of $\beta \in [0,1]$ and any desired $\alpha$.

C. Bounds on the Objective Function and Convergence Time

In this section, we analyze the objective function ($f(\lambda)$) of problem $P_1$. We first derive upper and lower bounds on $f(\lambda)$. The bounds are useful for various purposes, e.g., a HetNet operator could use these bounds and the current state of the network to estimate the gap from optimality and optimize network operation as needed (e.g., aggregate some of calculations across BSs with better backhauling between them). We can also use these bounds to derive upper bounds on WFRA’s convergence time. We start by analyzing $f(\lambda)$:

**Lemma 2** Let $f(\lambda)$ be $P_1$’s objective function. Then,

\[
f(\lambda) \leq \begin{cases} 
\omega_{\text{max}}R_{\text{max}}^{1-\alpha} \times N^{\alpha} \times M^{1-\alpha} & 0 < \alpha < 1 \\
\omega_{\text{min}}R_{\text{min}}^{1-\alpha} \times N^{\alpha} \times M^{1-\alpha} & \alpha > 1 
\end{cases}
\]

\[
(7)
\]

From linear algebra, if $a$ and $b$ are two positive numbers, then $(a-b) \times \left(\frac{1}{a} - \frac{1}{b}\right)$ (where $\alpha > 0$) is always $\leq 0$, with equality happening when $a = b$. Since $\omega_{k}$ and $\omega_{l}$ in Eq. (7) are two positive numbers, it follows that each of the multiplications of the parentheses in Eq. (7) is zero, i.e.,

\[
\forall k, l \in \mathbb{N}, r_{k}^{\text{opt}}, r_{l}^{\text{opt}} \Rightarrow \frac{r_{k}^{\text{opt}}}{r_{l}^{\text{opt}}} = \gamma
\]

In other words (using Eq. (8)), $\forall i \in N, r_{i}^{eq} = r_{i}^{opt}$.

Finally, while $r^{eq}$ is unique, we prove that there could be infinitely many resource allocations ($\lambda_{i,j}$s) to realize it.

**Proposition 1** Consider problem $P_1$ and let $r^{opt}$ denote the unique optimal outcome. Then, there could be infinitely many possible resource allocations to realize $r^{opt}$.

**Proof:** Our proof is based on an example. Consider a HetNet topology with two BSs and two clients. Let $\omega_{i} = 1 \forall i$ and $R_{i,j} = 1 \forall i,j$. Then, $\lambda_{1,1} = \lambda_{2,2} = \beta$ and $\lambda_{1,2} = \lambda_{2,1} = 1 - \beta$ results in the optimal allocation (i.e., $r_1 = r_2 = 1$) for any value of $\beta \in [0,1]$ and any desired $\alpha$. 

**Lemma 2** Let $f(\lambda)$ be $P_1$’s objective function. Then,

\[
f(\lambda) \geq \begin{cases} 
\omega_{\text{max}}R_{\text{max}}^{1-\alpha} \times N^{\alpha} \times M^{1-\alpha} & 0 < \alpha < 1 \\
\omega_{\text{min}}R_{\text{min}}^{1-\alpha} \times N^{\alpha} \times M^{1-\alpha} & \alpha > 1 
\end{cases}
\]

\[
(8)
\]
Hence, as BS $f$ is bounded by the difference between maximum and minimum values of $f$, the key idea to bound the convergence time, is to find a lower bound on $f$’s increments through:

**Lemma 3** Let $U(x) = \frac{x^{1+\alpha}}{1-\alpha}$ with $x > 0$ and $\alpha > 0$. Let $\epsilon$ be a small positive number. Then

\[ U(x) - U(x - \epsilon) \geq \epsilon U'(x) - \frac{\epsilon^2}{2} U''(x) \]

\[ U(x) - U(x + \epsilon) \geq -\epsilon U'(x) - \frac{\epsilon^2}{2} U''(x + \epsilon) \]

Here, $U'$ and $U''$ denote the 1st and 2nd derivative of $U$.

We use the Taylor theorem [16] and our objective function properties to derive the lemma properties [9].

