UNIVERSITY OF CALIFORNIA
RIVERSIDE

Flexible Resource Manangement in Wireless Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Shailendra Singh

March 2016

Dissertation Committee:

Dr. Srikanth V. Krishnamurthy, Chairperson
Dr. Jiasi Chen
Dr. Michalis Faloutsos
Dr. Eamonn Keogh

The Dissertation of Shailendra Singh is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost I want to thank my advisor Dr. Srikanth V. Krishnamurthy. I am deeply indebted to Srikanth for his fundamental role in my doctoral work. Srikanth provided me with every bit of guidance, assistance, and expertise that I needed during my first few semesters; then, when I felt ready to venture into research on my own and branch out into new research areas, Srikanth gave me the freedom to do whatever I wanted, at the same time continuing to contribute valuable feedback, advice, and encouragement. In addition to our academic collaboration, I greatly value the close personal rapport that Srikanth and I have forged over the years. I quite simply cannot imagine a better adviser.

I would like to thank Dr. Harsha V. Madhyastha for his invaluable collaboration and his close interest in my work during the short time that we have worked together. I would also like to thank Dr. Eamonn Keogh, Dr. Michalis Faloutsos and Dr. Jiasi Chen for participating in my defense committee. In short, I feel privileged to be recognized as a doctor by such a distinguished committee. In addition, I wish to thank Dr. Karthikeyan Sundaresan and Dr. Sampath Rangarajan for giving me a chance to have an internship at NEC Laboratories America Inc., and for their excellent mentoring and constant support. I would also like to thank my friends and colleagues who have made this long journey a lot more enjoyable and memorable: Golnaz Ghiasi, Dr. Smruti Parichha, Ahmed Atya, Dr. Jiasi Chen, Dr. Masoud Akhoondi, Dr. Michael Butkiewicz, Dr. Huy Hang, Dr. Indrajeet Singh, Tuan Dao and many more. I cannot thank enough my brother Kapil Malik and cousin Dr. Anil Malik for their support and encouragement throughout my PhD. I must acknowledge with tremendous and deep thanks to Moloud Shahbazi. Thank you for supporting me for everything, and especially I cant thank you enough for encouraging me throughout this experience. Lastly, and most importantly, I wish to thank my parents, Mahendra Singh Malik and Krishna Malik. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

In loving memory of my father, Mahendra Singh Malik

ABSTRACT OF THE DISSERTATION

Flexible Resource Manangement in Wireless Networks

by

Shailendra Singh

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, March 2016
Dr. Srikanth V. Krishnamurthy, Chairperson

Surging data demand driven by the proliferation of smartphones is revolutionizing the wireless scene. By 2020, the number of smartphone subscribers will grow to 9 billion and the demand in mobile data traffic will grow 10 folds. Capacity (bandwidth) and Latency are the key contributing factors to a positive user experience. For example, streaming a video from Netflix or Youtube is bandwidth limited while loading or browsing the Youtube or Netflix website is latency limited. One can argue that by using more spectrum(wider wireless channels) can solve the capacity problem however according to Federal Communications Commission(FCC) we are already facing a spectrum deficit. Newer network technologies like 4G and LTE have improved the network performance(both capacity and latency) but 60world's mobile subscribers will still be using the slower 3G or 2G connections for at least a decade. Another challenge that mobile devices are posing is that the networks previously dominated by the static users (or static channel conditions) with devices like laptops are increasingly becoming more heterogeneous(mix of static and mobile users with varying channel conditions).

In this dissertation, I address the issue of managing wireless resources to support various multimedia applications, in wireless networks with a high degree of user mobility. Driven by necessity for flexible wireless resource management, I design, build and evaluate three wireless resource management schemes. First, I present JPRA, a joint power

and rate adaptation scheme which adaptively distributes uneven power levels of OFDMA sub-carriers to cope with frequency selectivity in fading. Thus, improves the total network capacity. Second, I present TRINITY, a practical transmitter cooperation scheme to handle heterogeneous user profiles in wireless networks. Sophisticated MIMO-based transmission strategies, based on transmitter cooperation, have emerged recently. However, different types of users profiles (e.g., static vs mobile, stable vs dynamic channels) that make up todays enterprises, require different MIMO transmission strategies(CMSA, Network MIMO and DAS). With the wrong strategy, a user could even see a degradation in performance. TRINITY can simultaneously cater to a heterogeneous mix of users, by intelligently combining a plurality of MIMO transmission strategies wherein the transmitters at different nodes can cooperate to deliver significant performance gains.

Next, I present FluidNet, a scalable, light-weight framework for realizing the full potential of Cloud-based radio access networks(C-RAN). Cloud-based radio access networks (C-RAN) have been proposed as a cost-efficient way of deploying small cells. In this work, we argue that the intelligent configuration of the front-haul network between the baseband processing units(BBUs) and remote radio heads, is essential in delivering the performance and energy benefits to the RAN and the BBU pool, respectively. FluidNet deploys a logically re-configurable front-haul to apply appropriate transmission strategies in different parts of the network and hence cater effectively to both heterogeneous user profiles and dynamic traffic load patterns.

Finally, I present FlexiWeb, a network aware compaction scheme for accelerating mobile web transfers. Latency is often cited as the main reason for poor mobile web performance and to reduce the page load times, middleboxes that compress page content are commonly used today. Unfortunately, this can hurt performance in many cases. Our measurements reveal that the middlebox should be used only when network conditions are bad; otherwise, most objects in the web page should be directly fetched from the source web server. Based on this observation we build FlexiWeb, a framework that supports network-

aware middlebox usage and performs dynamic network-aware compression to provide further performance gains.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networks are increasingly becoming important in enabling convenient Internet access for users. Currently, there are as many mobile subscriptions as people in the world, and every second, 20 new mobile broadband subscriptions are activated [2]. This has led to the increase in mobile users, data consumption has also increased. Consequently, mobile data traffic will grow ten folds by 2020, largely driven by increased video consumption and mobile web usage on mobile devices. Almost 60 percent of all mobile data traffic will be from video by 2020 [2]. Apart from mobile phones, there will also be a multitude of other connected devices communicating. According to forecasts [2] out of 28 billion connected devices more than 15 billion will be Internet of Things(IoT) and consumer electronic devices by 2021 [2].

On the other hand GSM/EDGE and 3G subscriptions currently represent the largest share of mobile subscriptions, and they will continue to dominate for another decade [2]. In developed markets, there has already been a wide adoption of more advanced technologies like LTE, but on a global scale this has resulted in a marginal decline in GSM/EDGE and 3G subscriptions. The latency of GSM/EDGE and 3G networks is very high (Avg. RTT 800ms) while the bandwidth is lower in comparison to 4G and LTE networks. These slower networks severely impacts the performance of applications like mobile web and video. Typically videos rely more on the bandwidth than the latency while mobile web depends more on the latency than the bandwidth. Furthermore, we are running out of wireless spectrum to handle the increased capacity demands. According to FCC we are already in a spectrum deficit. In summary, in order to satisfy the capacity and latency requirements of mobile applications we need to manage our wireless resources efficiently. In

this dissertation, we first present resource allocation techniques to increase wireless capacity by exploiting the frequency selective fading to reduce the capacity wastage. Then, we present a scheme to improve wireless capacity by mapping different user profiles to appropriate transmission strategies. The last capacity improving scheme that we propose leverages the heterogeneity in both user and traffic in cloud based radio access networks for small cells. To handle the latency issue, specifically in mobile web, we present a scheme which allows for fine grained proxy control to improve page load time. We show that our techniques offer significantly higher performance than their state-of-the-art alternatives. Next, we detail our contributions in solving each of these problems.

1.1 Resource Management and Its Application to Increase Capacity

1.1.1 Frequency selective fading

In today's OFDMA networks, the transmission power is typically fixed and the same for all the sub-carriers that compose a channel. The sub-carriers though, experience different degrees of fading and thus, the received power is different for different sub-carriers; while some frequencies experience deep fades, others are relatively unaffected. In this paper, we make a case for redistributing the power across the sub-carriers (subject to a fixed power budget constraint) to better cope with this frequency selectivity. Specifically, we design a joint power and rate adaptation scheme (called JPRA for short) wherein power redistribution is combined with sub-carrier level rate adaptation to yield significant throughput benefits. We further consider three variants of JPRA: (a) JPRA-Basic where, the power is redistributed across sub-carriers so as to support a maximum common rate across all the sub-carriers (b) JPRA-Intermediate where, the power is redistributed across sub-carriers so as to support a maximum common rate across a "subset" of sub-carriers such that the aggregate rate is maximized. (c) JPRA-Adaptive where, the goal is to redistribute power such that the transmission time of a packet is minimized. While the first two variants decrease transceiver complexity and are simpler, the third is geared towards achieving the maximum throughput possible. We implement all three variants of JPRA on our WARP radio testbed. Our extensive experiments demonstrate that JPRA can provide a 35% improvement in total network throughput in testbed experiments compared to FARA, a scheme where only

sub-carrier level rate adaptation is used. We also perform simulations to demonstrate the efficacy of JPRA in larger scale networks.

1.1.2 Mapping Users with Appropriate Transmission Strategies

The proliferation of smartphones and tablet devices is changing the landscape of user connectivity and data access from predominantly static users to a mix of static and mobile users. While significant advances have been made in wireless transmission strategies (e.g., beamforming, network MIMO, etc.) to meet the increased demand for capacity, such strategies primarily cater to static users. To cope with growing heterogeneity in data access, it is critical to identify and optimize strategies that can cater to users of various profiles to maximize system performance and more importantly, improve users' quality of experience.

Towards this goal, we first show that users can be profiled into *three* distinct categories based on their data access (mobility) and channel coherence characteristics. Then, with real-world experiments, we show that the strategy that best serves users in these categories varies distinctly from one profile to another and belongs to the class of strategies that emphasize either *multiplexing* (eg., network MIMO), *diversity* (eg., distributed antenna systems) or *reuse* (eg., conventional CSMA). Two key challenges remain in translating these inferences to a practical system, namely: (i) how to profile users, and (ii) how to combine strategies to communicate with users of different profiles simultaneously. In addressing these challenges, we present the design of *TRINITY* - a practical system that effectively caters to a heterogeneous set of users. We implement and evaluate a prototype of TRINITY on our WARP radio testbed. Our extensive experiments show that TRINITY's intelligent combining of transmission strategies improves the total network rate by 50%-150%, satisfies the QoS requirements of thrice as many users, and improves PSNR for video traffic by 10 dB compared to individual transmission strategies.

1.1.3 User and Traffic Heterogeneity in Small Cells

Cloud-based radio access networks (C-RAN) have been proposed as a cost-efficient way of deploying small cells. Unlike conventional RANs, a C-RAN decouples the baseband processing unit (BBU) from the remote radio head (RRH), allowing for centralized operation of BBUs and scalable deployment of light-weight RRHs as small cells. In this work, we argue that the intelligent configuration of the front-haul network between the BBUs and RRHs, is

essential in delivering the performance and energy benefits to the RAN and the BBU pool, respectively.

We then propose *FluidNet* - a scalable, light-weight framework for realizing the full potential of C-RAN. *FluidNet* deploys a logically re-configurable front-haul to apply appropriate transmission strategies in different parts of the network and hence cater effectively to both heterogeneous user profiles and dynamic traffic load patterns. *FluidNet* 's algorithms determine configurations that maximize the traffic demand satisfied on the RAN, while simultaneously optimizing the compute resource usage in the BBU pool. We prototype *FluidNet* on a 6 BBU, 6 RRH WiMAX C-RAN testbed. Prototype evaluations and large-scale simulations reveal that *FluidNet* 's ability to re-configure its front-haul and tailor transmission strategies provides a 50% improvement in satisfying traffic demands, while reducing the compute resource usage in the BBU pool by 50% compared to baseline transmission schemes.

1.2 Resource Management and Its Application to Decrease Latency in Mobile Web

To reduce page load times and bandwidth usage for mobile web browsing, middleboxes that compress page content are commonly used today. Unfortunately, this can hurt performance in many cases; via an extensive measurement study, we show that using middleboxes to facilitate compression results in up to 28% degradation in page load times when the client enjoys excellent wireless link conditions. We find that benefits from compression are primarily realized under bad network conditions. Guided by our study, we design and implement FlexiWeb, a framework that determines both *when* to use a middlebox and *how* to use it, based on the client's network conditions. First, FlexiWeb selectively fetches objects on a web page either directly from the source or via a middlebox, rather than fetching all objects via the middlebox. Second, instead of simply performing lossless compression of all content, FlexiWeb performs network-aware compression of images by selecting from among a range of content transformations. We implement and evaluate a prototype of FlexiWeb using Google's open source Chromium mobile browser and our implementation of a modified version of Google's open source compression proxy. Our extensive experiments

show that, across a range of scenarios, FlexiWeb reduces page load times for mobile clients by 35–42% compared to the status quo.

1.3 Outline of this dissertation

The rest of this dissertation is organized as follows. In chapter 2, we detail the design JPRA - a joint power and rate adaptation scheme to alleviate frequency selective fading in wireless networks. In chapter 3, we describe our experimental study that reveals an interesting set of observations and guidelines for mapping different user profiles to an appropriate transmission strategy. Chapter 4 presents FluidNet - a scalable, light-weight framework for realizing the full potential of C-RAN. Finally in Chapter 5, we present FlexiWeb - a framework that determines both when to use a middlebox and how to use it, based on the clients network conditions.

Chapter 2

Adaptive Sub-carrier Level Power Allocation in OFDMA Networks

2.1 Introduction

The surge in demand for methods to transmit large amounts of data over the wireless channel has led to recent interest in orthogonal frequency division multiplexing (OFDM). The time dispersive and frequency-selective [3] nature of the wireless channel causes intersymbol interference (ISI) and signal degradation (due to fading) which are serious impediments in wireless communications. OFDM combats frequency-selectivity fading [3], by dividing the channel into subcarriers much narrower than the channel coherence bandwidth. This reduces a frequency-selective fading channel into a set of flat fading subcarriers, Multiple measurement studies [4, 5] have validated the existence of frequency diversity in practice. The current implementations as specified by the IEEE standards [6] for such systems do not include schemes for adaptively assigning powers to subcarriers based on CSI (Channel State Information); instead, these systems evenly distribute the total transmission power budget across all subcarriers (waterfilling [7] with equal power without CSI knowledge).

However, at a receiver, since each sub-carrier typically experiences a different fade the transmission rates that can be supported by the different sub-carriers can differ. Figure 2.1 shows such a scenario where different subcarriers show a difference in SNR of up to 12dB on a single link.

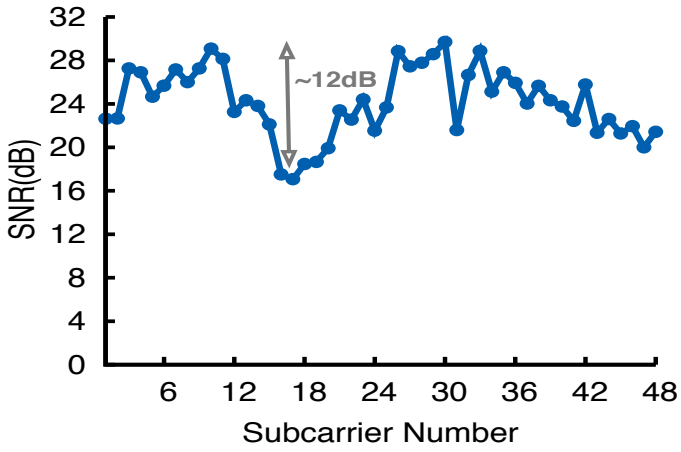


Figure 2.1: Frequency selective fading across 20 MHz of spectrum; the SNRs of the different subcarriers can differ by as much as 12 dB on a single link.

Spatial and temporal diversity can further complicate the communications between a single sender and multiple receivers at different times. Each receiver will likely have a different multipath fading profile given the differences in receiver locations. Even for the same receiver, the fading effects experienced by the different sub-carriers can be expected to differ in time.

Traditional rate-adaptation algorithms try to cope with fading by varying the transmission rate in response to either packet losses (e.g., [8]) or signal-to-noise ratio (SNR) variations (e.g., [9]). However, these are not done at a per sub-carrier level. Thus, poor received quality on a few sub-carriers can affect the supported transmission rate. To cope with this, Rahul *et al.* [4], propose FARA, a scheme that supports per sub-carrier rate adaptation. While FARA provides gains over traditional rate-adaptation schemes, it does not address the problem of frequency-selectivity directly and does not utilize the available power budget efficiently.

In this paper, we propose redistributing the power across the sub-carriers to better cope with frequency selectivity. Specifically, we argue for an adaptive, scenario specific, uneven distribution of transmission powers across sub-carriers. By appropriately assigning the transmission powers to the different sub-carriers (while adhering to the total power budget for the transmission), and by combining this with the choice of appropriate per sub-carrier transmission rates, we envision achieving significant gains in throughput. As our main contribution, we design a joint power and rate adaptation (JPRA) scheme towards realizing these envisioned benefits.

Calibration phase in JPRA: In order to determine the right power levels for each sub-carrier, a calibration phase is necessary for JPRA. The goal of calibration is twofold; (i) to identify the *practical* decoding thresholds for each transmission rate, and (ii) to correlate the required change in the received signal of a sub-carrier (in order to support a rate) with an appropriate change in the corresponding transmit power at the sender. We perform several measurements to understand if such a calibration phase is viable. We use the error vector magnitude (EVM) [10] to determine the minimum power level needed for sustaining a specific transmission rate. We find that these minimum powers are consistent over a large number of scenarios and node locations. This suggests that infrequent benchmark measurements are sufficient to calibrate JPRA and utilize it for adaptive sub-carrier level power allocation during data-transfers.

Variants of JPRA: We propose three variants of JPRA. The first variant, named JPRA-Basic, selects the power-assignments such that a 'single best rate' can be used on all the sub-carriers. This best rate corresponds to the highest rate for which the EVM threshold is satisfied for all the sub-carriers at the receiver. As one can easily see, since we operate on a fixed power budget, the scheme entails the transfer of power from relatively good sub-carriers (unaffected by fading) to those that are in deep fades. The advantage of this approach is that it is simple and reduces the transceiver complexity (since they only need to decode at a single rate). However, this may not yield the maximum achievable throughput with power re-allocation. As shown later however, it does offer an additional benefit in ensuring that transmission rate transitions are less likely than in traditional approaches (stabilizes the rate in use). In addition, it can potentially provide significant power savings for a target PDR, compared to traditional systems.

An implicit assumption made with JPRA-Basic was that all the sub-carriers can at least support the minimum rate; if not the link under consideration may not be able to carry any data whatsoever. Furthermore, there may be carriers that are in deep fades and if power is allocated to such carriers, it may be detrimental to the overall throughput performance. One may want to retain the simplicity of JPRA-Basic, but at the same time, try to improve the throughput performance that it yields. Towards this, we propose a second variant JPRA-Intermediate. Wherein, we sequentially eliminate the worst sub-carrier (meaning that no power is assigned to that sub-carrier) and re-distribute the power across all the other sub-carriers. After this, the transceiver can then compute the achievable common transmission rate. Note here that there is an inherent trade-off. Since we eliminate

sub-carriers, there is a decrease in the raw capacity available on the link. On the other hand, a higher transmission rate is likely to be possible on the remaining sub-carriers. Our objective is to find the best set of sub-carriers such that the aggregate rate is maximized. Essentially this variant of JPRA constitutes an intermediate step between JPRA-Basic and JPRA-Adaptive (described next).

Towards maximizing throughput via sub-carrier level rate allocation, we propose a third variant JPRA-Adaptive. Our goal with this scheme is to minimize the total transmission time of a packet by appropriate power re-distribution. Essentially, with JPRA-Adaptive, powers are assigned to sub-carriers such that the total number of bits (as mapped to symbols on a constellation) transmitted on the sub-carriers in each symbol-duration is maximized. To illustrate with an example, consider a case where we have just two sub-carriers. Assume that the power assignments are such that either both sub-carriers can be modulated using QPSK (2 bits per symbol) or, one sub-carrier can be modulated with BPSK (1 bit per symbol) while the other can be modulated with 16-QAM (4 bits per symbol). In this case, the packet airtime is minimized with the latter choice since more bits are transmitted per symbol duration (5 bits instead of 4 bits) and is consequently chosen by JPRA-Adaptive. We wish to point out here that in order to support a higher modulation, a higher received power is required. Thus, in the latter case the power on the second sub-carrier (supporting 16-QAM) is increased compared to the power used in the former case (when it supported QPSK). Similarly, the power on the first sub-carrier has to be decreased in the latter case compared to the former one (and thus, it can support only BPSK as opposed to QPSK).

