CLIENT-CENTRIC RADIO ACCESS NETWORK
SELECTION IN HETEROGENEOUS NETWORKS

MICHAEL WANG

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
ELECTRICAL ENGINEERING
ADVISER: MUNG CHIANG

NOVEMBER 2016
Abstract

Heterogeneity of modern wireless network radio access technologies (RATs) (e.g., 3G/4G/LTE, Wi-Fi, Bluetooth, 5G) is an intrinsic part of providing seamless connectivity and network access in next-generation wireless networks. With modern mobile edge devices increasingly equipped with a variety of wireless interfaces to access these heterogeneous networks (HetNets), these edge clients are capable of dynamically switching between different access networks to optimize their performance.

Such freedom comes at a cost: specifically, the added requirement and complexity of determining which network a client should select at any time. Traditionally, solutions for RAT selection in HetNets focus on the network-centric approach, with a centralized controller solving a global optimization, but this approach is not only non-scalable due to required signaling, but also the practical problem of non-cooperative network operators. However, with the increasing processing power of client edge devices (e.g., smartphones, tablets, etc), it is worth considering if and how RAT selection should be done at the client instead of in the network.

In this dissertation we study RAT selection in HetNets from a client-centric perspective under varying degrees of network-provided information, where the client wishes to maximize throughput on its selected RAT: (1) perfect network knowledge (clients have perfect information on other client-RAT association configurations), (2) partial network knowledge (clients have time-averaged statistics for their own client-RAT channels), and (3) no network knowledge (clients have no statistics provided by the RAT).

In (1), we model the problem as a non-cooperative game between different players under perfect information, and design the client-centric distributed RAT Selection Algorithm using binary exponential backoff and hysteresis (local memory of previous RATs) to guarantee convergence to a bounded Nash Equilibrium.
In (2), we model the problem as a multi-armed bandit with switching costs where the clients only know each channel’s spectral (eigenvalue) gap of its transition matrix, and develop the mHS algorithm that obtains optimal $O(\ln(t))$ total regret.

In (3), we solve the multi-armed bandit problem in the absence of network-provided information by applying reinforcement learning to map locally-obtainable measurements to empirically-obtained past throughputs in our WIFFN algorithm to achieve $O(\ln(t))$ regret.
Acknowledgements

First and foremost, I would like to thank my adviser, Dr. Mung Chiang, of the Princeton EDGE Lab for the vision, guidance, and the multitude of opportunities that he has provided throughout my time at Princeton. Thanks to Dr. Chiang, I have had the honor and the wonderful opportunity over these past five years to work on a diverse variety of interesting projects with real-world applications, some of which are included in this very work.

I would also like to thank the members of my thesis committee: Thank you, Dr. Prateek Mittal of Princeton University, for being such a helpful part of this process, and thank you Dr. Tian Lan of George Washington University with whom I have had the great pleasure of working with over the entirety of my graduate career. I would also like to thank my readers for this thesis: Dr. Bharath Balasubramanian, who has provided copious amounts of knowledge and levity during our collaboration, as well as Dr. Aveek Dutta, a great source of insight and understanding about the fundamentals of our shared work.

A great number of people have helped and taught me immensely during my time in the EDGE Lab: I would to specifically thank our Princeton mom Lisa Lewis for being there for us, as well as Dorothy Coakley for all of their help, assistance, and advice that we were able to call upon when in need. I’d also like to thank Jack Brassil, who not only provided the opportunity for my internship at HP Labs, but also shared his great insight and knowledge with me during my time there. There are a huge number of fellow graduate students and post-docs that I’d like to thank for the shared friendships and persistent encouragement. In alphabetical order (I don’t think there’s ever a way I could rank all of you) with the exception of two, I’d like to thank the members of the EDGE Lab, both past and present: Ehsan Aryafar, Swapna Buccapatnam, Aakanksha Chowdhery, Jae-yoon Chung, Maria Gorlatova, Felix Huang, Carlee Joe-Wong, Yixin Sun, Shirley Wang, Felix (Ming Fai)
Wong, Liang Zheng. The reason why I called out two members of the EDGE Lab is because they deserve special acknowledgement. Special thanks go to my good friend, Chris Brinton: ever since we first shared classes at the start of our time at Princeton in late 2011, he has been a true friend to me, and has been a reliable sounding board for ideas and thoughts ever since. I’d also like to thank Jiasi Chen, my good friend and office-mate since early 2012, for being an amazing person and for her mentorship, advice and encouragement over the four years that we shared. It’s been a wild ride, and I’m going to miss working alongside you all.

Over the course of my time in the EDGE Lab, I worked on a variety of projects. The first major project I worked on was the “Mission-Oriented Resilient Clouds (MRC)” project with Applied Communication Sciences (now Vencore Labs): I’d like to thank Stu Wagner, Andrei Ghetie, and particularly Eric van den Berg of ACS for the chance to learn about and implement the research that we did as part of the project, and for our ongoing collaboration. The second major project was the “Higher, Denser, Wilder” project for 5G networks with Intel Labs (as well as the precursor project dealing with client-side control of HetNets). I’d like to thank Nageen Himayat, Sarabjot Singh, and Shu-Ping Yeh of Intel Labs, as well as members of the team at the University of Southern California and New York University for their feedback and advice for our HetNets work in the HDW project.

I’d like to thank some friends from outside of Princeton: Kate Barber, Ethan Chiang, Charles Muscarella, Erika Tsuda, and Dejan Veskovic: I have known you all for much longer than my time at Princeton, and without you, I probably would have lost myself in work.

Finally, and most importantly, I’d like to thank my family for the patience they’ve had with me during my time here, their unconditional love and understanding, and whose support without which I would not have been able to complete this work.
To my parents, Shiehlie and Hsiu-mei, for their unwavering support, and to my sisters, Lillian and Jessica, for being there.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>2 Ideal Network Assistance: Convergence to Nash Equilibria</strong></td>
<td>6</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>6</td>
</tr>
<tr>
<td>2.2 System Model</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Network Model</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Throughput Model</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 RAT Selection Games</td>
<td>11</td>
</tr>
<tr>
<td>2.2.4 Distributed RAT Selection Algorithm</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Convergence</td>
<td>14</td>
</tr>
<tr>
<td>2.3.1 Single-Class RAT Selection Games</td>
<td>14</td>
</tr>
<tr>
<td>2.3.2 Mixed-Class RAT Selection Games</td>
<td>17</td>
</tr>
<tr>
<td>2.4 Pareto-Efficiency</td>
<td>21</td>
</tr>
<tr>
<td>2.5 Performance Evaluation</td>
<td>25</td>
</tr>
<tr>
<td>2.6 Related Work</td>
<td>29</td>
</tr>
<tr>
<td>2.7 Conclusions</td>
<td>31</td>
</tr>
</tbody>
</table>
### 3 Partial Network Assistance: Minimizing Regret and Switching Costs

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>33</td>
</tr>
<tr>
<td>3.2 System Model</td>
<td>36</td>
</tr>
<tr>
<td>3.2.1 Millimeter Wave Radio and HetNets</td>
<td>36</td>
</tr>
<tr>
<td>3.2.2 Online Learning and Regret in highly variable HetNets</td>
<td>38</td>
</tr>
<tr>
<td>3.3 Online Learning for Stochastic RAT Selection</td>
<td>40</td>
</tr>
<tr>
<td>3.3.1 Problem Formulation</td>
<td>40</td>
</tr>
<tr>
<td>3.3.2 Network Selection Algorithm</td>
<td>43</td>
</tr>
<tr>
<td>3.3.3 Upper Bound on the Total Regret</td>
<td>45</td>
</tr>
<tr>
<td>3.4 Performance Evaluation</td>
<td>48</td>
</tr>
<tr>
<td>3.4.1 Simulation Results</td>
<td>49</td>
</tr>
<tr>
<td>3.5 Related Work</td>
<td>51</td>
</tr>
<tr>
<td>3.6 Conclusion</td>
<td>52</td>
</tr>
<tr>
<td>3.7 Appendix</td>
<td>53</td>
</tr>
<tr>
<td>3.7.1 Proof of Theorem 8</td>
<td>53</td>
</tr>
<tr>
<td>3.7.2 Throughput Model for Simulations</td>
<td>59</td>
</tr>
</tbody>
</table>

### 4 Absent Network Assistance: Inference and Reinforcement Learning

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>61</td>
</tr>
<tr>
<td>4.2 System Model</td>
<td>64</td>
</tr>
<tr>
<td>4.2.1 Network Model</td>
<td>64</td>
</tr>
<tr>
<td>4.2.2 Radio Access Technology Model</td>
<td>68</td>
</tr>
<tr>
<td>4.2.3 Tradeoffs in Client-driven Inference and RAT Selection</td>
<td>69</td>
</tr>
<tr>
<td>4.3 Wireless Inference for Fog Networks (WIFFN) Algorithm</td>
<td>71</td>
</tr>
<tr>
<td>4.3.1 Passive Measurements and Binning</td>
<td>71</td>
</tr>
<tr>
<td>4.3.2 Reinforcement Learning and Regret</td>
<td>74</td>
</tr>
</tbody>
</table>
# Table of Contents

4.3.3 WIFFN Algorithm ..................................................... 76
4.4 Performance Evaluation .................................................. 80
  4.4.1 Simulation Results ................................................. 81
4.5 Related Work ............................................................ 84
4.6 Conclusion ................................................................. 85
4.7 Appendix ................................................................. 86
  4.7.1 Proof of Theorem 9 ................................................... 86

5 Related Works ................................................................. 90
  5.1 HetNet Selection ......................................................... 90
  5.2 Hybrid Control .......................................................... 93
  5.3 Fast Time Varying RATs (mmWave) .................................. 96
  5.4 Noisy Inference of Client Metrics ................................... 98
  5.5 Wi-Fi Offloading ........................................................ 100
  5.6 Game Theoretic Analysis .............................................. 101
  5.7 Probabilistic Analysis .................................................. 103
  5.8 Current State of the Industry ....................................... 104

6 Conclusion ................................................................. 107
  6.1 Future Work .............................................................. 110

7 Published Works ............................................................. 113

Bibliography ................................................................. 115
List of Tables

2.1 Chapter 2 Notation ................................................. 14

3.1 Chapter 3 Notation ................................................. 41

3.2 Switching Scheduler Algorithm: Block/Frame Lengths ............. 44

3.3 Parameters for PHY Layer Throughput Distribution ................. 48

4.1 Passive Measurement Ranges for Edge Network RATs ............... 66

4.2 Chapter 4 Notation .................................................. 71

4.3 Parameters for PHY Layer Downlink Throughput Distribution based on MCS. 81
# List of Figures

1.1 Heterogeneous Network Model .................................................. 3

2.1 An example heterogeneous network .......................................... 9

2.2 System state evolution in class-2 RAT selection games .................. 17

2.3 An example infinite improvement path ........................................ 18

2.4 An example of Hysteresis ...................................................... 19

2.5 Simulation Results for the Ideal Algorithm ................................. 26

3.1 Temporal variations in LTE/802.11 vs mmWave ............................ 34

3.2 3-State mmWave Markovian Channel State Model ......................... 37

3.3 Heterogeneous Network Example with mmWave access .................... 38

3.4 Example of switching schedule for (M=3)-RAT HetNets .................. 44

3.5 Performance Evaluation of mHS Algorithm .................................. 49

4.1 Example of Edge Network with Heterogeneous Radio Access Technologies 66

4.2 Coarsifying passive measurements through “binning” .................... 72

4.3 Measurement and Association in each timestep ........................... 77

4.4 Simulation Results for WIFFN Algorithm ................................... 82

5.1 Difference between Network-controlled and Client-controlled HetNets 91
Chapter 1

Introduction

Heterogeneity of modern wireless network radio access technologies (RATs) (such as 3G, 4G/LTE, Wi-Fi, Bluetooth, and potential 5G technologies) is a critical component for ensuring network access and communication in current- and next-generation wireless networks. With so many different networking options available, and with modern mobile and edge devices sufficiently equipped with multiple wireless interfaces to take advantage of these different networks, these mobile devices are able to switch between different networks in an opportunistic way to perform services useful to the user.

However, this increased access to different networks comes with the added requirement and complexity of determining which network a user (termed “client”) should connect with at any given time. In a chaotic radio environment (e.g. a bustling urban downtown neighborhood), channel conditions change so frequently that fine control of the clients in the network is required in order to prevent certain behaviors such as frequent switching between networks from damaging overall network performance. The main question to be answered in the Heterogeneous Networks (HetNets) scenario is “How should a client select a RAT at any given time?” For example, there are many efforts under way to specify solutions for networking cellular technologies (LTE,
3G, etc.) and IEEE 802.11 (Wi-Fi) technologies in the Third Generation Partnership Project (3GPP) [3,4] (for a survey refer to [102]).

As an analogy, consider the problem of cars changing lanes on a multi-lane highway during rush hour. Here, a car represents a mobile client, whose objective is to minimize its transit time, representing download or upload time. To optimize this objective, a car wishes to obtain the fastest possible speed on the highway, which represents maximizing throughput—to get from point A to point B, with a minimal passage of time. However, the highway is not an unlimited resource: each lane on the highway represents a different RAT, and the speed of a car in each lane represents its current throughput. If cars continually switch lanes in pursuit of increasing speed (clients opportunistically switching in the hopes of better throughput), this can often lead to oscillations, chaos, and a reduction in average speed (throughput)—from this, we can realize that the answer to the main question for RAT selection in HetNets may not be so straight-forward.

One traditional approach taken by network operators and academic research on HetNets is to allocate authority to a centralized agent or controller [32,34,109], which is then able to distribute edge devices and clients over different networks based on some network objective such as load balancing. This approach has the advantage of finding a global optimal operating point by assigning devices to RATs. However, with increasing numbers of mobile devices at the network edge, the message passing required from edge devices to the centralized agent to communicate available RATs and channel qualities for each device can quickly grow exponentially, resulting in an untenable situation for high-density areas like urban environments. Furthermore, different network operators seldom have an incentive to cooperate: for example, Boingo Wireless has no financial benefit in offloading its own WiFi customers onto Verizon’s cellular network, and vice versa.
Figure 1.1: Heterogeneous Network Model
An example of a Heterogeneous Network with clients able to access 5G millimeter wave RATs

Meanwhile, with the advent of smartphones and other edge devices with high computational capacity, it is increasingly common to find devices released by device vendors with some form of association control given to the client, such as control loops that compare received signal powers from different RATs to select the optimal RAT. Localizing multiple aspects of the network association control plane to the edge device allows for a more accurate local view of not only the ambient radio environment around the client, but also the current applications that are running the client device, any client-specific preferences for data and energy usage, as well as the remaining battery life of the device itself. Furthermore, pressing issues present in a centralized decision-making scheme such as ownership and control of the intelligence for RAT selection over multiple BSs are avoided by allowing individual clients to determine which network they should connect to.

The heterogeneity (e.g. latency, bandwidth, packet loss, availability, etc.) of networks, coupled with the rise of increased computational power on modern mobile and edge devices, has increasingly led to the question of not only become which RAT to select, but also where the intelligence for RAT selection in HetNets should be located, and it is now not inconceivable to place some functionality for RAT association at the network edge–on the client device itself. In fact, with interest
in client-driven network control (used in applications such as edge networking, or Fog networking \cite{28}) increasing in the past few years, network-controlled centralized solutions can no longer be applied to problems such as client-controlled data transfer, storage and processing on the network edge.

The subsequent chapters detail three approaches localizing RAT selection and association for HetNets with varying degrees of network-provided information, defined as knowledge of both the wireless channel between the client and the base station/access point, as well as the behavior of other clients on the network. Chapter \ref{chapter:2} introduces the basic problem formulation of client-centric RAT selection in Heterogeneous Networks given perfect global information on the client-RAT association of all other clients, as well as their potential PHY layer rates on their accessible networks. The problem is modeled as a non-cooperative game and a baseline algorithm is designed for clients to converge to a local Nash Equilibrium, which leverages the concept of hysteresis, or local memory of prior throughputs, to prevent oscillation. We present bounds on the goodness of such Nash Equilibria in terms of the “pareto-efficiency gain,” or the potential improvement the clients could achieve if RAT selection were determined by a centralized controller that achieved some pareto-optimal client-RAT allocation.

Chapter \ref{chapter:3} describes the more likely case when the network is unwilling (or unable) to provide exact, up-to-date information on other clients in the network, when the client can only obtain some time-averaged statistics on channel states over time. In such a scenario, RATs can be modeled as markov chains with unknown transition matrices and reward distributions—except for the value of the spectral (eigenvalue) gap for each network, the only information needed in our approach for minimal regret association. We model the problem as a multi-armed bandit, and present a learning policy that balances the clients’ desire to explore the available RATs against their
desire to exploit the optimal RAT, while minimizing the cost of switching between networks. Our mHS algorithm achieves asymptotically optimal $O(\ln(t))$ total regret.

Chapter 4 details how a client may leverage reinforcement learning in the absence of any network information. In the case where a client needs to rely on local passive measurements (e.g., RSRP/RSRQ in LTE, RSSI in 802.11 Wi-Fi and Bluetooth), we model the problem as a multi-armed bandit with unknown rewards and unknown statistics, and develop the WIFFN algorithm that leverages local historical measurements to generate a mapping of measurements to potential throughput. We introduce the concept of “binning” passive measurements, which allows the client to trade off between minimizing initial regret versus long-term regret by increasing the number of bins and refining the empirical mean throughput in each range of passive parameters. This algorithm is shown to also be asymptotically optimal in terms of regret, and outperforms other existing measurement-based RAT selection algorithms.

Related work is presented in Chapter 5, detailing previous network-centric RAT selection approaches as well as potential hybrid (shared) control between the network and mobile clients. Background is provided on fast time varying RATs such as 5G millimeter wave (mmWave) networks, noisy inference, as well as Wi-Fi Offloading. A short overview of game-theoretic and probabilistic analysis is provided, as well as a concluding commentary on the current state of the industry with respect to controlling RAT selection in HetNets. We conclude in Chapter 6 and provide a list of publications by the author in Chapter 7.
Chapter 2

Ideal Network Assistance:
Convergence to Nash Equilibria

2.1 Introduction

In this chapter, we examine the problem of client-centric RAT association in HetNets
given perfect information on other clients (users) in the network. We assume that,
in this non-cooperative game formulation, although the clients may not communicate
with each other, the network is able to provide each client with a global big-picture
view of the current client-RAT association and the peak physical (PHY) layer rates
for each potential client-RAT pairing.

The main challenge in analyzing the behavior of these games is to incorporate
realistic models that (i) capture the multi-rate property of heterogeneous networks
(i.e., each client has a distinct transmission rate for each access technology), and
(ii) accurately model the impact of each client’s decision on other clients’ received
throughputs. We divide the throughput models of different access networks into two
general classes. In class-1 throughput models, clients on the same base station (BS)
achieve the same throughput, however, different client combinations result in distinct
throughput values. This class of throughput models is especially suitable to model throughput-fair access networks such as Wi-Fi [49]. In class-2 throughput models, each client receives a client-specific throughput value that depends on the number of other clients sharing the same BS. This class of throughput models is especially suitable to model time/bandwidth/proportional-fair access networks in 3/4G networks. We next analyze some of the most important properties of the equilibria in these games, such as convergence and Pareto-efficiency. We further perform extensive measurement-driven simulations to investigate the performance of distributed RAT selection in practice.

Our main contributions in this chapter are as follows [14]:

- **Convergence:** We prove that in single-class RAT selection games, convergence to Nash equilibria is guaranteed [Theorems 1, 2]. When a mixture of classes is considered, we provide an example 2-client game in which an improvement path can be repeated infinitely without reaching an equilibrium. Thus motivated, we introduce a hysteresis mechanism to RAT selection games, and prove that by applying appropriate hysteresis policies, convergence to equilibria can still be guaranteed [Theorem 3].

- **Efficiency:** We investigate the optimality of Nash equilibria with respect to the set of Pareto-dominant points. We show conditions under which the Nash equilibria are also Pareto-optimal [Theorems 4, 5]. When the conditions are not met, we introduce a metric termed *average Pareto-efficiency gain* to quantify the distance between the Nash points and the set of Pareto-dominant points. We show that in class-1 games, the distance between a Nash point and Pareto-dominant points can become unbounded [Theorem 4]. However, we provide tight constant approximation bounds for class-2 games [Theorems 6, 7].

- **Practicality:** We perform hundreds of measurements across multiple access technologies (e.g., HSPA, HSPA+, Wi-Fi) to obtain information on the availability and quality of access networks in an indoor environment. With extensive
measurement-driven simulations, we show that RAT selection games converge to equilibria with a small number of switchings. We also show that the appropriate selection of switching threshold provides a balance between convergence time and the efficiency of equilibria. Finally, we investigate the impact of noisy throughput estimates and propose solutions to handle them.

This chapter is organized as follows. We present our system model in Section 2.2. In Sections 2.3 and 2.4, we investigate the convergence and Pareto-efficiency properties of RAT selection games, respectively, and we present the results of our measurement-driven simulations in Section 2.5. We discuss the related work in Section 2.6. Finally, we conclude this chapter in Section 2.7.

2.2 System Model

In this section, we present the system model and propose a generic, distributed RAT selection algorithm with autonomous actions by each client.

2.2.1 Network Model

We consider a heterogeneous wireless environment which consists of $M$ base stations (BSs) and $N$ clients. Here, BS is simply a generic term to collectively represent NB in 3G, eNB in 4G, AP in Wi-Fi, femtoBS in femto-cells, etc. The set of BSs and clients are denoted by $\mathcal{M} = \{1, ..., M\}$ and $\mathcal{N} = \{1, ..., N\}$, respectively. We denote the set of clients connected to BS $k$ by $\mathcal{N}_k$. Fig. 2.1 shows an example of such a heterogeneous network in which BSs consists of multiple access networks (LTE, 3G, and Wi-Fi). We assume that all BSs are interference-free by means of spectrum separation between BSs that belong to different access networks, and frequency reuse among same kind BSs. Each client has a specific number of RATs, and therefore has access to a subset of BSs. Note that due to the frequency separation between BSs, each RAT can receive
beacon signals from at most one BS. If a client’s wireless interface is able to receive beacon signals from multiple BSs, we model this functionality by assuming multiple RATs for such an interface. For example, an 802.11b wireless card that is able to receive signals from channels 1, 3, and 11 (in 2.4 GHz band), is denoted as a 3-RAT interface. Different access networks in heterogeneous networks have many different characteristics such as packet sizes, physical layer technology, modulation and coding scheme (MCS), etc. Given today’s consumer device capability, while a client can switch its selected RAT (based on its expected performance on other RATs), we assume that each client uses only a single RAT at any given time.

Figure 2.1: An example heterogeneous network.

2.2.2 Throughput Model

The throughput achieved by a client $i$ on a BS $k$, denoted as $\omega_{i,k}$, depends on the client’s selected access network, the client-specific parameters (e.g., transmission rate) and the other clients that are connected to the same BS. The instantaneous PHY rate $R_{i,k}(t)$ of client $i$ on BS $k$ depends on its selected MCS and the channel conditions at time $t$. We assume stationary channel conditions without considering mobility.

The different access networks in heterogeneous networks have different medium access (MAC) protocols to share the bandwidth among the clients. We divide the medium access protocols into two classes:
Class-1 Throughput Models: In this class, the throughput of a client $i$ on BS $k$ depends on the specific clients that are connected to $k$. However, all clients that share the same BS achieve the same throughput, i.e., with abuse of notation

$$\omega_{i,k} = f_k(R_{1,k}, R_{2,k}, \ldots, R_{n_k,k}) \quad \forall i \in N_k \quad (2.1)$$

Here, $n_k$ is the number of clients that are connected to BS $k$. An example of such MAC protocols is the distributed coordination function (DCF) implemented in 802.11, in which a Wi-Fi BS provides fair access opportunity to uplink clients [22,49]. The throughput of the clients on the downlink depends on the queuing technique implemented on the BS. The most common technique uses a round-robin scheme. Thus, the downlink throughput of a Wi-Fi client can be expressed as

$$\omega_{i,k} = \frac{L}{\sum_{j \in N_k} \frac{L}{R_{j,k}}} \quad \forall i \in N_k \quad (2.2)$$

Here, $L$ is the packet size. Throughput models similar to Eq. (2.1) can also be derived for the uplink [22].