We now proceed to derive a lower bound on $f$’s increments (i.e., $\Delta$). With abuse of notation, let $\{1, 2, \ldots, k\}$ denote the set of clients with non-zero PHY rates to BS $j$. Let client 1 be the one with minimum $R_{min}$ across all these clients. Hence, as BS $j$ executes WFRA (by choosing a new $\lambda$ vector), it increases the throughput of client 1 and we would have

\[ \frac{r_1 + \epsilon R_{1,j}}{(\omega R_{1,j})^\alpha} \leq \frac{r_2 - \epsilon R_{2,j}}{(\omega R_{2,j})^\alpha}, \ldots, \frac{r_k - \epsilon R_{k,j}}{(\omega R_{k,j})^\alpha} \]

In Eq. (9), $\epsilon_1$ is a positive number equal to $\epsilon_2 + \ldots + \epsilon_k$. The above inequality is because as $j$ executes WFRA, the level of the water fill at client 1 cannot exceed other clients’ level. Further, since $\epsilon_1 = \sum_{i=2}^{k} \epsilon_i$, from Eq. (9) we would have

\[ \forall i \in \{2, \ldots, k\} \quad \frac{\omega_i R_{1,j}}{(r_1 + \epsilon R_{1,j})^\alpha} \geq \frac{\omega_i R_{i,j}}{(r_i - \epsilon R_{i,j})^\alpha} \]

\[ \implies \epsilon_1 \frac{\omega R_{1,j}}{(r_1 + \epsilon R_{1,j})^\alpha} \geq \sum_{i=2}^{k} \epsilon_i \frac{\omega R_{i,j}}{(r_i - \epsilon R_{i,j})^\alpha} \]

We derive a lower bound on $f$’s increments through:

\[ f(\lambda)^{new} - f(\lambda) = \omega_1(U(r_1 + \epsilon R_{1,j}) - U(r_1)) + \sum_{i=2}^{k} \omega_i(U(r_i - \epsilon R_{i,j}) - U(r_i)) \]

In Eq. (10), $\alpha > 1$ and $\epsilon$ is small. We use the Taylor theorem [16] and our objective function to derive the bounds in Theorem 4 by making these simplifying assumptions and leveraging $\Delta$, and Lemma 2.
Case 2: $\alpha \to 1$. This value of $\alpha$ corresponds to proportional fairness. In particular, for $\alpha = 1$ the utility function definition in $P_1$ is commonly changed to $U(r_i) = \log(r_i)$. We prove the convergence and optimality of WFRA (with $\alpha = 1$) in [8].

Case 3: $\alpha \to \infty$. Here, the objective function in $P_1$ is commonly changed to a max-min function. In [17], we studied the max-min fair resource allocation problem and presented a slightly different water filling algorithm that converges to an equilibrium. However, we proved that the resulting equilibrium is not necessarily optimal. We can apply the same methodology to prove the convergence of WFRA (as $\alpha \to \infty$), however, the outcome is still not necessarily optimal. For instance, consider the example given in [17]: Assume 2 clients with $\omega_i = 1$, and 2 BSs. Let $R_{1,1} = 1$, $R_{1,2} = 2$, $R_{2,1} = 4$, and $R_{2,2} = 3$. Then, $\lambda_{1,1} = 1$, $\lambda_{1,2} = 0.4$, $\lambda_{2,1} = 0$, and $\lambda_{2,2} = 0.6$ is an equilibrium, which results in $r_1 = r_2 = 1.8$. However, the optimal outcome is $r_1 = r_2 = 2.4$, which can be achieved by $\lambda_{1,1} = 0.4$, $\lambda_{1,2} = 1$, $\lambda_{2,1} = 0.6$, and $\lambda_{2,2} = 0$.

IV. PERFORMANCE EVALUATION

In this section, we present the results of extensive simulations to characterize the performance of WFRA. Discussion and comparison against DComp (an alternative distributed algorithm based on dual decomposition) are presented in [9].

Simulation Setup. We used a comprehensive multi-RAT system level simulator to conduct simulations with three access technologies: (i) LTE, (ii) WiFi, and (iii) 5G New Radio (NR). Each RAT is calibrated with industrial benchmarks [18] to ensure its conformance to the 3GPP standard specifications. Fig. 4 summarizes some of the key simulator parameters.

Fig. 5(a) shows our simulation topology, which consists of 7 LTE macrocells each with three sectors. We randomly drop a single WiFi, a single 5G NR small cell, and a varying number of clients in each sector. Each of our clients has a weight ($\omega_i$) of 1 and has three RATs (WiFi, LTE, and 5G NR). We associate each client’s RAT to the corresponding BS that provides the highest downlink SNR. We generate channels between each client and all the BSs according to the 3GPP specifications [18]. Channel variations in our simulator are captured through a Doppler mobility parameter, which creates slow variation (fading) around an average $R_{avg}$ between every client and BS. Each of our data points is an average of 100 simulation trials. We implement the following algorithms:

Conventional. This scheme combines traffic across multiple RATs without changing the Conventional scheduling algorithm on each BS. In particular, we use a round-robin (RR) queuing mechanism at the WiFi BSs and a proportional fair scheduler [8] at the LTE/NR BSs. In the RR scheduler, the BS maintains a different queue for each client and sequentially serves a single packet from each queue at every round.