We implement all the versions of JPRA on our six node WARP radio testbed. We also implement FARA [4] for comparison. We perform extensive experiments which show that JPRA-Adaptive outperforms FARA by as much as 35 % in terms of throughput. The gains are much more significant compared to traditional OFDMA systems (75 %). We also show the efficacy of JPRA in larger scale settings via simulations.

The main properties of JPRA are summarized below:

- *Throughput Efficiency*: JPRA-Adaptive achieves a 75% increase in more network throughput as compared to standard SNR based rate adaptation. It also outperforms the state of the art OFDMA rate adaptation scheme, FARA[4] by 35% in terms of total network throughput.

- *Potential Power savings*: JPRA-Basic is more power efficient compared to systems with no power re-distribution. Specifically, for a fixed BER (bit error rate), we observe up to 4.5 dBm in power savings.
- *Stability of Transmission Rates* : JPRA-Basic reduces the number of rate changes by 30% in comparison to standard SNR based rate adaptation.
- *Impact on Carrier Sensing*: JPRA does not affect the RSS (received signal strength) of packets since the total power budget remains fixed.

We acknowledge that our system is currently applicable only on (quasi) static topologies. The calibration phase required for the deployment of our scheme cannot be as effective in scenarios that include mobility. However, we seek to examine mobility scenarios as part of our future work.

Organization: The paper is organized as follows. In Section 2, we provide relevant background and overview related work. The calibration phase of JPRA is described in Section 3. Details of the two JPRA variants for power/rate allocation are in Section 4. Section 5 describes both our experimental and simulation results. Section 6 concludes the paper.

2.2 Background and Related Work

In this section, we first describe studies related to our work and then, we briefly provide background on the EVM calculation required by our system.

Frequency Selective Fading: There are studies that employ rate adaptation to cope with frequency selective fading. For example, in [4] the authors propose Frequency Aware Rate Adaptation (FARA) to improve system performance. They assign subsets of sub-bands to each sender-receiver pair and based on the SNR reported by the receiver on these sub-bands, the sender performs rate adaptation. However, the authors do not propose a solution to improve the performance of sub-carriers experiencing frequency selective fading or low SNR. Barthia *et al* [5] propose a smart mapping of symbols to sub-carriers. This supports partial recovery of symbols if they are lost due to frequency selective fading. They also propose an extra layer of FEC codes on top of Physical layer FEC. One of the main limitations of this work is that the proposed solution is only compatible with block FEC schemes and its not clear how it will work with convolutional or turbo code FECs [6]. A large

amount of feedback information is also required for partial symbol recovery. In contrast our scheme is not limited by the choice of FECs and the amount of feedback information is low (we only need to send sub-carrier power values and rates).

Rate Adaptation : There is a large volume of studies on rate adaptation (e.g., [11, 9]). SampleRate [11], proposed by Bicket *et al.*, probes the performance at a random rate every 10 frames, and selects the rate that minimizes the expected transmission time (including retransmissions). Wong *et al.* [8] develop Robust Rate Adaptation Algorithm (RRAA), which uses short term loss ratios to opportunistically change rate. It further incorporates an adaptive RTS filter to prevent collision losses from lowering data rates. All these (and many similar) existing schemes adapt rate according to frame loss rates. RBAR [12] uses the RTS/CTS exchange to estimate the SNR at a receiver, and picks the transmission bit rate accordingly. OAR [13] further builds on RBAR, by opportunistically transmitting back-to-back frames when the channel quality is good. CHARM [14] leverages the reciprocity of the wireless channel to estimate the average SNR at the receiver using packets overheard from the latter. The overhead of RTS/CTS (present with RBAR and OAR) is thus avoided and implementation on commodity cards is enabled. Sen *et al* [15] propose the use of EVM (Error Vector Magnitude) to perform rate adaptation. Since the above rate adaptation schemes use information such as loss ratio, SNR and EVM averaged over a packet, they fail to capture the effects of frequency selectivity. On the contrary, we try to directly address issues related to frequency selective fading by using per subcarrier EVM measurements.

Multi-User OFDM : There exist a few studies on subcarrier, power and rate assignments in multi-user scenarios. In this case a single channel is shared among multiple users. In [16] Javidi *et al* propose a scheme to outperform "water filling based multi-user subcarrier assignment" by introducing a subcarrier allocation scheme which takes the queue lengths of different users into account. Adaptive power allocation for a multiuser OFDM environment has been proposed in [17, 18]. An OFDM channel is divided into multiple subbands and these subbands are assigned to different users. To alleviate frequency selective fading experienced by different users, redistribution of power and modulation on these subbands is done according to the SNR experienced by the users. However, the majority of these studies are evaluated only via simulations by making assumptions about channel conditions. Furthermore, power adaptation is done for each user and not on a per OFDM sub-carrier basis.

Error Vector Magnitude(EVM): In this work we use the EVM per subcarrier as the CSI (carrier state information) feedback from the receiver, for the sender to perform power redistribution and rate selection. EVM is a vector measurement taken in terms of peak (or rms) percentages between the ideal symbol position and the actual measured position in the constellation space for a particular modulation. The error vector is a vector in the I-Q plane between the ideal constellation point and the data interpretation by the receiver [10]. In other words, it is the difference between the actual received symbols and the ideal symbols. The average power of the error vector, normalized to the signal power, is the EVM. It can be expressed as a percentage:

$$\text{EVM}(\%) = \sqrt{\frac{P_{error}}{P_{reference}}} * 100$$

where, P_{error} is the RMS power of the error vector and $P_{reference}$ is defined as the reference constellation average power. In contrast to SNR, higher EVM values correspond to bad channel conditions, while lower EVM values represent good channel conditions. Note that in JPRA, SNR can be used in lieu of EVM with very minimal changes. It only requires changing of the EVM based rate table to a SNR based rate table and replacing the EVM with SNR in the calibration phase; Specifically, in this phase changes in SNR with changes in transmission powers will be recorded.

2.3 Calibrating Phase of JPRA

JPRA is a measurement driven system and requires a set of calibrating measurements. These measurements are:

1. *Per sub-carrier EVM for each received packet.* This provides the system, knowledge with respect to both the sub-carriers that are experiencing deep fades and those that are relatively unaffected.

2. *EVM threshold for supporting each available rate.* This identifies the modulation that a sub-carrier can support.

3. *A mapping between a power adjustment and the corresponding EVM change.* This is critical in determining the effect of a power change on the rate that can be supported by a sub-carrier. In other words, it seeks to answer: By increasing (or decreasing) the

transmission power on a sub-carrier by a certain magnitude, what is the higher (or lower) rate that that can be achieved on that sub-carrier ?

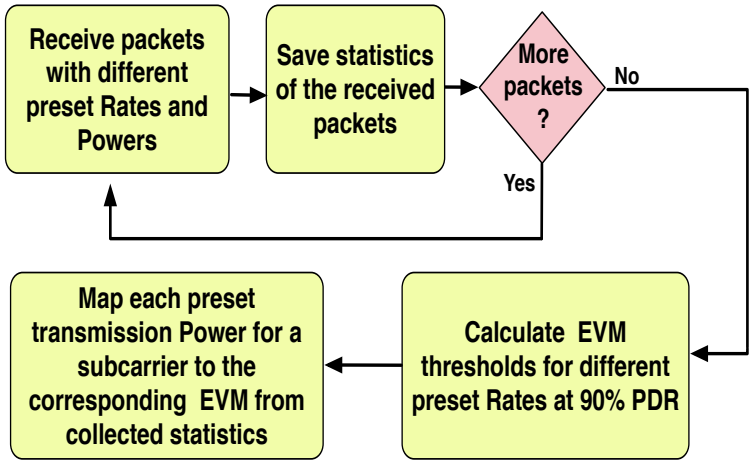
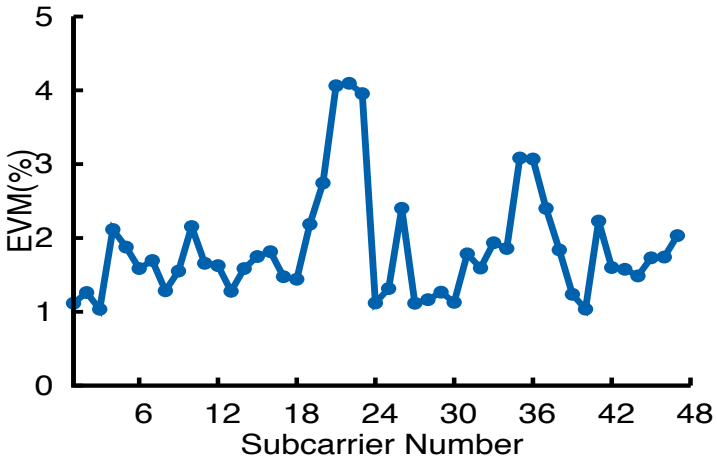


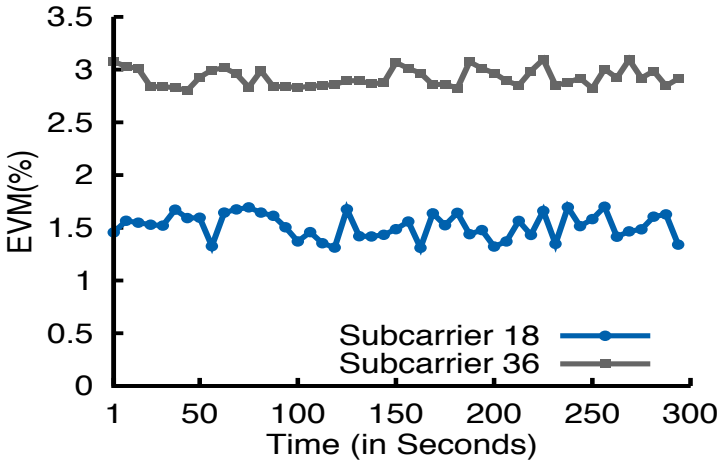
Figure 2.2: Calibration phase of JPRA

In what follows we provide more details on the way we obtain these measurements. Figure 2.2 outlines the main steps involved in calibrating JPRA. We defer a detailed description of our testbed to later; simply put, we have a six node WARP testbed that we use for all of these calibrations.

Calculating per sub-carrier EVM: EVM is an indicator of modulator or demodulator performance in the presence of impairments. For a receiver to calculate the EVM, both received and transmitted packets are required. The WARP boards provide the per sub-carrier EVM for every correctly received packet. If the packet passes the CRC check, the receiver can reconstruct the exact signal that was transmitted. The bits are re-mapped on to a constellation space to obtain ideal symbol positions based on the modulation used for the transmission. With the ideal symbol positions, it is easy to calculate the EVM as described in Section 2.2. With the above process it is not possible to calculate the EVM for packets that fail the CRC check; the receiver cannot deduce what was transmitted and thus, cannot obtain the ideal symbol positions. Thus, the EVM estimates are updated only based on packets that are correctly received. Note that the EVM for a corrupt packet can be calculated using a simple pilot based EVM calculation scheme; in fact, such a scheme is already being used in some of the commercial atheros based wireless cards [19].



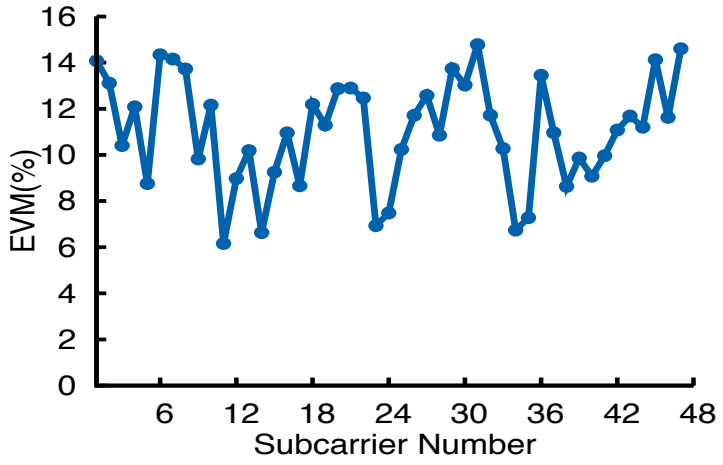
(a) EVM values for different subcarriers for a static channel



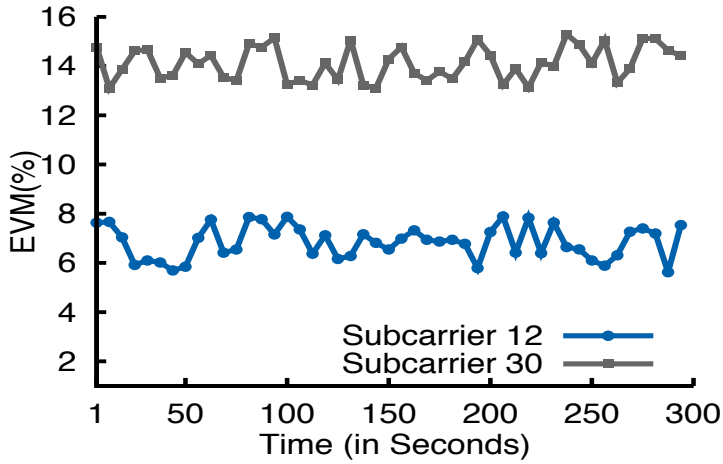
(b) EVM variation over time for two sample subcarriers

Figure 2.3: Stability of EVM in a static environment

Stability of EVM: Frequency diversity is a known property of wireless channels. However, if the EVM of a subcarrier changes too quickly (say every few milliseconds), it will be hard to track it without excessive overhead. Exploiting frequency diversity in rate and power adaptation requires the performance of the subcarriers to change slowly in comparison with the adaptation timescale. In order to measure the stability of EVMs under static and slow changing environments, we perform a set of experiments where we placed 6 warp boards at different locations to create links of varying quality. We fix one of these boards as transmitter and remaining nodes act as receivers. To create a slow changing



(a) EVM values for different subcarriers for a slow changing channel



(b) EVM variation over time for two sample subcarriers

Figure 2.4: Stability of EVM in a slow changing environment

environment we slowly move the antenna around the node. Each run lasts for 5 minutes, and is repeated 5 times. The receivers continuously measure the EVM per subcarrier and report the values as a function of time. The reported results are the averages of 5 runs. Figures 2.3(a) and 2.4(a) show the per subcarrier EVM for both static and slow changing channels. We see that EVM values differ significantly across subcarriers for both cases. The differences in EVM values can be as high as 6% to 8% in the case of the slow changing channel. The static channel is less diverse but still the subcarrier EVMs can vary by as much as 3-4%. Figures 2.3(b) and 2.4(b) show how the EVM in a representative subcarrier

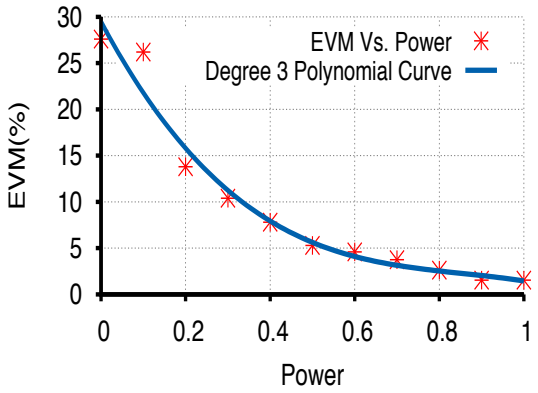


Figure 2.5: Average change in EVM(%) with change in sub-carrier Power from 0 to 1 for QPSK.

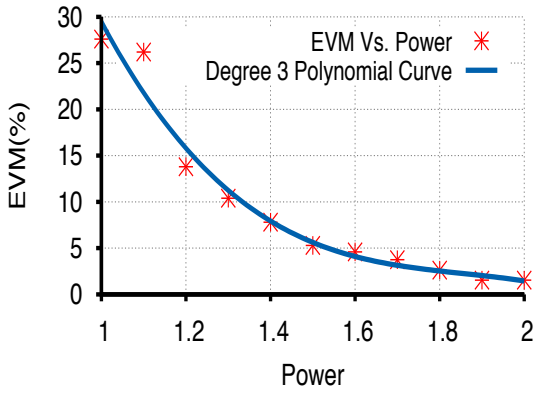


Figure 2.6: Average change in EVM(%) with change in sub-carrier Power from 1 to 2 for QPSK.

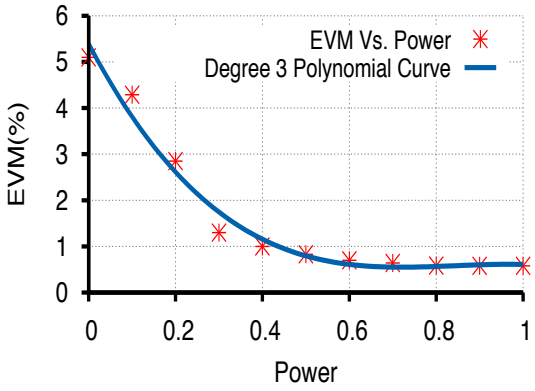


Figure 2.7: Average change in EVM(%) with change in sub-carrier Power from 0 to 1 for 16QAM.

varies over time. As can be seen, the EVMs in static environment largely vary within only a narrow interval over a period of few tens of seconds. In slow changing environment, the subcarrier EVM values changes more rapidly but still stay stable over multiple seconds. Furthermore, note that the magnitude of EVM variations over time is (Figure 2.3(b) and 2.4(b)) much lower than the variations in magnitude of subcarrier EVMs (Figure 2.3(a) and 2.4(a)). This is important for the calibration process to be effective. These results show that a rate and power adaptation scheme like JPRA can successfully harness the frequency diversity in both static and slowly changing scenarios.

Mapping rate to the EVM threshold: In order for a symbol on a sub-carrier to be demodulated with high probability, its EVM at the receiver needs to be below a specific threshold. This threshold is different for different modulation schemes. Similar thresholds has been determined in terms of SNR [4], but cannot be directly used with JPRA. The mapping between the EVM threshold and rate are calculated empirically via measurements, by varying the transmission power and the locations of the sender-receiver pair. The maximum EVM value that yields a target PDR (90% in our case) provides us with the required information. We observe that this mapping does not change from one setting to another. However, we recognize that it may vary for different hardware platforms. Thus, although we use one time pre-deployment measurements for calibration, in practice, they may have to be periodically repeated. The mappings that we obtain are presented in Table 2.1 and used to select the transmission bit rates with JPRA.

Estimating power adjustments for specific EVM changes: To perform joint rate and power adaptation we need to estimate the magnitude of increase/decrease in the EVMs of the received sub-carriers, for given changes in the transmit power. This mapping will facilitate the power redistribution across the available sub-carriers. In the current implementation of OFDM on WARP Boards, the total transmission power (called the power budget) is distributed equally among the available sub-carriers. The radio board applies power to the analog baseband/RF waveforms. In other words, the time domain signal after the IFFT stage is processed for power allocation. Since, in the time domain the individual sub-carriers are not explicitly visible, the power is uniformly distributed across all frequencies used. To achieve a non-uniform power allocation across the sub-carriers, we process each sub-carrier prior to applying the IFFT. The coefficient used to establish the power level of a sub-carrier does not correspond to actual transmission value but is a scaling factor. It takes real values between 0 and 2, with 1 representing the default power (uniform distribution of the power budget across sub-carriers), 0 mapping to zero transmission power and 2 to the maximum transmit power allowed (double the default transmission power for that sub-carrier).