Class-2 Throughput Models: In this class, the throughput of a client $i$ on BS $k$ depends only on the total number of clients that share the same BS (i.e., $n_k$), instead of the specific client combination. However, the throughput of each client can be different from that of other clients, i.e.,

$$\omega_{i,k} = R_{i,k} \times f_k(n_k) \quad \forall i \in N_k \quad (2.3)$$

Time-fair TDMA MAC protocols are an example of class-2 throughput models. Here the wireless medium is time-shared among all the clients such that each client has the same time duration to access the medium. Therefore, the throughput of a client $i$ connected to a time-fair BS $k$ is given by
\[ \omega_{i,k} = \frac{R_{i,k}}{n_k} \quad \forall i \in N_k \] (2.4)

OFDMA based MAC protocols with fair subcarrier sharing (e.g., WiMAX) are another example of class-2 throughput models. With fair spectrum sharing, clients receive a similar number of sub-carriers. Hence, the throughput of a client \( i \) is roughly dependent only on the total number of clients sharing the same BS, and would be similar to Eq. (2.4).

Another example of Class-2 models is proportional-fair scheduling (PFS) in 3G networks. Here, the PFS algorithm schedules at the next slot, the client that has the highest instantaneous rate relative to its average throughput (for details refer to [68]). With PFS, the closed-form expression of the average throughput of a client with Rayleigh fading is

\[ \omega_{i,k} = \frac{R_{i,k}}{n_k} \times \sum_{j=1}^{n_k} \frac{1}{j} \quad \forall i \in N_k \] (2.5)

Here, \( \sum_{j=1}^{n_k} \frac{1}{j} \) appears due to channel fading [68].

### 2.2.3 RAT Selection Games

We model the RAT selection problem in heterogeneous networks as a non-cooperative game, in which clients select RATs in a distributed manner to increase their own individual throughputs. Thus, the player set is the set of clients, i.e., \( N \). Player strategies are the choice of the RATs (or the corresponding BSs). We denote player \( i \)'s strategy by \( \sigma_i \). The strategy profile of all clients is denoted by \( \sigma = (\sigma_1, \sigma_2, ..., \sigma_N) \).

A strategy profile \( \sigma \) is said to be at Nash equilibrium if each player considers its chosen strategy to be the best under the given choices of other players. Therefore, at Nash equilibrium, no client will profit from deviating its strategy unilaterally.
We define a path as a sequence of strategy profiles in which each strategy profile differs from the preceding one in only one coordinate. If the unique deviator in each step strictly increases its throughput, the path is called an improvement path.

### 2.2.4 Distributed RAT Selection Algorithm

We propose a generic distributed RAT selection algorithm. While the algorithm is simple as we wished, its performance analysis is actually consequently more difficult. Consider synchronized slotted time for now. In RAT selection games, each client uses only one RAT at any given time for communication. However, a client is able to decode the traffic on its other RATs. For example, if RAT $j$ of client $i$ is tuned to BS $k$, then client $i$ is able to decode the packets transmitted by $k$, and therefore has the information on the number of clients on $k$ and their rates. Thus, each client can estimate its expected throughput if it decides to use another RAT for communication.

Algorithm 1 summarizes the RAT selection algorithm operated by each client. In order for client $i$ to make a switch at time $t+1$ from BS $k$ to BS $k'$ (by changing its RAT), the expected gain defined as $\frac{\omega_{i,k'}[t+1]}{\omega_{i,k}[t]}$ should be higher than a given threshold ($\eta$) for the past $T$ time slots (Line 2).

Here, $T$ corresponds to the frequency of measurement prior to switching. Note that if multiple clients switch to a BS concurrently, their expected throughputs would be different from their achieved throughputs. In order to minimize the number of concurrent switches to the same BS, we assume that clients switch probabilistically with probability $p < 1$ (Line 4). The randomization parameter, $p$, depends on the congestion in the network and acts similarly to the 802.11 contention window mechanism. Similar to the binary exponential back-off in the 802.11 DCF, we assume that when concurrent migrations to a BS happen, a client sets its randomization parameter to $p^{m_i+1}$ (Line 6), in which $m_i$ is the number of past consecutive concurrent migrations observed by $i$. 
Algorithm 1: RAT Selection Algorithm

Input: Client $i$’s parameters: $\eta$, $T$, $p$, $h$, Set of RATs
Output: Decision to switch, and the selected RAT

for each RAT $k'$ do

if $\frac{\omega_{i,k'}[t+1]}{\omega_{i,k}[t]} > \eta$, \(\forall t = t - T + 1, ..., t\) then

if $\text{class}(k') = \text{class}(k)$ then

if $\text{rand} < p^{m_i+1}$ then

switch to $k'$

if concurrent move then increment $m_i$

else reset $m_i$ to 0

else

if $\omega_{i,k'} > h$ then

if $\text{rand} < p^{m_i+1}$ then

switch to $k'$, update $h$

if concurrent move then increment $m_i$

else reset $m_i$ to 0

Since clients in the RAT selection games selfishly switch their selected RAT to increase their own throughput, it is possible for some of the clients to keep switching without reaching an equilibrium. A system design mechanism to dampen oscillations and guarantee convergence is to employ hysteresis. The hysteresis parameter in the RAT selection games, $h$, denotes the dependence of the RAT switching to the history of past switches that a client has made. Algorithm 1 shows a hysteresis policy in which a client that changes its class of BSs (e.g., from class-1 to class-2) needs to have an expected throughput higher than its hysteresis value (Lines 8-9). In section 2.3 we define our hysteresis policy in detail and demonstrate how it can guarantee convergence to equilibria in RAT selection games.
Table 2.1: Chapter 2 Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Number of clients</td>
</tr>
<tr>
<td>$M$</td>
<td>Number of BSs</td>
</tr>
<tr>
<td>$n_k$</td>
<td>Number of clients on BS $k$</td>
</tr>
<tr>
<td>$N_k$</td>
<td>Set of clients on BS $k$</td>
</tr>
<tr>
<td>$R_{i,k}$</td>
<td>PHY rate of client $i$ to BS $k$</td>
</tr>
<tr>
<td>$\omega_{i,k}$</td>
<td>Throughput of client $i$ to BS $k$</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>Strategy profile of client $i$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Switching threshold</td>
</tr>
<tr>
<td>$p$</td>
<td>Randomization parameter</td>
</tr>
<tr>
<td>$T$</td>
<td>Frequency of measurement prior to switching</td>
</tr>
<tr>
<td>$h$</td>
<td>Hysteresis parameter in the switching algorithm</td>
</tr>
</tbody>
</table>

2.3 Convergence

In this section, we investigate the convergence properties of the RAT selection games. We first consider the case in which all BSs belong to the same class of throughput models. Next, we consider the case when a mixture of the two classes exists. The behaviors qualitatively differ, as we will show.

In RAT selection games, different clients can occasionally join and/or leave a single BS concurrently. However, due to the presence of the randomization parameter $p$, such events happen infrequently and diminish rapidly when there is network congestion (due to the exponential decrease of $p$ with congestion). For the rest of this section, we ignore these events.

2.3.1 Single-Class RAT Selection Games

We first consider the case in which all BSs belong to the same class of throughput models. Note that in our model each client has a different rate for each BS. In addition, each BS has a BS-specific model to share the throughput among clients.

**Theorem 1.** Class-1 RAT selection games converge to a Nash equilibrium.
Proof. Our proof is in essence similar to the proof given in [38]. Here we apply it to the RAT selection problem and present it for completeness. Denote the throughput of client \( i \) by \( \omega_i \). For simplicity, assume the following ranking of client throughputs

\[ \omega_1 \leq \omega_2 \leq \ldots \leq \omega_N \]  

(2.6)

Define a function \( g \) on the ordered throughput values as

\[ g = \omega_1 \times S^{N-1} + \omega_2 \times S^{N-2} + \ldots + \omega_N \]  

(2.7)

Here, \( S \) is a very large number (i.e., \( S \gg \omega_i, \forall \text{ possible } \omega_i : i \in \mathbb{N} \)). Now assume that client \( i \) migrates from BS \( a \) to BS \( b \). Note that in class-1 throughput models, all same-BS clients achieve the same throughput. Thus, due to \( i \)'s migration, the throughput of all clients on BSs \( a \) and \( b \) would be affected. The throughput of clients on BS \( a \) would increase, since a client has left \( a \). The throughput of clients on BS \( b \) would decrease, since a client has joined. However, the throughput of clients on \( b \) would be higher than \( \omega_{i,a} \), or else client \( i \) would not have migrated. Thus, in the new ranking of client throughputs, the value of \( g \) in Eq. (2.7) strictly increases. As the number of clients and BSs is finite, function \( g \) cannot increase indefinitely and would terminate at a point, i.e., the Nash equilibrium.

We next focus on class-2 RAT selection games.

Theorem 2. Class-2 RAT selection games converge to a Nash equilibrium.

Proof. Our proof is based on contradiction. Define the system state of the network as the set of BSs and their connected clients. Now assume that there is a loop in the system, i.e., there exists a system-state sequence with identical start and end states, as shown in Fig. 2.2.
Next, consider the throughput inequalities of the migrating clients between any two consecutive states. As in the definitions of the models in Section 2.2, the throughput of a client $i$ on BS $k$ is equal to $R_{i,k} \times f_k(n_k)$, in which $n_k$ is the number of clients on BS $k$. For the example depicted in Fig. 2.2, we have the following inequalities for the intermediate states

$$R_{i,k} \times f_k(n_k') > ...$$

(2.8)

$$...$$

(2.9)

$$... > R_{i,k} \times f_k(n_k'')$$

(2.10)

Next, we multiply all the terms on the right hand sides, and all the terms on the left hand sides of the inequalities. Note that when a client $i$ migrates to BS $k$, we have an inequality similar to Eq. (2.8). Similarly, when eventually client $i$ migrates from BS $k$ (in order to have a cycle), an inequality similar to Eq. (2.10) exists. Therefore, when we multiply the right and left hand sides, all the $R_{i,j}$ terms will cancel each other.

On the other hand, between any two consecutive states, the number of clients on any given BS $j$ goes up or down by 1, each time a client joins or leaves the BS $j$, respectively. Therefore, whenever the number of clients on BS $j$ becomes equal to $n_j$ by a joining client (i.e., there exists an $f_j(n_j)$ term on the left hand side), an $f_j(n_j)$ term would later appear on the right hand side when a client leaves BS $j$. Therefore, after multiplying the right and left hand sides of the inequalities, the $f_j(n_j)$ terms will also cancel each other. After all the cancellations we have $1 > 1$, which is a contradiction. Since a cyclic system state sequence can not exist, every class-2 RAT selection games terminates at an equilibrium, i.e., the Nash equilibrium. 

\[ \square \]
2.3.2 Mixed-Class RAT Selection Games

In this section, we investigate the convergence properties when there is a mixture of the two classes. We first provide an example 2-player 4-BS game, in which a cycle exists, and therefore convergence to an equilibrium cannot be guaranteed. Next, we show how adding appropriate hysteresis policies can guarantee convergence.

Fig. 2.3 shows an example 2-player RAT selection game in which an improvement path can be repeated infinitely. The BSs are shown as a, b, c, and d, and the players are displayed as 1 and 2. BSs b and d are throughput-fair and belong to class-1 (Eq. (2.2)), while BSs a and c are time-fair and belong to class-2 (Eq. (2.4)). The $R_{i,j}$ value of clients on RATs/BSs is shown in Fig. 2.3.

Initially, clients 1 and 2 are connected to BSs a and b, respectively. During each stage of the game, one of the clients migrates to another BS in order to increase its throughput. In the example depicted in Fig. 2.3, the improvement path starts from (a:1, b:2) strategy profile in which client 1 is connected to BS a, and client 2 is connected to BS b. The path continues as (b:1, b:2), (b:1, d:2), (c:1, d:2), (c:1, c:2), (a:1, c:2), and finally back to (a:1, b:2).

The transition inequalities for the migrating client is also depicted in Fig. 2.3. All these inequalities hold for the selected $R_{i,j}$ values (and can further hold for an infinite

---

Figure 2.2: System state evolution in class-2 RAT selection games
Here $x_{nk}^i$ denotes the $i$'th client on BS $k$ and $n_k$ denotes the number of clients on BS $k$. In the state evolution sequence shown above, the beginning and end states are the same, i.e. a cycle happens.
An example improvement RAT in a 2-player, 4-strategy RAT selection game with both class-1 and class-2 BSs. BSs $a$ and $c$ are class-2, whereas BSs $b$ and $d$ are class-1. The unique deviator client is shown through arrows in each step. This cyclic path is generated by the six strategy profiles shown, and it can be endlessly repeated. The inequality relevant to each step, i.e., the one that guarantees the RAT switching client strictly increases its throughput is shown on the right. The six inequalities can all be validated for an infinite combination of $R_{i,j}$s. One such example is $R_{1,a} = 7.2$, $R_{1,b} = 9$, $R_{1,c} = 10.1$, $R_{2,b} = 48$, $R_{2,c} = 23.4$, $R_{2,d} = 9$. The selected rates are according to 802.11a for class-1, and 3G HSDPA for class-2.

number of client-rate combinations). The existence of such a cyclic improvement path demonstrates that in generic mixed-class RAT selection games, an improvement path can be repeated infinitely.

The above example emphasizes the need to design system parameters that can stop infinite oscillations by clients and guarantee convergence. We therefore introduce hysteresis, a mechanism that enforces the dependence of the system not only on its current selection, but also on its past selections.

In order to define hysteresis, we classify all the BSs according to their throughput class as depicted in Fig. 2.4. We next define the hysteresis value of a client $i$ in a given class as its last achieved throughput in that class prior to switching to a different class of BSs. For example, if a client switches from a BS $a$ in class-1 to a BS $b$ in class-2, its hysteresis parameter in class-1 is defined as its throughput on BS $a$. 

Figure 2.3: An example infinite improvement path
**Definition 1.** **Hysteresis Policy:** Assume a client $i$ that has moved from a class of BSs to another class of BSs. In order for $i$ to return to a BS in the previous class, its expected throughput should be higher than the corresponding hysteresis value.

Fig. 2.4 shows an example client $i$ that has moved from BS $a$ in class-1 to BS $b$ in class-2. It next changes its selected BS in a series of selfish moves within class-2. Now if client $i$ wants to go to BS $d$ in class-1 from its position on BS $c$, not only $\omega_{i,d}$ should be higher than $\omega_{i,c}$, but also $\omega_{i,d}$ should be higher than $\omega_{i,a}$ (i.e., the hysteresis value).

Our next theorem demonstrates that such a hysteresis policy guarantees convergence to an equilibrium.

![Figure 2.4: An example of Hysteresis](image)

The hysteresis value of client $i$ in class-1 is its achieved throughput prior to leaving class-1, i.e., $\omega_{i,a}$ in the above example.

**Theorem 3.** **Mixed-class RAT selection games, with hysteresis policy, converge to an equilibrium.**

*Proof.* Our proof is based on contradiction. Define the system state of the network as the set of BSs and their connected clients. Assume there is a loop in the system state evolution that can be repeated infinitely. Consider the second repetition of this loop. Note that in order to have a loop, every client that leaves its class has to return back to its class at a later time. Further, due to the loop repetition, such clients have a history of being in both classes.

For simplicity, assume that the cycle starts when a client leaves a class-1 BS. Fig. 2.4 shows an example of such a client that leaves class-1, and returns back to
class-1 (to form a cycle). Denote the throughput of client \( i \) prior to leaving class-1 as \( \omega_{i,a} \), and immediately after returning to class-1 as \( \omega_{i,d} \).

Now assume that for every leave and return by any client in class-1, there exists a virtual BS. Each of these virtual BSs (e.g., virtual BS \( v \) handling client \( i \)), handles only one specific client (e.g., by having zero rates for all other clients), and offers a throughput equal to the average of the client’s throughputs before leaving class-1 and immediately after returning to class-1. For example, the throughput of virtual BS \( v \) for client \( i \) is equal to \( \omega_{i,v} = \frac{\omega_{i,a} + \omega_{i,d}}{2} \). Note that due to the hysteresis policy, \( \omega_{i,d} \) is greater than \( \omega_{i,a} \), and therefore we have the following inequality

\[
\omega_{i,a} < \omega_{i,v} < \omega_{i,d}
\]  

Eq. (2.11) shows that client \( i \) gains by leaving BS \( a \) and joining virtual BS \( v \), and also gains later by returning to BS \( d \).

Now, consider the clients that join class-1 from class-2. Such clients would later return to class-2 due to the loop. Let \( j \) denote an example of such a client that joins BS \( e \) on class-1, from a BS in class-2. Note that \( j \) has a prior history of visiting a class-1 BS (e.g., BS \( g \)). Thus, we can assume that client \( j \) visits class-1 from a virtual BS \( v' \). The throughput of virtual BS \( v' \) for client \( j \) is equal to \( \omega_{j,v'} = \frac{\omega_{j,e} + \omega_{j,g}}{2} \). Note that due to the hysteresis policy, \( \omega_{j,e} \) is greater than \( \omega_{j,g} \), and hence \( \omega_{j,e} \) is also greater than \( \omega_{j,v'} \). Thus, we can correctly assume that client \( j \) visits class-1 from a virtual BS \( v' \). Similarly, we can construct another virtual BS, that client \( j \) joins when leaving class-1. Thus, for any client that visits class-1 from class-2, we can construct the corresponding virtual BSs. Now, note that each virtual BS accommodates only one client, and therefore it can belong to both class-1 and class-2 throughput models (BSs). Thus, we can assume that all virtual BSs belong to class-1. Now by considering class-1 and all the virtual BSs, it follows that the loop is happening within class-1.
However, in Section 2.3.1 we proved that single-class RAT selection games do not have cyclic behavior, which is a contradiction.

### 2.4 Pareto-Efficiency

Beyond convergence properties, we analyze the Pareto-efficiency of Nash equilibria in RAT selection games. We show that in some cases the Nash equilibria are necessarily Pareto-optimal. When this is not the case, we quantify the improvement of the Pareto-optimal solutions, with respect to the Nash equilibria. In order to do this, we first present the formal definitions of some of the concepts used in this section.

**Definition 2.** Let $G$ be a game with a set $N$ of players. We say that strategy profile $\sigma'$ Pareto-dominates strategy profile $\sigma$ if it holds that

$$\forall i \in N : \omega_{i,\sigma'} \geq \omega_{i,\sigma}, \quad (2.12)$$

**Definition 3.** Let $G$ be a game with $N$ players. Let $\sigma'$ denote a strategy profile that Pareto-dominates strategy profile $\sigma$. We define the average Pareto-efficiency gain of $\sigma'$ to $\sigma$ as

$$\frac{\sum_{i=1}^{N} \frac{\omega_{i,\sigma'}}{\omega_{i,\sigma}}}{N} \quad (2.13)$$

For example, assume that strategy profile $\sigma'$ has an average Pareto-efficiency gain of $\alpha$ with respect to $\sigma$. This means that clients observe an average of $\alpha$ factor increase in their throughputs by changing from strategy profile $\sigma$ to $\sigma'$. We next proceed to analyze the Pareto-efficiency of RAT selection games. We first do this for class-1 throughput models.
Theorem 4. Let $G$ be a class-1 RAT selection game with $N$ clients. Let $\sigma^p$ denote a Pareto-optimal strategy profile, and $\sigma^n$ denote a Nash profile. Let $\gamma = \frac{R_{\text{max}}}{R_{\text{min}}}$ denote the ratio between maximum and minimum rates across all the clients. Then

1) $G$ has a Pareto-optimal Nash equilibrium,

2) The average Pareto-efficiency gain of $\sigma^p$ to $\sigma^n$ can become unbounded as $\gamma \to \infty$.

Proof. Part 1. Consider the client-BS profile, $\sigma^1$, that maximizes function $g$ in Eq. (2.7). Since the value of $g$ can not be further increased, $\sigma^1$ is a Nash equilibrium. Now assume $\sigma^1$ is not Pareto-optimal. Then, there exists a strategy profile $\sigma^2$ in which all clients achieve higher or equal throughputs with respect to $\sigma^1$, and at least one client achieves a higher throughput. Hence, the value of function $g$ in profile $\sigma^2$ would be higher than its value in profile $\sigma^1$, which is a contradiction.

Part 2. While the best Nash is always Pareto-optimal, the distance between the worst Nash and a Pareto-optimal point can be very large. We provide an example for the throughput model of Eq. (2.2) to prove this. Assume 2 clients and 2 BSs $a$ and $b$ such that

$$R_{1,a} = 1, R_{1,b} = \gamma, R_{2,a} = \gamma, R_{2,b} = 1 \quad (2.14)$$

The profile (1 in BS $a$, 2 in BS $b$) is a Nash point in which each client’s throughput is equal to 1. On the other hand, the profile (1 in BS $b$, 2 in BS $a$) is a Pareto-optimal point in which each client’s throughput is equal to $\gamma$ ($\gamma > 1$). Thus, there exists a Pareto-optimal point in which each client increases its throughput by a factor of $\gamma$ and has an average Pareto-efficiency gain of $\gamma$, increasing up to 54 in 802.11 a/g. □

We next investigate the Pareto-efficiency of RAT selection games in class-2 throughput models. Specifically, we focus on the time-fair and proportional-fair throughput models of Eq. (2.4) and Eq. (2.5), respectively. We first prove that when
each client has a similar rate across different RATs (note that different clients can have different rates), all Nash points are also Pareto-optimal. Next, we provide approximations on Pareto-efficiency gains when each client has a distinct rate for each RAT.

**Theorem 5.** Let $G$ be a class-2 time-fair (or proportional-fair) RAT selection game with $N$ clients. If each client has the same rate across different RATs, then a Nash equilibrium is also Pareto-optimal.

**Proof.** Assume the contrary. Let $s(i)$ denote the selected BS of client $i$ in the Nash outcome and $n_k$ denote the number of clients on BS $k$ in the Nash outcome. Further, let $q(i)$ denote the selected BS of client $i$ in the Pareto outcome and $p_j$ denote the number of clients on BS $j$ in the Pareto outcome. Assume a time-fair throughput model (similar argument holds for proportional-fair model).