OMMA. Opportunistic Multi-MAC aggregation [19] is a delay-equalizing resource allocation algorithm, which splits traffic across multiple RATs so as to equalize the packet delay across the RATs, and hence minimize the maximum delay per client. OMMA does not account for fairness among the clients, but similar to WFRA only uses local information at each BS.

### Table: Name and description of common parts across all three RATs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation duration</td>
<td>100 seconds</td>
</tr>
<tr>
<td>Channel model</td>
<td>Urban street canyon</td>
</tr>
<tr>
<td>Traffic model</td>
<td>Full buffer</td>
</tr>
<tr>
<td>Number of clients</td>
<td>5/10/15 per sector (i.e., total number of 105/210/315 clients)</td>
</tr>
<tr>
<td>Link adaptation</td>
<td>Ideal MCS feedback every 10 ms</td>
</tr>
<tr>
<td>Doppler</td>
<td>3 km/hr environmental mobility</td>
</tr>
<tr>
<td>MAC level packet size</td>
<td>500 bytes</td>
</tr>
<tr>
<td>LTE parameters</td>
<td>2 GHz band, 20 MHz bandwidth; DL FDD mode; 44 dBm Tx power; up to 64 QAM modulation</td>
</tr>
<tr>
<td>NR parameters</td>
<td>6 GHz band, 20 MHz bandwidth; DL FDD mode; 30 dBm Tx power; up to 256 QAM modulation</td>
</tr>
<tr>
<td>WiFi parameters</td>
<td>802.11n, 5 GHz band, 20 MHz bandwidth; 11 channels, RTS/CTS disabled, 4 ms TxOP, DL mode; 20 dBm Tx power; up to 64 QAM modulation</td>
</tr>
</tbody>
</table>

![Fig. 4. Some of the key simulation parameters.](image-url)
to 0, there would be no channel variations or packet losses (i.e., ideal conditions). We next set the Doppler to 3 km/hr and measure the vector of throughput values (r) across all the clients for each scheme. Finally, we calculate $\sum_{i=1}^{N} \frac{r_i}{P_i}$ and divide it by $N$ (total number of clients) to quantify the distance of each scheme to the ideal $\alpha$-fair allocation. This metric allows us to compare two n-dimensional vectors. A value close to 1 means an outcome is close to the ideal outcome. Fig. 5(d) depicts the distance of each scheme to the ideal $\alpha$-fair allocations with 15 clients/sector (we observed similar results with 5/10 clients). We observe that even with channel dynamics, WFRA gets very close to the ideal allocations.

**Convergence Time.** Figs. 6(a) and 6(b) depict WFRA’s convergence time properties. Here, we set the Doppler to 0 to avoid channel dynamics. Note that if channel conditions are continuously changing, WFRA would not converge as it would continuously adapt to system dynamics. Fig. 6(a) shows the average number of required steps for WFRA to converge. There are two observations: (i) WFRA converges in a small number of steps, and (ii) there is a small change in the convergence time for different client numbers. This is due to WFRA’s water filling operation, in which a BS finds the $\lambda_{i,j}$s to all of its clients in one shot and also because irrespective of the number of clients, half of the BSs adjust their $\lambda_{i,j}$ at each scheduling interval ($p_j = 0.5$).

Fig. 6(b) shows sample evolutions of the previously defined distance metric $(\sum_{i=1}^{N} \frac{r_i}{P_i})/N$ as function of step number $t$ (i.e., scheduling interval) in WFRA. The samples correspond to simulation realizations with 15 clients per sector. We observe that irrespective of the value of $\alpha$, the distance metric is more than 0.8 within 4 steps, which shows that WFRA can get very close to the optimal outcome within a few iterations.

V. CONCLUSION

We studied the theoretical aspects of generic $\alpha$-fair throughput aggregation in HetNets. We showed that if a BS wants to unilaterally maximize the system objective, it should allocate its resources according to a water filling operation. We used this observation to design a distributed algorithm that achieves $\alpha$-fairness. We analyzed several theoretical aspects of the algorithm and showed that our algorithm can quickly adapt to system dynamics and realize the desired fairness outcome.

VI. ACKNOWLEDGEMENTS

This research was supported in part by NSF grants CNS-1910517 and CNS-1942305.

REFERENCES

[18] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz,” in 3GPP Technical Report, 38.901 (Release 15), 2018.