In order to determine the scaling factor described above, we perform another set of calibration measurements. We use a pair of WARP board transceivers and we vary the transmission power of a sub-carrier by scaling its transmission power as explained above, with a step size of 0.1. At each step the transmitter sends back to back packets for 3 minutes; we perform 10 trials for each step. Figures 2.5 and 2.6, show the average EVM(%) change

when QPSK is used when decreasing and increasing the transmission power of a sub-carrier, respectively. Analyzing the obtained data we find that a third-degree polynomial can fit the data fairly accurately (the corresponding R^2 value[1] is 96%). Consequently, this polynomial can be used to predict the change in EVM(%) with respect to sub-carrier power. We also looked at different modulations and we observe that qualitatively the results do not change with the change in modulation. To illustrate, Figure 2.7, depicts the results obtained when 16-QAM is used. A polynomial of degree 3 can again be used to fit the data (R^2 value is 95%). As one might expect though, the coefficients of the polynomial are different as compared to the QPSK case since the EVM(%) range is different. Finally, we repeated the above experiments using different sub-carriers obtaining similar results. In particular, the fitting polynomial coefficients for different sub-carriers for the same modulation are very similar. The above measurements are obtained for static links and the results do not significantly change for small variations in the transceivers' locations. However, for large deviations from the initial measurements positions the results will not hold; in particular, the polynomial coefficients will be different. Therefore, we need to perform these calibrating measurements when a new link appears or when the topology changes significantly. This limits the applicability of the current version of JPRA to static or slowly changing topologies. Extending JPRA to more dynamic settings is deferred to the future.

Note that in JPRA, SNR can be plugged in with very minimal changes. It only requires changing of EVM based rate table to a SNR based rate table and replacing the EVM with SNR in the calibration phase where changes in SNR with changes in transmission powers are recorded.

Calibration overhead of JPRA: Mapping EVM thresholds to bit rates is a one time process and is only performed when system is initiated. Many hardware manufacturers already perform this calibration and can provide it as part of hardware specification sheets [21]. The main contributing factor towards calibration overhead is the estimation of power adjustments for specific EVM changes. Power adjustments for specific EVM changes are mapped to a third degree polynomial equation for each modulation as described above. This step is executed each time a new link appears or in a periodic fashion when the channel is changing rapidly. For each calibration, we exchange $M \times 10$ packets between the transmitter and receiver, where M is the number of available modulation schemes;

[1] R^2 value[20] shows the goodness of fit of a model.

Table 2.1: EVM to bit-rate(modulation + coding) mapping

Maximum EVM(%)	Modulation	Coding
18.0	BPSK	1/2
10.2	BPSK	3/4
6.6	QPSK	1/2
4.0	QPSK	3/4
1.67	16-QAM	1/2
1.26	16-QAM	3/4
1.1	64-QAM	1/2

Specifically with each modulation scheme 10 packets are exchanged. Since we vary the transmission power from 1 to 0 (1 being the default) or 1 to 2, with a step size of 0.1, we need 1 packet for estimating EVM at each power level. As mentioned earlier once estimated, this mapping does not change significantly from one subcarrier to another; thus, the same polynomial equation is used for estimating the EVM at all subcarriers, for a particular modulation. The total time required to execute this step is of the order of milliseconds. For example, in the case of 802.11g, there are 4 available modulations. To calculate all the third degree polynomial equations to predict the change in EVM with change in power, we need to transmit 40 packets. In 802.11g the average packet transmission time is 428μs [22] (including the acknowledgment). In this case the total time required to send 40 packets is $\approx$ 22 milliseconds. We believe that an overhead of few tens of milliseconds is acceptable when subcarrier EVM values stay stable for few tens of seconds, as shown in figure 2.3(a).

2.4 Power/Rate Allocation with JPRA

In this section we describe the three versions of JPRA in detail. In a nutshell, for all schemes, each receiver upon packet reception calculates the per sub-carrier EVM. Using this information JPRA redistributes the power among the sub-carriers depending on the objective (i.e., sub-carriers use the same maximum possible common rate with JPRA-Basic and JPRA-Intermediate *or* the the packet air time is minimized while allowing sub-carriers to be modulated at different rates with JPRA-Adaptive). Figure 2.8 depicts the high level functionalities of JPRA.

Upon executing the appropriate JPRA version, the receiver obtains a tuple for each sub-carrier specifying (i) the magnitude of increase or decrease in transmission power and (ii) the corresponding bit rate to use; this is represented as $\langle carrier\#, TXpower, Rate \rangle$. This is transmitted to the sender in the ACK (or NACK). The sender can then adjust its sub-carrier's transmission powers and bit-rates for the next packet. Note that in the case of JPRA-Basic and JPRA-Intermediate, all tuples will have the same rate information. If the channel conditions remain constant for a train of packets, the receiver may omit sending this information with every packet and the sender will use the latest sub-carrier settings for the next packet.

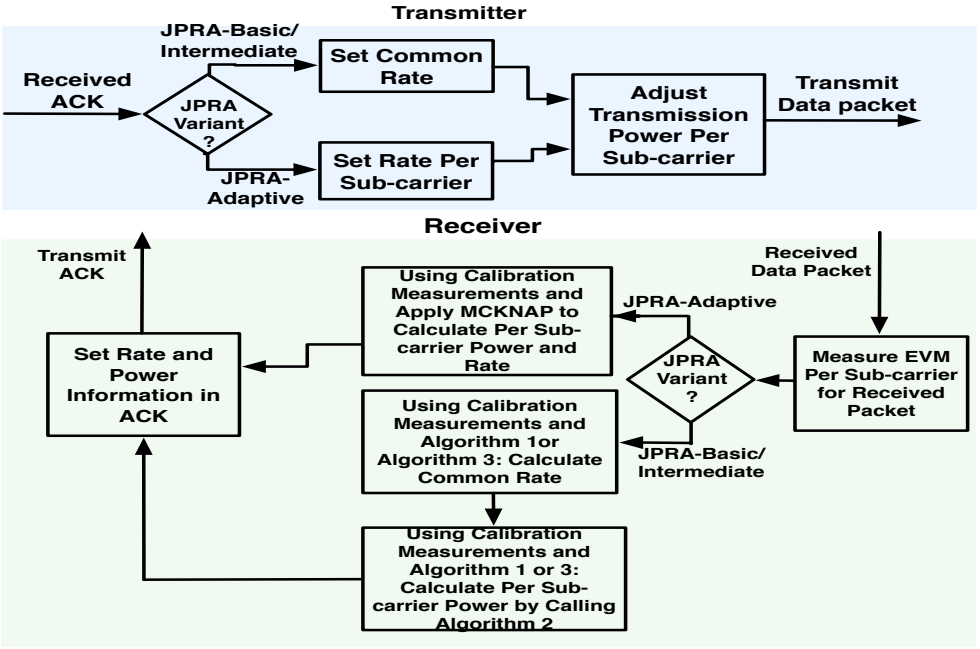


Figure 2.8: Power and Rate allocation with JPRA

2.4.1 Achieving a common bit-rate with JPRA-Basic

Due to frequency selectivity, for a specified bit-rate, some of the sub-carriers' EVMs will be lower than the minimum tolerable EVM for that rate (class 1 sub-carriers); on the other hand, some others will have EVMs higher than the threshold (class 2 sub-carriers). In conventional OFDMA systems, the modulation used is conservative. In particular, all sub-carriers are modulated with the minimum supported rate from among them. With

JPRA we make a case for redistributing the *extra* power from class 1 sub-carriers to class 2 sub-carriers. This in turn will reduce the EVM of the latter and can potentially allow the use a higher bit-rate modulation on all the sub-carriers.

Upon receiving a packet, the receiver iterates over the list of available rates that are greater than the currently supported lowest rate, to identify the maximum bit-rate that all sub-carriers can satisfy after power redistribution. In each iteration, it has to solve the problem of per sub-carrier power redistribution for the specific rate considered. One can map this problem on to a form of the well-known Knapsack problem (which we describe later) and based on this, we provide a solution to our problem.

In every iteration, the receiver first subtracts the EVM of every sub-carrier(EVM_i) from the EVM threshold(EVM_{th}^j) of the rate under consideration (obtained from the calibration phase). For some sub-carriers this value will be negative, while for others it will be positive. If there are no negative values (the EVM of every carrier is lower than the rate's EVM threshold), all sub-carriers can be modulated using this rate. If not, it maps the EVM difference to the corresponding sub-carrier transmission power increase required in order to meet the threshold (negative weights), or the excess transmission power present in sub-carriers whose EVM adheres to the required threshold (positive weights). Again, this mapping is done based on the measurements from the calibration phase. The positive weights are summed to get the total extra energy available (in Knapsack problem terminology, the total weight that can be carried by the knapsack). Each of the absolute values of the negative weights is considered as the weight of the corresponding sub-carrier (item). Our objective is to select as many items as possible with the given total weight. Ideally, we want all items to be included in our Knapsack. If this can be achieved, all sub-carriers can be modulated with the rate considered and we further examine the next available higher rate. If this cannot happen, we terminate our iterations; as shown later it is guaranteed that no other higher rate can be supported.

The problem of finding the maximum number of sub-carriers that can satisfy the given EVM threshold requirement (for the considered rate j) at each iteration, can be formalized by the following optimization problem. Let us assume that n sub-carriers are above the EVM threshold for rate j and have negative weights (these are the items to choose from). We introduce a slack variable for each such item i, x_i^j, which is 1 if the sub-carrier is chosen for placement in the knapsack (assignment of power for rate j), and 0 if not. Then we need to solve the following problem:

$$maximize \sum_{i=1}^{n} x_i^j \tag{2.1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_i^j \cdot w_i^j <= W^j \tag{2.2}$$

where $x_i^j \in \{0,1\}$ for each $i \in \{1,2,...,n\}$ and, W^j is the total excess power available.

However, we are not interested in finding the maximum number of sub-carriers that can be assigned additional powers to meet the rate requirement (items that can be put in the knapsack); our goal is to only check if all such sub-carrier requirements can be accommodated. Specifically, we need to check if the value of the objective function is equal to n. If this is the case, rate j can be supported; if not, rates greater than or equal to j cannot be supported. Thus, the receiver only needs to check if upon setting $x_i = 1, \forall i \in \{1,2,...,n\}$, the constraint (2.2) is satisfied. If so, it moves to the next higher rate; if not, it chooses the immediately lower rate and is done with rate selection.

To see this, note that in each iteration we iterate over the rates in the decreasing order of their EVM thresholds. For rate j , we can rewrite constraint (2.2) in terms of EVM values as follows:

$$(EVM_{th}^j - EVM_1) + (EVM_{th}^j - EVM_2) + (EVM_{th}^j - EVM_3)$$

$$+..... + (EVM_{th}^j - EVM_n) \geq 0$$

$$\Rightarrow n \times EVM_{th}^j - (EVM_1 + EVM_2 + ... + EVM_n) \geq 0 \tag{2.3}$$

where, EVM_{th}^j is the EVM threshold for rate j and EVM_i is the EVM of sub-carrier i(from among "n" sub-carriers). Similarly for rate $j+1$, constraint (2) can be rewritten as:

$$n \times EVM_{th}^{j+1} - (EVM_1 + EVM_2 + ... + EVM_n) \geq 0$$

To show that we cannot support rates higher than or with the rate at which we terminate our iterations, let's assume that constraint (2) fails for rate j. This happens when the

inequality in (2.3) does not hold true. This implies that,

$$n \times EVM_{th}^{j} - (EVM_1 + EVM_2 + ... + EVM_n) < 0$$

$$\Rightarrow n \times EVM_{th}^{j} < (EVM_1 + EVM_2 + ... + EVM_n) \tag{2.4}$$

Since $n \times EVM_{th}^{j+1} < n \times EVM_{th}^{j}$ (the EVM for a higher rate is smaller than that of a lower rate) the following inequality must hold true:

$$n \times EVM_{th}^{j+1} < (EVM_1 + EVM_2 + ... + EVM_n)$$

This proves that if constraint (2) fails for rate j, then it will also fail for rate $j+1$. By induction, it can be easily shown that constraint (2) will fail for all possible rates higher than j.

Algorithm 1 summarizes the steps of our common rate selection algorithm.

Once it has decided on the maximum common rate that can be supported, the receiver performs the power redistribution. It transfers power from the *fading-immune* sub-carriers, to the affected ones. Algorithm 2 presents the steps taken. Recall that with Algorithm 1, we divide the sub-carriers in two classes (lists). The first list contains the sub-carriers with the excess power, while second list contains sub-carriers that require more power for meeting the EVM threshold for the chosen rate. These lists are returned by Algorithm 1 and used by Algorithm 2 to perform power redistribution. Upon executing this algorithm, the receiver first subtracts the excess power from the sub-carriers in List-1. It then adds the extra power as required to each sub-carrier in List-2 (in decreasing order of the required power). If total extra power is more than the required power then left over power is distributed evenly among all the sub-carriers.

2.4.2 Achieving a common rate with JPRA-Intermediate

Selecting a common rate as above for all the sub-carriers may not be optimal since there may exist carriers (in deep fades) that may be unable to carry data at acceptable rates. With JPRA-Intermediate our objective is to find a common rate on a *subset* of sub-carriers such that the aggregate rate is maximized. This constitutes removing sub-carriers which

Algorithm 1 Rate selection using JPRA-Basic.

Input: Sub-carrier EVM_i, Rate $j \in S$, Rate threshold $EVMTH_j$, EVM to Power Mapping.
Output: Rate j, List $L_{j,1}$,List $L_{j,2}$
for each Rate j **do**

  for each Sub-carrier i **do**
    $diff_i = EVMTH_j$ - EVM_i
    Map $diff_i$ to power p_i

    if $p_i > 0$ **then**
      $W =+ p_i$
      Put p_i in List $L_{j,1}$

    else
      $w =+ \mod(p_i)$
      Put $\mod(p_i)$ in List $L_{j,2}$
    end if
  end for
  if $w \leq W$ **then**
    return Rate j , $L_{j,1}$,$L_{j,2}$
  else
    return Rate $j-1$,$L_{j-1,1}$,$L_{j-1,2}$
  end if
end for

Algorithm 2 Power Redistribution after selecting rate.

Input: Initial sub-carrier Power $p_i^{'}$,List $L_{j,1}$,List $L_{j,2}$.
Output: Adjusted sub-carrier Power $p_i^{''}$.
for each sub-carrier in List $L_{j,1}$ **do**
    x=+ p_i
    $p_i^{''} = p_i^{'}$ - p_i
end for
Sort List $L_{j,2}$ in decreasing order

for each Sub-carrier in List $L_{j,2}$ **do**
    $p_i^{''} = p_i^{'} + p_i$
    y=+ p_i
end for

if $x > y$ **then**
  $l = (x - y)/ n$
  where n is total number of sub-carriers

  for each sub-carrier in List $L_{j,1}$ and $L_{j,2}$ **do**
    Update $p_i^{''}$ =+ l
    return $p_i^{''}$
  end for
end if

require high powers per bit delivered, and redistributing their powers to other sub-carriers; at the same time, it is possible that powers from some of the best sub-carriers are also carried over to other sub-carriers towards achieving the above objective. Note that there is an inherent trade-off here; while eliminating poor sub-carriers can increase the common rate that can be supported on the other sub-carriers, it also decreases the number of available sub-carriers and thus, contributes to a lowering of the capacity. JPRA-Intermediate tries to find the best point of operation while accounting for this trade-off.

Let us assume, for simplicity, that all sub-carriers are considered for setting the common rate. When a packet is received, the receiver iterates over the list of available rates that are greater than the currently supported lowest rate, to identify the maximum bit-rate that all the sub-carriers can satisfy after power redistribution. The problem of identifying the maximum bit-rate that all the sub-carriers can satisfy after power redistribution is already solved in JPRA-Basic using a Knapsack formulation.

To maximize the aggregate rate after identifying the maximum common rate, JPRA-Intermediate removes subcarrier which requires the highest power per bit delivered and adds its power to the excess power budget. We then, again perform power redistribution and calculate the common rate for remaining subcarriers. We remove subcarriers until the aggregate rate achieved is maximized.

Algorithm-3 is the main algorithm in JPRA-Intermediate. It iteratively invokes Algorithm 1, to identify a *set* of sub-carriers that support the maximum number of bits to be transmitted per symbol, using a common rate. To begin with it considers all sub-carriers (as described above), and calls Algorithm 1, to find the maximum common rate that can be supported in this case (say $R(n)$, where n is the number of sub-carriers). The total number of transmitted bits per symbol is then $n \times R(n)$ (after performing power redistribution). It then removes the sub-carrier that requires the highest power per bit delivered (*poorest* sub-carrier) and adds its power to the excess power budget (as discussed); the common aggregate bit-rate with the remaining $(n-1)$ sub-carriers is now calculated by invoking Algorithm 1 to be $(n-1) \times R(n-1)$, where $R(n-1)$ is the maximum common rate that is now supported. If $(n-1) \times R(n-1) < n \times R(n)$, it is easy to see that the algorithm has converged, since no further optimization is possible. Note that $R(n)$ and $R(n-1)$ are calculated after performing power redistribution using Algorithm 2; thus these are only available as this algorithm is executed iteratively. If not, from among the remaining sub-carriers, the poorest sub-carrier is removed and the process is repeated. The iterations continue until a point is reached, say

with the number of sub-carriers m, where $m \times R(m) < (m+1) \times R(m+1)$. This implies that the optimal set of sub-carriers are the $(m+1)$ sub-carriers in the prior to last iteration and, $R(m+1)$ is the maximum common rate to be used.

Algorithm 3 Common rate selection and Power redistribution using JPRA-Intermediate.

1: **Input:** Total number of sub-carriers n, Per sub-carrier EVM_i, Rate $j \in S \equiv \{6, 12...\}$, Rate threshold $EVMTH_j$, EVM to Power Mapping, Initial Per sub-carrier Power $p_i^{'}$.
2: **Output:** Rate $R(m)$, Per sub-carrier Power $p_k^{''}(m)$.
3: $R(n) =$ **Algorithm-1**(EVM_i, Rate $j \in S$, $EVMTH_j$, EVM to Power Mapping)
4: Aggregate Rate $AG_n = R(n) \times n$
5: *Remove Sub-carrier which requires highest amount of power per bit delivered and add it's power to total excess power W*
6: $m \leftarrow n - 1$
7: **while** $m > 1$ **do**
8: $R(m) =$ **Algorithm-1**(EVM_i, Rate $j \in S$, $EVMTH_j$, EVM to Power Mapping)
9: Calculate $AG_m = R(m) \times m$
10:
11: **if** $AG_m < AG_{m+1}$ **then**
12: $p_k^{''}(m+1) =$ **Algorithm-3**($p_i^{'}$, List $L_{j,1}$, List $L_{j,2}$)
13: **return** $R(m+1)$ and $p_k^{''}(m+1)$
14: break
15: **else**
16: Update $S = S$ - $R(m)$
17: *Remove Sub-carrier which requires highest amount of power per bit delivered add it's power to total excess power W*
18: Update $m \leftarrow m - 1$
19: **end if**
20: **end while**
21: $p_k^{''}(m) =$ `Algorithm-2(`$p_i^{'}$`,List` $L_{j,1}$`,List` $L_{j,2}$`)`
22: **return** $R(m)$ and $p_k^{''}(m)$

Once the maximum common rate and the sub-carrier set to be used have been determined as above, the receiver performs the power redistribution. It transfers power from the *fading-immune* sub-carriers, to the affected ones. Algorithm 2 presents the steps taken. Upon executing this algorithm, the receiver first subtracts the excess power from the sub-carriers in List-1. It then adds the extra power as required to each sub-carrier in

List-2 (in decreasing order of the required power). If the total extra power is more than the required power, then the left over power is distributed evenly among all the sub-carriers.

Computational Complexity: The run time for JPRA-Intermediate is $O(n^2 \cdot l)$, where n is the number of sub-carriers and l is the number of available data rates. In brief, JPRA-Intermediate (i.e., Algorithm 1) iterates over the n sub-carriers and executes algorithm 2 in every iteration. The latter includes a nested loop which executes over all the sub-carriers and over all the l transmission rates, thus, requiring $n \cdot l$ running time. Since, algorithm 3 (whose complexity is $O(n)$), is executed outside the loop in Algorithm 1, the time complexity of JPRA-Intermediate is $O(n^2 \cdot l)$.

2.4.3 Minimizing Transmission Air Time with JPRA-Adaptive

The transmission time of a packet depends on the modulation and coding rate used, Higher modulation schemes lead to shorter packet times. In turn, shorter packet transmission times lead to throughput improvements. Ideally, in order to minimize the packet air time one would select the highest modulation on all the sub-carriers. However, due to frequency selectivity in fading all sub-carriers might be unable to support this high rate.