From the definition of Pareto-optimality, for each client $i$ we have that $R_{s(i)} \geq R_{q(i)}$. Therefore, at least for one client $j$ we have $n_{s(j)} > p_{q(j)}$, and for the rest of the clients (i.e., $\forall k \in \mathbb{N}$ and $k \neq j$) we have $n_{s(k)} \geq p_{q(k)}$. These inequalities show that each BS in the Pareto-point has a smaller (or equal) number of clients than in the Nash point (with at least one BS having a smaller number). However, the total number of clients across all BSs is equal to $N$, which is a contradiction. \qed

**Theorem 6.** Let $G$ be a time-fair RAT selection game with $N$ clients and $M$ BSs. Let $\sigma_n$ denote a non-Pareto-optimal Nash profile and $\sigma_p$ denote a Pareto-dominant profile with respect to $\sigma_n$. Then, the average Pareto-efficiency gain of $\sigma_p$ to $\sigma_n$ is bounded by

$$
\begin{cases} 
2 & \text{if } N \leq M \\
\frac{N+M}{N} & \text{if } N \geq M
\end{cases}
$$
Proof. We use the same notation as in the proof of Theorem 5. From Pareto-dominancy definition we have $R_{i,q(i)} \times f_q(p_q(i)) \geq R_{i,s(i)} \times f_s(n_s(i))$. From Nash equilibrium property we have $R_{i,s(i)} \times f_s(n_s(i)) \geq R_{i,q(i)} \times f_q(n_q(i) + 1)$. Thus, by replacing $f$ with the corresponding value in Eq. (2.4), the improvement factor of client $i$ is

$$\frac{R_{i,q(i)} \times f_q(p_q(i))}{R_{i,s(i)} \times f_s(n_s(i))} \leq \frac{R_{i,q(i)} \times f_q(p_q(i))}{R_{i,q(i)} \times f_q(n_q(i) + 1)}$$

(2.15)

$$\leq \frac{n_q(i) + 1}{p_q(i)}$$

(2.16)

Next, the sum of improvement factors of all clients is

$$\sum_{i=1}^{N} \frac{n_q(i) + 1}{p_q(i)} \leq \sum_{p_k, p_k \neq 0} \frac{n_k + 1}{p_k} \times p_k \leq$$

$$\begin{cases} 
2 \times N & \text{if } N \leq M \\
(N + M) & \text{if } N \geq M
\end{cases}$$

(2.17)

(2.18)

The average gain is derived by dividing the above by $N$.

Theorem 6 provides a tight bound on the average Pareto-efficiency gain of time-fair RAT selection games. In order to observe this, consider a 2 player example with 2 BSs $a$ and $b$ with the following rates

$$R_{1,a} = 1, R_{1,b} = 2 - \epsilon, R_{2,a} = 2 - \epsilon, R_{2,b} = 1$$

(2.19)

The profile (1 in BS $a$, 2 in BS $b$) is a Nash profile, in which each client’s throughput is 1. The profile (1 in BS $b$, 2 in BS $a$) is a Pareto-dominant profile, in which each
The average Pareto-efficiency gain is equal to \(2 - \varepsilon\), which can become arbitrarily close to 2 as \(\varepsilon \to 0\).

**Theorem 7.** Let \(G\) be a proportional-fair RAT selection game with \(N\) clients and \(M\) BSs. Let \(\sigma_n\) denote a non-Pareto-optimal Nash profile and \(\sigma_p\) denote a Pareto-dominant profile. Then, the average Pareto-efficiency gain of \(\sigma_p\) to \(\sigma_n\) is bounded by

\[
\begin{cases}
  2 \times (1 + \ln(N)) & \text{if } N \leq M \\
  N M^{-1} \times (1 + \ln(N)) & \text{if } N \geq M
\end{cases}
\]

**Proof.** We use the same steps as in the proof of Theorem 6. By placing the proportional-fair throughput model of Eq. (2.5) in Eq. (2.15), the improvement factor of client \(i\) is

\[
\leq \frac{n_q^{(i)} + 1}{p_q^{(i)}} \times \frac{\sum_{k=1}^{p_q^{(i)}} \frac{1}{k}}{\sum_{k=1}^{n_q^{(i)}+1} \frac{1}{k}}
\]  

(2.20)

The bound is next achieved due to the following inequality

\[
\frac{\sum_{k=1}^{p_q^{(i)}} 1/k}{\sum_{k=1}^{n_q^{(i)}+1} 1/k} \leq \sum_{k=1}^{N} 1/k \leq (1 + \ln(N))
\]  

(2.21)

\[\square\]

### 2.5 Performance Evaluation

In this section, we study the performance of RAT selection games through measurement-driven simulations. We first perform hundreds of measurements to obtain SNR values of multiple wireless access technologies in an indoor building. We next analyze the performance of these games in realistic environments.
Figure 2.5: Simulation Results for the Ideal Algorithm
(a) Average number of equilibria with 9 clients and 3 RATs, (b) Pareto-optimal/non-Pareto-optimal equilibria, (c) CDF of Pareto-efficiency gain; (d) CDF of the cardinality of Pareto-dominant sets; (e) Impact of $\eta$ on Pareto-efficiency gain; (f) Average number of per-client switchings with varying clients/RATs/$\eta$; (g) Maximum convergence time with varying clients/RATs/$\eta$; (h) Impact of noisy throughput estimations on the average number of per-user switchings.

Measurement Driven Simulations. We use the field test application in the iPhone to obtain information on the number of wireless towers, their frequency of operation and technology, and the received SNR at the receiver. Our measurements were conducted over AT&T’s cellular network. We randomly select 100 locations spread across three floors of a large university building. The measured SNR value across all locations is between -68 dBm and -104 dBm. Each client in these locations has access to UMTS/HSPA, while many locations also have access to HSPA+. The average number of towers observed across the clients (locations) is 4.

In addition to cellular statistics, we also measure the received SNR of the Wi-Fi BSs, their frequency of operation and technology (802.11 a/b/g). The average number of Wi-Fi BSs observed across all the clients is 5. These SNR values are then converted to a data-rate based on the SNR-Rate table of the corresponding technology, and are fed to our simulation.
Equilibrium Analysis. Figs. 2.5(a) and 2.5(b) correspond to the number of equilibria and their Pareto-optimality, respectively. With $M$ BSs and $N$ clients, there exists $M^N$ system states, defined as the set of BSs and the clients connected to them. We consider 9 clients each with 3 RATs: 2 Wi-Fi RATs and a 3G RAT. Thus, the total number of system states is $3^9 = 19683$. We randomly select these 9 clients from our database of 100 clients (locations), and repeat this selection for 20 times. For each realization, we consider all system states and count the number of Nash equilibria, and their Pareto-optimality. Note that while the number of Nash equilibria is dependent on $\eta$ in our RAT selection algorithm, the number of Pareto-optimal points is not, and averages to 6033 across all realizations.

Fig. 2.5(a) depicts the total number of equilibria as a function of $\eta$ for 3 different throughput models. With throughput-fair, the throughput model of all technologies is according to the relationship in Eq. (2.2), while in time-fair, the throughput model of all technologies is according to the relationship in Eq. (2.4). In the mixture mode, all Wi-Fi RATs are throughput-fair, while the 3G RATs are time-fair. With $\eta = 1$, there is an average of 200, 4 and 8 Nash equilibria in the throughput-fair, time-fair, and mixture models, respectively. Thus, only a very small number of states form the equilibria in these games. As $\eta$ increases, the number of equilibria increases rapidly, and the gap between time-fair and throughput-fair models decreases.

Fig. 2.5(b) depicts the number of Pareto-optimal and non-Pareto-optimal equilibria as a function of $\eta$ in the mixture model. We observe that by varying $\eta$, the ratio between Pareto and non-Pareto equilibria remains similar, while the individual values increase. Since increasing $\eta$ can significantly increase the number of equilibria, it has the potential to reduce convergence times without compromising Pareto-efficiency gains, shown later.

Average Pareto-Efficiency Gain. We next evaluate the Pareto-efficiency gains of Pareto-dominant points with respect to Nash equilibria. We consider prior con-
figuration setup with 9 clients and 3 RATs. For each Nash equilibrium, we consider the set of Pareto-dominant points and measure the average Pareto-efficiency gain for each Pareto-dominant point, as well as the cardinality of the Pareto-dominant set. Figs. 2.5(c) and 2.5(d) depict the corresponding CDF plots across all Nash points. We observe that the average Pareto-efficiency gain in the time-fair model is close to 1, suggesting that in time-fair models Nash points are mostly close to the Pareto-dominant points. The situation in the throughput-fair model is quite the contrary, in which for a small number of Nash points (less than 1% in Fig. 2.5(c)) the average Pareto efficiency gain can be as high as 10 with a large number of Pareto-dominant points.

Fig. 2.5(e) depicts the impact of increasing $\eta$ on the average Pareto-efficiency gains of the mixture model. As $\eta$ increases, the number of equilibria increases rapidly. However, Fig. 2.5(e) shows that limiting $\eta$ to less than 2 only slightly increases the average Pareto-efficiency gains.

**Convergence Time.** Figs. 2.5(f) and 2.5(g) depict the impact of system parameters (number of clients/RATs/$\eta$) on the average number of per-client RAT switchings and the maximum convergence time in the mixture model. Here we randomly select a given number of clients from our client database and execute our RAT selection algorithm. The randomization parameter ($p$) and the frequency of measurement prior to switching ($T$) are set to $\frac{1}{2}$ and 4, respectively. The simulation is repeated for 300 initialization points.

Fig. 2.5(f) shows that increasing the number of RATs from 2 to 5, slightly increases the average number of per-client switchings. Similarly, the number of clients has a small impact on the number of per-client switchings. Fig. 2.5(g) shows a similar trend on the maximum convergence time. Figs. 2.5(f) and 2.5(g) also show that by increasing $\eta$ to 2, the average number of per-client switchings decreases by 1. Thus, the average number of per-client time-slots to reach convergence decreases
significantly. Note that a small increase in $\eta$ does not cause serious degradation in average Pareto-efficiency gains (as observed in Fig. 2.5(e)), and therefore one can select an appropriate $\eta$ value for a given network to balance between convergence time and the desirability of the equilibria.

Further, note that the total number of states ($M^N$) provides an upper bound on the maximum convergence time. However, our results in Figs. 2.5(f) and 2.5(g) show that with appropriate parameter selection ($\eta, p, T$), the number of concurrent switchings and oscillations would be negligible, and the system will converge to an equilibrium in a very small number of steps.

**Impact of Noisy Measurements.** Since the RAT selection algorithm relies on correct throughput prediction on RATs, sensitivity to noisy estimates can become a bottleneck. In Fig. 2.5(h) we plot the impact of such noise on the average number of switchings for the mixture model. We model the noise by assuming that the predicted throughput is according to a Gaussian distribution in which the mean is equal to the actual throughput and the standard deviation is equal to the product of the noise value and the actual throughput.

Fig. 2.5(h) shows that increasing the noise power increases the average number of switchings. Further, it is possible for some of the clients to keep on switching without reaching convergence. This problem can be addressed by adapting the $\eta$ value according to the noise power. By increasing the $\eta$ value, a client requires higher throughput values to make a change, compensating for noisy throughput estimates.

### 2.6 Related Work

There exist a large number of studies on network selection in HetNets. We highlight the crucial differences in the models and analysis between the work in this chapter and the most relevant samples.
Congestion Games and RAT Selection. Congestion games model the congestion externalities when clients compete for limited resources. The idea here is that each client pays a client-specific cost $c_i(x)$ when it uses resource $r$, which depends on the congestion level $(x)$ and the specific preference of the client for $r$. The congestion impact of a client on a resource $r$ is denoted by a weight. The congestion level $(x)$ of resource $r$, is then the sum of the weights of the clients that select $r$. Each client in these games aims to minimize its own cost. Over the last few decades several papers have studied the convergence properties of different classes of these games. Majority of these proofs is based on giving potential functions (functions in which the gain (loss) observed by any client’s unilateral move, is the same as the gain (loss) in the potential function). The convergence properties of a subclass of these games with separable preferences and player-independent costs was studied in [38]. Our proof in Theorem 1 is an application of [38] to the class-1 RAT selection games. The convergence properties of congestion games with separable preferences and player-independent weights was studied in [39]. Our class-2 throughput models have similarities to the games studied in [39]. However, unlike [39] (and the majority of convergence proofs in related work such as [38,71,76,95]), we present a new proof methodology [Theorem 2] that does not rely on potential functions. More importantly, a key issue we must face in RAT selection games is that different technologies have different classes of throughput models. None of the prior work in game theory has studied the equilibria properties when a mixture of classes in considered.

Fairness and Pareto-Efficiency. Pareto-efficiency is a desirable outcome for non-cooperative games. Over the last few decades several fairness concepts that achieve Pareto-optimality have been introduced [19,75]. Our metric of average Pareto-efficiency gain quantifies the distance of Nash equilibria with respect to Pareto-dominant points. Similar concepts have been recently introduced in [16,57] for load balancing, but do not apply to RAT selection games. Other work introduced the
concepts of price of anarchy (PoA) and price of stability (PoS). PoA bounds the distance of any Nash point with respect to an optimum defined by a social welfare function (e.g., sum of throughputs). PoS bounds the distance of the best Nash from the social optimum. In contrast, the Pareto-efficiency metric is more general and fits the questions about RAT selection better. However, one can still derive upper bounds on PoA and PoS in RAT selection games based on our proposed techniques on average Pareto-efficiency gains.

**Game Theory Applications in Network Selection.** Congestion game based network selection was considered in [26, 53]. However, the model in [26] does not capture the multi-rate property of HetNets, while the model in [53] assumes only a single BS in each class of throughput models. As we will show later, in general multi-rate, multi-BS RAT selection games, convergence cannot be always guaranteed. Other work considered evolutionary game models to study the problem of network selection [84,97]. In evolutionary games, a group of players form a population, and players from one population may choose strategies against clients from other populations. These games assume a large number of clients in which each of them has a negligible impact on others. This is not the case with RAT selection games in which an individual client has a major impact on the performance of all other clients.

### 2.7 Conclusions

In this chapter, we studied the dynamics of RAT selection games in heterogeneous wireless networks, where control of RAT association has been placed at the client-side with the network providing big-picture information regarding the client-RAT association and peak physical layer rates for all other clients. We investigated the convergence properties of these games and introduced hysteresis as a system parameter that can guarantee convergence for all clients. Tight bounds are also provided for the average
Pareto-efficiency gains of RAT selection games. Finally, through measurement-driven simulations, RAT selection games are shown to converge to Nash equilibria within a small number of re-associations or switches.

However, the assumption that the network is able to provide an accurate and timely big-picture view of all other clients’ behavior is a strong and potentially inaccurate assumption. Furthermore, the assumption that throughput can be exactly estimated using both Class-1 and Class-2 without concern for channel conditions is useful for the theoretical bounds in this Chapter, but may not easily translate to accurate throughput estimations. In the next chapter, we address these problems by (1) relaxing the assumption that the network provide accurate information on all other clients in the HetNet, and (2) develop a RAT selection algorithm based on the multi-armed bandit that operates with minimal time-averaged statistics.
Chapter 3

Partial Network Assistance:
Minimizing Regret and Switching Costs

3.1 Introduction

There exist many new technologies coming into common use for HetNets: in particular, millimeter wave (mmWave) radio is of growing interest for deployment in next-generation 5G networks [9,24,90]. With the potential for extremely high throughput compared to sub-28GHz [9], mmWave can exploit the enormous amount of spectrum available in these bands. It is likely mmWave Radio Access Technologies (RATs) will co-exist with existing technologies such as 3G, LTE, and 802.11(Wi-Fi) in a heterogeneous network (HetNet).

A client in this HetNet scenario can access different RATs to download data; however, due to the drastically different timescales involved for each technology and the signaling overhead involved for practical implementation, it is unlikely that the networks will provide the big-picture view or the peak physical layer rates of other
Figure 3.1: Temporal variations in LTE/802.11 vs mmWave
Example of slower throughput fluctuations ($x_j$) in LTE/802.11 (left) and faster fluctuations in Millimeter Wave (right).

clients as in the previous chapter. At best, the client may be able to know some link-specific information regarding the wireless channel. We address this scenario of optimal access network selection in HetNets with only partial information from the network in this chapter, where we answer the important question of how should a client optimally select the best access network in a HetNet with mmWave to maximize throughput and minimize switching costs?

We solve this problem using a client-centric approach which does not require extensive signaling and coordination among the different access networks, an unrealistic assumption if the networks are owned by different operators. In addition, we recognize that the client is often better positioned to monitor both the set of accessible RATs and changes in the client’s own data demands. However, there is a cost associated to switching between RATs: client-incurred RAT-specific overhead due to wireless network handoff, and such costs may add up to be substantial.

As a client switches between different RATs, the lack of perfect information available to the client results in suboptimal throughput and switching costs when maximizing throughput. We model this problem of maximizing throughput and minimizing switching costs with the model of online learning in a stochastic Markovian environment [72], where the client must learn accessible RATs’ variable statistics through experimentation. Specifically, we formulate RAT selection in mmWave HetNets by extending the standard Multi-Armed Bandit Problem (MABP) to incorporate two critical elements: (1) HetNet switching overheads and (2) Markovian characteristics of
mmWave RATs, thereby accounting for access to high-throughput-but-highly-variable mmWave RATs as shown in Fig. 3.1.

We present a distributed algorithm for RAT Selection in HetNets with mmWave that balances maximizing aggregate throughput and minimizing wireless switching overheads. It operates on the UE client device, using past empirically-derived throughput values from past RATs the UE has associated with. Our metric is the performance gap between our algorithm and an offline-optimal algorithm, and show that we obtain optimal $O(\ln(t))$ total regret (loss due to uncertainty and switching costs), instead of only the regret component characterizing throughput in [101]. Furthermore, our algorithm achieves better total obtained throughput in simulations compared to other candidates that do not operate under any such switching constraint. This better throughput result while under switching constraints is a direct consequence of the tighter Chernoff bound [65] used by parameters in our algorithm.

Our main contributions in this chapter are as follows [103]:

- We model the temporal behavior of mmWave RATs as a finite-state, irreducible, aperiodic Markov Chain with a general non-reversible, unknown transition matrix $P$. (Section 3.2) We formulate the problem of Stochastic RAT Selection in HetNets with mmWave RATs under unknown statistics as a rested MABP (Section 3.3.1);

- We develop an online learning policy that is distribution-independent (mmWave HetNet Selection Algorithm, Theorem 8) and solves the RAT Selection problem while minimizing cost of switching (Section 3.3.2) by grouping channel access;

- We show that the total regret (loss due to uncertainty and switching costs) of mHS is upper-bounded by $O(\ln(t))$ in time, optimal by [62] (Section 3.3.3); and

- We use real mmWave characteristics in our numerical results to compare our policy with those existing in literature [101], show our solution outperforms
existing policies in the total obtained throughput, and discuss the client implications of implementing this system. (Section 3.4)

From our results, a client using mHS in a HetNet with highly-variable-throughput mmWave RATs obtains the best of both worlds, i.e., order-optimal throughput performance and switching minimization, irrespective of whether mmWave or other traditional RATs are optimal.

3.2 System Model

3.2.1 Millimeter Wave Radio and HetNets

Next-generation mobile networks are expected to make extensive use of millimeter wave (mmWave) radio technology [24]. We model the 3-state mmWave RAT as a 3-state discrete-time stochastic process, illustrated in Fig. 3.2 with transition probabilities \( \{ P_{ij} \} \). The mmWave channel is composed of Line-of-Sight (LOS), non-LOS and outage states. In LOS, the mobile device has an unobstructed path for the signal to propagate [51]—potentially obtaining peak rates tens of Gbps. In non-LOS, the mmWave channel severely degrades to a much lower data rate [9, 90] due to obstructed signal paths. Outage occurs when very little signal is observed due to physical obstacles and atmospheric absorption [40], and can be treated as if no signal were received. The mmWave RAT channel is in one of these states at all times, and the non-deterministic transitions between individual states obey Markovian properties [72] when accessed by a mobile client (user). The channel’s data rate is also non-deterministic because of fast-fading and shadowing effects, and can be modeled as a random variable dependent on the number of clients and the mmWave channel state.

Thus, any mobile device that performs RAT selection must do so by predicting the state of the mmWave RAT based on the channel’s stochastic parameters, which
may not be known to the client. Furthermore, mmWave channels change state and channel quality on the order of a millisecond [90] or less, much faster than comparable technologies like LTE/802.11. These fast state changes demand RAT selection on a timescale precluding traditional centralized, network-operated control schemes for UE-RAT association, as there is very little time to aggregate all UEs’ channel state information, centrally calculate an optimal association, and return UE-RAT configurations to all UEs. These differences necessitate new UE-centric control of HetNets with mmWave.

Due to frequent transitions between mmWave RAT states, many RAT selection policies that converge to an optimal client-BS assignment encounter issues from mmWave’s non-i.i.d. nature on the same timescale that LTE/802.11 may be considered independent in time. In the next section we examine a system model and Stochastic RAT selection policy that (1) makes RAT selection decisions on the same timescale as mmWave state transitions, (2) leverages properties of the mmWave transition matrix $P$, and (3) considers LTE/802.11 RATs consisting of a single state to be degenerate Markov chains used to describe mmWave.

We model a client’s heterogeneous wireless environment to be composed of $M$ Base Stations (BSs). Here, “BS” represents a generic Node-B in 3G, eNodeB in 4G, Access Point in 802.11, and mmWave Base Stations, and $M = \{1, ..., M\}$ denotes the set of BSs. $n_j$ denotes the number of clients on BS $j$, and each client has access to a subset of the BSs. We assume that all RATs are interference-free due to frequency
Figure 3.3: Heterogeneous Network Example with mmWave access
Each client has access to a subset of BSs

separation between different-RAT BSs, and frequency reuse among same-kind-RAT BSs. An example of this HetNet environment is shown in Fig. 3.3. We model beacon signals belonging to different BSs received on the same RAT by a single client as being received from multiple RATs to simplify analysis, using the term “RAT” to mean “BS” interchangeably.\(^1\) Furthermore, due to battery considerations for mobile devices, we assume that each client is willing to use at most one RAT at any given time. In addition, we assume that the mmWave RATs are rested: the mmWave channel does not undergo transitions unless the client interacts with it. This can be assumed because the client is stationary over the timescale of the algorithm, which can be performed on the order of sub-milliseconds, and client mobility is the primary driver of mmWave state changes.

3.2.2 Online Learning and Regret in highly variable HetNets

In the HetNets scenario, there are multiple choices of RATs (arms) for a client (player), but the client can only associate with a one RAT in any given discrete time step. At each time step, the client is able to download data (obtain a reward) from its associated RAT. The obtained throughput for a client from each RAT is unknown \textit{a priori} to that client and differs between RATs. The goal of the client is therefore to maximize the expected sum data over a period of time—in other words,

\(^1\)For example, an 802.11g interface may be capable of receiving data on Channels 1, 6 and 11 in the 2.4GHz ISM band, but we consider the client as having 3 distinct RATs.
how should one select a RAT to download data from by learning the statistics of each RAT from its past observations. Furthermore, this is complicated by the existence of fixed, RAT-specific overhead costs incurred from switching.

This is an example of the fundamental tradeoff between exploration and exploitation. The client needs to explore, or sample/associate with, all available RATs to discover the best option—and to prevent repeated sampling of suboptimal RATs that it wrongly believes to be best. On the other hand, the client also needs to maximize total obtained throughput, as well as to avoid excessive switches. The client’s total regret is a concept typically used to measure how well an algorithm performs.

Regret is the difference between the expected total obtained throughput from an ideal policy with perfect knowledge, and the expected total obtained throughput of the policy in question. It is a measure of the inefficiency generated by uncertainty in the network and the decision-making process. We focus on regret relative to the infeasible “ideal best single RAT” policy, which relies on perfect a priori knowledge of the throughput (reward) distributions of all RATs. The key challenge is to upper-bound regret as tightly as possible.

Our design goals for such a UE-centric RAT selection algorithm for HetNets with mmWave require the following:

- A UE-centric design to remove inefficient signaling and delay overhead present in network-centric control designs, such that UEs may respond more quickly to channel conditions changing on the faster mmWave timescale.

- Maximize aggregate throughput obtained in order to efficiently obtain data for UE operation.

- Minimize total switching costs from moving between different RATs (e.g., soft/hard handover, session handover).
There exists an extensive body of work on applying online learning-type algorithms to regret minimization in MABPs \([8,15,62,101]\). However, many of these are insufficient to meet all the design goals: \([15,62]\) provide policies to maximize aggregate throughput but do not take into account the Markovian nature of state transitions or aggregate switching costs, essential to minimizing overhead in today’s highly-loaded wireless network infrastructure. \([8]\) minimizes switching between available options assuming i.i.d. rewards from all options, without accounting for frequent and fast-changing mmWave RATs in HetNets. The work in \([101]\) comes closest to ours: they address reward maximization given Markovian (not i.i.d.) rewards, but any analysis of switching costs and modeling their interaction with throughput maximization is absent.