Adaptive modulation and coding schemes have been proposed to cope with frequency selectivity (e.g., [4]). With such schemes, the bit-rate for each sub-carrier is selected based on its CSI (Channel State Information). However, power redistribution is not performed. Power redistribution helps in maximizing the total number of bits transmitted by the sub-carriers in a symbol duration. Let us consider a toy example with two sub-carriers A and B, initially with default power allocations. A can only support BPSK while B can support QPSK. In total A and B can carry 3 bits since BPSK and QPSK can modulate 1 and 2 bits, respectively. If one completely shuts down A and transfers its power to B, B may be able to use 16QAM with the extra power. Thus, the total number of bits transmitted in a symbol duration is now 4 instead of 3 (a 16QAM symbol maps on to 4 bits). Thus, we gain 1 extra bit per symbol by doing power redistribution. With a large number of sub-carriers, where each sub-carrier has a different EVM and requires a different level of extra energy to support a higher modulation scheme, the gains can be (and actually are) more significant.

Formalizing the problem: The problem of appropriately allocating powers (and thus, transmission bit-rates) to sub-carriers in order to minimize the packet air-time can be formally defined as follows: Lets assume that there are N sub-carriers x_1,x_2 ...x_N, and R+1 (one for each rate level and a null rate level for no assignment) bit-rate levels for each sub-carrier. On each sub-carrier only one specific bit-rate out of the R+1 available bit-rates can be used; for each bit-rate $j \in x_i$ there is a corresponding *profit* $r_{i,j}$, which is the bit-rate itself and a weight that corresponds to the power p_{ij}, needed to achieve this bit-rate. We estimate this power using our measurements from the calibration phase. Maximizing the profit, essentially translates to using the highest bit rates on the sub-carriers; as one can easily see, this maximizes the number of bits transmitted per symbol and as a consequence, minimizes the transmission air time. However, the maximization of the profit as above is constrained by the total power budget available (P).

The above problem can be mapped to the well known Multiple Choice Knapsack Problem (MCKP). In MCKP, we are given m classes N_1, N_2, . . . , N_m of items that are to be packed in a knapsack of capacity C. Each item $j \in N_i$ has a profit a_{ij} and a weight w_{ij} , and the problem is to choose at most one item from each class such that the profit sum is maximized without the weight sum exceeding C.

With our problem, a sub-carrier x_i corresponds to a class, that consists of items that correspond to the $R+1$ bit rates. The profit is the bit-rate value r_{ij} and weight is the required power p_{ij}. The total capacity of the knapsack is the total transmission power P, available to the transmitter.

Using this mapping, our problem can be formulated as the following MCKP:

$$\text{Maximize} \quad \sum_{i=1}^{N} \sum_{j \in x_i} r_{ij} k_{ij} \tag{2.5}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} \sum_{j \in x_i} p_{ij} k_{ij} \leq P \tag{2.6}$$

$$\sum_{j \in x_i} k_{ij} = 1, \quad i = 1, .., N$$

$$k_{ij} \in \{0, 1\}, \quad i = 1, ..., N, \quad j \in x_i$$

Here k_{ij} is a slack variable for each r_{ij}; if a particular r_{ij} is selected to be included in the knapsack it is 1 and otherwise 0. The total number of items considered is $n = \sum_{i=1}^{N} R + 1 = N \cdot (R + 1)$.

Solving the problem: The MCKP problem is NP-hard [23] as it contains the knapsack problem as a special case [23]. However, it has been shown that it can be solved in pseudo-polynomial time [24]. Pisinger [25] proposes an algorithm (MCKNAP) which has been shown to outperform other algorithms for solving the MCKP problem. The algorithm first extends the partitioning algorithm of Balas and Zemel [26] for the original knapsack problem, to the case of MCKP. The partitioning algorithm identifies a set of classes (called the *core*) that include the optimal solution with high probability; this can be thought of as a reduction phase. A dynamic programming algorithm is then applied on the core in order to identify the optimal solution in pseudo linear time. This can be thought as an expanding phase, since it might add classes that do not belong to the core. The resulting solution defines the classes and the corresponding elements that are included in the knapsack (i.e., the elements with $k_{ij} = 1$). Based on the assigned rates, the total transmit power is allocated among the carriers.

Computational Complexity: The computational complexity of MCKNAP is $O(n + c \sum_{x_i \in c} r_i)$ for a minimal core c; thus, the algorithm executes in linear time for small cores and pseudo polynomial time for large cores. The proofs are available in Section 7 of [25].

2.5 Performance Evaluation

In this section, we examine the performance of JPRA via extensive real testbed as well as simulation experiments.

2.5.1 Testbed Setup and Implementation

Our testbed utilizes the Wireless Open-Access Research Platform (WARP) developed at Rice University. The WARP platform consists of three main components; (a) A Xilinx Virtex-II Pro FPGA, (b) A 2.4/5GHz Radio Board, which supports wideband applications such as OFDM, and (c) A 10/100 Ethernet port, which serves as the interface between the board and the wired Internet. The Xilinx module implements the MAC and

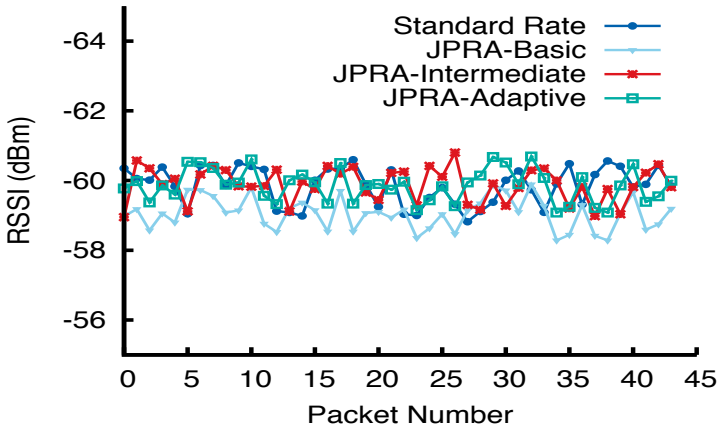


Figure 2.9: RSSI values per packet

PHY layer protocols. MAC protocols can be implemented in C, while the PHY layer protocols are implemented within the FPGA fabric using MATLAB Simulink. The current physical layer design uses an OFDM implementation that is loosely based on the PHY layer of the 802.11a standard. The WARPMAC/WARPPHY modules provide basic building blocks towards implementing more advanced MAC/PHY protocols. The basic configuration of WARP's OFDM design is given in Table 2.2.

We conduct our experiments on a six node WARP indoor testbed. We randomly select three sender/receiver pairs, and send saturated traffic with packets of 1472 bytes. We consider cases where the connections interfere with each other and cases where they don't. Each experiment lasts for 6 minutes and the reported results are the averages of 5 runs unless otherwise stated. Each node is connected to a laptop through an Ethernet switch, which acts as a controller. The controller is also used for analyzing and collecting traces during the experiment.

Implementation: We implement JPRA on top of WARP's standard OFDM design, which runs on the board's FPGA in real time. We run JPRA at the MAC layer to calculate the rates to be used and the power redistribution. The receiver communicates this information to the transmitter through ACKs or NACKs (as mentioned in Section 2.4). We modify the transmitter to perform power redistribution and rate adaptation per sub-carrier, upon obtaining this information from the receiver. We also implement (for comparison), a standard EVM based rate adaptation scheme, which selects a single rate

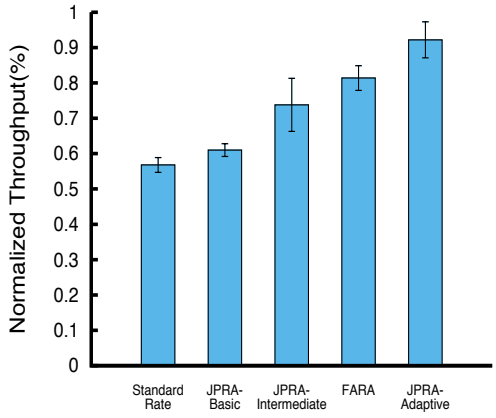


Figure 2.10: Normalized network throughput with 10MHz channel width.

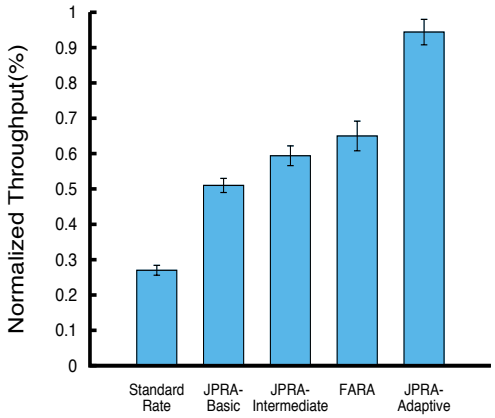


Figure 2.11: Normalized network throughput with 20MHz channel width.

Table 2.2: MAC/PHY parameters for JPRA implementation

Carrier Frequency	2427MHz
RF Bandwidth	10MHz and 20MHz
Number Of Sub-carriers	64
Modulation Schemes	BPSK, QPSK, 16QAM
Payload Length	1470 bytes
Tx Power	19dBm
MAC Protocol	CSMA/CA

for all the sub-carriers. This scheme referred to as *Standard rate*, selects the appropriate rate for each packet by comparing the average packet EVM with the threshold values in Table 2.1 . Standard rate adaptation does not perform any kind of power adaptation. Finally, for comparison purposes as well, we also implement FARA as described in [4]. In brief, FARA selects a rate based on the EVM or SNR threshold for each sub-carrier. In our implementation, FARA examines the individual sub-carrier EVM and selects the rate which can be supported. If a carrier cannot satisfy even the basic (minimum) rate it is left unused (no symbols are mapped on to that sub-carrier).

2.5.2 Experimental Results

Effects of power redistribution on RSSI value: We first want to ensure that a power redistribution with JPRA does not affect key network functionalities like carrier

sensing. Typically the RSSI value is used in most of the wireless systems to determine the amount of energy on the channel. It is used as an indicator of whether the channel is idle or not. In WiFi, packet detection is based on RSSI values as well. Thus, we measure the RSSI values per packet with and without JPRA. Figure 2.9 depicts a part of a representative trace that we collected with Standard Rate adaptation, JPRA-Basic, JPRA-Intermediate and JPRA-Adaptive. As one can see the RSSI values are similar in all scenarios, thus leading us to believe that JPRA will leave key higher functions such as carrier sensing or packet detection unaffected. The reason behind this observation can be explained as follows. The RSSI value is essentially the average energy present on the antenna during the reception of the preamble. Since JPRA is not applied on the preamble, we do not observe any significant RSSI variations.

Network throughput: Next, we examine the total network throughput. Figure 2.10 compares the throughput with JPRA with that using Standard Rate Adaptation and FARA. The throughputs are normalized by the highest throughput value among all observations (highest throughput = 1).This provides an immediate assessment of the relative performance with the schemes that we compare. The channel bandwidth in this experiment was 10MHz. We observe that JPRA-Intermediate outperforms standard rate adaptation by up to 28% and it outperforms JPRA-Basic by 8% . As one may expect, FARA outperforms both JPRA-Basic and JPRA-Intermediate, since it opportunistically employes higher rates on a subset of sub-carriers. However, the gains are marginal ($< 8\%$); further, JPRA-Basic and JPRA-Intermediate only require feedback corresponding to the common rate to be used unlike in FARA where the rate for each sub-carrier has to be specified. JPRA-Adaptive outperforms both standard rate and FARA by, above 42% and 20% respectively. The main source of gain for JPRA-Adaptive in comparison to FARA and Standard rate adaptation, is the per sub-carrier power redistribution. The power redistribution allows the transmission of a higher number of bits per symbol as discussed earlier. Note here that we show the 95% confidence intervals for all results.

Wider channels exhibit a higher degree or frequency selectivity. We also perform the same set of experiments with a 20 MHz channel; this channel width is typical in WiFi systems. With the 20 MHz channel the observed normalized throughput with JPRA-Intermediate is 43% higher as compared with that of the Standard Rate adaptation (Figure 2.11). Using single bitrate on all the subcarriers reduces the aggregate througput of JPRA-Basic by 13% in comparison to JPRA-Intermediate. FARA still outperforms JPRA-

Intermediate by about 7%. With JPRA-Adaptive we observed higher gains with the 20MHz channel width. In particular, our results (Figure 2.11) indicate that JPRA-Adaptive delivers upto 75% more network throughput than Standard Rate Adaptation. It also outperforms FARA by 35%. Thus, *per sub carrier power adaptation enhances network performance in comparison to schemes without power adaptation*. With the increase in channel width, gains also increase. This becomes especially noteworthy, given the increasing number of technologies that use wide channels (e.g., 802.11n with 40 MHz channels and white space networking with channels of at least 100MHz).

Stability of rate: One of the main reasons why JPRA-Basic could still be attractive is the stability of rate that it can offer. In addition JPRA-Basic and JPRA-Intermediate requires a much simpler transceiver design since they select a single rate for a given transmission which can be supported on all or a chosen subset of sub-carriers; therefore a single rate or modulation is applied to all or a subset of the sub-carriers like in standard OFDMA. In contrast FARA and JPRA-Adaptive, in the worst case, use l (total number of rates) different modulations on the different sub-carriers for a given transmission, which leads to a more complex transceiver design. In particular the transceiver has to implement additional bookkeeping to track the rates on the different sub-carriers with these other schemes.

To reiterate, a single rate is chosen, which all sub-carriers can support. Since some carriers might experience fading while others may not, on average with power redistribution, the maximum common rate that can be supported is not expected to change significantly from one transmission to the next (even though the sub-carriers that experience fading might be different). In order to examine the performance of JPRA-Basic with respect to rate stability, we created 10 diverse links in our lab by changing the positions of the sender and the receiver. The sender transmits 1472 byte size packets for 3 minutes and we monitor the throughput on the receive side. We also log the number of rate changes applied by the transmitter. The results are shown in Figure 2.12. We plot the average number of rate changes over 3 trials with JPRA-Basic, JPRA-Intermediate, Standard Rate, JPRA-MT and FARA. As one can see JPRA-Basic resulted in up to 30% fewer rate changes in comparison to Standard Rate Adaptation, it also outperforms the other schemes. JPRA-Adaptive performs upto 8% and 14% better than FARA and Standard Rate Adaptation. With FARA and JPRA-Adaptive, we consider a rate change to have occurred if the transmission bit-rate on any sub-carrier is changed between packet transmissions. We believe this is reasonable since the receiver has to now decode packets with the new rate on the specific sub-carrier.

Note here that FARA also performs better than standard rate because it shuts down sub-carriers which are in deep fades, these are the carriers where the fluctuations are most likely. *To summarize, power redistribution helps in stabilizing rate changes since it improves performance on sub-carriers experiencing selective fading (which results in increased rate changes).*

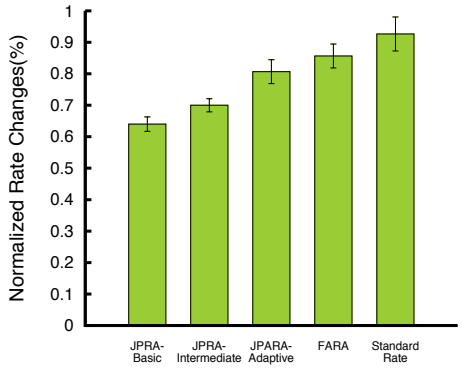


Figure 2.12: Normalized number of rate changes with 20 Mhz.

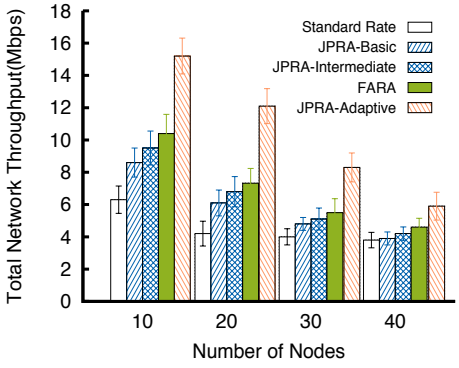


Figure 2.13: Network throughput with packet size 512 bytes.

Power savings: In our next experiments, we examine the possible power saving gains with JPRA-Basic. In particular, we seek to answer the following question "Can we use lower total transmission power in order to achieve a target PDR?". We set a target PDR of 90% and we vary the total transmission power of the sender. On an average over 5 trials we observe that with JPRA-Basic we need 4.5dBm less power compared to Standard Rate adaptation in order to achieve the preset target packet delivery ratio. These gains are important, since potentially JPRA-Basic can reduce interference; moreover, it can lower energy costs since it requires less transmission power to achieve a certain performance.

Overheads: A receiver needs to send feedback information to the transmitter to perform rate and power reallocation. WARP's CSMA/CA MAC layer uses standard 802.11 DATA and ACK packet formats, but allows us to piggyback rate and power information on the ACK packets (as with 802.11n ACK packets [27]). Bit-rate information is encoded using 6 bits (up to 64 values). Each sub-carrier power value is a signed real value between 0 and 2.0, which can be represented using 7 bits. When the channel remains stable for multiple packets we just send a normal ACK (without any additional information) which tells the receiver to use power and rate values from the previous instance. For 48 data sub-carriers to

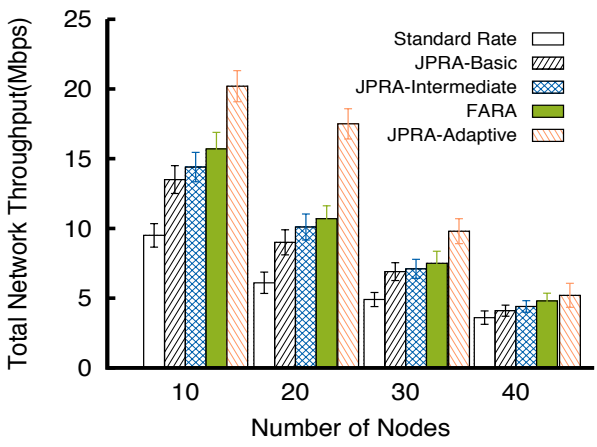


Figure 2.14: Network throughput with 1472 bytes packet size .

encode bit-rate and power information we require 48*6 bits and 48*7 bits, respectively. In a worst case scenario, JPRA-Adaptive incurs a 5.2% overhead and JPRA-Intermediate incurs a 2.5% overhead compared to Standard Rate adaptation for a payload size of 1472 bytes. The worst case overhead for JPRA-Basic is also 2.5% (as with JPRA-Intermediate). Note that the corresponding overhead increase with FARA is 2.4%. In our testbed the average overheads that we observed for JPRA-Adaptive, JPRA-Intermediate and JPRA-Basic were 4.1%, 1.40% and 1.1%, respectively. The average overhead of FARA was 1.82%, which is more than JPRA-Basic and JPRA-Intermediate since these are more stable and require less feedback. We wish to point out here that our throughput results, already account for this overhead i.e., the gains are in spite of the overhead .

2.5.3 Simulations

In order to evaluate our scheme in larger settings we perform simulations in NS-3. In order to have a realistic OFDM PHY layer model, we use a modified version of NS-3 which includes a detailed physical layer model called PhySim-Wifi[28]. PhySim-WiFi includes a physical layer implementation of the OFDM PHY specification for the 5 GHz band as well as a wireless channel emulation. Specifically in our simulations we use NS-3.9 with the PhySimWiFi 1.1 module. Key Physical and MAC layer parameters are given in Table 2.3 . We further modified PhySim-WiFi in order to implement JPRA-Basic, JPRA-Intermediate, JPRA-Adaptive and FARA. Standard rate adaptation is already implemented in NS-3 but we modified it to use EVM instead of SNR as rate adaptation metric. We implemented

Table 2.3: MAC/PHY parameters used in simulation

Carrier Frequency	5320MHz
RF Bandwidth	20MHz
Number Of Sub-carriers	64
Modulation Schemes	BPSK, QPSK, 16QAM, 64QAM
Payload Length	512/1472 bytes
Tx Power	20dBm
MAC Protocol	CSMA/CA

these schemes in a similar fashion as we implemented them on our testbed. We repeat the calibration phase in the simulator in order to calculate per sub-carrier EVM, EVM to sub-carrier Power mapping and EVM to rate thresholds. EVM to sub-carrier power mapping follows a similar third degree polynomial curve that we observed on our testbed.