### 3.3 Online Learning for Stochastic RAT Selection

In this section, we describe the client-centric RAT selection problem with mmWave, our proposed mHS (mmWave HetNet Selection) algorithm, and show our main result of optimal \(O(\ln (t))\) regret in Theorem \([8]\).

#### 3.3.1 Problem Formulation

We formulate the problem as a rested Multi-armed Bandit, where a UE selects a single RAT to associate with at every time slot. At each time slot, the UE obtains a throughput dependent on the distribution for that given RAT but does not know the state of the RAT. Under the rested assumption, RATs with Markovian state transitions do not evolve or change states over time when not played: the RAT evolves according to the underlying Markov law only when played.

We describe the parameterized throughput distributions for each of the \(M\) RATs by \(f_1(x; \theta_1, n_1), ..., f_M(x; \theta_M, n_M)\) over a measure \(\nu\) for the channel throughput. Both \(f_j(\cdot; \cdot)\) and \(n_j\) are known from the RAT, and \(\theta_j\) are unknown channel conditions
Table 3.1: Chapter 3 Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Number of RATs</td>
</tr>
<tr>
<td>$n_j$</td>
<td>Number of clients on RAT $j$</td>
</tr>
<tr>
<td>$f_j(x; \theta_j, n_j)$</td>
<td>Throughput reward distribution given $\theta_j, n_j$</td>
</tr>
<tr>
<td>$\theta_j$</td>
<td>Hidden (from the UE) parameters for RAT $j$</td>
</tr>
<tr>
<td>$x_j$</td>
<td>Throughput obtained by client on RAT $j$</td>
</tr>
<tr>
<td>$\mu(\theta_j) = \mu^j$</td>
<td>Mean value of $x_j$</td>
</tr>
<tr>
<td>$\alpha(t)$</td>
<td>RAT chosen by selection policy at time $t$</td>
</tr>
<tr>
<td>$\mu^*$</td>
<td>Highest mean throughput over all RATs</td>
</tr>
<tr>
<td>$X(t)$</td>
<td>Total data by client over $t \in [1, t]$</td>
</tr>
<tr>
<td>$\bar{X}^j(t)$</td>
<td>Sample mean data for RAT $j$ over $t \in [1, t]$</td>
</tr>
<tr>
<td>$R_\alpha$</td>
<td>Total Regret</td>
</tr>
<tr>
<td>$T^j(t)$</td>
<td>Number of samples of RAT $j$ by policy $\alpha$ over $t \in [1, t]$</td>
</tr>
<tr>
<td>$R^S_\alpha$</td>
<td>Sampling Regret</td>
</tr>
<tr>
<td>$R^S_\alpha$</td>
<td>Switching Regret</td>
</tr>
<tr>
<td>$m_j$</td>
<td>Peak throughput obtainable on RAT $j$</td>
</tr>
<tr>
<td>$P_j$</td>
<td>Probability Transition matrix for RAT $j$</td>
</tr>
<tr>
<td>$c_a^j$</td>
<td>Cost of Association to RAT $j$</td>
</tr>
<tr>
<td>$c_d^j$</td>
<td>Cost of Dissociation from RAT $j$</td>
</tr>
<tr>
<td>$s_{ja}^j$</td>
<td>Number of Switches to RAT $j$</td>
</tr>
<tr>
<td>$s_{jd}^j$</td>
<td>Number of Switches away from RAT $j$</td>
</tr>
<tr>
<td>$N_{fk}$</td>
<td>Timeslots the client can switch RATs</td>
</tr>
<tr>
<td>$f$</td>
<td>Frame number</td>
</tr>
<tr>
<td>$b_f$</td>
<td>Length of block in frame $f$</td>
</tr>
<tr>
<td>$k_f$</td>
<td>Number of blocks in frame $f$</td>
</tr>
<tr>
<td>$t(f, k)$</td>
<td>First time slot in frame $f$, block $k$</td>
</tr>
<tr>
<td>$N_f$</td>
<td>Last time slot at the end of frame $f$</td>
</tr>
</tbody>
</table>

Belonging to some bounded parameter set $\Theta$. Assuming that the distribution is bounded, the mean is

$$
\mu(\theta_j) = \int_{-\infty}^{\infty} x f(x; \theta) dv(x)
$$

and we define the optimal mean over all $j$ to be

$$
\mu^* = \max\{\mu(\theta_1), \ldots, \mu(\theta_M)\} = \mu(\theta^*) = \mu(\theta_{j^*})
$$
The total data obtained by a RAT selection policy $\alpha = \{\alpha(1), \alpha(2), \ldots \}$ by time $t$, where $\alpha(t) \in \{1, \ldots, M\} \forall t$, is

$$X(t) = \sum_{i=1}^{t} x(i)$$

(3.3)

The expected total data obtained by time $t$ is therefore

$$E_\alpha[X(t)] = \sum_{i=1}^{t} \mu(\theta_j) E_\alpha[T_j(t)]$$

(3.4)

where $T_j(t) = \sum_{i=1}^{t} 1(\alpha(i) = j)$ is the total number of time slots that policy $\alpha$ has sampled the distribution $f_j(x; \theta_j, n_j)$.

We characterize $R_\alpha$, total regret for the policy $\alpha$, as the sum of the sampling regret $R^T_\alpha$ (loss due to suboptimal throughput) and the switching regret $R^S_\alpha$ (loss due to total switching costs):

$$R_\alpha = R^T_\alpha + R^S_\alpha$$

(3.5)

where

$$R^T_\alpha(t) = t\mu^* - E_\alpha[X(t)]$$

$$= \sum_{j: \mu < \mu(\theta_j)} (\mu^* - \mu(\theta_j)) E_\alpha[T_j(t)]$$

(3.6)

and

$$R^S_\alpha(t) = \sum_{j=1}^{M} c^a_j \sum_{j=1}^{M} c^d_j E_\alpha[s^a_j(t)] + \sum_{j=1}^{M} c^d_j E_\alpha[s^d_j(t)]$$

(3.7)

where $c^a_j, c^d_j$ are constant RAT-specific costs of association and dissociation and $s^a_j(t), s^d_j(t)$ are the number of times up to $t$ the client has associated and dissociated from RAT $j$. 
The UE wishes to maximize its total throughput less the switching regret, \( E_\alpha[X(t)] - R^S_\alpha(t) \): this is equivalent to minimizing the total regret over time \( R_\alpha(t) \).

Throughout the rest of the chapter, we make the following assumptions:

1. The parameters \( \{\theta_j\} \) are such that there exists \( \mu(\theta_j) = \mu^j < \mu^* = \mu(\theta_{j^*}) \), for all \( j \neq j^* \).
2. \( |x_j(t) - E[x_j(t)]| \leq m_j \forall j \).
3. \( P_j \) and \( P_j^*P_j \) are irreducible, where \( P_j^* = \text{adjoint}(P_j) \).

The first assumption implies there exists a unique “best” RAT amongst set of available RATs: it is unlikely for multiple RATs to be perfectly identical. The second implies that the deviation of the throughput values are bounded by \( m_j \), because throughputs must be finite. Finally, the last assumption is weaker than reversibility of Markov Chains. Note that we do not require the transition matrices \( P_j \) to be reversible, since reversibility is an unrealistic assumption for all mmWave RATs.

### 3.3.2 Network Selection Algorithm

For this type of exploration vs exploitation problem with constant RAT-specific costs per switch, it is intuitive to see that any asymptotically efficient policy must ensure that samples from the same RAT are grouped as much as possible to minimize the number of switches between different RATs. We use a Switching Scheduler Algorithm (Alg. 2) inspired by the block switching algorithm in [8]. This algorithm outputs \( N_{fk} \), a schedule of time slots at which a UE is allowed to switch RATs to minimize switching costs.

The Switching Scheduler scheme for mmWave HetNets, inspired by [8], first divides discrete time into “frames” \( f = 0, 1, 2, \ldots \). Each frame is further divided into “blocks” of equal duration \( b_f \), numbered \( k = 1, 2, \ldots, k_f \), labeled by \( (f, k) \). Table 3.2 shows how to determine block and frame lengths. \( N_f \) is the final time slot in frame
Algorithm 2: Switching Scheduler Algorithm

\textbf{Input:} Number of accessible RATs $M$

1. Initialization: Switching Schedule $N_{fk} = \{1, \ldots, M\}$, frame index $f = 1$, block index $k = 1$

2. \textbf{while} $f \geq 1$ \textbf{do}

3. \hspace{1em} \textbf{while} $k \leq k_f = (N_f - N_{f-1})/b_f$ \textbf{do}

4. \hspace{2em} $t(f,k) = N_{f-1} + ((k-1)b_f + 1)$

5. \hspace{2em} $N_{fk} = N_{fk} \cup t(f,k)$

6. \hspace{2em} $k = k + 1$

7. \hspace{1em} \textbf{end}

8. \hspace{1em} $f = f + 1$

9. \textbf{end}

Table 3.2: Switching Scheduler Algorithm: Block/Frame Lengths

<table>
<thead>
<tr>
<th>Frame $(f)$</th>
<th>$b_f$</th>
<th>$N_f - N_{f-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>$M$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$\left\lceil \frac{e_{12}-e_{02}}{M} \right\rceil \cdot M \cdot 1$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>$\left\lceil \frac{e_{22}-e_{12}}{M} \right\rceil \cdot M \cdot 2$</td>
</tr>
<tr>
<td>$f$</td>
<td>$f$</td>
<td>$\left\lceil \frac{e_{f2}-e_{(f-1)2}}{M} \right\rceil \cdot M \cdot f$</td>
</tr>
</tbody>
</table>

$f$, and $t(f,k)$ is be the first time slot in the block $(f,k)$. An example schedule for an $(M=3)$-RAT system is shown in Fig. 3.4.

The algorithm initializes $N_{fk}$ to be all time slots $1, \ldots, M$, to allow the UE to sample all $M$ RATs. Next, in each frame, the number of blocks is determined by dividing the frame duration $N_f - N_{f-1}$ by $k_f$, the block duration. The intuition for Alg. 2 is to restrict switching opportunities as time goes on, because the need to refine throughput estimates of other RATs decreases with time due to averaging in the sample mean throughput in mHS (Alg. 3).

Once $N_{fk}$ is determined \textit{a priori}, it is fed to the online mHS Algorithm (Alg. 3), which calculates which RAT to switch to at the allowed switching opportunities $t \in N_{fk}$. The algorithm starts by sampling throughputs for all $M$ RATs during the first $M$ time slots to initialize sample means for each RAT. Next, at each $t \in N_{fk}$, the algorithm calculates a ranking index, $\text{index}^j(t)$, for all RATs, and switches to the RAT

Figure 3.4: Example of switching schedule for (M=3)-RAT HetNets.
with the highest rank. This ranking index balances the client’s need for exploitation (selecting the RAT with the largest $\bar{X}_j(t)$), and an exploration term that encourages occasional samples of suboptimal RATs to update sample mean throughputs. This term depends on $m_j$, the maximum throughput obtainable, $L_j$, a parameter depending on the state transition matrix of the UE-RAT channel, and $T^j(t)$, the number of samples of RAT $j$ up to time $t$. This exploration term grows with $\sqrt{\ln(t)}$, causing the UE to associate with less-sampled RATs whenever it becomes large relative to all other ranking indexes in order to update sample throughputs.

**Algorithm 3: mHS: mmWave HetNet Selection Algorithm**

**Input:** Switching Schedule: $\{N_{fk}\}$, RAT-specific parameters: $\{m_j, L_j\}$

1. Initialization: time $t = 1$
2. while $t > 0$ do
3.  if $t \in N_{fk}$ then
4.    if $t \leq M$ then
5.      Play RAT $t$
6.    else
7.      Calculate for all $j \in M$:
8.      $\bar{X}^j(t) = \frac{x_j(1)+x_j(2)+...+x_j(T^j(t))}{T^j(t)}$
9.      index$^j(t) = \bar{X}^j(t) + m_j \sqrt{\frac{L_j \ln(t)}{T^j(t)}}$
10.     Play the RAT with $\arg\max_j \{\text{index}^j(t)\}$
11.   else
12.     Continue playing the RAT played in the previous timeslot $t - 1$
13. end
14. end
15. end

### 3.3.3 Upper Bound on the Total Regret

Using existing policies [101], it is possible to achieve $O(\ln(t))$ sampling regret for maximizing throughput for Markovian HetNets. However, with the addition of switching costs between RATs, these policies can incur arbitrarily high total regret due to unconstrained switching. We show that our mHS policy in Alg. 3 still has a total regret of $O(\ln(t))$ for mmWave HetNets with a $o(\ln(t))$ switching regret. The upper-bound on total regret in Theorem 8 holds for $L_j$ sufficiently large, which depends on the
smallest value of the spectral (eigenvalue) gap $\varepsilon(Q_j)$ over all RATs $j$ in the HetNet for $Q_j = P_j^*P_j$, where $P_j$ is the transition matrix for the $j$th RAT. To calculate $L_j$, both a limit on the absolute deviation $|x_j - E[x_j]| \leq m_j$, and the spectral (eigenvalue) gap can be obtained from each accessible RAT (e.g., estimation, historical values). Note that the choice of $L_j$ does not change the logarithmic behavior of regret: however, picking a larger $L_j$ will result in more frequent exploratory actions, and change both the constant $O(1)$ and the coefficient of $O(\ln(t))$. By picking smaller values of $L_j$, the index policy emphasizes the sample mean $\bar{X}_j(t)$: in the limit where $L_j = 0$, the policy simply selects the RAT with the largest sample mean throughput at all times, resulting in very fast (but potentially suboptimal performance due to very few samples from alternate RATs); conversely, larger values of $L_j$ emphasize the exploration term, which results in slower convergence times.

However, in our main result, we show that lower-bounding the exploration parameter $L_j$ guarantees an upper-bound on the total regret uniformly over time, with regret coefficient dependent only on $(m_j, L_j)$. The lower-bound of $L_j(m_j, \varepsilon_{\text{min}}) \geq \left( \frac{48+180/m_j}{\varepsilon_{\text{min}}} \right)^2$ with constants from the Chernoff bound in [65], and smallest spectral (eigenvalue) gap $\varepsilon_{\text{min}} = \min_j \{\varepsilon(P_j^*P_j)\}$ gives us our result with $C'(m_j, L_j) = \frac{4m_j^2L_j}{(\mu^* - \mu)^2}$ and $C(m_j, L_j) = \frac{4m_j^2L_j}{\mu^* - \mu^*}$.

We present our main result, which upper-bounds the total regret for the mHS algorithm in mmWave HetNets:
Theorem 8. Under the mHS RAT selection algorithm, for all $j \in M$ such that $\mu(\theta_j) < \mu^*$,

The expected number of samples of a suboptimal RAT $j$ is upper-bounded by

$$i) E_\alpha[T^j(t)] \leq [C'(m_j, L_j) + o(1)] \ln(t),$$  \hspace{1cm} (3.8)

the expected number of switches to and from RAT $j$ is upper-bounded by

$$ii) E_\alpha[s^j_d(t)] \leq E_\alpha[s^j_a(t)] \leq o(\ln(t)),$$  \hspace{1cm} (3.9)

and the total regret over all $M$ RATs is upper-bounded by

$$iii) R_\alpha(t) \leq \left[ \sum_{j: \mu_j < \mu^*} C(m_j, L_j) + O(1) \right] \ln(t)$$  \hspace{1cm} (3.10)

Proof (Sketch): In order to bound the total regret, first we bound the expected number of times a client samples a suboptimal RAT, $T^j(t)$, at the end of a frame $d$ in our Switching Scheduler Algorithm. We choose a parameter $L_j$ sufficiently large to ensure convergence of the expectation $E_\alpha[T^j(t)]$ according to the Chernoff bound in [65]. We extend this result to all time in the frame, which gives us (3.8). Next, we bound the number of times a client switches to and from ($s^j_a(t)$ and $s^j_d(t)$ respectively) a RAT at time $N_{d-1} < t < N_d$: taking the expectation and using the bounds on $N_d$ in [8], we obtain the upper bound on $E_\alpha[s^j_a(t)]$ and $E_\alpha[s^j_d(t)]$ (3.9). Finally, we combine (3.8) and (3.9) using (3.5) to obtain the $O(\ln(t))$ bound (3.10). See Appendix 3.7.1 for the full proof.

Theorem 8 shows that, for the more general case of non-reversible Markovian behavior of throughput of a given RAT (e.g., mmWave RATs), the mHS algorithm is able to achieve the optimal $O(\ln(t))$ total regret as in the i.i.d. case with switching costs in [8].
Remark 1. Policies like mHS that achieve $O(\ln(t))$ regret achieve the optimal time-averaged total obtained throughput when compared with an “ideal best single RAT” policy. This can be seen in the asymptotic lower bound

$$\frac{R_\alpha(t)}{t} \to 0 \implies \frac{E_\alpha[\text{total reward}]}{t} \to \mu^*$$

(3.11)

In general, any policy that has sub-linear regret growth over time achieves the optimal time-averaged throughput.

It is well known that $O(\ln(t))$ sampling regret is order-optimal asymptotically for a policy that is uniformly good for any Markovian throughput (reward) distribution [62], [101]. However, mHS is order-optimal uniformly over time for total regret, rather than only asymptotically as in [62] and [8].

3.4 Performance Evaluation

<table>
<thead>
<tr>
<th>RAT</th>
<th>Rate Vector (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTE [1]</td>
<td>1.16,1.477,1.914,2.406,2.73,</td>
</tr>
<tr>
<td>(@10MHz BW)</td>
<td>3.32,3.9,4.52,5.12,5.56</td>
</tr>
<tr>
<td>2 802.11g</td>
<td>[6,9,12,18,24,36,48,54]</td>
</tr>
<tr>
<td>3 802.11n</td>
<td>[60,120,180,240,360,480,540,600]</td>
</tr>
<tr>
<td>4 802.11g</td>
<td>[6,9,12,18,24,36,48,54]</td>
</tr>
<tr>
<td>5 802.11n</td>
<td>[60,120,180,240,360,480,540,600]</td>
</tr>
<tr>
<td>6 mmWave-NLOS</td>
<td>[20,40,60,80,100]</td>
</tr>
<tr>
<td>mmWave-LOS [0]</td>
<td>[1000,1100,1200,1300,1400]</td>
</tr>
</tbody>
</table>

We compare the performance of mHS with the “Baseline” UCB policy in [101], a UCB-based algorithm for the Markovian MABP that does not consider switching costs, and a “Naive” policy that selects the RAT with the best last-observed throughput. Fig. 3.5 show simulation results for 6 RATs in a mmWave HetNet averaged over 100 runs, with parameters given in Table 3.3 and both association/dissociation
mHS achieves (a) lower total regret over time, (b) lower switching regret, and (c) greater total obtained throughput than Baseline and Naive policies. (d) Increasing the number of clients increases regret more quickly during exploration; (e) Greater throughput gap between optimal and other RATs result in lower regret; (f) Increasing number of accessible RATs increases regret.

switching costs of 25% of the minimum PHY throughput on each RAT. The mmWave transition matrix $P$ is valued as in [72].

### 3.4.1 Simulation Results

Fig. 3.5a shows the total regret over time: the mHS algorithm has a slower growth of regret compared to the other policies. Fig. 3.5b shows that the switching costs incurred by mHS is significantly less than both comparison policies: it is only after time $t = 5350$ for these experiments that the switching regret exceeds the Naive policy, because the Naive policy eventually locks onto a single (possibly incorrect) RAT and no longer incurs switching regret. Fig. 3.5c completes the comparison by examining the total throughput of the three policies, and our method obtains over 24% more total obtained throughput [101].
Note that in these figures, the mHS algorithm undergoes a long period up to about $t = 5350$ on average in our simulations, where total regret is constant. The high throughput available on the mmWave RAT results in a large $m_j$, causing clients to remain on mmWave instead of exploring other RATs until after the transient period. The flat regret curve is a consequence of the policy selecting the “optimal” (max mean throughput) RAT, and therefore incurring no regret from selecting the optimal RAT. However, these periods of nonincreasing regret are transient: by design, the exploration terms in mHS for other RATs grow until they force a period of exploration, and the UE switches to different RATs to refine its knowledge of the RATs’ average throughputs.

In Fig. 3.5d we plot the average total regret of clients over time in a HetNet system with a variable number of other clients randomly distributed on the $M$ RATs, which move to and from RATs under a Poisson probability with mean interarrival/departure times of 10. Throughputs on each RAT are shared in a time-fair manner (see Appendix 3.7.2). By increasing the number of clients, regret increases more quickly over time when the UE explore alternative RATs: this is caused by clients obtaining less throughput per RAT on average, so the exploration term causes clients to explore more frequently to obtain better throughputs.

We simulate the performance of our algorithm when we scale the potential throughputs on the mmWave RAT by multiplying by $\{1/40, 1/30, 1/20, 1/10\}$ in Fig. 3.5e. By reducing peak mmWave throughputs to values comparable to other RATs, total regret incurred by mHS increases. When a client has to learn the statistics of similar-throughput-RATs, more regret is incurred as it takes longer to differentiate between them: our algorithm functions better when there is a larger gap between the available throughputs on different RATs. In Fig. 3.5f the number of RATs is increased causing regret to grow over a longer time period, as well as obtain a higher total regret (indicative of the extra time to learn each RAT).
For a UE with access to highly-variable-throughput RATs, such as mmWave, in HetNets, mHS achieves an optimal tradeoff between maximizing expected throughput and minimizing switching costs, to minimize the total regret for a client, as well as significantly outperforming other techniques in terms of the total data downloaded. In practical terms, for time step durations on the order of channel coherence times for mmWave (1 ms [90]), clients can identify and lock onto the optimal mmWave RAT in only a few ms to minimize regret. In general, our technique is most applicable when mmWave RATs allow extremely high throughputs, while allowing the client to occasionally explore alternatives. It obtains $O(\ln(\tau))$ regret even if the throughput advantage of mmWave is significantly degraded to below that of LTE/802.11n.

### 3.5 Related Work

**RAT Selection in HetNets:** Optimal selection of wireless access networks in HetNets has been studied in various contexts. Many approaches invoke optimal association of clients to RATs [109], or use RAT-centric techniques to guide client decisions towards an optimal system state [32,34]. Alternatively, client-centric approaches [14] seek to shape client perspectives such that the outcome of local client decisions approaches a global optimum. However, these strategies generally assume that channel statistics change slower than algorithm convergence. In our work, we consider both stochastic behavior for HetNet channels and optimality from the client perspective in terms of the average regret observed during RAT selection.

**Millimeter Wave RATs:** Studies in mmWave frequencies [9, 92, 100] show its potential as a possible high-throughput access network. However, there is a need for network selection algorithms that are able to function with its dynamic channel statistics, which may change during a single session. Specifically, mmWave RATs have been shown [55] to exhibit highly variable SNR and bitrates, which motivates the need for adaptive network selection algorithms that incorporate this behavior. Authors in [72]
have modeled the mmWave RAT as a Markov chain with pre-determined PHY-layer rates and perform handovers assuming steady-state channel conditions and known state transition probabilities. In our work, we do not assume perfect knowledge of transition probabilities, allow for parameterized distributions of rates in each state, and do not assume steady-state channel conditions.

**Multi-Armed Bandits:** For the classical multi-armed bandit problem with \textit{i.i.d} rewards for the arms/actions, the seminal work of \cite{62} shows that the asymptotic lower bound on the regret of any uniformly-good policy scales logarithmically in time (O(\ln(t))). The authors in \cite{8} provide a distribution-dependent block-based online policy to minimize switching costs for the MABP with stationary and \textit{i.i.d} reward distributions. The authors in \cite{101} propose a distribution independent index based policy (similar to the UCB policy in \cite{15}) for the MABP with Markovian rewards, which can incur a high switching cost. In our work, we consider the more general case of Markovian rewards under irreversibility characteristic of mmWave and provide a simpler online learning policy, which is distribution-independent and incurs only logarithmic regret with sub-logarithmic switching regret.

### 3.6 Conclusion

In this chapter, we studied the problem of stochastic RAT selection in HetNets with mmWave RATs by modeling it as a client-centric multi-armed bandit problem. In our model, mmWave RATs are realistically modeled as irreversible Markov chains and we also take into account the overhead incurred upon frequent switching between RATs. We developed the mHS policy and proved that it achieves the optimal O(\ln(t)) regret with a sub-logarithmic switching regret over time. Finally, we evaluated the performance of our policy through simulations and showed that it not only achieves a significantly better total regret and throughput performance, but also incurs very
low switching overhead compared to existing baseline policies developed for MABP with Markovian rewards, while only requiring network-provided information on the wireless channels’ spectral (eigenvalue) gap instead of perfect client-RAT association and peak rate information as in Chapter 2.

In the next chapter, we examine how to perform client-centric RAT selection in the absence of the spectral gap. We explore how a client must learn about the network and adapt its behavior in the absence of any network-provided knowledge by relaxing (1) the assumption that the network is able to provide information to the client, and (2) allowing the client to trade off between initial incurred regret and accuracy of its learned statistics.

3.7 Appendix

3.7.1 Proof of Theorem 8

Proof of 8.i: First, we prove that for a time index at the end of a frame \( d > 0, \ t = N_d \), Theorem 8.i holds:

\[
E_{\alpha}[T^j(N_d)] \leq \left( \frac{4m^2_jL_j}{(\mu^* - \mu_j)^2} + o(1) \right) \ln N_d
\]

(3.12)

and then extend the result for all \( t \).
Let $\bar{X}^j(T^j(t))$ denote the sample mean of the reward collected from arm $j$ over the first $t$ time indexes, and let $c_{t,s} = m_j \sqrt{L_j \ln(t)/s}$ and integer $l > 0$.

$$T^j(t) = \sum_{i=1}^{n} 1(\alpha(i) = j) = \sum_{f=1}^{d} \sum_{k=1}^{k_f} 1(\alpha(t(f,k)) = j)$$

$$\leq l + \sum_{f=1}^{d} \sum_{k=1}^{k_f} 1(\phi^j(t(f,k),l), T^j(t(f,k) - 1) \geq l)$$

$$\leq l + \sum_{f=1}^{d} b_f \sum_{k=1}^{k_f} 1(\min_{0<s<t(f,k)} (\bar{X}^*(s) + c_{t(f,k)-1,s})$$

$$\leq \max_{l<s_j<t(f,k)} (\bar{X}^j(s_j) + c_{t(f,k)-1,s_j})$$

$$\leq l + \sum_{f=1}^{\infty} \sum_{k=1}^{k_f} \sum_{s=1}^{t(f,k)-1} \sum_{s_j=l}^{t(f,k)-1} 1(\bar{X}^*(s) + c_{t(f,k),s} \leq \bar{X}^j(s_j) + c_{t(f,k),s_j})$$

(3.13)

with indicator $1(.)$ and $\phi^j(t(f,k),l)$ is the event when

$$\bar{X}^*_{T^*(t(f,k)-1)} + c_{t(f,k)-1,T^*(t(f,k)-1)}$$

$$\leq \bar{X}^j_{T^j(t(f,k)-1)} + c_{t(f,k)-1,T^j(t(f,k)-1)}$$

Next, if $\bar{X}^*(s) + c_{t(f,k),s} \leq \bar{X}^j(s_j) + c_{t(f,k),s_j}$ holds, then one of the following must hold:

$$\bar{X}^*(s) \leq \mu^* - c_{t(f,k),s}$$  
(3.14)

$$\bar{X}^j(s_j) \geq \mu^j + c_{t(f,k),s_j}$$  
(3.15)

$$\mu^* < \mu^j + 2c_{t(f,k),s_j}$$  
(3.16)
By choosing \( s_j \geq \frac{4m_j^2 L_j \ln(t)}{(\mu^* - \mu^j)^2} \), we have that (3.16) is false. Choosing \( l = \frac{4m_j^2 L_j \ln(t)}{(\mu^* - \mu^j)^2} \) and taking expectation for (3.13):

\[
E_\alpha[T^j(t)] \leq \left( \frac{4m_j^2 L_j \ln(t)}{(\mu^* - \mu^j)^2} + 1 \right)
+ \sum_{f=1}^\infty \sum_{k=1}^{k_f} \sum_{s=1}^{t(f,k)-1} \sum_{s_j = l} P(\bar{X}^j(s_j) \geq \mu^j + c_{t(f,k),s_j})
+ \sum_{f=1}^\infty \sum_{k=1}^{k_f} \sum_{s=1}^{t(f,k)-1} \sum_{s_j = l} P(\bar{X}^* (s) \leq \mu^* - c_{t(f,k),s}) \tag{3.17}
\]

For the second term, the probability in the summation is bounded by

\[
P(\bar{X}^j(s_j) \geq \mu^j + c_{t(f,k),s_j})
\leq Z_q \exp\{-s_j \varepsilon(Q) c_{t,s_j}^2 \over 8b^2(1 + h(5c_{t,s_j}/b^2))\} \leq \frac{1}{\pi_{\text{min}}} \exp\{-\sqrt{L_j} \varepsilon(Q) \sqrt{\ln(t)} \over 8 + 30m_j\} \tag{3.18}
\leq \frac{1}{\pi_{\text{min}}} t^{-s_{\text{min}} \sqrt{\ln(t) \over 16 + m_j}} \tag{3.19}
\]

where (3.18) follows from Theorem 3.3 of [65], which provides a Chernoff bound independent of transition matrix reversibility, and in (3.19), we bound \( Z_q^j = \| q^j_y \|_2 \) as

\[
Z_q^j = \left| \left( \frac{q^j_y}{\pi_y} \right)_{y \in S^j} \right|_2 \leq \sum_{y \in S^j} \left| \frac{q^j_y}{\pi_y} \right|_2 \leq \frac{1}{\pi_{\text{min}}}
\]
and (3.19) holds if $L_j > 1$ as $\ln(t) > 1, \forall t > M$. Next, the third term can be bounded same as the second term. Now, (3.17) can be written as:

$$E_\alpha[T^j(t)] \leq \frac{4m^2_j \ln(t)}{\mu^*-\mu^j} + 1$$

$$+ 2 \sum_{f=1}^{\infty} b_f \sum_{k=1}^{k_f} \sum_{s=1}^{s_j} \sum_{s_j=1}^{1} \frac{1}{\pi_{min}} t^{-\frac{\epsilon_{min} \sqrt{L_j}}{16+60/m_j}}$$

$$\leq \frac{4m^2_j \ln(t)}{\mu^*-\mu^j} + 1 + \frac{2}{\pi_{min}} \sum_{f=1}^{\infty} b_f \sum_{k=1}^{k_f} t^{-\frac{\epsilon_{min} \sqrt{L_j}}{16+60/m_j}}$$

(3.20)

where (3.20) bounds the double summation over $s, s_j$ with $t^2$. From the mHS algorithm, we use $b_f = f$, and $k_f \leq (e^{f^2} - e^{(f-1)^2} + f)M/f$:

$$E_\alpha[T^j(t)] \leq \frac{4m^2_j \ln(t)}{\mu^*-\mu^j} + 1$$

$$+ \frac{2M}{\pi_{min}} \sum_{f=1}^{\infty} (e^{f^2} - e^{(f-1)^2} + f)t^{-\frac{\epsilon_{min} \sqrt{L_j}}{16+60/m_j}}$$

Next, rewriting $t = N_d$ in terms of frames $f$:

$$N_d = M + \sum_{f=1}^{d} (N_f - N_{f-1}) \leq M(e^{d^2} + f^2/2 - f/2)$$

(3.21)

Using (3.21), we can upper bound $E_\alpha[T^j(t)]$ as:

$$E_\alpha[T^j(t)] \leq \frac{4m^2_j \ln(t)}{\mu^*-\mu^j} + 1$$

$$+ \frac{2M}{\pi_{min}} \sum_{f=1}^{\infty} (M(e^{f^2} + f^2/2 - f/2))^{-\frac{\epsilon_{min} \sqrt{L_j}}{16+60/m_j}}$$

$$\leq \frac{4m^2_j \ln(t)}{\mu^*-\mu^j} + 1 + \frac{2M}{\pi_{min}} \sum_{f=1}^{\infty} e^{-f\frac{\epsilon_{min} \sqrt{L_j}}{16+60/m_j}}$$
which converges if $L_j$ is chosen such that $L_j \geq \left( \frac{3(16+60/m_j)}{\varepsilon_{\min}} \right)^2$. Thus,

$$E_\alpha[T^j(t)] \leq \frac{4m_j^2 L_j \ln(t)}{(\mu^* - \mu)^2} + 1 + DB; \text{ where}$$

$$D = \frac{2}{\pi_{\min}} M^{1/2}(\varepsilon_{\min} \sqrt{L_j} - 2), \quad B = \sum_{f=1}^{\infty} e^{-f(\varepsilon_{\min} \sqrt{L_j} - 3)}$$

Extending this result for general $t$. Let $d : N_{d-1} < t \leq N_d$. Then,

$$\frac{E_\alpha[T^j(t)]}{\ln(t)} \leq \frac{E_\alpha[T^j(N_d)]}{\ln N_{d-1}} \leq \frac{4m_j^2 L_j}{(\mu^* - \mu)^2} \ln N_{d-1} + \frac{DB}{\ln N_{d-1}}$$

which completes our proof of Theorem 8.ii. Proof of 8.ii: Next, we show that the switching regret is bounded by $o(\ln(t))$:

$$R^S_\alpha(t) = \sum_{j=1}^{t} c_a(j) E_\alpha[S^j_a(t)] + \sum_{j=1}^{t} c_d(j) E_\alpha[S^j_d(t)]$$

where

$$S^j_a(t) = \sum_{i=2}^{t} 1(\alpha(t) = j, \alpha(t - 1) \neq j);$$

$$S^j_d(t) = \sum_{i=2}^{t} 1(\alpha(t) \neq j, \alpha(t - 1) = j)$$

with constant arrival and departure costs $c_a(j), c_d(j)$ for each RAT $j$. Letting $N_{d-1} < t \leq N_d$,

$$S^j_a(t) \leq 1 + \sum_{f=1}^{d} \frac{T^j(N_f) - T^j(N_{f-1})}{f}$$

$$= \frac{T^j(N_d)}{d} + \sum_{f=1}^{d-1} T^j(N_f - 1) (\frac{1}{f} - \frac{1}{f+1})$$

$$\leq \frac{T^j(N_d)}{d} + \sum_{f=1}^{d-1} T^j(N_f) / f^2$$
Taking the expectation,

\[ E_\alpha[s^j_d(t)] \leq \frac{E_\alpha[T^j(N_d)]}{d} + \sum_{f=1}^{d-1} \frac{E_\alpha[T^j(N_f)]}{f^2} \]  

(3.22)

From (3.8) and (4.2, 4.3) of [8], which upper- and lower-bounds \( N_d \) in terms of \( M \) and \( d \),

\[ E_\alpha[T^j(N_d)] \leq \frac{4m^2L_j\ln N_d}{(\mu^* - \mu)^2} + o(1) \ln (N_d) \]

\[ = \left[ \frac{4m^2L_j}{(\mu^* - \mu)^2} + \zeta \right] \ln (Me^d + Md^2) \]

\[ \leq K(\zeta)d^2 \text{for some } K(\zeta) \]

For some time \( t > N_{f_0} \)

\[ E_\alpha[s^j_d(t)] \leq \frac{E_\alpha[T^j(N_d)]}{d} + \sum_{f=1}^{d-1} \frac{E_\alpha[T^j(N_f)]}{f^2} \]

\[ \leq K(\zeta)d + \sum_{f=f_0}^{d-1} K(\zeta) + \sum_{f=1}^{f_0-1} \frac{E_\alpha[T^j(N_f)]}{f^2} \]

\[ \leq 2K(\zeta)d + M(\zeta) \]

where \( M(\zeta) = \sum_{f=1}^{f_0-1} \frac{E_\alpha[T^j(N_f)]}{f^2} \). Comparing with \( \ln (t) \),

\[ \frac{E_\alpha[s^j_d(t)]}{\ln (t)} \leq \frac{E_\alpha[s^j_d(t)]}{\ln N_{d-1}} \leq \frac{2K(\zeta)d + M(\zeta)}{(d - 1)^2} = o(1) \]

which concludes the proof of \( E_\alpha[s^j_d(t)] = o(1) \ln (t) \).

For \( E_\alpha[s^j_d(t)] \), recognize that there is a symmetry in associating with and dissociating from RATs: in order to switch to a new RAT at time \( t \), a client must depart the previous RAT at the same time. This means that \( s^j_d(t) \leq s^j_a(t) \) must hold true, with \( s^j_d(t) = s^j_a(t) \) if \( \alpha(t) \neq j \) and \( s^j_d(t) = s^j_a(t) - 1 \) if \( \alpha(t) = j \). This means that \( E_\alpha[s^j_d(t)] \leq E_\alpha[s^j_a(t)] = o(1) \ln (t) \), completing the proof of Theorem 8.ii.
Proof of 8.iii: From (3.8),

\[ R_T^\alpha(t) = \sum_{j: \mu^j < \mu^*} \left[ \frac{4m^2 L_j}{(\mu^* - \mu^j)} + o(1) \right] \ln(t) \]  

(3.23)

Similarly,

\[ R_S^\alpha(t) \leq \sum_{j=1}^M c_j^d E_\alpha[s^j_\alpha(t)] \]

\[ = \sum_{j \neq j^*} (c_j^d) E_\alpha[s^j_\alpha(t)] + E_\alpha[s^*_\alpha(t)] c_{\Sigma}^* \]

\[ \leq \sum_{j \neq j^*} (c_j^d) E_\alpha[s^j_\alpha(t)] + \sum_{j \neq j^*} (c_j^d) (E_\alpha[s^j_\alpha(t)] + 1) \]

\[ \leq \sum_{j \neq j^*} (c_j^d + c_{\Sigma}^*) E_\alpha[s^j_\alpha(t)] + (M - 1)c_{\Sigma}^* = o(\ln(t)) \]

where \( c_j^d = c_j^a + c_j^d \). Combining the two for the total regret,

\[ R_\alpha(t) = R_T^\alpha(t) + R_S^\alpha(t) \]

\[ \leq \sum_{j: \mu^j < \mu^*} (\mu^* - \mu^j) E_\alpha[T^j_\alpha(t)] \]

\[ + \sum_{j \neq j^*} [E_\alpha[s^j_\alpha(t)](c_j^d + c_{\Sigma}^*)] + (M - 1)c_{\Sigma}^* \]

\[ \leq \sum_{j: \mu^j < \mu^*} 4m^2 L_j \ln(t) \left( \frac{1}{(\mu^* - \mu^j)} \right) + \sum_{j: \mu^j < \mu^*} (\mu^* - \mu^j)(1 + DB) \]

\[ + \sum_{j \neq j^*} [E_\alpha[s^j_\alpha(t)](c_j^d + c_{\Sigma}^*)] + (M - 1)c_{\Sigma}^* \]

Hence, Theorem 8.iii follows.

3.7.2 Throughput Model for Simulations

In our simulations, we model the downlink MAC-layer throughput of a client on BS \( j \), denoted \( x_j \), as a function of \( n_j \), and the instantaneous physical (PHY) layer rate \( R_j(t) \) of a client on BS \( j \), where \( x_j = R_j/n_j \). Examples of RATs with this throughput-sharing function are Time-Fair TDMA-type MACs like single-user MIMO mmWave
deployments, and Proportional-Fair schedulers (PFS) like those used in interference-limited OFDMA/LTE systems and 3G. For LTE and 802.11 networks, we assume time invariant channel conditions; for mmWave networks, we consider a Markovian model described in Section 3.2.1 on the sub-second timescales of RAT selection. The throughput achieved by each client is dependent upon the instantaneous physical (PHY) layer rate $R_i(t)$ of client $i$ on BS $j$, and depends on the modulation and coding scheme (MCS) and instantaneous channel conditions at time $t$, and we ignore the impact of mobility on instantaneous throughput.
Chapter 4

Absent Network Assistance: Inference and Reinforcement Learning

4.1 Introduction

With the exponential growth of mobile device use such as smartphones and laptops at the network edge, there has been a corresponding increase in traffic between these different devices: over 4000x over the past 10 years alone [29]. This traffic has been largely a result of an explosion of different applications that run on top of these mobile devices—tasks and applications such as real-time healthcare and monitoring, video streaming, and file transfer are only some of the actors that are predicted to cause mobile traffic to increase nearly 8x by 2020 [29].

A significant portion of this traffic will be situated on the network edge: to transfer a file from one mobile device to another, to periodically send an update from a wearable peripheral to a laptop or smartphone. With the rise of internet of things and wearable devices, both human-initiated communication and machine-type com-
munication will be sent on an increasingly congested wireless network edge. Without further improvements to how data is sent over the network edge, congestion and interference can result in abnormally abysmal system performance, and for certain applications with requirements for low-latency or tight temporal deadlines, may even result in failure.

One of the main reasons for this inefficiency on the network edge is the imprecise information that the mobile device has about its radio environment. Fluctuations in the channel conditions, interference from other radio interfaces in the area, coupled with uncertainty in the load on the network results in very rough estimates of resources available for the device on a given Radio Access Technology (RAT). Furthermore, different RATs leverage different protocols which have different ranges and granularity of measured statistics: these signals and metrics cannot be directly compared. Finally, many of these signals and metrics are instantaneous values such as SINR, and thus making decisions purely based on the statistics provided by the protocol can be an extremely noisy endeavor.

In this chapter, we present an online, fully-distributed, incrementally-deployable reinforcement learning algorithm for an edge-based device to learn the mapping of signals and statistics obtainable through passive measurement to empirical throughputs obtained by the mobile device. This algorithm is designed to actively refine its knowledge (online) based on local information accessible to the device (fully-distributed), and benefits each individual device even if no others are running such a scheme (incrementally-deployable). The algorithm operates on the client device at the network edge: by using reinforcement learning to map the coarsified passive signaling observed by the client to the empirical obtained throughputs, the edge client can build up its local knowledge of what throughput distributions are obtainable in different states of channel quality on different RATs.
We introduce the client-control mechanism of “binning” passive measurements, which trades off between favoring many small bins with fewer but more accurate mean throughput measurements in each state, versus a small number of very large bins with relatively inaccurate mean recorded throughput. By leveraging the multi-armed bandit approach to intelligently sampling accessible RATs as in the previous chapter, and coupling these empirical values to the observed signals from the protocol used by the RAT, the mobile client can better estimate the achievable performance it can obtain compared to making decisions solely based on the passive measurements themselves.

Our main contributions in this chapter are as follows:

- We model wireless network selection in edge (Fog) networks as a fully-distributed, client-controlled optimization in the absence of perfect channel information and throughput prediction (Section 4.2). We formulate the problem of client-controlled stochastic wireless channel inference and network selection in heterogeneous networks as a multi-armed bandit with unknown transition matrices;

- We introduce the coarsification of passive measurements at the client as a control mechanism that trades off between minimizing initial vs long-term throughput inefficiency at the cost of additional complexity for passive measurements (Section 4.3.1);

- We develop an online learning policy that is reward-distribution independent (WIFFN Algorithm, Section 4.3.3), which solves the wireless channel inference and RAT selection problem by leveraging the unique RAT-specific passive signaling observable at the client;

- We show that the regret (loss due to wireless channel uncertainty) of WIFFN is upper-bounded by $O(\ln(t))$ in time, and is asymptotically optimal (Theorem 9);
• We use established standards for heterogeneous networks in our numerical simulations to compare our policy with existing WiFi-only and WiFi-First online policies, and show our solution allows clients to obtain more efficient use of network resources (Section 4.4).

This paper is organized as follows. Section 4.2 describes the edge-based (Fog) network model for data transfer between multiple edge clients, as well as the passive signaling available to each type of RAT from its protocol specifications. Section 4.3 describes the reinforcement learning techniques and multi-armed bandit algorithm we develop for augmenting passive measurements. Simulation results are presented in Section 4.4 and we conclude in Section 4.6.

4.2 System Model

In this section, we discuss the network model for Edge (Fog) networks, the trade-offs present in client-driven inference and RAT selection, and the characteristics of common RATs in the heterogeneous edge networks.

4.2.1 Network Model

Edge-centric networking, or Fog networking, is characterized by a significant portion of wireless traffic between multiple mobile and smart devices on the network edge. Data transfer between these edge devices may consist of a wide variety of types, ranging from short updates to check for new information, to long-duration file transfers such as downloading or streaming video recorded from one device to another. Most of all, however, edge-centric networking is characterized by the clients’ desire to avoid passing their data transfer through the network core if at all possible: due to security or privacy concerns, it is desirable for clients to rely as much as they can on direct wireless links, or on a limited number of routers or relays on the network edge.
Examples of edge-centric networking includes any and all types of traffic between two mobile devices, as long as the clients themselves are interested in avoiding sending data and storing it in a centralized server in the network backbone. Examples of this include backing up data from a client’s smartphone (e.g. syncing calendars, etc) to his or her laptop by way of leveraging 802.11 Wi-Fi or Bluetooth links in order to avoid having to synchronize using a third-party server such as Dropbox, and sharing payment authentication from a client’s mobile device to a friend’s mobile device when they are geographically close to each other.

The primary characteristics of Edge-centric networking are the following. First, there exist a limited number of hops. Edge-centric networking is useful when clients do not wish to store their information “in the cloud” on a third-party server or network, and thus edge-centric networking generally involves a set of clients roughly located in the same geographical area that share common RATs. This geographic proximity manifests itself in a significant reduction in the number of “hops” that packets must travel between source and destination. Second, there is an emphasis on shallow networking, where clients at the edge prefer not to send their data deep “into the cloud” network, and may seek alternatives such as device-to-device (D2D) communication and other non-cellular networks if possible, as seen in Fig. 4.1.

Let \( M \) be the set of unique radio interfaces on an edge device that can be used for data transfer. These radio interfaces correspond to the different types of RATs that are accessible to the edge device, and depending on the technology, can connect to different Base Stations and other client devices sharing a similar interface (e.g. Bluetooth). Here, the term “Base Station” (BS) is used as a generic term to represent a 5G base station in 5G RATs, eNB in 4G/LTE, AP in 802.11, etc., and all communication links for a single client (whether to a BS or to another client) are assumed to be non-interfering due to frequency reuse or spatial separation between same-RAT BSs and frequency separation between different-RAT BSs. Each client
Figure 4.1: Example of Edge Network with Heterogeneous Radio Access Technologies

Edge network clients have multiple links leveraging heterogeneous RATs to communicate data from one edge device to another.

device can simultaneously communicate with a subset of BSs up to the number of RAT interfaces it is equipped with.\textsuperscript{1}

Table 4.1: Passive Measurement Ranges for Edge Network RATs

<table>
<thead>
<tr>
<th>Technology</th>
<th>Peak Rate</th>
<th>Measurement (dBm)</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>4G/LTE</td>
<td>325 Mbit/s</td>
<td>RSRP</td>
<td>-140</td>
<td>-44</td>
</tr>
<tr>
<td>802.11</td>
<td>54 Mbit/s (802.11g, 1x1)</td>
<td>RSSI</td>
<td>-90</td>
<td>-30</td>
</tr>
<tr>
<td></td>
<td>65 Mbit/s (802.11n, 1x1)</td>
<td>RSSI</td>
<td>-90</td>
<td>-30</td>
</tr>
<tr>
<td>Bluetooth 4.2</td>
<td>25 Mbit/s</td>
<td>Received Power</td>
<td>-70</td>
<td>-20</td>
</tr>
</tbody>
</table>

The goal of an edge client is to optimally learn about the different wireless networks accessible to the device, and maximize the data rate that it can achieve in support of ongoing applications’ data transfer in the absence of any network-side controller.