To create frequency selectivity, the sub-carrier gains $|h_n^2(t)|$ are generated based on path loss and fading. The path loss model is characterized by a standard model $|h_{PL}|^2 = K * 1/d^{\alpha}$ parameterized by $K = 40.14dB$ and the path loss exponent $\alpha = 3.5$. The fading component is implemented based on [29] and parameterized with the values provided in [30] (Table 1 of the reference). The resulting fading gains feature correlation in time and in frequency. The environment is further characterized by an RMS delay spread of 25ns.

Simulation setup: Our simulation topology consists of 40 nodes, arranged in a 8×5 grid. The inter-node distance in the grid is 20m. We consider randomly selected transceiver pairs (also referred to as traffic pairs) to send data packets. We consider 4 sets of experiments while varying the number of active links. In particular, we experiment with 5, 10, 15 and 20 traffic pairs, active simultaneously. Since the topology is fixed, a higher number of traffic pairs implies higher traffic intensity. We use CBR (constant bit rate) to generate packets of size 512 bytes and 1472 bytes. Simulation time for each run is 120 seconds, and our results are the average of 10 of such simulation runs.

Simulation results: Figures 2.13 and 2.14 present the results for network throughput with different node densities with each scheme. With JPRA-Intermediate, the total network throughput is 71% higher as compared with Standard rate adaptation, when the node density is 20. The network throughput of JPRA-Basic lies between that with JPRA-Intermediate and Standard Rate Adaptation and it outperforms Standard Rate Adaptation by 59%.

JPRA-Adaptive improves the total network throughput by upto 185% in comparison to Standard rate and also registers a gain of 64% over FARA. Similar to our experimental results FARA outperformed JPRA-Intermediate by 9% in simulations; again, this is due to per sub-carrier rate adaptation which leads to higher throughput. As the number of competing links increase, the gain for each scheme increases up to a certain traffic intensity. After this *sweet point* the gains are diminished since interference and contention become the dominant impairments.

With increased interference we observe an increasing number of packet collisions. Interference reduces the opportunity to gain from power redistribution since it affects all the sub-carriers.

2.6 Conclusions

In this paper, we make a case of adaptively distributing uneven power levels to OFDMA sub-carriers to cope with frequency selectivity in fading. To validate our thesis that this will provide significant throughput benefits, we design JPRA, for jointly selecting the transmission power and bit-rate for the sub-carriers that compose a channel. We design three versions of JPRA: (a) JPRA-Basic, a simpler scheme, which while less effective in improving throughput, provides increased stability in the rate in use, simplicity in transceiver design and potential power savings (b) JPRA-Intermediate, which provides a balance between simple transceiver design and throughput gains and (c) JPRA-Adaptive which provides significant throughput gains albeit with lower rate stability. We implement all the variants on our WARP radio testbed and also perform simulations to showcase its benefits in larger settings. We show that JPRA-Adaptive can provide upto a 35% increase in throughput compared to the state of the art, FARA scheme that only performs per sub-carrier rate adaptation.

Chapter 3

TRINITY: Tailoring Wireless Transmission Strategies to User Profiles in Enterprise Wireless Networks

3.1 Introduction

Two trends are becoming evident in enterprise wireless networks. First, enterprises that used to be dominated by static users due to devices like laptops and notebooks are increasingly being populated with mobile users owing to the proliferation of smart phones and BYOD initiatives. Second, wireless transmission strategies have also improved significantly to cope with the increased demand for capacity. Specifically, the past decade has seen strategies go from conventional CSMA to more sophisticated MIMO-based strategies. While standards have come as far as multi-user MIMO (IEEE 802.11ac), research has advanced to the point of building network (distributed) MIMO systems [31, 32].

Network MIMO (netMIMO) represents the highest form of multiplexing, allowing multiple data streams to be transmitted concurrently from distributed transmitters to users by converting interference into a multiplexing gain via transmitter cooperation. However, its benefits are realizable only if the whole process of channel estimation of all users' channels, their feedback and use for netMIMO operation, is completed within the coherence time of

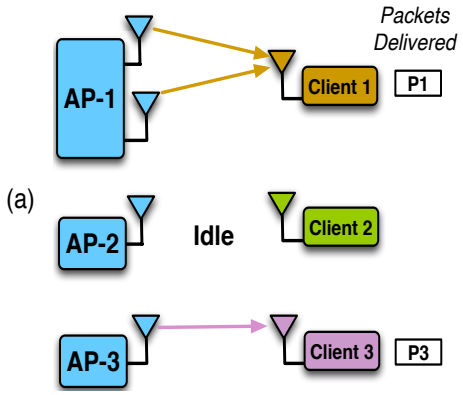


Figure 3.1: Reuse: (Two slot schedule: AP1,AP3 active simultaneously in slot 1, sending same data to one client in each slot.) AP2 in slot 2.)

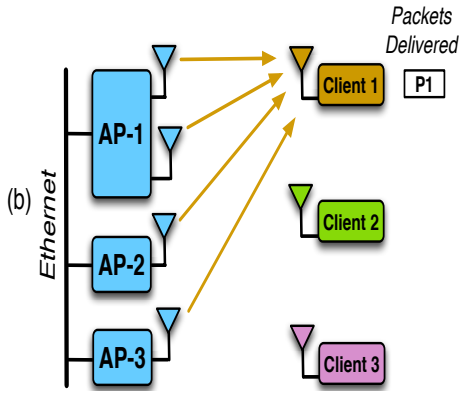


Figure 3.2: DAS: (3 slot schedule: 3 APs

all the users' channels. Indeed, applying netMIMO for mobile users or for whom the latter condition is not satisfied, can even hurt performance by as much as 70% for a 3 client system as shown later. Given the mix of users with diverse channel and mobility characteristics (referred to as *profiles*) in next generation enterprise networks, it is both important and timely to understand how transmission strategies must be tailored to user profiles so as to maximize system performance and improve user quality of experience.

A potential approach could be to differentiate between static and mobile users, apply netMIMO only for static users, while employing conventional CSMA approaches as in current enterprise networks for mobile users. However, such an approach has two drawbacks: (i) netMIMO could hurt performance even for static users if their channel fluctuations (e.g., environment changes) result in small coherence times; and (ii) while conventional CSMA would work for mobile users, it could be highly sub-optimal due to handover delays and the lack of transmitter cooperation gain.

We argue that users in an enterprise network can be profiled more generally into one of three distinct categories. First, they can be categorized into those with large and small channel coherence times. Further, among those with small coherence times, we can further classify them based on the contributing factor - user mobility or environment dynamics. Similarly, the gamut of wireless transmission strategies can be grouped into one of three distinct categories. At the top level, we have those that allow transmitter cooperation and those that do not. We refer to the latter as *Reuse* strategies, an example being the

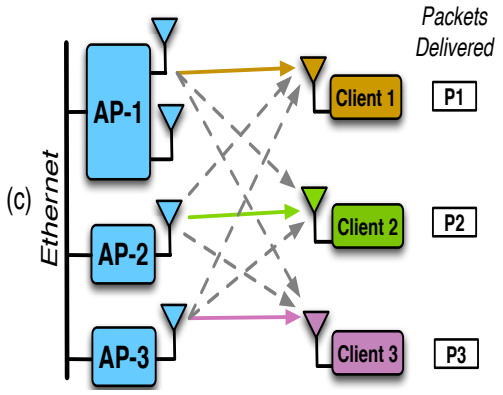


Figure 3.3: netMIMO: (3 APs sending 3 data streams to 3 clients in each slot.)

current enterprise CSMA scheme, where transmitters reuse the channel based on interference avoidance[1]. Schemes that enable transmitter cooperation can be further classified into two categories - *multiplexing*: those that exploit interference (through cooperation) and channel state information (CSI) to transmit multiple independent streams to users (e.g., netMIMO), and *diversity*: those that bypass interference by sending multiple versions of a stream through different transmitters without the need for CSI feedback to provide a diversity gain (e.g., space-time block codes, Distributed Antenna System, etc.)

With the help of implementation of the various strategies and real-world experiments with users of various profiles, we show that there is no single strategy that can cater to all users effectively and more importantly, not applying the right strategy for a user can even hurt performance. For example, for a 3 transmitter system, we find that multiplexing schemes (netMIMO) yield the best performance for static users with large coherence times with gains as high as 69%. However, when applied to mobile users and even static users with small coherence times, it degrades performance by as much as 71%. On the other hand, diversity schemes, with their increased coverage and diversity gain, yield the best performance for mobile users with gains as much as 96%. Interestingly, for the third category of static users with small coherence times, transmitter cooperation either degrades performance or results in capacity under-utilization, thereby resulting in conventional reuse schemes being the best suited strategy with gains as much as 83%.

While our findings would apply to larger topologies as well, translating them to a practical system encounters two key challenges: (i) accurate categorization of users into

[1]Note that this does not preclude each of the transmitters from individually employing MIMO to their users.

various profiles is central to the whole process, and (ii) how to combine multiple strategies to effectively cater to users of various profiles simultaneously. To this end, we present *TRINITY* - a practical system to address the afore-mentioned challenges and cater effectively to users of various profiles simultaneously. Briefly, TRINITY is deployed at the central controller (managing enterprise networks) and incorporates three key design elements.

- It enables simultaneous operation of all strategies by multiplexing them in the frequency domain in OFDM networks, where the available sub-carriers are split between different strategies. This allows for power pooling benefits that are not available with time domain multiplexing.
- The resources (e.g., # sub-carriers) allocated to a strategy depends on the traffic load of the corresponding user profile and is closely integrated with the user categorization process itself. A measurement based approach coupled with sensor hints (eg., accelerometer [33]) is employed to accurately categorize users into profiles.
- To maximize network performance, an efficient approach for clustering transmitters is proposed that addresses the performance-complexity tradeoff with multiplexing and the coverage-capacity tradeoff with diversity schemes.

We implement and evaluate TRINITY via extensive experiments on our 13 node WARP [34] testbed and with large-scale simulations. We see that TRINITY categorizes users with 90-95% accuracy. Tailoring transmission strategies to heterogeneous user profiles at finer time-scales compared to application layer dynamics, TRINITY can benefit both throughput and delay sensitive applications. It improves network rate by 50%-150%, satisfies QoS requirements for *thrice* as many users, and improves PSNR for video traffic by 10 dB, compared to individual transmission strategies.

3.2 Overview of Strategies

The gamut of transmission strategies can be categorized based on the level of cooperation between the transmitters at the top level. Note that, in all the schemes we discuss below, each transmitter (AP) can individually employ single-user or multi-user MIMO with its own clients. However, for ease of exposition, we will restrict our discussion to single antenna APs and users.

3.2.1 Non-cooperating Transmitters

The case of non-cooperating transmitters would correspond to the conventional CSMA paradigm - APs avoid interference in the presence of a carrier (sensing), while spatial reuse is automatically leveraged otherwise. In the example in Fig. 3.1, APs 1 and 3 can transmit in tandem on the same channel, while AP 2 time shares the medium with 1 and 3. Such schemes and their variants (e.g., with MIMO APs) will collectively be referred to as *Reuse* schemes.

3.2.2 Cooperating Transmitters

When transmitters are allowed to cooperate and are synchronized (at symbol level), additional benefits can be leveraged depending on the nature of cooperation.

Diversity

The goal of these schemes is to send multiple, dependent versions of a data stream through multiple transmitters to provide transmit diversity. A popular example would be the distributed Alamouti space-time (ST) codes [35] - when APs 1 and 2 employ the 2×1 Alamouti code to client 1, the resulting diversity gain allows the SNR at client 1 to scale as $|h_{11}|^2 + |h_{21}|^2$, where h_{ij} is the complex channel gain between AP i and user j.

A simpler form of transmit diversity is to transmit the same version of the data stream from multiple transmitters as shown in Fig. 3.2, wherein client 1 receives a coherent combination of the streams over a composite channel $\tilde{h} = h_{11} + h_{21} + h_{31}$. The power pooled from the multiple transmitters contributes to a combining (SNR) gain on average. This latter form of transmit diversity is similar in principle to broadcast and is often referred to as traditional distributed antenna systems (DAS). Further, unlike ST diversity, DAS does not require the receiver to estimate the individual channels from different transmitters (and associated pilot overhead). This has made it a popular transmit strategy for deployments in stadiums, universities, casinos, hospitals, etc. [36] for both WiFi and cellular signals.

Allowing the data to be accessible from multiple transmit points simultaneously not only provides coverage (during mobility) but also a diversity gain. Further, it does not use CSI (between APs and clients) and hence does not rely on feedback from clients. We refer to such schemes broadly as *diversity* schemes.

Multiplexing

In multiplexing, multiple independent data streams are transmitted concurrently to different users by converting interference into a multiplexing gain through transmitter cooperation - a classic example of which is network (distributed) MIMO [31, 32]. The data streams for different users are shared at each of the transmission points, which are in turn tightly synchronized (at the level of symbol phases). From the PHY layer perspective, this can be realized using a precoding algorithm called zero-forcing beamforming (ZFBF) [37]; this applies a precoding matrix (V, computed from channel matrix inverse) to send a linear combination of the data streams through each AP, such that unwanted streams (interference) cancel each other at each client, thereby leaving only the desired stream. A simple scenario is shown in Fig. 3.3.

Network MIMO (netMIMO) can allow the capacity to scale with the number of cooperating transmitters. However, this comes at the cost of leveraging CSI that needs to be fed back from the clients in a timely manner and tight phase synchronization between APs. We refer to variants (e.g., other forms of precoding) of these netMIMO schemes as multiplexing schemes.

Related Work

The availability of experimental software radio platforms has made it possible for network practitioners to take sophisticated physical layer multiplexing schemes like beamforming [38], multi-user MIMO [39, 40, 41], interference alignment [42, 32], network MIMO [31], etc. from theory to practice. Similar efforts have been made with respect to Reuse [43] and diversity [44] strategies as well. While these works have made great strides in highlighting the potential and limitations of such schemes in practice, their focus has understandably been on static clients with stable channels (for reliable CSI estimation) as a first step. On the theory front, works [45] are now exploring how to effectively leverage outdated CSI. However, their use in netMIMO is still in its early stages and is hence not considered here.

Recent works [33, 38] have looked at the impact of mobility on the performance of user's rate adaptation. While [33] employs sensor hints to detect mobility and tune rate adaptation accordingly, [38] adapts the rate table subject to channel coherence for

beamforming purposes. Both these works focus on the rate adaptation process assuming a given transmission strategy.

Going forward, given the heterogeneity of user profiles that a system must cater to in both enterprises and outdoor cellular (e.g., small cell LTE, WiMAX) networks, it becomes important to understand which *strategies* are appropriate for which user profiles and how to intelligently combine them, thereby forming the focus of this work.

3.3 Mapping Strategies to User Profiles

3.3.1 User Profiles

Users in enterprise networks can be categorized into one of the following three categories based on their mobility and channel coherence characteristics.

- *Mobile users with Short Coherence Time (Mobile)*: Coherence time (Tc) varies based on the speed of mobile client which can range from walking speed of 3-4 Kmph (Tc = 10 ms) to vehicular speeds of 75 Kmph^2 (Tc = 1.1 ms).
- *Static Users with Short Coherence Time (Short-Tc)*: Static users can also experience short Tc (10-20 ms or less) due to a dynamic environment where objects or other subjects (users) are mobile.
- *Static Users with Long Coherence Time (Static)*: Clients in static settings and in the absence of mobility experience a more stable channel (longer Tc, 100 ms or more).

3.3.2 Experimental Study

Our experimental set-up shown in Figs. 1 to 3 is deployed in an indoor lab with three transmitters and three clients using a WARP testbed. For mobility we placed the WARP clients on a cart and we moved the cart at walking speeds. We move the cart at the same walking speed on the same path for all mobile experiments. Aggregate network rate (throughput normalized to unit bandwidth, bits/s/Hz) is used as the metric and depends on both client SNR as well as the schedules indicated in Figs. 1 to 3. The reported results

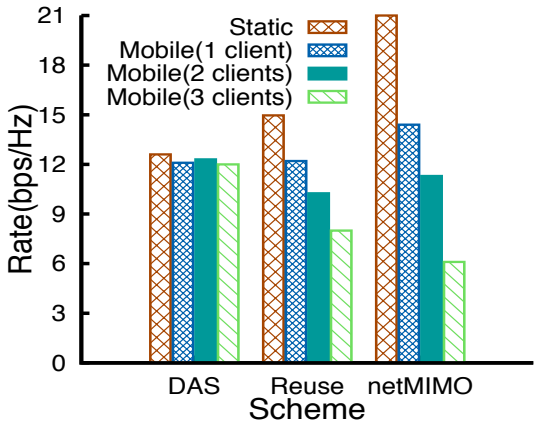


Figure 3.4: Static vs. Mobile.

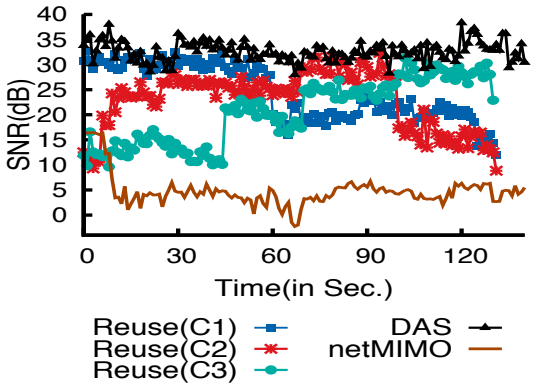


Figure 3.5: Impact of Mobility.

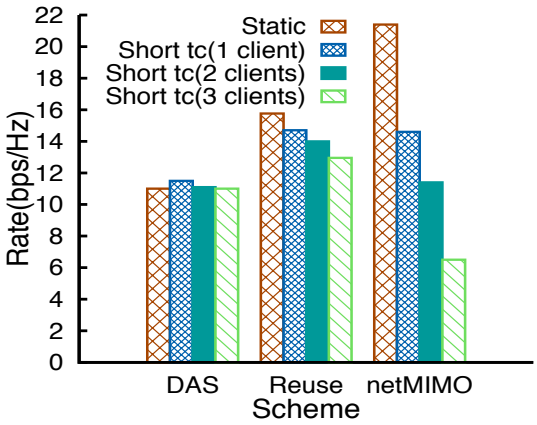


Figure 3.6: Static with Short Tc.

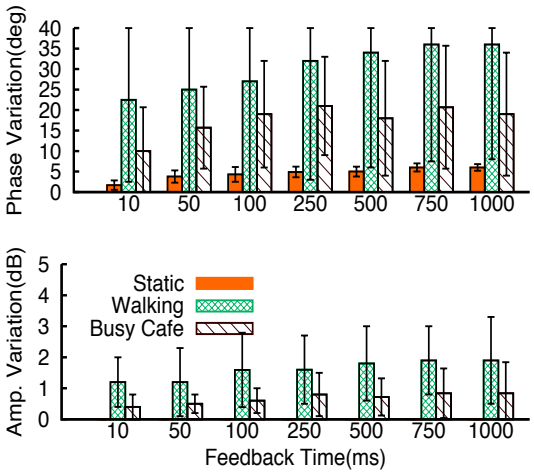


Figure 3.7: Channel variations.

are averaged over multiple runs. Further details on testbed, implementation and metrics are given in section 4.6.

Static Clients: From Fig. 5.5, it can be seen that *when all the clients are static, netMIMO is the most appropriate strategy*, outperforming Reuse and DAS by up to 69.8%. netMIMO achieves high network rate by multiplexing three data streams to three clients at the same time. Since the channel coherence time in our static environment is large, all clients can decode their data with high reliability. Further, *since the benefit of spatial reuse*

[2]Vehicular speeds are considered as our schemes are also applicable to small cell networks outdoors as discussed in Section 3.8.

(absent in DAS) outweighs the diversity gain in static environments, Reuse performs better than DAS.

Mobile Clients: However, as we vary the number of mobile clients in the network, the performance of both netMIMO and Reuse start to degrade; the degradation for netMIMO being especially severe. On the other hand, *owing to coverage and combining gain in SNR from three transmitters without reliance on any CSI, DAS is unaffected by client mobility.* Thus, DAS outperforms netMIMO and Reuse by upto 96.7% depending on the number of mobile clients.