For any given client $i$, the following optimization problem is performed locally on the client device:

\textsuperscript{1}It is assumed that, due to frequency separation of the BSs, each RAT interface can observe signals from at most one BS or client at any time. For interfaces that receive signals from multiple BSs and clients, the functionality is assumed to be modeled by treating each BS as a distinct RAT: e.g. an 802.11 interface with Channels 1,6,11 is considered to be 3 distinct interfaces.
maximize

\[ U_i(\omega_{i,1}, \omega_{i,2}, \ldots, \omega_{i,M}) \]  \hspace{1cm} (4.1)

subject to

\[ \sum_k X_{i,k} \leq K_i \quad \forall k \in M \]  \hspace{1cm} (4.2)

variables

\[ X_{i,k} \quad \forall k \in M \]  \hspace{1cm} (4.3)

where \( U_i(.) \) is some utility function, that depends on the throughputs obtained by the client on potential RATs, and is readily observable at the client, \( K_i \), the client-specified choice of the maximum number of multi-homed (or single-homed, in the case of \( K_i = 1 \)) connections that it wishes to support, and the client has full visibility into what its past association behavior (\( X_{i,k} \forall k \in M \)) has been.

However, in practical situations, the client has visibility on its current statistics and its past observations but it is unlikely that any client will have perfect knowledge of future potential throughputs on all its accessible heterogeneous RATs. Specifically, although the client may be able to dictate the function \( U_i(.) \) to better serve its ongoing applications, the inputs to this utility function \( \{\omega_{i,1},\omega_{i,2},\ldots,\omega_{i,M}\} \) are imperfectly known, unlike the perfect-knowledge case in [14].

The approach in this chapter for discovering these potential throughputs is to leverage reinforcement learning to map locally discoverable measurements such as RSRP in 4G/LTE and RSSI in 802.11 to empirically obtained throughputs in order to better predict obtainable throughput on each RAT.
4.2.2 Radio Access Technology Model

There are a multitude of public and proprietary RATs that edge clients are increasingly equipped with. In this work, we consider the following heterogeneous candidate RATs for data transmission commonly found on such edge devices: 4G/LTE, 802.11 Wi-Fi, Bluetooth, and 5G Millimeter Wave (mmWave). These technologies span a wide variety of different spectrum bands, spectral efficiencies, SINR quantification, and ultimately different achievable peak rates for each RAT. Each of these technologies define specific measurements that are accessible by the client through local measurements, e.g., various measures of received signal power and allocated bandwidth. The passive measurements are summarized in Table 4.1

LTE

LTE, or Long-Term Evolution is a high-speed wireless communication standard that is present on most mobile devices, such as smartphones and tablets, as well as other devices via wireless USB dongles, capable of peak download rates of 300 Mbits/s and upload rates of 75 MBits/s under certain configurations (4x4 antenna array using 20 MHz of spectrum) in LTE-Advanced, Release 8. Passive measurements available include Reference Signal Received Power (RSRP), ranging from $-140$ dBm to $-44$ dBm in 1-dBm granularity, Reference Signal Received Quality (RSRQ) between $-19.5$ to $-3$ dB in 0.5dB granularity, and Resource Blocks ($N$) between 6 to 100 in a single 20 MHz band.

802.11

The IEEE 802.11 standard, commonly known as “Wi-Fi”, is a set of wireless communication standards for implementing local area network communication in the 900 MHz, 2.4, 3.6, 5 and 60 GHz frequency bands. The most common “flavors” of Wi-Fi nowadays include 802.11g, which uses 2.4GHz but allows peak rates of 54 Mbits/s,
and 802.11ac, which allows for wider channels in the 5GHz band for potentially up to 1300 Mbits/s. Acceptable signal strengths for communication require at a minimum -70 dBm for reliable packet delivery, -67 dBm for data-intensive applications such as video and VoIP. It is also interesting to note that although the standard has defined a range of 0-255 for RSSI, it allows individual vendors to define the “maximum” RSSI value in this range, resulting in incompatible comparisons between vendors like Cisco (0-100) and Atheros (0-60).

**Bluetooth**

Bluetooth is a short-range wireless communication standard present on most mobile devices and computers (Class 2 Bluetooth devices) that relies on unlicensed spectrum in the ISM band to transfer data from one device to another. The standard is designed for low-power data transmission, with a typical effective range of about 10 meters. With the current Bluetooth 4.2 specification [23], the range of measured power at the receiver ranges from -70 dBm reference floor, up to -20 dBm or greater. RSSI measurements have also been standardized, but not all Bluetooth devices are RSSI-measurement enabled and thus it has been omitted from this work.

### 4.2.3 Tradeoffs in Client-driven Inference and RAT Selection

One of the most readily-visible problems in channel inference and RAT selection is simply that these heterogeneous technologies make use of the underlying physical and MAC layers differently. Thus, while client device on 4G/LTE may measure RSRP and RSRQ to infer RSSI, these RSSI values translate into different throughputs compared to the RSSI values available in the 802.11 Wi-Fi standard. Although standards for these technologies define peak achievable rates for both downlink and uplink, in practice these peak rate estimations are rarely, if ever, achievable, given the listed
measurements due to variability in the rest of the link between source and destination, as well as the load on the router and interference from other source-destination pairs.

In this chapter, we seek to capture this mismatch between observed, standardized passive measurements, and the actual throughputs that a client can expect, by leveraging reinforcement learning to estimate sample mean throughputs for different RATs given different states of the network (inferred through local passive measurements). By leveraging the client’s history of passive measurements on each RAT and their empirical throughput obtained, it is possible for the client to directly compare two disparate RATs based on actual historical performance.

There are two tradeoffs that exist in this approach: (1) the tradeoff between exploration and exploitation of wireless channels, and (2) the tradeoff between using coarse and fine-grained mapping of passive measurements to empirical sample mean throughputs.

First, selecting the optimal RAT association is a classic example of the tradeoff between exploration, where the client desires to sample as many RATs as possible to better understand its options, and exploitation, where the client desires to maximize its time spent obtaining throughput from what it believes to have the greatest mean reward. The metric used to quantify this tradeoff is external regret, which quantifies the gap between the performance of an online algorithm and some other offline algorithm with perfect knowledge: a measure of the “loss” incurred from uncertainty about the overall radio wireless environment.

Second, in order to create a mapping of (RAT-type, passive measurements) to sample mean throughputs, there exists a tradeoff between coarsification and granularity of the number of entries in this mapping, which directly trades off speed of convergence and optimality of an algorithm that uses it. If the client attempts to define a great number of these (RAT-type, passive measurement)-pair bins to store its historical sample mean throughputs, it could potentially take a very long time to
Table 4.2: Chapter [4] Notation

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Number of accessible RATs</td>
</tr>
<tr>
<td>$\omega_j(t)$</td>
<td>Throughput obtained on RAT $j$ at time $t$</td>
</tr>
<tr>
<td>$\phi_j(t)$</td>
<td>Passive parameters for RAT $j$ at time $t$</td>
</tr>
<tr>
<td>$\theta_j(t)$</td>
<td>Binned passive parameters for RAT $j$ at time $t$</td>
</tr>
<tr>
<td>$n(j, \theta_j)$</td>
<td>Number of samples of RAT $j$ with parameters $\theta_j$</td>
</tr>
<tr>
<td>$\bar{\omega}_j(\theta_j)$</td>
<td>Empirical mean throughput of RAT $j$ with parameters $\theta_j$</td>
</tr>
<tr>
<td>$\bar{\omega}_j(\theta_j, T)$</td>
<td>$\bar{\omega}_j(\theta_j)$ after $T$ samples</td>
</tr>
<tr>
<td>$\beta_j$</td>
<td>Exploration Parameter for RAT $j$</td>
</tr>
<tr>
<td>$b_j$</td>
<td>Number of Bins in RAT $j$</td>
</tr>
<tr>
<td>$\mu^j(\theta_j)$</td>
<td>Mean throughput on RAT $j$ with parameters $\theta_j$</td>
</tr>
</tbody>
</table>

fill each bin depending on the vagaries of the wireless channel itself. However, if the client attempts to minimize the number of these (RAT type, passive measurement)-pair bins, the sample mean throughputs obtained may not be good predictors of wireless throughput on the RATs given such a wide range of passive measurements for each sample mean value.

4.3 Wireless Inference for Fog Networks (WIFFN) Algorithm

In this section, we discuss how to handle heterogeneous RATs and their passive measurements, how to apply reinforcement learning to RAT selection in edge-based wireless networks with channel inference, and provide the WIFFN algorithm to maximize time-averaged throughput.

4.3.1 Passive Measurements and Binning

We handle the problem of indirect comparison between the different passive measurements and their ranges available on each RAT by mapping these locally observable parameters to a directly comparable dependent variable: throughput. By recording the empirical mean throughput and associating observed values to the passive pa-
rameters seen by the client, a localized translation can be learned. This is shown in Fig. 4.2.a, where each value for the parameters of the RATs are matched to the empirical mean throughputs observed in those states through the straightforward iterative update step of:

\[ \bar{\omega}_j(\phi_j) \leftarrow \frac{n(j, \phi_j)\bar{\omega}_j(\phi_j) + \omega_j(t)}{n(j, \phi_j) + 1} \]
\[ n(j, \phi_j) \leftarrow n(j, \phi_j) + 1 \]  

(4.4)

However, many RATs define their passive parameters over a very fine scale (e.g. 1dBm measurement intervals for passive RSRP values in 4G/LTE for a total of 98), certain problems arise: (1) there is no guarantee that the RAT will ever be observed under a particular value for passive measurement, and (2) if the RAT were observed in a state where no historical data had been recorded, the question of how should the client deal with this situation arises. The resulting decision may potentially delay convergence: either because the client ignores the unknown RAT, or selects the new RAT for sampling–both of which could be a suboptimal decision.

To solve this problem of (potentially many) combinations of RATs and their passive measurements throwing off convergence, we allow the coarsification of calculating the empirical mean throughputs: by recognizing that two adjacent values for passive

Figure 4.2: Coarsifying passive measurements through “binning”
An example of “binning”. (a) The original granularity of the raw passive measurement \( \phi_j(t) \), and (b) the binned measurements \( \theta_j(t) \), which reduces the number of scalars needed to map measurement to throughputs \( b_1 \) from 8 to 4; and \( b_2 \) from 6 to 2.

To solve this problem of (potentially many) combinations of RATs and their passive measurements throwing off convergence, we allow the coarsification of calculating the empirical mean throughputs: by recognizing that two adjacent values for passive
measurements may still result in a similar throughput due to shared modulation and
coding schemes, the client can decide how many amalgamated intervals, or “bins”, \( b_j \)
for passive measurement ranges it can allow for RAT \( j \).

Algorithm 4 takes as input the current raw passive measurements \( \phi_j(t) \) on all
RATs \( j \in M \), the maximum and minimum measurement values \( \phi_{j,max}, \phi_{j,min} \), and
the desired number of bins for each RAT \( b_j \), and outputs \( \theta_j(t) \), the bin index for
the passive measurement, with \( \theta_j = 1 \) corresponding to the lowest range of passive
measurements, and \( \theta_j = b_j \) corresponding to the highest values.

\[ \text{Algorithm 4: mBin (Measurement Binning) Algorithm} \]

**Input**: Current passive measurements \( \phi_j(t) \), Number of Bins for RAT \( j \) \( b_j \), Min/Max
passive measurement values \( \phi_{j,min}, \phi_{j,max} \)

**Output**: Binned passive measurements \( \theta_j(t) \)

\begin{align*}
\text{for } j = 1 \ldots M \text{ do} \\
d_j &= \frac{\phi_{j,max} - \phi_{j,min}}{b_j} ; \\
\theta_j(t) &\leftarrow \left\lfloor \frac{\phi_j(t)}{d_j} \right\rfloor ; \\
\text{end}
\end{align*}

This is seen in Fig 4.2.b, where the client condenses the full range of 8 values of
\( \phi_1 \) to \( b_{j=1} = 4 \) bins, and the 6 of \( \phi_2 \) to \( b_{j=2} = 2 \) bins. By coarsifying the granularity of
the passive measurements recorded by the client, both time and internal memory is
saved when performing channel inference to predict throughput, as well as reducing
the likelihood that any bin can throw off convergence in the future by remaining
non-sampled. The update equation for mean throughput under the mBin algorithm
remains as in (4.4), where \( \theta_j \) refers to the new bins defined by the client and replaces
the raw measurements \( \phi_j \). Although these bins do not necessarily have to be equal
in the range of parameter values, in this work we assume that each bin has a range
equal to the range of the parameter values divided by the number of bins: bin size
\( d_j = \frac{\phi_{j,max} - \phi_{j,min}}{b_j} \).
4.3.2 Reinforcement Learning and Regret

However, the coarsification of passive measurements in Algorithm 4 in order to reduce the number of times an unknown network configuration can throw off convergence is only one part of the client-centric problem to determine network association. To guarantee optimal performance for the client, a method must be used to identify which network(s), given their observed passive parameters, are optimal, for the client to connect to the best network. We leverage techniques from reinforcement learning to allow the client to collect information about the network in an online fashion: by recording the empirical mean throughput observed by the client on each RAT in different network conditions, defined by the bins passive measurement bins. Specifically, we model the client’s choice of RAT selection using throughput inference based on historical network performance as a multi-armed bandit problem, where the client needs to infer the mapping between the observed passive measurement of the network to its performance—in this chapter, throughput.

We use the metric of external regret to quantify the difference between the expected total throughput obtained by an optimal policy with ideal knowledge and the expected total throughput obtained by an online policy—in this case, our proposed fully-distributed, incrementally-deployable reinforcement learning algorithm based on the multi-armed bandit problem (MABP). The optimal policy we compare against is the (infeasible) ”ideal best single RAT” policy generated by an offline-optimal oracle, assumed to have perfect a priori knowledge of which RAT will have optimal time-averaged throughput $\mu^*$ over the course of some arbitrarily-long time period. The key challenge in this work is to upper-bound the regret as tightly as possible, to ensure that the expected total throughput from our online algorithm is as close as possible to the ideal. The regret is defined as the difference between the optimal aggregate throughput and the expected total aggregate throughput $\sum_t \omega^\alpha(t)$ of some online algorithm $\alpha$: 
\[ R(t) = t \cdot \mu^* - E[\sum_t \omega^\alpha(t)] \] (4.5)

where the expectation is taken over the full run time of the algorithm up to time \( t \). Clearly, the goal for such an algorithm that seeks to maximize the obtained aggregate throughput under uncertainty is to minimize the regret over time—to decrease the performance gap of the online algorithm with respect to the offline oracle.

The design goals for a fully-distributed, incrementally-deployable network inference and RAT selection algorithm for client-centric Fog networking requires the following:

- A client-centric design to minimize signaling and delay present in network-centric approaches such that the client has full control over its own network inference and RAT selection capabilities.
- Algorithmic reliance upon locally-available and client-discoverable passive measurements.
- Maximize aggregate throughput obtained by the client.

For the rest of the chapter, we make the following assumptions:

1. For each RAT \( j \), the number of bins \( b_j \geq 1 \).
2. Mean RAT throughputs \( \mu^j : \mu^j < \mu^* \) for all \( j \neq j^* \) exists, and there exists exactly one optimal RAT \( j^* \).
3. \( |\omega_j(t) - E[\omega_j(t)]| \leq \omega_{max,j} \forall j \).
4. The client \( i \) is single-homed: \( K_i = 1 \).

The assumptions include (1) the positivity of the number of bins, and (2) the existence and uniqueness of a RAT with maximal mean throughput, as it is unlikely
that there exists multiple RATs with the exact same mean throughput. (3) requires
the deviation of throughput from the mean at any given time be bounded by the
maximum potential throughput achievable on the RAT. Finally, for simplicity, (4)
assumes that each client is single-homed $K_i = 1$, though we mention how to deal
with $K_i > 1$.

4.3.3 WIFFN Algorithm

We formulate the problem as a multi-armed bandit, where the client selects an edge-

network to associate with at any given time. Each edge-network can be modeled as

a multi-state markov chain, where each state is described by a particular value of

the passive parameter denoting the observed channel conditions between the client

and the RAT. In the absence of any network assistance, the probability transition

matrices of the RATs’ states are unknown. However, based on the commonly-known

wireless standards for the RATs involved, it is possible to know the maximum range

of these passive measurements, as well as the peak rates of each technology under
different modulation and encoding schemes.

To discover these passive measurements, during each time slot, the client makes

a local measurement of the network at the outset of the time interval: a passive

sensing of the channel between the client and the RAT. Based on the observed passive

measurements, the client selects a network to associate with for the remainder of the
time interval. The time interval is assumed to be short enough to ensure that the

passive parameters of the channel do not change significantly over its duration. Thus,
at each time slot, the client has an estimate of the state of the network (Fig 4.3).

However, knowledge of the state doesn’t guarantee the client perfect knowledge

of throughput on each RAT. In each state, there exists a (potentially unique)
distribution of throughput achievable for the RAT given such channel conditions.
This (state)-(throughput distribution) mapping can be considered an abstraction
The client observes passive measurements on all accessible RATs during the first fraction, and then selects and associates with a RAT for the remainder. For the actual proprietary and operator-specific mapping of passive measurements (e.g. RSRP/RSSI) to Channel Quality Indicator (CQI) value, which is then mapped deterministically to a specific MCS, which then achieves some unknown distribution of throughput for that RAT under such channel conditions.

The reinforcement learning algorithm we propose in this work balances both exploration and exploitation: it strives to minimize the incurred regret (Eq 4.5) by building up a local memory of the average behavior (mean empirical throughput) of each accessible RAT under different observed passive parameters, in order to dynamically adapt to different channel states as the channels change over time (exploitation). Furthermore, the algorithm incorporates an exploration term that grows with time, causing the client to select RATs with potentially non-optimal throughputs in order to verify that the historical sample means are unchanged and correct (exploration).

At each time slot, Algorithm 5 determines the passive parameters $\phi_j$ on each RAT $j$, and runs mBin to determine the associated bin index $\theta_j$. For the first $M$ timeslots, the algorithm is initialized (Lines 4 – 7) as the client samples each of the $M$ accessible RATs, recording the empirical throughput ($\omega_j(\theta_j)$) and number of visits ($n(j, \theta_j)$) for the RAT $j$ under passive parameter bin $\theta_j$. Once the initialization phase is done, for all subsequent timesteps (Lines 9 – 15) the client generates an index value $\text{index}_j(t)$ for each RAT based on the sample mean throughput of RAT $j$ in bin $\theta_j$, and an exploration term that depends on the number of bins $b_j$, $n(j, \theta_j)$, and the elapsed
time \( t \). \( \beta_j \) is a RAT-specific parameter that depends on the maximum potential throughput on the RAT \( j \), and is determined by the client. In the case where the combination of \((j, \theta_j)\) have not yet been explored, the index is calculated using the mean throughput for RAT \( j \) over all visited states, and an exploration parameter with the denominator set to unity to encourage exploration.

Algorithm 5: WIFFN Algorithm

**Input**: RAT-specific passive parameters \( \phi_j(t) \), Obtained throughput \( \omega_j(t) \), Number of accessible RATs \( M \), Number of bins for all RATs \( \{ b_j \} \)

**Output**: Optimal RAT association \( j = \arg\max_j \text{index}_j(t) \)

1. Initialization: time \( t = 1 \)
2. while \( t > 0 \) do
   3. Run \( m\text{Bin} \), obtain \( \theta_j(t) \)\forall j at current time \( t \);
   4. if \( t \leq M \) then
      5. Play RAT \( j = t \);
      6. \( n(j, \theta_j) \leftarrow 1 \);
      7. \( \bar{\omega}_j(\theta_j) \leftarrow \omega_j(t) \)
   else
       8. if \( \bar{\omega}_j(\theta_j(t)) > 0 \) then
           9. \( \text{index}_j(t) = \bar{\omega}_j(\theta_j(t)) + \beta_j b_j \sqrt{\frac{\ln(t)}{n(j, \theta_j)}} \)
       else
           10. \( \text{index}_j(t) = \bar{\omega}_j(\theta_j; n(j, \theta_j) > 0) \theta_j(t) + \beta_j b_j \sqrt{\ln(t)} \)
       end
       11. Play RAT \( j = \arg\max_j \text{index}_j(t) \)
       12. \( \bar{\omega}_j(\theta_j) \leftarrow \frac{n(j, \theta_j) \bar{\omega}_j(\theta_j) + \omega_j(t)}{n(j, \theta_j) + 1} \)
       13. \( n(j, \theta_j) \leftarrow n(j, \theta_j) + 1 \)
   end
3. end

Once the indexes have been calculated for each RAT \( j \), the client then associates with the RAT with the maximal index value for \( K_i = 1 \) (Line 14 – 16). In the case where \( K_i > 1 \), then the client simply needs to associate with the RATs with the \( K_i \) largest index values. At the end of the timeslot, the client updates the empirical mean throughput of the \( \theta_j \) bin of RAT \( j \), as well as the recorded number of visits \( n(j, \theta_j) \).

In our main result, we show that using Algorithm 5 with a lower-bound on the exploration parameter \( \beta_j \) can guarantee an upper bound on the metric of regret (Eq. 4.5):
Theorem 9. Under the WIFFN RAT inference and selection algorithm, for all \( j \in M \) such that \( \mu(\theta_j) \), the expected number of samples of a suboptimal RAT \( j \) is upper-bounded by

\[
E[T^j(n)] \leq [C(\beta_j, b_j) + o(1)] \ln(n),
\]

(4.6)

and the external regret over all \( M \) RATs is upper-bounded by

\[
R(n) \leq \left[ \sum_{j : \mu_j < \mu^*} C(\beta_j, b_j) + O(1) \right] \ln(n)
\]

(4.7)

where \( C(\beta_j, b_j) = \frac{4\beta_j^2 b_j^2}{(\mu^* - \mu_j)} \).

Proof (Sketch): First, we bound the number of times that a suboptimal RAT is visited under the WIFFN algorithm. By choosing the parameters \( \beta_j \) with respect to the maximum throughputs on each RAT \( (\omega_{\max,j}) \) and the number of bins \( (b_j) \) chosen for each RAT, the bound is finite and exists as a generalized harmonic number. Taking the expectation over these suboptimal visits, and using the definition of regret in (4.5), the regret can be order-bounded by \( \ln(n) \). See Appendix 4.7 for the full proof.

Theorem 9 shows that for a Fog network where the only information obtained by clients are the passive measurements observed on potential RATs and the empirical throughput obtained during data transmission, the WIFFN algorithm is able to achieve the optimal \( O(\ln(n)) \) regret as in [62]:

Remark 2. Using the WIFFN algorithm, clients may achieve the same optimal time-averaged throughput as an offline-optimal "best RAT" policy by the following:

\[
\lim_{t \to 0} \frac{R(t)}{t} = 0 \quad \Rightarrow \quad \lim_{t \to 0} \frac{E[\sum_1^n \omega^\alpha(t)]}{t} = \mu^*
\]

(4.8)
Remark 3. Increasing the number of bins $b_j$ trades off between the accuracy of throughput estimation $\bar{\omega}_j(t)$ and peak incurred regret.