To better understand the behavior of these transmission schemes, we also recorded the received SNR of the symbols transmitted from each transmitter. Fig. 5.6 shows the average SNR of the mobile clients under Reuse, DAS and netMIMO over multiple runs. For DAS and netMIMO, we only report the SNR values of a single client because other clients also exhibit a similar trend. In the Reuse scheme, client-1(C1) is associated with AP-1, client-2(C2) with AP-2, client-3(C3) with AP-3. During experiments client-1 is moved from left to right while client-3 is moved from right to left. Client-2 is moved in both directions. It can be inferred from Fig. 5.6 that each client in Reuse, experiences a high SNR only when they are near their respective APs. In DAS, a mobile client experiences a high SNR throughout the experiment due to the coverage and the signal combining effect from the three transmitters, with about 5-6 dB of SNR gain over the highest SNR possible with Reuse (with ideal handoffs).

Why not Reuse or netMIMO for Mobile Clients? Note that *in addition to link degradation, mobility also impacts the benefits of spatial reuse in Reuse.* Even for centralized schemes [43] to leverage reuse, interference conflicts have to be estimated from transmitters to various clients to determine appropriate reuse schedules - a process that is executed only once every several packets to eliminate overhead and complexity. In the presence of client mobility, tracking interference conflicts of mobile clients accurately is not practical, which in turn affects the potential of reuse from such clients.

In netMIMO, the SNR goes down as soon as the client becomes mobile. The reason is evident from Figs. 5.7, where in contrast to a static client, a large variation in channel phase (20deg-40deg) and magnitude (1-2 dB) is observed for a walking client even with a channel (CSI) feedback rate of 10ms (similar values were also reported in [38]). Since precoding employs CSI to remove the interference between the concurrent streams, *stale CSI resulting from such variations during mobility has a more pronounced impact on*

netMIMO performance. Further, if there are N clients/AP, M APs and S sub-carriers (or sub-channels) in a OFDM transmission, to apply netMIMO, one needs to measure channels from each of M APs (sequentially, M transmissions) to all the MN clients in all the S sub-carriers (SMN channels), which then has to be fed back from each of the MN clients (MN transmissions) to construct the channel matrix required by the precoding algorithm. Even with channel reciprocity (available only in TDM systems), one has to incur the MN transmissions from clients for estimation. Further, adding a few measurements per client that are typically needed for stable channel estimates, it is hard to accomplish this entire process within the channel coherence time of even walking clients, thereby explaining the acute degradation in netMIMO performance for mobile clients.

Thus, *DAS ends up being the best suited scheme for mobile clients.*

Short-Tc Clients: A client can experience fluctuating wireless channel conditions even when it is not mobile due to various reasons (such as mobility in environment, multi path etc.). This fluctuation in channel conditions can result in a small channel coherence time (around 10 ms) even though the client is static as observed in Fig. 5.7 for a static client in a busy indoor cafeteria. Fig. 5.8 shows the aggregate network rate achieved by each scheme for such static clients in a busy cafe. With phase variations as high as 20deg even for a CSI feedback rate of 10 ms, netMIMO is highly susceptible to stale CSI and the rate degrades in a similar fashion as in mobile scenario. However the corresponding degradation in Reuse is now less pronounced. While link quality is impacted, since the topology does not change, spatial reuse is not impacted and thus, Reuse outperforms DAS (higher gains expected in larger topologies). Hence, interestingly, *a simple Reuse scheme serves the best for static clients that experience fluctuating channel conditions (small coherence time).*

3.4 Design of TRINITY

Two key challenges arise in realizing a practical system that can leverage our inferences from Section 3.3: (i) How to categorize users into various profiles? and (ii) How to intelligently combine various strategies to cater to a heterogeneous set of users simultaneously and manage resources effectively between strategies? We now present the key elements in TRINITY designed to address the above challenges. TRINITY is implemented at the central controller that manages the distributed transmitters.

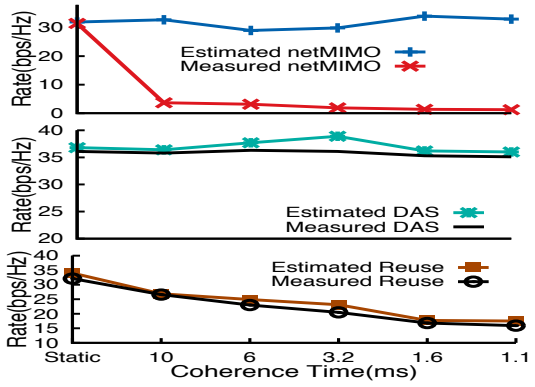


Figure 3.8: : Estimated vs. Measured

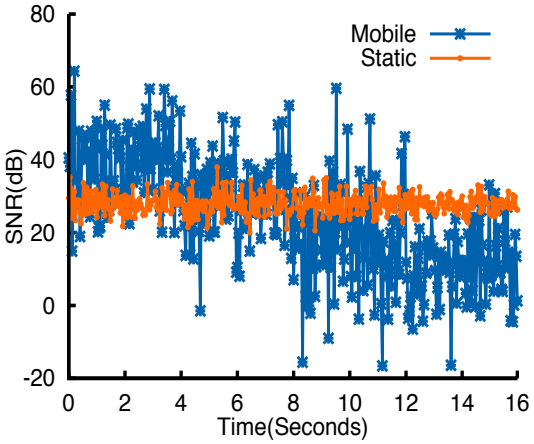


Figure 3.9: : SNR: Mobile Vs. Static

3.4.1 Categorization

While channel coherence is hard to measure, the mobility state of a user that influences the former can be inferred with the help of sensor and SNR hints as follows.

Triggers

Sensor hints: Most mobile devices such as smartphones and tablets are equipped with GPS and accelerometer. While GPS can provide a good estimate of user speed outdoors (especially for vehicular users), accelerometer readings have been successfully employed [33, 46] to decipher user mobility indoors with high accuracy. While accelerometer data is used to determine whether a mobile device is static or mobile in [33], it is used to determine user mobility at walking speed in [46]. When accelerometer data is available, TRINITY employs the approach in [33]. Here, for each new accelerometer sample, the standard deviation of the sample's magnitude is computed over a sliding window (w) of samples. If the standard deviation in a window exceeds a threshold (a), movement is detected, while if it is within the threshold for n successive windows, the device is reported to be stationary. We adopt the values reported in [33], where experimental evaluations indicated that with $w = 5,\ a = 0.15\ m/s^2$, and $n = 10$ device movement can be detected with high accuracy over 95%.

SNR hints: While sensor hints provide high accuracy, they may not be available from all devices. In such cases, TRINITY employs SNR hints to decipher user mobility with accuracy over 85% using approaches similar to [47, 48]. The variation in the average SNR

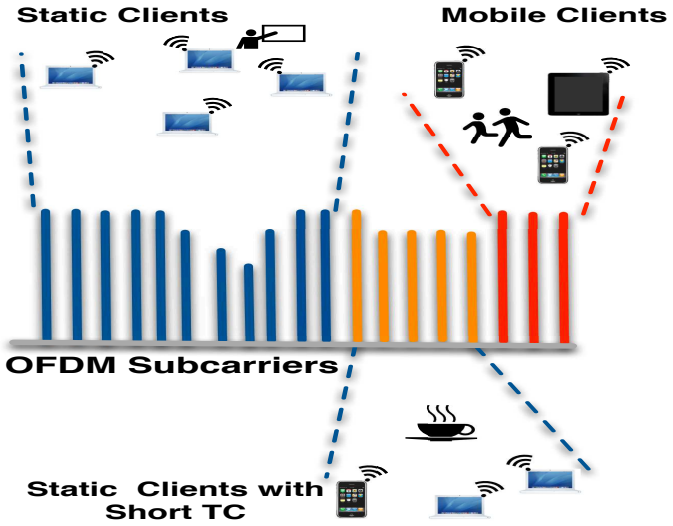


Figure 3.10: : Strategy Multiplexing

of a client during a given time period can be used for this purpose. Consider a simple experiment where a mobile client is moving away from the AP at walking speed and we record its SNR variation with time. We performed 20 such trials and SNR trace from one such trial is compared against that of a static client in Fig. 3.9. As expected, the SNR variation for a mobile client is quite high due to the rapidly changing environment. More importantly, we found that the SNR variation for the mobile client is consistently more than 10 dB in all 20 trials. Similar values based on experimental measurements are also reported in [47, 48]. Using this value as threshold, one can estimate the mobility state of a user (mobility hint) by taking the standard deviation of a moving window (2 seconds) of ten samples of average SNR, each computed over a 200 ms (t_o) interval.

$Std.Dev(SNR(t-9t_o),\ldots,SNR(t-t_o),SNR(t)) > Th_{moving}$

where $Th_{moving} = 10dB$.

Sensitivity of netMIMO Similar to sensor and SNR hints, netMIMO's sensitivity to fluctuating channels can be used effectively in categorization. When channels are measured from the transmitters to users and the precoding matrix is computed for netMIMO, one can easily estimate the SINR *expected* at users when netMIMO is executed (calculated using the proposed technique in [49], see Section 3.5.3). When netMIMO is executed, the resulting SINR or rate can be *measured* and then compared against the estimated value. A significant drop in measured rate is expected if netMIMO can no longer be supported by the user's channel. To understand the validity of our claim, we simulate a scenario with

different channel coherence times to create mobility. Instead of transmitting the symbols over the air we pass them through a flat AWGN channel, so as to keep it constant over the coherence time (channel feedback rate is 20ms), after which it changes independently to a new realization. It is seen from Fig. 3.8 that netMIMO rate drastically degrades for a client when it changes its state from being static to mobile even at walking speed. Unlike DAS or Reuse, the deviation between the estimated and measured rates is very large for netMIMO (more than 50% consistently) due to its strong reliance on CSI.

Thus, while sensor and SNR hints are strong indicators of mobility, significant rate discrepancies can easily predict lack of support for netMIMO.

Algorithm

Leveraging the above two indicators as triggers, TRINITY categorizes users with high accuracy using the following approach.

Step 1: When a user joins the network, with the help of sensor and/or SNR hints, it first determines if the user is mobile and if so categorizes it to be a DAS user. Otherwise, it aggressively assumes the user to be a netMIMO user.

Step 2: In addition to the user's netMIMO SINR, it can also estimate (from the measured channels) the SNR (and hence rate) the user would receive if the transmitters were to instead operate in DAS and reuse (single transmission per user) modes. If there is a significant difference between the estimated and observed netMIMO rates (by more than 50%) for the user, then the user is removed from the netMIMO category. If the user cannot support netMIMO and if its sensor and/or SNR hints do not indicate it to be a DAS user, only then the user is classified as a Reuse user.

Step 3: *Re-categorization during network dynamics-* Since a user's profile can change from time to time, categorization cannot be a one-time process but must be implicitly integrated into the system. If the user is a netMIMO or Reuse user and if its profile degrades to a lower multiplexing category (netMIMO→reuse→DAS), then this would automatically be reflected in its rate (for netMIMO) and sensor/SNR hints (for Reuse). Using these triggers, the user can then be appropriately assigned to DAS or reuse. However, to determine if a non-netMIMO user can be moved to a higher multiplexing category (DAS→reuse→netMIMO), TRINITY periodically (every 5 secs) moves the user to netMIMO category and performs categorization to determine its appropriate category.

3.4.2 Multiplexing Strategies in Frequency Domain

The ideal scenario would be to partition the network into disjoint regions, where only one strategy needs to be applied in each region. This would allow frequency and time resources to be reused by strategies across the network. While such scenarios can occur (e.g., a big conference hall with only static users on one end and a cafeteria with only mobile users on the other end of a floor), they are not common. In reality, users of different profiles are inter-twined in various regions of the network (e.g., static and mobile users in a cafeteria). Hence, it becomes inevitable to multiplex different strategies either in the time or frequency domain to serve users of different profiles in any given region.

TRINITY employs multiplexing in the frequency domain, which allows it to leverage power pooling benefits from the transmitters. In an OFDM system with say N sub-carriers, these sub-carriers would be split orthogonally (eg. N_m to netMIMO, N_d to DAS, $N - N_m - N_d$ to Reuse) between the various strategies in TRINITY (eg., Fig. 3.10) and used by the transmitters to serve their respective clients anywhere in the network. On the downlink (AP→users), when an AP (with fixed transmit power) has users that do not cover all the profiles, then the unused power on the sub-carrriers assigned to the unused strategie(s) will be pooled to the sub-carriers assigned to the strategies in operation. This in turn results in a higher SNR on the sub-carriers in operation, a gain termed as *power pooling gain*. Note that when strategies are multiplexed in the time-domain, all sub-carriers are used for a given strategy at a time and hence there is no room for power pooling.

3.4.3 Resource Management

Once the users are categorized, the traffic load for each of the strategies is determined based on the traffic carried by users in the respective category in the entire network. The different user profiles (i) are weighted (v_i based on priority or fairness), and the allocation of number of sub-carriers to each of the strategies across the entire network is made proportional to their weighted traffic load (eg. $N_i = \frac{v_i N}{\sum_i v_i}$). Note that other weight choices can also be employed. Among the sub-carriers allocated to a strategy, users in the respective profile are then scheduled based on a desired fairness model (proportional fairness in our case). Since resource management is tightly coupled with user categorization, it continuously adapts to dynamics in user profiles. Once resources have been allocated to different

strategies (user profiles), the next step is to determine how the transmitters serving users in different categories access these resources.

3.5 Medium Access in TRINITY

While the reuse strategy can be realized in a completely distributed manner, cooperation strategies like netMIMO and DAS do require a central entity (eg. central controller). Given this, TRINITY adopts a centralized approach to downlink medium access, whereby transmissions are scheduled by the central controller. Further, since the resources allocated to the strategies are orthogonalized in the frequency domain, it is now sufficient to consider scheduling within each of the strategies in isolation. We will later discuss (Section 3.5.5) how the various strategies are integrated at a transmitter.

3.5.1 Clustering for netMIMO and DAS

Need for Clustering: For netMIMO, the ideal operation would be to execute one large netMIMO between all the transmitters in the network and the users in the netMIMO category. While this would provide the maximum number of concurrent streams (scaling with the # of transmitters, say T), this would also incur the overhead of synchronizing all the transmitters, measuring and feeding back CSI from all transmitters to all netMIMO users and sharing of all netMIMO users' data streams across all transmitters. Clearly this is not feasible beyond a few transmitters (as considered in [42, 31]), especially for large networks, where the central controller manages several tens of transmitters.

Similarly in the case of DAS, involving all transmitters in one big DAS set-up eliminates the need to identify appropriate transmitters for clients, especially when they are mobile and indoors. However, the cost incurred is that the data served to a user on a sub-carrier is broadcast to all the transmitters, thereby significantly limiting reuse in the network. Given that users' mobility may be restricted to regions of the network for a given period of time, this could amount to gross under-utilization of the spectrum.

Approach: To address these issues, TRINITY employs a clustered approach to executing netMIMO and DAS. TRINITY decomposes the set of transmitters into smaller, contiguous clusters of transmitters (Fig. 3.12), wherein netMIMO and DAS are executed only within each cluster and interference between clusters is avoided either in the time or frequency domain. Executing cooperation strategies at the cluster granularity ensures

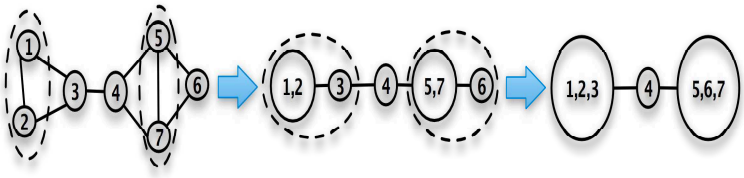


Figure 3.11: Graph coarsening with $Q = 3$

feasibility for netMIMO, while it allows for reuse of spectrum across clusters for DAS. The maximum cluster size (Q, decided based on practical considerations) as well as the clusters themselves can be different for netMIMO and DAS.

Clustering Algorithm: In determining these clusters, we design a topology-aware clustering algorithm for TRINITY.The underlying principle is that transmitters within each cluster should have strong mutual interference (ideally they should form a clique in the conflict graph), in order to maximize the cooperation or combining gain. Moreover, transmitters across different clusters should be weakly coupled, so as to maximize frequency reuse opportunities. TRINITY adopts a graph coarsening approach to clustering (executed by the central controller), whereby neighboring vertices are consolidated (merged) recursively based on a desired metric.

Step 1: For each strategy, the entire network can be represented by a conflict graph $G(V, E)$, where transmitters constitute the vertices V and an edge $(u, v) \in E$ indicates interference (to clients) between transmitters u and v operating under the given strategy.

Step 2: For each randomly selected vertex v, the algorithm finds the neighbor u that together with v suspends the maximum clique locally. Then, it merges u, v and their edge into a single vertex u', which is added into the conflict graph.

Step 3: This procedure is applied recursively on the new conflict graph, until no more vertices can be merged or the cluster size constraint Q is reached. Each vertex in the resultant conflict graph contains multiple transmitters in the original network, which then form the desired clusters (see Fig. 3.11).

Employing *heaviest clique* to be the coarsening metric helps create clusters with good cooperation gain, while also reducing the interference across clusters.

3.5.2 Interference Avoidance across Clusters

An inevitable feature of clustering in netMIMO and DAS is that transmitters on the edge of the clusters will receive interference from neighboring clusters, albeit only from transmitters operating in the same strategy. Note that, instead of allocating sub-carriers to strategies across the entire network, one can envision varied allocations for different clusters based on their local traffic demand. However, this would result in cross-strategy interference across clusters that is challenging to address and is hence avoided in our design.

To eliminate inter-cluster interference, TRINITY takes the following approach.

Step 1: It considers a conflict graph with clusters themselves as vertices (an edge exists between adjoining clusters) and performs a simple, (degree-based) greedy multi-coloring on the set of sub-carriers allocated to the strategy.

Step 2: The orthogonal sets of sub-carriers assigned to the clusters are then used by their cluster-edge transmitters alone to remove inter-cluster interference, while all sub-carriers allocated to the strategy are used by the cluster-interior transmitters, thereby allowing for efficient reuse of resources across all the clusters. In the example in Fig. 3.12, APs 3 and 4 are cluster-edge transmitters and hence operate on orthogonal subcarriers (first 24 subcarriers), while the other APs operate on all sub-carriers (see Fig. 3.13) for their netMIMO clients.

In the case of reuse strategy, there is no need for clustering (or equivalently cluster size is one with every transmitter being an edge transmitter). Thus, a conventional conflict graph with transmitters can be formed and a greedy multi-coloring on the set of sub-carriers allocated to reuse strategy will determine the set of frequency resources for operation at each transmitter.

3.5.3 Client Scheduling

Once the set of sub-carriers for operation at each transmitter for a given strategy is determined, the next step is to schedule clients. Given that there may be multiple clients associated with a transmitter, scheduling must be performed to ensure long-term throughput fairness for clients.

Reuse and DAS: Each client i is associated with a weight w_i and a time-averaged throughput R_i. In each time frame t, each transmitter schedules a client (among those in the same strategy) with the maximum weighted throughput ($\arg\max_i w_i r(t)$), and performs the fol-

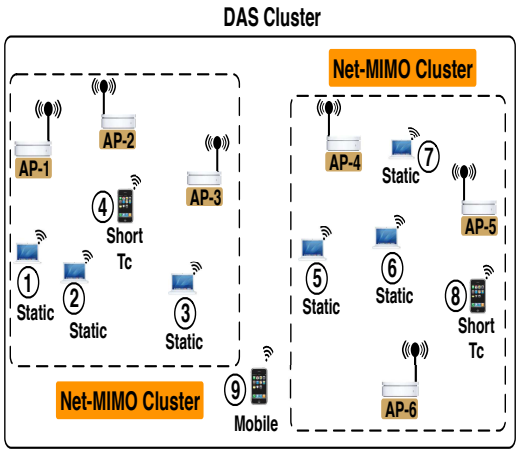


Figure 3.12: : Sample network topology

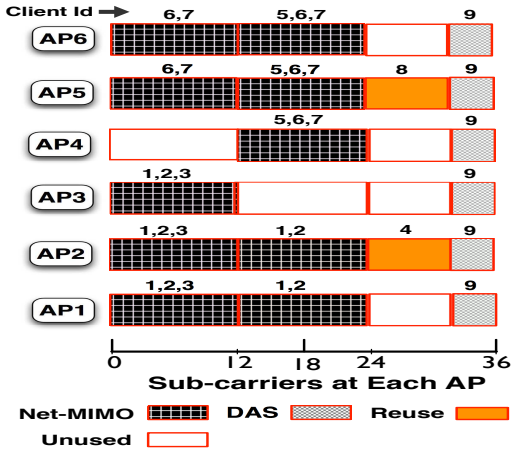


Figure 3.13: : Execution of Trinity on sample topology

lowing update after packet transmission:

$$R_i \leftarrow \alpha r(t) + (1-\alpha)R_i, \text{ and } w_i \leftarrow R_i^{-1}$$

where $r(t)$ is the throughput achieved in the current time frame and α is an exponential filtering coefficient. By configuring w_i to the inverse of R_i, the above update is proven to achieve long-term proportional fairness for clients and is equivalent to maximizing the aggregate utility of users in the strategy (with each user's utility given by $\log(R_i)$) [37].

netMIMO: Here, scheduling is more sophisticated as each transmission involves multiple clients. Different grouping of clients results in different crosstalk interference and hence different individual throughput. TRINITY groups clients using a simple greedy approach, executed in the beginning of each time frame (more sophisticated approaches are also possible). It iteratively searches for the next client that provides the maximum increase to the weighted sum throughput $R(t)$, until no clients can be added. A client's weight w_i is updated in a similar way to reuse and DAS.