By increasing $b_j$, the algorithm can get a much finer mapping of $\theta_j$ to potential throughput. However, as there are more combinations of $(j, \theta_j)$ for the client to explore, Algorithm 5 incorporates a proportionally-larger number of RAT associations that favor exploration, most of which may not produce the optimal throughput during that timestep. The increase in inefficient exploration required results in comparatively greater regret incurred by the client at any given timestep. (Sec 4.4)

Remark 4. The choice of number of bins $\{b_j\}$ for RATs $j \in M$ does not impact the asymptotic temporal behavior of regret for the WIFFN RAT inference and selection algorithm if $\beta_j$ is chosen to be greater than $\frac{\omega_{\text{max},j} b_j}{\sqrt{3} 2}$. 

Proof: The coefficient for logarithmic regret in the WIFFN algorithm is $C(\beta_j, b_j) = \frac{4\beta_j^2 b_j^2 \ln(n)}{(\mu^* - \mu^j)}$. By choosing $\beta_j > \frac{\omega_{\text{max},j}}{b_j} \sqrt{\frac{3}{2}}$ to guarantee convergence in Theorem 9, the $b_j$ values cancel out in the numerator and the denominator of $C(\beta_j, b_j)$, leaving it as a function of $\omega_{\text{max},j}$, the maximum throughput obtainable on each RAT $j$. ■

4.4 Performance Evaluation

We compare the performance of the WIFFN algorithm with two other commonly used algorithms: the naive “Wi-Fi Offloading” algorithm and the “Wi-Fi Assist” algorithm, based on Apple’s Wi-Fi assist option. Simulation results are averaged over 200 clients over the time period shown. Clients are restricted to only knowing the bin values ($\omega_j$) for each RAT $j$, and the actual throughput of their selected RAT at each timeslot—they are unable to observe the throughputs of the non-selected RATs. The parameters used in the simulations are shown in Table 4.3.
Table 4.3: Parameters for PHY Layer Downlink Throughput Distribution based on MCS.

<table>
<thead>
<tr>
<th>ID</th>
<th>RAT</th>
<th>Peak Data Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LTE  <a href="20MHz">50</a></td>
<td>[14.9,29.9,44.8,67.2, 89.7,129.1,172.1,325.1]</td>
</tr>
<tr>
<td>2</td>
<td>802.11g [88] (20MHz)</td>
<td>[6,9,12,18,24,36,48,54]</td>
</tr>
<tr>
<td>3</td>
<td>802.11g (20MHz)</td>
<td>[6,9,12,18,24,36,48,54]</td>
</tr>
<tr>
<td>4</td>
<td>802.11n [88] (20MHz)</td>
<td>[6.5,13,19.5,26,39,52,58.5,65]</td>
</tr>
<tr>
<td>5</td>
<td>802.11n (20MHz)</td>
<td>[6.5,13,19.5,26,39,52,58.5,65]</td>
</tr>
</tbody>
</table>

4.4.1 Simulation Results

The “Wi-Fi Offloading” algorithm models a generic algorithm where the network operator wishes to offload clients to accessible 802.11 Wi-Fi networks whenever possible in order to reduce load on their managed cellular (3G/4G/LTE) networks. “Wi-Fi Offloading” simply looks at the measured passive parameters for accessible 802.11 RATs (measured in RSSI), and selects the RAT with the highest measured RSSI value for association at any given time. The “Wi-Fi Assist” algorithm seeks to model the behavior of Apple Wi-Fi Assist [13], where the client attempts to select the best 802.11 Wi-Fi network based on RSSI, but is now allowed to select the cellular network if the Wi-Fi channels are sufficiently degraded (as determined from passive measurements) to some value below a threshold. Both of these two algorithms have equal access to passive measurements obtained by the client, but perform no reinforcement learning on the relationship between these measurements and actual throughput as the WIFFN algorithm does.

Fig. 4.4a shows a semilog plot of external regret over time for the WIFFN algorithm, the Wi-Fi Offloading, and the Wi-Fi Assist algorithms. It is clear that the WIFFN algorithm, which achieves $O(\ln(t))$ regret (Theorem 9), significantly outperforms both the Wi-Fi Offloading and the Wi-Fi Assist algorithms in terms of minimal regret growth due to the learning of each network’s time-averaged performance. The other two algorithms are shown to have super-logarithmic regret growth over time, meaning relatively poor throughput performance compared to WIFFN.
WIFFN achieves (a) lower regret than the Wi-Fi Offloading and Wi-Fi Assist algorithms, (b) asymptotically optimal regret, and (c) obtains an increase of over 97% throughput. Increasing the number of bins $b_j$ in WIFFN results in (d) a greater initial regret with a lesser slope in regret growth, and has (e) a similar slower rate of throughput acquisition that increases in slope to be greater than that of $b_j = 1$.

Figure 4.4: Simulation Results for WIFFN Algorithm

The goodness of this logarithmic regret is shown in Fig. 4.4b, where the total regret is averaged over elapsed time. It is clear that the WIFFN algorithm asymptotically goes to zero (Remark 4.8), meaning it achieves asymptotically optimal regret. However, both Wi-Fi Offloading and Wi-Fi Assist algorithms have a time-averaged regret that remains relatively constant. As expected, since both algorithms make decisions based on the passive measurements without performing any learning on these measurements, there is no change in the per-timeslot regret incurred by these algorithms. Wi-Fi Offloading and Wi-Fi Assist seem to converge asymptotically to 61 and 54 Mbps per timeslot respectively, which reflects the gap between the mean throughput of the offline optimal policy (selecting the RAT with optimal $\mu^*$ for all $t$)
and both the average throughput over all 802.11 RATs (for Wi-Fi Offloading) and the average throughput over all 802.11 RATs (above an RSSI threshold of $-60$ dBm in our simulations) and 4G/LTE RATs when all Wi-Fi RSSI values fall below that threshold (for Wi-Fi Assist). In addition to the time-averaged regret, Fig. 4.4c shows the total throughput obtained by each of the three algorithms considered in this work. By applying reinforcement learning to readily-observable passive measurements, a clear increase in obtained per-timeslot throughput has been obtained of over 120% when compared to Wi-Fi Offloading, and 97% when compared to Wi-Fi Assist.

The effects of increasing the number of bins $b_j$ over all RATs is examined in Fig. 4.4d, where we have focused on the first 100s for clarity (omitting the first $M = 5$ seconds as they are essentially the initialization for the algorithm. As the number of bins is increased from 1 to 20, the initial regret incurred increases proportionally with number of bins: the more bins on a RAT, the greater the initial incurred regret: this increase in regret with number of bins is due to Line 12 in the WIFFN algorithm: if the RAT $j$ has not been sampled in a given bin index $\theta_j$ before, then the exploration term of the index tends to force the client to sample it. However, after the initial rise in regret due to exploration, the regret begins to “plateau” out to a much smaller slope—and this slope is visibly lesser than the slope of the $b_j = 1$ regret curve. The tradeoff that exists for increasing the number of bins is a decreased regret growth over time. This reduced regret growth results from the finer granularity of passive parameter bins, which the client device can make more accurate predictions of throughput once it has made similar measurements in the past. The total obtained throughput over the same time interval is also presented for comparison in Fig. 4.4c. Similar to Fig. 4.4d as the number of bins is increased, the initial rate of increase in total throughput is decreased as evidenced by the $b_j > 1$ curves, but their eventual slope is greater than that of the $b_j = 1$-bin curve by at least 10% (in the case of $b_j = 2$).
4.5 Related Work

**Heterogeneous Networking:** Optimal radio access network selection in HetNets has been studied in the past in both network-centric and client-centric forms \[14,32,34,44,59,103,109\]. The majority of approaches rely on either a network-centric \[32,34,44,109\] assignment of clients to RATs at any given time, or some form of dynamic network-based broadcast signaling to induce the movement of clients towards a desired distribution of clients. Others approaches \[14,59,103\], however, have focused on the fully-distributed case explored in this work: \[14\] and \[59\] are client-centric approaches that provides clients with a big-picture view of the global HetNet client-RAT configuration to induce convergence towards an equilibrium point. The work in \[103\] examines the case where only partial information about the networks are provided to the client regarding channel transition probabilities. However, in this work, no information is provided to the client outside of common information pertaining to the definition of wireless communication protocols, where the client must use reinforcement learning to discover network behavior.

**Wireless Inference:** Wireless inference is a difficult problem owing to the dual problems of determining the channel quality indicator (CQI) from channel estimation/measurements, and the uncertain translation of the selected modulation and encoding scheme (MCS) to actual throughput—which varies due to channel conditions and network load among other things. Generally, the determination of CQI is left up to the device or network operator: in LTE \[50\], as long as a measure is generated for the eNodeB, resources can be allocated to the client. In \[17\], a Euclidean distance metric is used based on SINR to generate a channel quality measure for mobile handoff, but focuses on cellular radio instead of the wider range of heterogeneous networks. The closest to our approach is \[106\], which affirms that the channel can never be perfectly sensed and works off of potentially-incorrect estimates. In our work, we similarly abstract the link between passive measurement and actual
throughput through coarsification of passive measurements and relying on historical
data to determine optimal client association instead of assuming exactly accurate
knowledge.

Reinforcement Learning: There is an extensive area of work concerning the
application of online learning algorithms to regret minimization in multi-armed ban-
dits [15,62,82,101]. These works explore how to balance the need for a client to both
explore available options and exploit the optimal option, but these works do not fully
meet our design goals. General maximum reward policies are studied in [15, 62, 82]
but do not take into account the possibility of having some partial information about
the network state in the form of passive measurements of the accessible channels. The
authors of [101] come closest to our approach in that they model reward maximiza-
tion given the Markovian nature of the networks, but some network information is
required to be transmitted to the client to select algorithm parameters, which is not
allowed by our design goals.

4.6 Conclusion

In this chapter, we have developed the WIFFN algorithm that addresses the prob-
lem of client-centric wireless inference for edge (Fog) networks by applying rein-
forcement learning techniques for the multi-armed bandit problem model without
network-provided information. In this model, the clients have no access to any
network-provided information, such as the temporal behavior of channel states, and
clients are forced to discover the mapping of locally-available passive measurements
to throughput by learning the edge-network environment based on historical client-
obtained data. Our WIFFN algorithm is shown to have a most \( O(\ln(t)) \) regret,
and has been shown to outperform existing passive-measurement-based algorithms in
standards-based simulations by over 97% in terms of throughput, as well as obtaining an asymptotic order-optimal regret.

4.7 Appendix

4.7.1 Proof of Theorem 9

Let $\bar{\omega}_j(T^j(n))$ denote the sample mean of the reward collected from arm $j$ over the first $n$ time indexes, let the exploration term be denoted as $c^j_{n,s} = \beta_j b_j \sqrt{\ln(n)/s}$ and integer $l > 0$.

$$
T^j(t) = \sum_{n=1}^{t} 1(\alpha(i) = j) \\
\leq l + \sum_{n=1}^{t} 1(\phi^j(n, l), T^j(n - 1) \geq l) \\
\leq l + \sum_{n=1}^{t} 1(\min_{0 < s < n} (\bar{\omega}_s(\theta^*_s(n), s) + c^*_s_{n-1,s})$

$$
\leq \max_{l < s_j < n} (\bar{\omega}_j(\theta^*_j(n), s_j) + c^j_{n-1,s_j})) \\
\leq l + \sum_{n=1}^{\infty} \sum_{s=1}^{n-1} \sum_{s_j=l}^{n-1} 1(\bar{\omega}_s(\theta^*_s(n), s) + c^*_s_{n,s} \leq$

$$
\bar{\omega}_j(\theta^*_j(n), s_j) + c^j_{n,s_j}) \tag{4.9}
$$

with indicator function $1(.)$ and $\phi^j(n, l)$ is the event when

$$
\bar{\omega}_s(\theta^*_s(n), T^*(n - 1)) + c^*_s_{n-1,T^*(n-1)} \\
\leq \bar{\omega}_j(\theta^*_j(n), T^j(n - 1)) + c^j_{n-1,T^j(n-1)}$$
Next, if \( \bar{\omega}_s(\theta_s(n), s) + c_{n,s}^* \leq \bar{\omega}_j(\theta_j(n), s_j) + c_{n,s_j}^j \) holds, then one of the following must hold:

\[
\begin{align*}
\bar{\omega}_s(\theta_s(n), s) & \leq \mu^* - c_{n,s}^* \quad (4.10) \\
\bar{\omega}_j(\theta_j(n), s_j) & \geq \mu^j + c_{n,s_j}^j \quad (4.11) \\
\mu^* & < \mu^j + 2c_{n,s_j}^j \quad (4.12)
\end{align*}
\]

Because \( \mu^* > \mu^j \) by definition, then \( s_j \geq \frac{4\beta^j b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} \), and we have that (4.12) is false. Choosing \( l = \left\lceil \frac{4\beta^j b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} \right\rceil \) and taking expectation for (4.9):

\[
E_\alpha[T^j(n)] \leq \left( \frac{4\beta^j b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} + 1 \right)
+ \sum_{n=1}^{\infty} \sum_{s=1}^{n-1} \sum_{s_j=l} P(\bar{\omega}_j(\theta_j(n), s_j) \geq \mu^j + c_{n,s_j}^j)
+ \sum_{n=1}^{\infty} \sum_{s=1}^{n-1} \sum_{s_j=l} P(\bar{\omega}_s(\theta_s(n), s) \leq \mu^* - c_{n,s}^*) \quad (4.13)
\]

The second term can be bounded by observing that throughput \( \omega \) is finite and strictly nonnegative (that is, \( \omega \in [0, \omega_{\text{max},j}] \)), and applying the one-sided Hoeffding bound on deviation above the mean:

\[
P(\bar{\omega}_j(\theta_j(n), s_j) \geq \mu^j + c_{n,s_j}^j) \leq \exp\left(-2\beta^j b_j^2 c_{n,s_j}^j / \sum_{n=1}^{s_j} (\omega_{\text{max},j}^2)\right)
\leq \exp\left(-2\beta^2 b_j^2 \ln n / \omega_{\text{max},j}^2\right)
\leq n^{-2\beta^2 b_j^2 / \omega_{\text{max},j}^2} \quad (4.14)
\]
Using this result, the triple summation can be written as:

$$\sum_{n=1}^{\infty} \sum_{s=1}^{n-1} \sum_{s_j=l} n \cdot P(\tilde{\omega}_j(\theta_j(n), s_j) \geq \mu^j + c_{n,s_j}^j)$$

$$\leq \sum_{n=1}^{\infty} n^2 n^{-2\beta_j^2 b_j^2/\omega_{\max,j}^2} \quad (4.15)$$

$$\leq \sum_{n=1}^{\infty} n^{-(2\beta_j^2 b_j^2/\omega_{\max,j}^2 - 2)}$$

where the double sum has been bounded by $n^2$ in (4.15). Bounding the third term in the same way, (4.13) can be written as:

$$E_\alpha[T_j(n)] \leq \left( \frac{4\beta_j^2 b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} + 1 \right)$$

$$+ 2 \sum_{n=1}^{\infty} n^{-(2\beta_j^2 b_j^2/\omega_{\max,j}^2 - 2)} \quad (4.16)$$

where (4.16) bounds the double summation over $s, s_j$ with $n^2$. Simplifying, we obtain:

$$E_\alpha[T_j(n)] \leq \left( \frac{4\beta_j^2 b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} + 1 \right)$$

$$+ 2 \sum_{n=1}^{\infty} n^{-(2\beta_j^2 b_j^2/\omega_{\max,j}^2 - 2)} \leq \left( \frac{4\beta_j^2 b_j^2 \ln(n)}{(\mu^* - \mu^j)^2} + 1 \right) + 2H_{\infty,Z}^j \quad (4.17)$$

where $Z = 2\beta_j^2 b_j^2/\omega_{\max,j}^2 - 2$, and (4.17) upper bounds the harmonic sum with exponent $Z$. The limit of this harmonic sum $H_{k,n} = \sum_{i=1}^{k} \frac{1}{n^i}$ exists as $k$ tends to $\infty$ if and only if the exponent $n > 1$. Thus, we choose $\beta_j > \frac{\omega_{\max,j}}{b_j} \sqrt{\frac{3}{2}}$ such that $Z > 1$ and the harmonic sum exists as a constant.
From the definition of regret (4.5),

\[ R(n) \leq \sum_{j: \mu^j < \mu^*} (\mu^* - \mu^j) E[T^j(n)] \]
\[ \leq \sum_{j: \mu^j < \mu^*} \frac{4\beta^2 b^2 \ln(n)}{\mu^* - \mu^j} \]
\[ + \sum_{j: \mu^j < \mu^*} (\mu^* - \mu^j)(1 + 2H^j_{\infty, Z}) \]

where only the first term depends on \( n \), and the second term is constant in \( n \). Hence, Theorem 9 follows. ■
Chapter 5

Related Works

5.1 HetNet Selection

There have been many different approaches to modeling the RAT selection problem in HetNets [11, 20, 21, 36, 56, 58, 66, 98, 108], where clients are allowed to switch radio interfaces for data transmission in order to improve some metric (Fig. 5.1a). These approaches have attempted to ask and answer the questions of where in the wireless network that RAT selection should take place, and how RAT selection should take place: specifically, what algorithm should be used to determine which RATs to connect to.

Traditionally, solutions to these questions have been primarily network-centric, setting the “where” to be inside the network core, and “how” to be some centralized agent running an optimization algorithm that makes a simultaneous global client-to-Base-Station/eNode-B association decision for all clients in the network. The advantages of such an architecture, where all decisions are made on the network side, are that the RAT selection algorithm can act upon a big-picture view of the network (can poll clients for their local channel conditions and have loading information on the RATs themselves) and converge to and achieve a globally-optimal result with
Figure 5.1: Difference between Network-controlled and Client-controlled HetNets

(a) Client with access to 3 Heterogeneous RATs; (b) Network-controlled HetNets: (1) The client reports information on its accessible RATs to the centralized agent, which (2) computes the globally optimal client-RAT association back to the client and (3) switches RATs; (c) Client-controlled HetNets: (1) The client gathers information on its accessible RATs and makes a local decision for its own client-RAT association decision and (2) switches RATs.

respect to some network metric relatively quickly. This approach is generally favored by organizations such as cellular service providers that seek to maintain control of network operation, reflecting the view that the network should optimize for some network-wide metric that the service provider wishes to focus on.

In network-controlled HetNets, these decisions are made in the following way (Fig. 5.1b): first, the centralized agent that performs the decision-making polls all clients seeking to connect to one or more of the RATs controlled by the agent, requesting information on the client’s local channel conditions. Based on this information, it calculates the optimal association of clients to RATs, and then transmits these associations back to each client. Once the client receives these instructions (by way of the RAT it is currently using), it then changes its RAT association as directed.

The exact optimization performed is unique to each network. Depending on the type of network and the network operator, different networks may be designed to optimize for different metrics depending on their business needs:

- Maximize aggregate throughput and fairness of resource allocation \[21,31,66\]
- Load balancing \[11,20,27,98\]
• Minimize outage probability \[36,56\]

• Maximize a measure of Quality of Service

A variety of techniques are used to achieve these goals such as cost functions, utility maximization, stochastic geometry and combinatorial optimization to determine the “best” \[48\] network for association \[67,102\].

However, there are downsides to centralized network-controlled HetNets. Chief among these are the issues of timeliness of switching, scalability of the system to multiple clients with multiple interfaces, and the issue of how to obtain global control of client-RAT association. Any centralized scheme would have to obtain client inputs (e.g., channel conditions, battery life, number and type of interfaces on the device, application type, etc.) in order to optimally allocate resources. Such polling would incur significant delay in each calculation due to latency between client and controller. Furthermore, all clients need to convey this information to the network—depending on the geographical area that the centralized agent controls, the sheer amount of required overhead traffic simply for passing clients’ parameters to the network could become a significant portion of data that the network transports, further limiting the efficiency of the HetNet. Finally, different RATs belonging to different business entities may not be willing to pool resources and allow outside control of proprietary networks. For example, to achieve true centralized network-controlled HetNets, companies such as Verizon and Boingo must be willing to hand off their existing customer traffic to each other when instructed to do so—raising questions about fairness of traffic allocation between their respective cellular and Wi-Fi networks, how to agree upon a method of traffic allocation, and even customer privacy.

In contrast to centralized network-controlled HetNets, the client-controlled HetNet association examined in this work focuses solely on the clients’ perspective. Based on local observations at the device, the client must choose which RAT to associate with in order to maximize some client-desired metric (e.g., throughput, some utility
function, etc.), evaluated locally (Fig. 5.1c). Although this form of distributed optimization run on individual clients may not obtain a client-RAT association with an achieved objective value as good that generated from a centralized network-controlled HetNet problem [14, 77], there are several advantages. First, the timeliness problem of the centralized case is avoided: as all measurements used are locally available on the client, client-centric RAT selection does not suffer from network latency. Second, client-centric RAT selection in HetNets is scalable, as each client only needs to calculate its own optimal association. Third, client privacy is preserved as the device no longer needs to transfer information about its traffic (e.g., type of data/traffic, utility function, battery power, etc.) to agents in the network. Finally, the problem of sharing network information and control between different business entities is avoided—as long as each client has permission to associate with a RAT, each network merely needs to allow connectivity when the client requests it.

### 5.2 Hybrid Control

Hybrid control extends the concept of client-centric control of RAT selection in HetNets to allow for network-assistance in the RAT selection process. This type of shared control preserves the client’s right to make the final decision on when and where to switch, but gives the network some input in the switching itself, such as providing information on network metrics such as load balancing. This added information is useful because the purely-local view of the network observed by fully-distributed client-centric RAT selection is not guaranteed to be accurate. Operating under this limited perspective can lead to suboptimal behavior when a client simply ignores the effect of its presence upon other users. Several works address this by allowing the network to inform clients with some global knowledge such as [32, 34, 77, 109] for example.
A broadcast technique to inform clients making local decisions of network conditions is used in several approaches [34][109]. In [34], the authors designed a BS association system for HetNets in which base stations broadcast both their current weighted load and price. These parameters are then used by clients to select and associate to the base station that best satisfies their utility. Under this service model, clients are assisted to make the best decision even under mobility, and can even assign individual applications to different interfaces.

This is taken one step further in [109] where the authors develop a low-complexity distributed algorithm that uses gradient descent dual-decomposition to split the multi-homed joint RAT association problem into two distinct subproblems of identifying the optimal BS at the client, and updating a BS-specific multiplier to be broadcast by the BS after each iteration. The Lagrangian multiplier acts as the price of the BS determined by its load, and performs a sort of load-balancing in the network itself.

The authors in [32] develop a distributed algorithm in which the clients do not directly compete to maximize throughput—instead, they associate to BSs in a way to maximize the total reward obtained from the BS in order to prevent selfish behavior. The reward assigned by each BS is dependent on the loss of throughput incurred for other users due to association based on *marginal cost pricing* [30].

Theoretical results for generalized distributed client-centric RAT selection in HetNets with prioritized service were analyzed in [77]. Both a purely client-centric and the hybrid association model were analyzed, and showed that purely client-centric association with generic weights can result in infinite oscillations; but under several specific classes of weighted priorities, convergence can be guaranteed for the system. Tight polynomial and linear bounds are found for the client-centric model, and that the proper selection of a potential function by the network can guarantee convergence of the system.
Assignment problem approaches for traffic offloading in HetNets for femto have also been proposed \cite{96,104} that find optimal association. Compared to the above works, matching schemes explicitly allow for indirect negotiation between clients and BSs in allocating network resources. Instead of providing a price parameter to clients, these approaches directly rank clients for each BS in terms of how well a potential BS may maximize the client or network metric.