When a set of clients are grouped for netMIMO transmission, the expected rate of each is computed assuming zero-forcing beamforming with a maximum power constraint on each transmitter. Suppose $\boldsymbol{M}$ is the channel matrix from all transmitters in the cluster to the set of selected clients I', with rows indexing the clients and columns the transmitters.

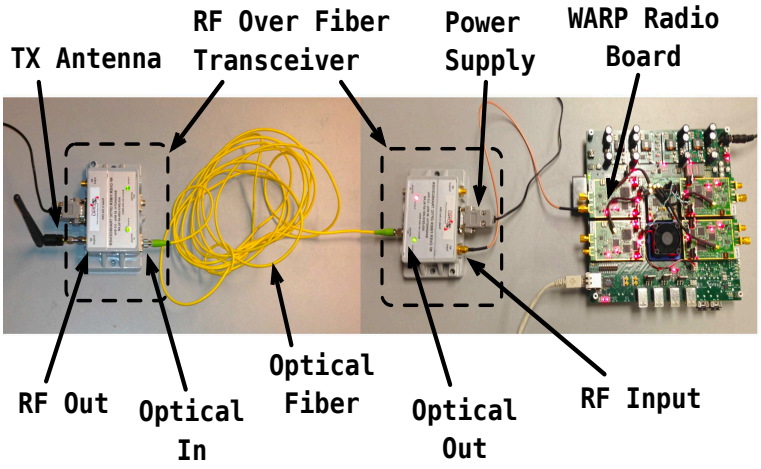


Figure 3.14: RF over Fiber Transmission

The zero-forcing precoding matrix is computed as the pseudo-inverse of $\boldsymbol{M}$ [37]: $D = \boldsymbol{M}'(\boldsymbol{M}\boldsymbol{M}')^{-1}$. To respect the per-transmitter power constraint, elements of D must be normalized [37] by $\sqrt{P}$, where $P = \arg\max_k \sum_{i=1}^{|I'|} |M_{ki}|^2$. Then for client k, the expected rate can be estimated as $R_k = W \log(1 + \sum_j M_{ij} D_{ji}/(\sqrt{P} N_0))$, where N_0 is the receiver's noise level and W the bandwidth.

3.5.4 Framing Transmissions

TRINITY multiplexes multiple strategies in the frequency domain at a transmitter, which amounts to transmitting to multiple clients in a single frame. This is the essence of OFDMA, which is popular in cellular networks (LTE, WiMAX). Similar to a conventional OFDMA frame, TRINITY's transmission frames are two-dimensional in time and frequency (time symbols and sub-carriers). The frame consists of a preamble followed by a control region spanning a couple of symbols (and all sub-carriers). The control region (spanning two symbols) contains information on which sub-carriers carry data meant for which user as well as the MCS (modulation and coding rate) to be used for decoding the data. It is the responsibility of the controller to ensure that data for users on a specific strategy are scheduled only on sub-carriers assigned for the strategy. Training or reference signals/pilots are also embedded in the frame (similar to WiMAX/LTE, 802.11n/ac) to enable frequency synchronization, channel measurement and CSI reporting for netMIMO.

Compared to OFDM (employed in 802.11n/ac), whereby transmission to a single user is made in a frame, OFDMA would incur slightly higher control overhead (due

to multi-user transmission). However, it has been shown that the benefits of user multiplexing in OFDMA and its associated frequency (multi-user) diversity and power pooling gains outweigh the additional overhead incurred. Further, if necessary, the number of users scheduled at a transmitter within a strategy can be limited to balance diversity gains vs. overhead.

3.5.5 TRINITY: Summary of Operations

We summarize the sequence of operations in TRINITY with the help of a simple example in Fig. 3.12, where there are 6 APs and 9 clients, six of which (1,2,3,5,6,7) are static, two (4,8) are short-Tc and one (9) is mobile.

1. With the help of sensor and/or SNR hints, a non-categorized user is first checked to see if it is a DAS user (client 9). Otherwise, it is assumed to be a netMIMO user. Channels are estimated for the netMIMO user, followed by its execution. Based on the estimated and observed rates for netMIMO operation, and coupled with sensor hints, the user is appropriately categorized as either netMIMO or reuse user (clients 1,2,3,5,6,7 are netMIMO; 4,8 are reuse users).

2. Sub-carriers in frequency domain are appropriately partitioned between strategies based on traffic load of users in the associated categories, priority and fairness. (Assuming unit traffic load for all clients, netMIMO, reuse and DAS strategies receive 24, 8 and 4 sub-carriers respectively out of a total of 36 sub-carriers.)

3. For a given cluster size limitation ($Q_{netMIMO} = 3$, $Q_{DAS} = 6$), transmitters are clustered for netMIMO and DAS strategies using the heavy clique graph coarsening approach (two clusters for netMIMO and one for DAS). Transmitters on the cluster interior (APs 1,2,5 and 6 for netMIMO; all APs for DAS) use all sub-carriers assigned to the strategy, while those on the cluster edge operate on a subset determined by multi-coloring to avoid interference (for netMIMO, APs 3 and 4 operate on orthogonal sets of 12 sub-carriers out of the allocated 24). For reuse, transmitters directly employ a multi-coloring approach to determine their frequency resources (APs 2 and 5 do not interfere and employ all 8 sub-carriers assigned for reuse). Clients are then scheduled at each transmitter for each strategy on the appropriate set of sub-carriers subject to proportional fairness.

4. Data for multiple clients belonging to different profiles are then frequency multiplexed and transmitted simultaneously from each AP (frame transmissions at each AP is shown in Fig. 3.13).

5. While netMIMO and reuse users are re-categorized automatically as and when their channel state changes to one with lesser multiplexing gain, reuse and DAS users are explicitly moved to netMIMO category periodically to see if they can access a state with higher multiplexing gain.

3.6 Implementation and Test-bed

3.6.1 Implementation

We have prototyped TRINITY on the WARP (FPGA-based) SDR platform using the WARPLab framework. Each main WARP board is capable of supporting up to four daughterboards with RF front-ends. Multiple WARP boards can be connected to a host PC via an Ethernet switch. The host PC is responsible for the (de-)construction and processing of baseband waveforms that are transmitted (received) to (from) the FPGA board with the RF front-end. TRINITY and all its components are executed on the Host PC, which also serves as the central controller, and downlink frames for each AP are determined. WARP boards are used for both the AP and clients.

Transmission Strategies:

The PHY layer implementation for reuse is a simplified, centralized version of 802.11g. A transmitter modulates data bits using BPSK, and then maps the modulated symbols to OFDM subcarriers. Every 64 subcarriers form an OFDM symbol, and multiple OFDM symbols form the data portion of a frame (packet). Then, the transmitter prepends a preamble to the data, which contains two random sequences known to the receiver. The receiver uses a correlation algorithm to detect these random sequences, identify the start of the packet, and estimate the channel state information (CSI), in a way similar to [50]. Then the OFDM symbols can be identified and demapped to BPSK symbols (by normalizing with the CSI), and then demodulated to the original information bits sent by the transmitter. As a frame duration may exceed the channel coherence time, 4 pilot subcarriers are inserted into each OFDM symbol to update the channel periodically.

If transmitters are operating in netMIMO, then the central controller performs a linear combination of the BPSK symbols (zero-forcing beamforming precoding [37]) to be sent by the different transmitters so as to pre-cancel their mutual interference. The channel matrix, whose inverse provides the precoding matrix, is obtained before netMIMO operation from the CSI piggy-backed in the ACK packets for the previous frame. Each receiver stores its latest channel estimation results in its ACK, based on pilots from the last few OFDM symbols of the previous frame. Due to the nature of WARPLab framework, the minimum feedback delay for CSI is 50 ms, which is still sufficient for enabling netMIMO for static clients. The DAS mode is a simple extension to the reuse PHY. The same data symbols are modulated and sent over the air by different transmitters simultaneously without precoding.

Categorization: While SNR hints can be directly obtained from the sent and received data bits, WARP devices do not provide sensor hints unlike modern mobile devices. Hence, to emulate the latter, a trigger is explicitly fed to the WARP device during a channel state change using WARP's User I/O Push buttons. Several WARP-based PHY layer designs [34] have employed these push buttons for events such as incrementing center frequency, switching antenna mode, changing modulation and coding etc. In TRINITY, we program one of the push buttons to convey the sensor hint to the central controller that the client is mobile once it starts walking.

3.6.2 Deployment and Synchronization

In TRINITY, we make the transmitters themselves light-weight, serving as just RF boards and transmitting the RF signal, while all the baseband processing is pushed to the central controller. Hence, all the processing from all APs happens at the central controller. This allows for efficient and practical realization of netMIMO, especially for synchronizing, estimating channels and sharing of data streams across APs over a large geographic area. With each WARP mother board serving 4 daughter RF boards, we move the mother boards to the central controller, while the daughter boards serve as the RF transmitters. The RF signals processed at the mother boards are carried to the RF daughter boards through optical fibers (as shown in Fig. 3.14) that have very low latency ($\approx 5\mu$s/KM), thereby preserving the symbol-level synchronization needed for netMIMO. The RF signal is modulated to optical through an off-the-shelf commercial RoF (radio over fiber) transceiver, carried over the fiber and then converted back to RF signal at the other end. Co-locating multiple

Table 3.1: Physical layer parameters

Frequency	2.48GHz	Bandwidth	10MHz
Feedback Rate	50ms	Symbols/Pkt	2048
Symbol Time	$20\mu s$	Modulation	BPSK

WARP mother boards also enables easier clock and phase synchronization between them (using approaches recently proposed in [51]) - a feature imperative for the functioning of netMIMO.

Note that the above distributed, thin AP deployment model (with an optical backhaul) has several benefits and has recently become a very cost-effective and popular option for cellular and WiFi coverage in convention and hospitality centers, hospitals, etc. [52].

3.6.3 Experimental Set-up

All our experiments are carried out in the afore-mentioned testbed that is deployed in an indoor office environment as shown in Fig. 3.15. Our testbed consists of 6 Transmitters and 7 client nodes. Different scenarios and network topologies are created by moving the transmitters and clients to different locations. For mobility we placed the WARP clients on a cart and move the cart at (timed and repeatable) walking speed on the same path across all mobile experiments. Since a controlled environment is not possible for short-Tc clients (eg. busy cafe), to allow for repeatable experiments with short-Tc clients, we emulate their channel variations by moving only their antenna in the proximity of their original location. Fig. 3.16 clearly shows that such emulation produces channel variations very similar to a busy cafe, while at the same time allowing for repeatability.

All clients and transmitters are single antenna WARP boards. To avoid external interference, all experiments are performed during night and on channel 14 (unused by other devices) in the 2.4 Ghz band. All nodes are connected to a central controller (a PC running the WARPLabs PHY layer signal processing modules) via an Ethernet switch. Each experiment lasts for 6 minutes and the reported results are averaged over multiple runs unless otherwise stated.

Metrics: The main metrics of evaluation are aggregate network rate (bits/s/Hz or bits/s as appropriate) and utility. A client's rate is obtained from its achievable Shannon

rate ($log(1+\rho)$) and depends on average received SINR (ρ, per sub-carrier) from OFDM symbols transmitted as well as number of sub-carriers allocated to it. Due to the inherent feedback latency in WARP, rate adaptation is not supported and hence throughput is not directly measured. In a way, measuring rate directly from SINR prevents the vagaries of rate adaptation algorithm from influencing the inferences on strategies. Utility is used to incorporate fairness among clients in the same category. To achieve proportional fairness in our client scheduling (Section 3.5.3), the utility of a user is given by the logarithm of its rate ($\log(R_i)$). To evaluate QoS, we consider video delivery. Specifically, we consider typical video file [53] (encoding rate = 1Mbps, frames per second = 30) downloads and estimate the achieved Peak-Signal-to-Noise-Ratios (PSNR) for quantifying the delivered QoS. PSNR is defined as a function of the mean squared error (MSE) between the pixels of the decoded video and that of the original version. It is computed as follows: $PSNR = 10\log_{10}\frac{2^L-1}{MSE}\ [dB]$, where L is the number of bits used to encode pixel luminance (8 bits).

3.7 Evaluation

3.7.1 Micro-benchmarks

We validate TRINITY's performance first by micro-benchmarking its individual components,

Categorization

We use a sample network topology containing 3 APs and 6 clients as shown in Figure 3.17. Two scenarios are considered: (1) categorizing a single client (client 4) in isolation, and (2) categorizing two clients (2 and 4) simultaneously. To emulate a realistic scenario, where new clients in the network need to be categorized, we assume the rest of the clients in the network are already categorized accurately and focus on the categorization accuracy of the new clients. Given that all devices may not provide sensor hints, we also consider two variants - one with sensor hints and one without sensor hints (only SNR hints and CSI).

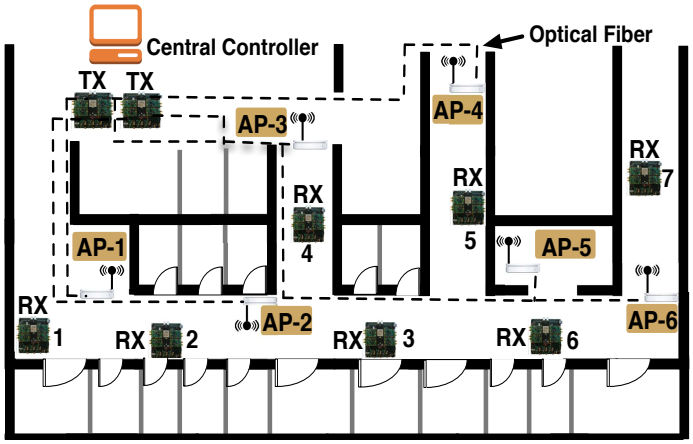


Figure 3.15: Testbed deployment

Accuracy with and without sensor hints: Client 4's channel state is set to be in one of the 3 states: static, mobile or short-Tc and 20 trials are performed for each of the states. In the case of two clients (4 and 2), experiments are conducted over all possible combinations of their profiles. Categorization accuracy is evaluated by comparing the profile categorized by TRINITY against *ground truth*, which in turn is the client's profile that yields the highest aggregate network rate. To determine the client's optimal profile, we perform exhaustive search by trying all possible combinations of transmission schemes (DAS, Reuse and netMIMO) for each client in the topology to find the combination which maximizes the overall network rate.

The accuracy results are presented in Fig. 3.18. TRINITY achieves a high accuracy of 95% for netMIMO users, 90-95% for reuse users and 80-85% for DAS users. The accuracy is highest for netMIMO, where rate discrepancies provide a clear indicator, while channel dynamics for DAS users contribute to some classification errors. The accuracy with sensor hints is about 7% higher than without it, indicating that SNR hints are sufficient to provide comparable accuracy in classifying DAS and reuse users. This is promising in that it reduces the dependence on sensor hints, allowing for operation with all devices. Further, introducing multiple clients for simultaneous categorization does not impact the accuracy of categorization. This is expected as categorization is specific to individual clients.

Impact on network rate and utility: We now try to understand the implication of the accuracy results by presenting their impact on aggregate network rate. Here, after categorization, the appropriate transmission schemes are used to transfer data to clients

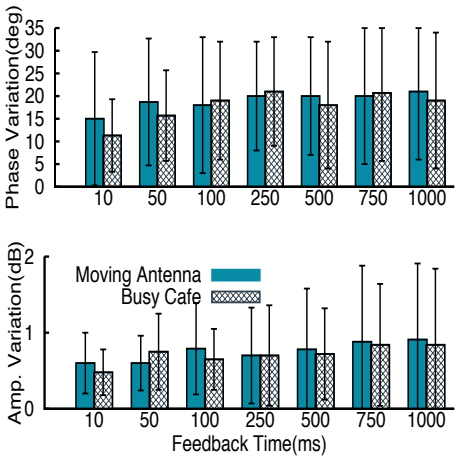


Figure 3.16: Channel Variation

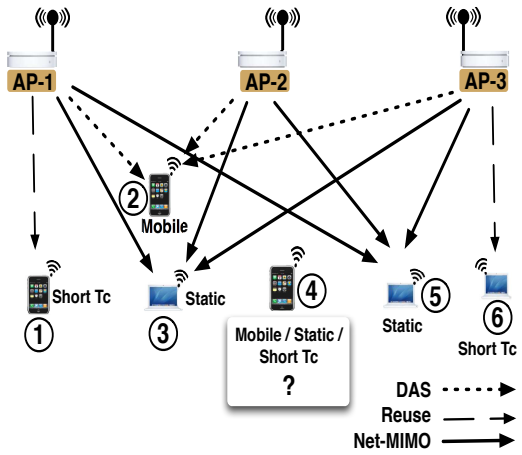


Figure 3.17: Sample topology for categorization

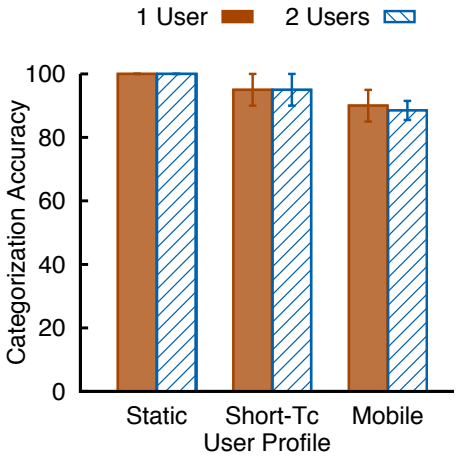


Figure 3.18: Accuracy of Categorization

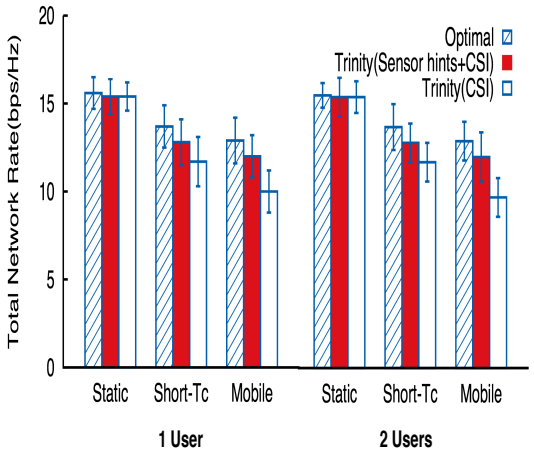


Figure 3.19: Impact on Network Throughput

and the aggregate network rate is measured. The results in Fig. 3.19 clearly indicate that the impact of making local categorization decisions for users, although sub-optimal compared to the exhaustive global approach (eg. some slow moving users may be under Reuse w.r.t. network rate), does not incur appreciable loss, while providing a large reduction in complexity.Given the high accuracy of categorizing netMIMO users, their loss in network rate is negligible. For reuse and DAS users, while the loss is small with sensor hints, in its absence the loss can get to 10% and 15% respectively but still remains within reasonable limits. Similar results are seen when the metric is modified to utility to capture fairness in Fig. 3.20.