In \cite{104}, the problem of achieving proportional-fair throughput for client-RAT association in HetNets is transformed into an equivalent matching problem, which can be solved in polynomial time. This exchange, where the BSs iteratively announce their price of association, and clients submit bids for resources, results in a global reduction for macrocellular traffic of up to 30% compared to several non-cooperative game-based strategies.

A preference list approach is used in \cite{96} to find a stable matching between clients and femtocells for uplink. In this setup, both clients and BSs rank each other based on preference functions that capture clients’ utilities which depend on packet success rate, delay, and small cells’ incentive to extend macrocell coverage. The game is solved using two phases involving admission games that allow transfers between BSs, followed by data transmission, with performance improvements of up to 23% compared to a best packet success rate algorithm.

Although the network may provide some additional information, the ultimate decision to switch still rests on the client itself—and in the absence of perfect, global knowledge, the client will need to discover how it should associate with the BSs to meet its objective. Online learning techniques such as reinforcement learning \cite{61} and multi-armed bandits \cite{103} are among some of the methods used to explore and exploit the spectrum resources that are accessible to a client.

Q-learning is used in \cite{61} to learn the client-specific bias values for received power in order to perform cell-range expansion for HetNets. By using these individualized
bias values over a common bias over all UEs, the system is a multi-agent system
that leverages distributed learning where information is never shared. The costs are
reported back to the clients from the BSs and are used to update the Q-values for
future ranking of RATs for association.

5.3 Fast Time Varying RATs (mmWave)

Several white papers by several industry groups predict \[47, 80\] 5G wireless systems
will provide, at the very least, higher data rates (between 10 Gbps in an indoor office
environment to 25 Mbps in a very dense crowd, with an average of 50 Mbps for
general use), end-to-end latency on the order of 1 ms, and handle up to 150,000/km\(^2\)
connections in a crowd-like environment.

The key elements of 5G identified by many telecom companies are \[37, 52, 89\]:

- **Peak Data Rate**: 10 Gbps per client, 4x that of 4G
- **End-to-End Latency**: 1 ms, 1/50 that of 4G
- **Scale of Connections**: 1 million/km\(^2\), 100x that of 4G

In order to meet these goals, 5G will not only require access to diverse (licensed,
shared licensed, and unlicensed) spectrum \[54, 89\], but leverage new spectrum bands
from 400 MHz to 100 GHz \[85, 86\] and to create capability to handle dense HetNets \[7\].

Millimeter-wave (mmWave) radio technologies is increasingly looked at as one
of the primary new RATs for 5G given the scarcity of spectrum at microwave fre-
quencies \[24, 25, 83\]. A combination of cost-effective hardware, high-gain and steer-
able antennas, larger carrier bandwidth allocations all translate to higher data rates
for mmWave communications for small-cell and indoor applications on the order of
200m \[87, 93\]. However, mmWave is also characterized by higher path-loss exponents
and an inability to penetrate obstructions such as the human body.
There exist many studies on mmWave wireless communication in the 60 GHz band \cite{33,43,46,91,107}. These studies characterize the free space propagation loss and the higher loss due to attenuation from non-Line-of-Sight (NLOS) channels. However, these bands may not be suitable for cellular due to the unlicensed nature of this band and the coexistence of 802.11ad, and the primary candidates are 28, 38, 71-76 and 81-86 GHz for indoor applications \cite{45}.

Outage studies for 38 GHz were done in Texas in 2012 \cite{78} and the first statistical channel models were measured for 28 and 73 GHz in dense urban neighborhoods (New York City) in 2013 \cite{9}, which analyzed a range of parameters: path loss, shadowing, mmWave line-of-sight (LOS)/NLOS/Outage probabilities, angular orientation of the BSs and clients, and number of clusters. Furthermore, the authors showed strong evidence for the existence of a third state, Outage, characterized by a complete loss of signal, and proposed modeling mmWave with a three-state channel model defined by the degrees of signal obstruction.

mmWave technology is particularly sensitive to obstruction, with drastic changes in the path-loss exponent for different mmWave frequency bands \cite{83}. These increase from values of 1.8 and 2 for 28 GHz and 73 GHz with a direct line-of-sight component to values upwards of 4.5 and 2.69 once the direct path is blocked. Furthermore, human-body blockage can have a significant effect on signal quality, with attenuation between 20-35dB \cite{41,69} with the loss of the line-of-sight.

However, with exceedingly high throughput upwards of 10+ Gbps for 73 GHz demonstrated only in the past year \cite{10}, a large path-loss exponent only helps the case that these high throughput mmWave BSs can be used as dense small cells in HetNets to handle extremely high throughput and high volume traffic.

In \cite{103}, the authors apply online learning using multi-armed bandits to RAT Selection for HetNets with fast-changing mmWave channels to maximize throughput while minimizing parameterized switching costs, with optimal total regret. Limited
feedback from the BS in the form of a parameter describing the channel state between client and BS is sent to each client, and the client then uses this knowledge of the channel to discover and exploit the “best” BS in terms of average throughput using a upper-confidence-bound-type algorithm in which the client is only allowed to switch BS associations at specific times. An alternative approach that also leverages the Markovian nature of channel state changes is considered in [72], which directly considers dynamic channel load and link quality but not switching costs. However, the use of mmWave in HetNets is a relatively new area of study, and there remain lots of work to be done to better understand how to interchangeably use mmWave alongside other existing networks.

5.4 Noisy Inference of Client Metrics

In order for client-centric BS association algorithms to function correctly, they require a method for differentiating one BS from another to determine which is optimal for association. These metrics depend on the concerns of the client, which may vary for different applications (e.g. maximizing throughput for bandwidth-hungry video or minimizing end-to-end latency for web applications). These metrics are highly sensitive to noisy estimates, and can become a bottleneck to optimal association of clients to BSs. In the ideal situation, this information could be accurately inferred at the client given some additional knowledge, or provided by the BS that performs some asynchronous calculation; however, in the general case this cannot be assumed.

Inaccurate inference can result in a variety of inefficiencies [14]. First, oscillations may result if a client can frequently switch between two different valuations of distinct BSs, causing it to repeatedly associate to one BS only to switch to the other. This frequent change in available throughput and load on the BSs in question can have an adverse impact on other clients in the network, resulting in a cascading effect where
one oscillating client can cause oscillations to propagate throughout the network as other clients see their throughputs and latencies change due to incorrect switching. Next, incorrect inference may not result in optimal achieved metrics for the client: inference errors may result in inefficient or incorrect BS associations, leading to sub-optimal metrics. Furthermore, additional costs can be incurred: switching between different BSs (and indeed different types of RATs as well) require that the client and network set up a new connection and tear down the old one in a handover process which consumes both time that could be otherwise used for downloading/uploading data, as well additional battery. For such power-limited devices such as smartphones and mobile devices, it is highly undesirable to allow noisy estimates and inferences to cause oscillations and incorrect switches.

One solution to this is to adapt the decision threshold for switching (e.g. $\eta$ in [14]) in a client-centric control algorithm to the ambient noise in the inference. By learning the distribution of inference noise on each BS, it is possible to negate its impact (e.g. by subtracting the worst-case empirically observed error) on the ranking of the BS. By increasing the required gain in predicted metrics required to initiate a switch between BSs, the client can switch more conservatively. However, the system becomes less likely to switch with an increased decision threshold—and it is less likely to make smaller corrections in client-BS association that would increase the overall efficiency of the network. Decision thresholds that control switching in client-centric control algorithms can be used as a control knob—a tuning mechanism that controls both convergence speed and resiliency to noisy metric inference. By increasing the switching threshold, clients switch less frequently and are more resilient to noise—but lose out on optimality.
5.5 Wi-Fi Offloading

Opportunistic client-centric switching based on local availability of non-cellular networks in HetNets has already been well-studied in the context of 802.11 Wi-Fi over different timescales (sub-second to tens of seconds and more) \[18, 64, 81\], and much of it remains applicable to HetNets in a 5G environment. Wi-Fi networks have been shown to be a high capacity option for offloading traffic from cellular networks, accepting 65% of total mobile client traffic while saving 55% of the battery by only offloading data to Wi-Fi when the network is available \[64\].

The Wiffler system \[18\] has also shown up to a 45% reduction in cellular workload for data with a delay tolerance of up to 60 seconds. By leveraging delay tolerance of different types of data and fast switching, Wiffler is able to route data from one RAT to another if the first BS is unable to satisfy the traffic delay requirements. The system is able to opportunistically leverage the offloading capacity of Wi-Fi networks (if present), and fall back upon other traditional cellular networks if no Wi-Fi networks that can satisfy data delay requirements is within reach.

The authors in \[81\] have gone further, developing client-stored models of daily mobility in order to perform predictive forecasts for the mobile client’s radio environment. By leveraging the habitual behavior of people taking similar paths during their daily lives and combining past wireless measurements, a system for determining typical Wi-Fi BS quality and client location can generate connectivity forecasts of which Wi-Fi BSs can be available to in the future, and can assist in opportunistic client-centric data offloading, though the system requires some training.

Many works also address Wi-Fi offloading in the presence of some network prediction. This has been studied for integration of cellular-and-WLAN networks \[2\], which supports network discovery and selection to help clients discover non-cellular networks. In \[94\], HotZones, uses prediction to download delay-tolerant content when close to Wi-Fi BSs. It creates a rank-ordered list of most frequently visited BS based
on past client behavior. This profile is shared with the network operator, and the total aggregate list is broadcast to all clients—in effect, bypassing the high overhead of opportunistic scanning for higher-throughput Wi-Fi RATs. Clients may then connect to various Wi-Fi BSs as they wish with the added knowledge of a measure of load on those BSs, and are found to be able to offload up to 70% of their cellular traffic to those BSs.

Large-scale city traces of mobile clients over the course of 30 days were used in MADNet [35] to evaluate the gains of citywide (San Francisco) Wi-Fi offloading using metropolitan BSs. It allows the cellular network to reduce load by signaling over cellular, but performing download/uploading over Wi-Fi based on explicitly defined client preferences. More than half of cellular traffic was offloaded, and file transfer delay was reduced by more than 50% in the majority of requests.

Many of these client-centric approaches remain valid for HetNets with 5G, as the ubiquitous deployment of Wi-Fi for both indoor and outdoor coverage and offloading is not expected to be replaced anytime soon. These techniques may also be extended for unmanaged 5G RATs deployed as small cells in both licensed and unlicensed spectrum, however these techniques must be updated so that they function on the faster timescale of 5G mmWave RATs (e.g. order of milliseconds) in an efficient way. Opportunistic use of these RATs in the home or office environment within a single room or a floor has the potential to maintain the benefits of load balancing and coverage of Wi-Fi offloading, but also to exploit the higher throughput potential of 5G technologies such as mmWave.

5.6 Game Theoretic Analysis

Non-cooperative game theory, in which individual clients make local decisions to maximize individual payoffs without a means to enforce restrictions on the behavior of
other clients, is often used for distributed coordination without any sort of management from a non-client party—which is the case for distributed RAT selection. In this model, the set of players is the set of clients, and the set of player strategies is the set of BSs (or RATs) that they may associate with at a given time. A game of this type is said to have converged to a Nash Equilibrium (NE) if each player considers its selected strategy to be optimal given the choices of all other players—that is, it cannot unilaterally improve its payoff by changing strategies.

A common class of techniques for distributed coordination is found in the area of non-cooperative congestion games [26, 53], where players select from a common set of strategies to play and the reward of each strategy is a monotonically non-increasing function of the total number of players playing that strategy [74]. In [53], by considering an entire type of RAT to be a single BS in a congestion game framework, a finite improvement path can be found for the congestion game where each asynchronous client can selfishly switch to reduce their cost until a NE is reached. However, to implement this, the authors caution that to reach a pure NE requires exact information from the BS on the client-incurred cost, which may be difficult to obtain in a timely and accurate manner. The authors in [26] model downlink access to multiple broadband BSs as a congestion game, which models the client- and BS-specific cost of association as the congestion impact on other clients sharing the same network. By abstracting away the multi-rate property of HetNets, tight analytical bounds for the price of anarchy and price of stability (ratio between the value of the “best/worst” equilibria points and an optimal solution) are found.

Evolutionary games, in which groups of clients select strategies to play against clients from other groups, have also been considered [84, 97]. In [97], a population game for multihomed RAT association is studied for 802.11 under evolutionary dynamics. Prices based on channel occupancy and total throughput in the cell are used to calculate payoff functions for each client to calculate a potential function for the
population game, and it is shown that the stationary points of such a game are asymptotically stable and maximizes throughput. In [84], an evolutionary game is used to perform client-driven network load balancing between different types of networks (specifically WMAN, cellular, and WLAN). Two solutions to obtaining evolutionary equilibrium are presented. The population evolution solution relies on coordination between clients to share knowledge of the average payoff in a given area, so that underperforming clients may change networks; and reinforcement learning leverages Q-learning to explore and rank the different networks for optimal empirical payoff.

5.7 Probabilistic Analysis

Markov Decision Processes (MDP) have also been used to model the HetNets RAT selection problem at the client side [42, 63, 99, 105]. Clients may internally store empirical knowledge on the rewards (e.g. throughputs) obtained in each state (e.g. BS) that are accessible, as well as their transition probabilities. Every time period, the client must make a decision on which action to take—which RAT or BS to associate with, in order to maximize some expected total reward.

[99] develop a MDP model for vertical handoff between different types of RATs, that considers the link reward for connecting to a BS, the signaling load and processing load when the handoff is performed. The algorithm relies solely on implicit feedback from the network in the link reward obtained after connecting to the BS, and shows improvement over several other algorithms for vertical handoff.

Markov chains can also be used to explicitly model ongoing voice and data sessions on individual RATs [42]. With the assumption that calls and data sessions begin and end sequentially (they arrive/depart individually) and that the total traffic offered to both networks are known, this work develops a 4D Markov Chain to model two
TDMA and WCMDA networks in order to simulate the performance of several RAT selection policies.

Application-specific models can also be used, such as for Video-on-Demand \cite{63}. In this work, multihomed clients optimize their choice of RATs for the minimization of video playback disruption costs and the communication cost of receiving a video chunk over a given RAT. The MDP is used to determine which RAT to send a chunk request to at a given time, and the resulting adaptive ATAC policy is shown to have lower costs than policies with static thresholds for request allocation.

\section*{5.8 Current State of the Industry}

The goal of HetNets is to enable the seamless transition between network association, transfer of data, switching between networks, and dissociation for a client with access to multiple wireless networks of different types, in a way that maximizes the aggregate utility of the clients and networks involved. With such a vast number of different types of RATs, each with different specifications, effective transmit/receive ranges, supported data-rates, and speed of channel changes, finding a simple unifying algorithm for HetNets has been a question that many organizations in industry have considered.

There are several obstacles to developing a common approach to the integration of new technologies such as 5G into the HetNets architecture, such as noise, fast temporal variations, hybrid control schemes, multi-homing and load balancing. Recently, industry groups such as 3GPP and IEEE have been pressing forward with standardization initiatives to enable HetNets for existing technologies, as well as laying the groundwork for next-gen 5G technologies such as mmWave.

In Release 12 \cite{6}, multiple enhancements were released to improve client mobility in HetNets for both LTE and UMTS, including cell discovery (client-based discovery,
network-based discovery and collaborative client and network-based discovery), and
general enhancements for small cell deployments. Furthermore, 3GPP has included
standardization for dual connectivity (simultaneous multihoming to both macro- and
microcells by clients) [70], allowing for dynamic traffic routing over multiple paths.

However, on the core philosophy of where HetNet control should reside within
the network, the debate is still ongoing in 3GPP: between centralized solutions that
place client-BS association decisions in the network, distributed solutions that place
client-BS association decisions strictly on the client, and the hybrid approach, where
the association decision is made by the network guiding clients in the association
process. Several different techniques have been proposed [5], ranging from client-
centric BS selection with network assistance (broadcasts) and with network assistance
and network-advised policies, as well as network-controlled BS selection with network-
determined policies.

Without a solid determination of who should control BS selection and where that
intelligence shall be placed, many industry leaders in the consumer electronics world
have been moving forward with their own homegrown approaches to managing access
to HetNets. In the past, Apple Inc. had built their flagship smartphone, the iPhone,
such that it automatically offloaded the client onto Wi-Fi wherever possible in order
to decrease use of the client’s subscribed data plan. Furthermore, in early 2016, Apple
introduced Wi-Fi Assist, a feature that detects a poor Wi-Fi signal and automatically
switches the mobile device back to using data from your cellular plan [13].

When the Wi-Fi Assist feature was released, however, there was a large backlash
against Apple because many device owners were either uninformed or unaware of the
feature which resulted in individuals incurring thousands of dollars worth of data
overage charges due to inadvertent data use [79]. This approach, although done in
the best interest of maximizing quality of service by finding the BS with the best
signal, clearly demonstrates the overall failure of the system when it is designed to
optimize an objective or metric that doesn’t match that of the client (in this case, total monetary cost was not considered) even though the BS association ostensibly gave full control to the user.
Chapter 6

Conclusion

Heterogeneity of wireless network architectures is everywhere in modern life: from the Bluetooth that connects one’s wearable activity trackers to their mobile devices or their devices to their cars, to the 802.11 Wi-Fi that connects their laptops to their wireless internet routers, from next-generation 5G millimeter wave office and home networks, to the traditional cellular 3G/4G/LTE networks that enable mobile data transfer, there is an ever-increasing number of radio access networks for a wireless client device to connect to. Traditionally, the problem of determining which network a client device should associate with has always been performed by placing the intelligence inside the cellular network (in most academic work) to enable the formulation of the problem as a global optimization problem that can be solved using existing techniques. However, with such schemes running into practical problems such as unsustainable scalability for channel measurement aggregation in support of a centralized solution, as well as the political problem of getting multiple network operators to agree to share information on customers and their proprietary networks, a fully centralized solution is virtually impossible to implement, no matter how elegant the mathematics can be for this centralized model.
On the industry side, there has been a long history of companies preferring a more simple approach to client-RAT association: to simply offload clients from cellular networks to Wi-Fi networks. By doing so, they can provide their customers with some data while decreasing the amount of data being sent over cellular networks. Although this type of solution aids the network providers in lessening the load on their networks, this type of solution is not ideal for their customers. By restricting clients from associating with cellular networks if Wi-Fi are available, clients may experience drastically decreased network performance: especially if many clients are similarly offloaded. Recently, as evidenced by the introduction of Apple Wi-Fi Assist earlier this year, hardware vendors are starting to take the client’s network experience into account—however, it is clear that such a scheme still depends on network providers or hardware vendors to determine how poor the Wi-Fi network needs to become prior to allowing client-cellular association.

In this dissertation, we have rejected this limited view that Radio Access Technology association should only be done by either the network operators via some centralized control mechanism or hard-coded into the device by hardware manufacturers to optimize for their own unknown metrics in favor of expanding potential RAT selection approaches. With the widespread proliferation of mobile client devices (e.g., smartphones, tablets, etc.) with ever-increasing processing power, it is now possible for the client device to make this decision at the network edge without involving the network core. With increased computational power on each mobile device, clients can potentially perform RAT selection in HetNets locally, in support of the user’s active applications and usage habits, instead of relying on the network operator to assign RAT association based on some arcane metric important to the vendor, but less so for the client. Even if client-centric control of RAT association may not result in an optimal pareto-efficient throughput allocation (as shown by the bound on Pareto-Efficiency Gain in Chapter 2), the question of how client-centric control could
be implemented, and how well it could perform is a question whose answer is very relevant to how future mobile devices should select RATs.

In the case of client-centric RAT selection for HetNets where the client can be provided with a perfect global big-picture view of all others in the HetNet and their peak physical-layer rates (Chapter 2), we modeled the problem as a non-cooperative game where each client had to pick one RAT from many in order to maximize the throughput obtained from the given RAT. We leveraged ideas from both non-cooperative game theory and binary exponential backoff, as well as introduced the concept of hysteresis, a local memory of the last-obtained-throughput on different RATs to prevent oscillations and guarantee convergence to a Nash Equilibrium network configuration. We then showed bounds on the goodness of such Nash Equilibria, as well as the convergence time of such an algorithm.

In the case of client-centric RAT selection for HetNets where the client can only obtain some partial information on the wireless channel (Chapter 3), we modeled each RAT in the HetNet as a Markov Chain with an unknown state transition probability matrix. Modeling the problem as a multi-armed bandit where the client had to learn the throughput distributions of each RAT, we developed the mHS algorithm that guaranteed order-optimal total regret. $O(\ln(t))$ total regret was obtained by minimizing switching costs through grouping samples of a particular network (Switching Scheduler algorithm), and selecting RATs by ranking them based on the client’s desire to exploit the optimal RAT and to explore alternatives in the hopes of finding a higher-throughput network.

In the case of client-centric RAT selection for HetNets in the complete absence of network-provided information (Chapter 4), we again modeled each RAT in the HetNet as a Markov Chain—but this time, neither the reward distribution of each network nor any information on state transitions was available to the client. Clients were assumed to only know the peak throughputs allowed by the protocol standards for each RAT,
as well as any passive channel measurements that could be locally obtained. By using reinforcement learning to “learn” the average “mapping” between observed passive channel measurements and the average empirical throughput obtained by the client in the past in that channel state, we developed the WIFFN algorithm and showed it was upper-bounded in regret by the same $O(\ln(t))$ and thus asymptotically optimal in time-averaged regret. Furthermore, we introduced the concept of “binning,” or coarsification, of passive measurements: by increasing the number of entries in the “mapping” generated by reinforcement learning, the client could decide how to trade off between a small number of passive measurement “bins” (resulting in minimal initial throughput and decreased initialization time) and a large number of “bins” (resulting in a greater initial incurred regret, but more accurate throughput estimation and reduced regret growth rate over time).

6.1 Future Work

As the growth of computational power on the mobile client device grows, it is increasingly possible to leverage the techniques discussed in this survey work to place the intelligence for RAT selection on the network edge, literally in the hands of the user. With massive amounts of machine-type communication and the rise of IoT on the horizon, traditional centralized decision schemes where a centralized controller gathers channel information from all users about their channel conditions and local radio environment is no longer tenable for the massive scale that can be expected from the rise of these new networking trends for smart home, smart office, and smart everything.

Therefore, new network selection approaches must be developed for HetNets in the context of 5G that can take advantage of the capabilities available on the network
edge—some of that work has already begun. However, many questions still remain to be studied in detail:

- **Network Assistance for Client-Centric HetNets.** To what degree should the client be autonomous in selecting a RAT for association? There is a vast gulf between fully-distributed approaches and hybrid solutions that leverage a degree of network assistance.

- **Objective Formulation.** How should the objective be designed for client-centric RAT association? Many approaches involve network-centric or client-centric objectives, but perhaps a combination of the two can balance the needs of both the network and the client.

- **Performance Gap.** How worse off will these client-centric solutions be compared with a centralized implementation with global knowledge? Can this gap be quantified in terms of Price of Anarchy or Stability?

- **Timescales.** The temporal variability of the channel conditions of newer RATs (e.g. mmWave) may be different from existing technologies: How should RAT Selection algorithms account for this difference in timescales of channel variation?

Furthermore, there have been additional criteria that tend to be assumed in discussions on HetNets—but are not necessarily guaranteed. These criteria simplify analysis, but cannot be assumed in the general case:

- **Quota.** The mobile device’s data quota is assumed to be infinite, and it always has a positive marginal utility for downloading or uploading more data.

- **Battery.** The battery capacity of the mobile device is assumed to be infinite, and many of the works in this survey do not consider the joint problem of RAT selection under explicit finite power constraints.
• **WiFi Coverage.** Smartphones and mobile devices are assumed to always have access to an alternative network, and it is always possible to access both cellular and Wi-Fi networks, which may not be true in certain scenarios (e.g. rural settings).
Chapter 7

Published Works


Bibliography