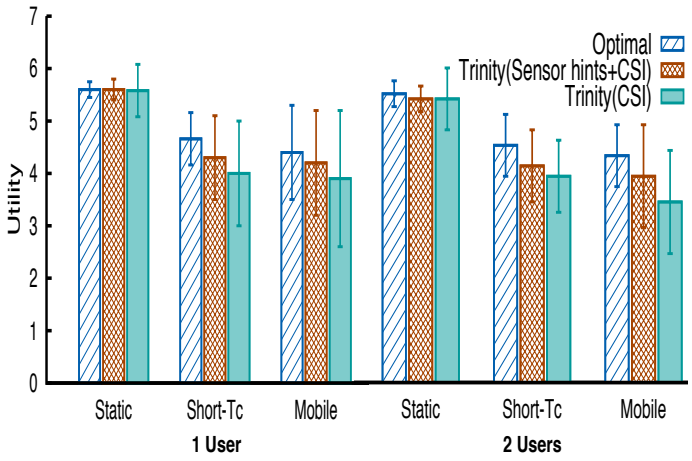


Figure 3.20: Impact on Network Utility

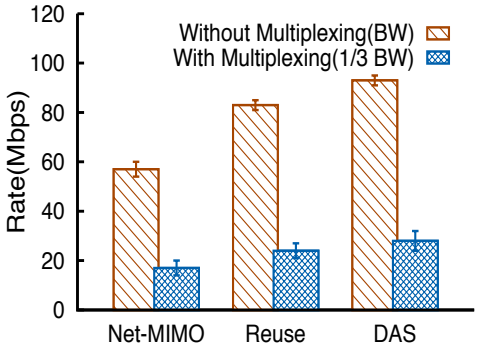


Figure 3.21: Frequency division multiplexing.

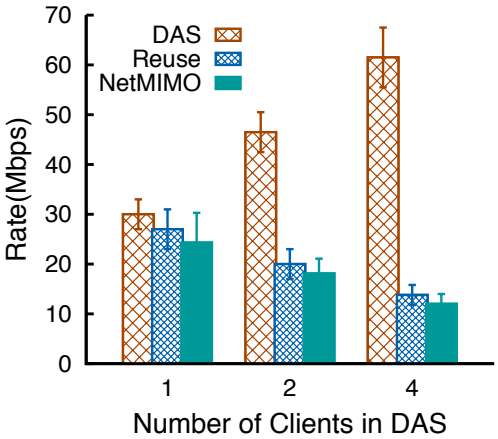


Figure 3.22: Proportional allocation.

The above experiments clearly indicate *TRINITY's ability to categorize users into their respective profiles with high accuracy using standard CSI and SNR hints (without reliance on sensor hints), thereby resulting in close-to optimal network performance.*

Multiplexing

We use the following topology for all the multiplexing experiments, wherein two APs jointly perform netMIMO to two netMIMO clients and DAS to a single DAS client. Also, one of the APs serves a single reuse client. Since different number of sub-carrier

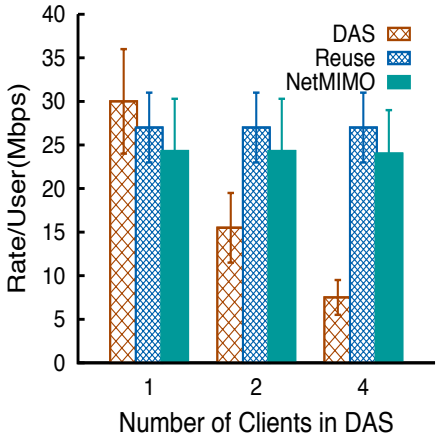


Figure 3.23: Subdivision allocation.

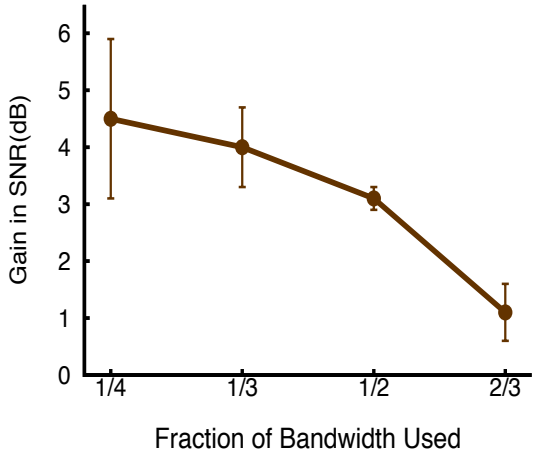


Figure 3.24: Power Pooling Gains.

resources may be employed in multiplexing, we use rate in bits/sec (accounting for # sub-carriers) as metric.

Feasibility of Frequency Domain Multiplexing (FDM): We first need to verify if multiplexing strategies (and hence users) simultaneously on orthogonal sub-carriers in the same frame can be made to work without any inter-carrier interference and hence throughput loss. We perform the following simple experiment. First, we evaluate the rate obtained by the clients when the strategies are operated in isolation, thereby operating on the entire bandwidth. Then, we execute TRINITY, wherein the three strategies are multiplexed at the two APs in each frame. Each strategy receives only one third of the entire bandwidth when multiplexed. The resulting rate of one client in each profile is then compared against that received when operating in isolation in Fig. 3.21. It can be clearly seen that the client in each profile is able to receive one third of its net rate when multiplexed, thereby indicating that clients are able to receive data when multiplexed simultaneously without suffering any throughput loss.

Different Allocation Policies: TRINITY can incorporate various allocation policies in determining the set of sub-carriers allocated to the different strategies. We now highlight two such policies. In one policy, we allocate sub-carriers proportional to the traffic in each profile. Assuming the same traffic load from each client, this is equivalent to an allocation proportional to the number of clients in each profile. Using the same topology as above, we now increase the number of DAS clients in the network. The resulting aggre-

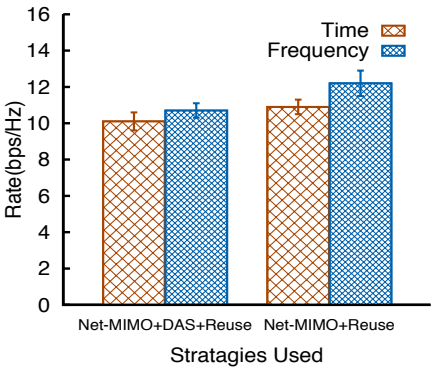


Figure 3.25: TDM Vs. FDM.

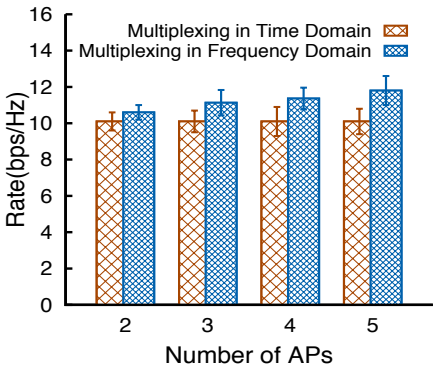


Figure 3.26: Impact of # APs on FDM.

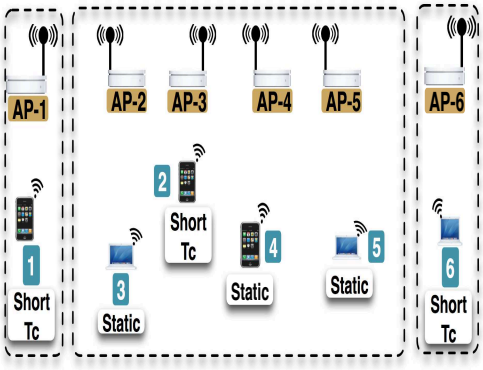


Figure 3.27: Sample network topology for case study 1.

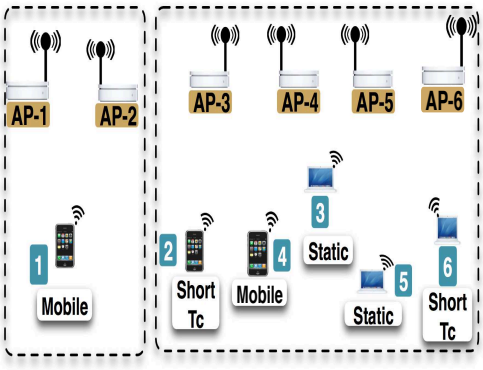


Figure 3.28: Sample network topology for case study 2.

gate network rate in each profile is presented in Fig. 3.22, clearly reflecting an allocation corresponding to the increasing traffic load in DAS profile at the cost of other profiles. In another policy, if one wants to maintain strict allocation guarantees between profiles, one could perform a deterministic allocation. For eg. in Fig. 3.23, we split the sub-carriers equally between the profiles and independent of their load. Hence, as the number of DAS users is increased, the available bandwidth for DAS profile gets shared amongst its users resulting in lower rate per DAS user without affecting users in other profiles.

Benefits of Frequency Domain Multiplexing (FDM): Recall that FDM allows for the power from un-used sub-carriers at an AP from one profile to be used towards clients in other profiles, thereby providing an SNR (power pooling) gain. We validate this gain by reducing the number of sub-carriers utilized for transmission to a reuse client and

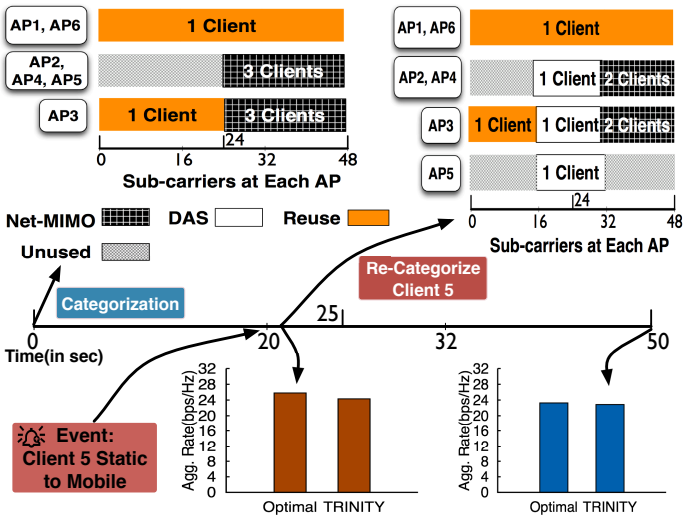


Figure 3.29: Execution of TRINITY on sample topology 1

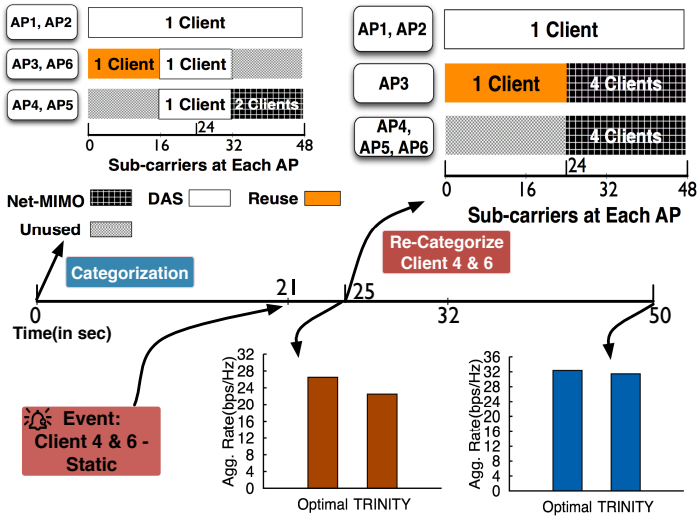


Figure 3.30: Execution of TRINITY on sample topology 2

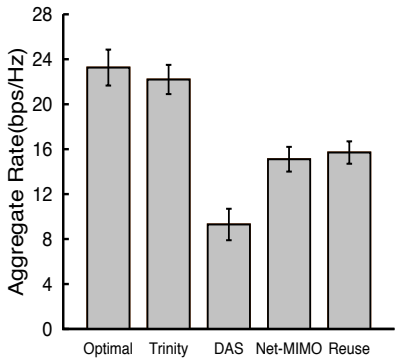


Figure 3.31: Performance of TRINITY: Throughput

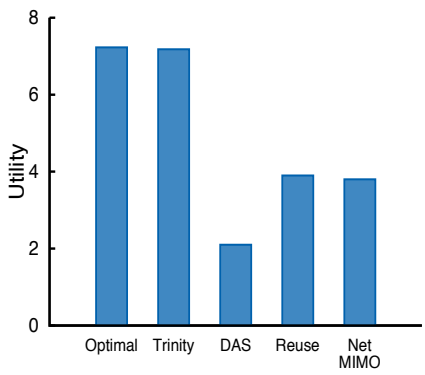


Figure 3.32: Performance of TRINITY: Utility

pooling the transmit power on only the active sub-carriers. Fig. 3.24, clearly shows the SNR gain as a function of the bandwidth utilized, where the SNR gain per sub-carrier can be as high as 4.5 dB when only a quarter of the bandwidth is utilized.

To understand the implication of this power pooling gain, we consider its impact on network rate compared to time domain multiplexing (TDM). In our experimental topology, when the reuse client is associated with AP1, the sub-carriers allocated for reuse at AP2 are un-used, whose power is redirected to those used to serve netMIMO and DAS users. Considering an equal split of sub-carriers between strategies, this results in about 10% gain in aggregate network rate compared to TDM in Fig. 3.25. When the DAS user is removed, its sub-carriers are re-allocated between netMIMO and reuse users, thereby allowing for more un-used sub-carriers (and their power) from reuse profile in AP2 to serve the netMIMO users. This further increases the rate gain to about 15%.

Increasing the number of APs, increases the possibility of more sub-carriers being left un-used at APs, since all APs may not be serving users from all profiles. To see if this contributes to increased network rate, we increase the number of APs within the interference range of the current two APs in our topology. Indeed, Fig. 3.26 shows that increasing the number of APs can leverage more power pooling gains from FDM and hence contribute to around 20% increase in network rate compared to TDM.

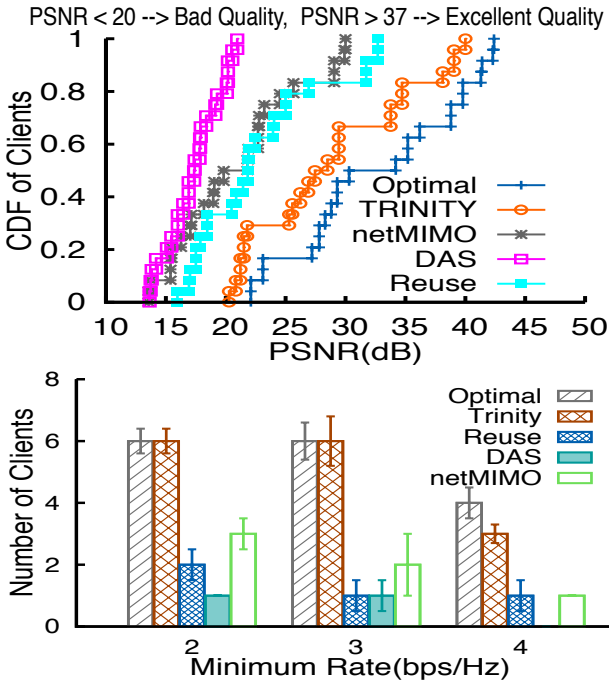


Figure 3.33: Quality of Service

3.7.2 Integrating TRINITY Components

We now evaluate TRINITY as a whole system with all its components and showcase how it caters effectively to users of various profiles.

Case studies in dynamic networks

We consider two case studies to highlight TRINITY's adaptation to network dynamics.

Moving to a lower multiplexing profile: The first experiment consists of 6 APs and 6 clients as shown in Figure 3.27. APs 1 and 6 do not interfere with other APs, whereas APs 2, 3, 4 and 5 are in a cluster, where they interfere with each other and can perform netMIMO or DAS. There are 3 static and 3 short-Tc clients at the start of the experiment. We depict TRINITY's operations and reactions on a time line as shown in Fig. 3.29. It can be seen that at the start of the experiment, TRINITY categorizes the clients accurately into different profiles - static clients are assigned to netMIMO, whereas short-Tc clients are assigned to reuse. TRINITY then runs its multiplexing algorithm to allocate these strategies on orthogonal frequencies as shown in Fig. 3.29 and starts transmissions.

APs 2, 3, 4 and 5 split their sub-carriers between netMIMO (served by APs 2,3,4,5) and reuse (served by AP 3) in the cluster. Hence, APs 2, 4 and 5 benefit from the power pooled from their un-used reuse sub-carriers.

At the 20th second, we change client 5's state from static to mobile by moving it at walking speed in the vicinity of APs 2, 3, 4 and 5 till the end of the experiment. The change in profile to a lower multiplexing one (from netMIMO) is identified automatically by reacting to netMIMO rate discrepancies in TRINITY. Hence, within a second, TRINITY re-categorizes client 5 accurately into the DAS profile and obtains the new sub-carrier mapping for the APs as shown in Fig. 3.29. To understand the impact on network rate during network dynamics, we compare TRINITY's rate against that of the optimal (exhaustive) solution, and is presented immediately after re-categorization as well as after 50s. It can be seen that the loss in network performance during the re-categorization process (latency to find the new profile) is small and gets further amortized when measured a longer time window.

Moving to a higher multiplexing profile: The second experiment consists of two clusters of APs, one with 2 APs (cluster 1) and other with 4 APs (cluster 2), as shown in Figure 3.28. There is one mobile client in each cluster, along with 2 short-Tc and 2 static clients in cluster 2.

Fig. 3.30 shows TRINITY's operations. At the beginning, TRINITY correctly categorizes client 1 under DAS (served jointly by APs 1 and 2), clients 3 and 5 are served under netMIMO from APs 4 and 5, and clients 2 and 6 are served under reuse from APs 3 and 6 respectively. The sub-carriers in this cluster are split between reuse and netMIMO, while they are all assigned to DAS in the other cluster. At the 21-st second, clients 4 and 6 both switch their state from mobile and short-Tc to static. Here, we do not consider the presence of sensor hints (only SNR and CSI). Hence, to detect the change in state to a higher multiplexing profile, TRINITY needs to wait for its periodic upgrade check. This happens every 5s (at 25s), where TRINITY upgrades its mobile and short-Tc clients to netMIMO and re-categorizes them, finding both of them to be in netMIMO profile. Since upgrading to a higher multiplexing state needs to be done actively, the latency involved results in a sub-optimal profile being used during that time and hence some loss in network rate (about 15-20%) when viewed over a short measurement as shown in Figure 3.30. However, this gets amortized when measured over a longer window.

Impact on Network Rate/Utility: To understand TRINITY's network-level performance, we compare it with four other schemes: optimal (exhaustive), netMIMO-

only, DAS-only and reuse-only. The latter three are baseline strategies that apply only one strategy to all users irrespective of their profile and is hence representative of state of the art schemes that do not differentiate between user profiles. Our evaluations include various network topologies with 6 APs and 6 clients in our testbed, similar to those in Figs. 3.28 and 3.27 but with a more comprehensive set of clustering patterns for the APs, namely topologies with (i) linear placement of APs (only neighboring APs interfere with each other's clients), (ii) two clusters of 3 APs each, (iii) three clusters of 2 APs each, (iv) clusters of size 1, 2 and 3 APs, and (v) a single cluster of 6 APs. In each topology, clients' locations and profiles are randomly configured, and the APs transmit back-to-back data for 180 seconds.

Figs. 3.31 and 3.32 show the average of the aggregate network rate and utility respectively that is achieved over all scenarios for each scheme. It can be seen that *TRINITY's performance is very close to that of optimal and outperforms other schemes by almost 50% in network rate and two folds in utility.* This clearly indicates the merit of catering transmission strategies to user profiles and the resulting potential of TRINITY.

Impact on Quality of Experience: Since we consider clients of various profiles, a given rate or throughput may not be valued the same by clients with different profiles. Hence, we also evaluate TRINITY in terms of client QoS satisfaction (using same set of topologies as previous experiment). To this end, we first consider a simple metric, wherein each client requires a minimum rate for its application for satisfactory experience and the goal is to see how many clients' requirements can be satisfied. Fig. 3.33 (bottom) shows the number of clients satisfied by a given scheme as a function of their increasing minimum rate requirement. We see that TRINITY is capable of satisfying as many clients as the optimal scheme in most of the cases. Further, *it can cater to thrice as many clients as the other individual strategies.* Next, we consider a standard encoded video of rate 1 Mbps with 30 frames per second for each client to understand the direct impact on application performance in Fig. 3.33 (top). Here, we see that *TRINITY is able to improve clients' video quality significantly by providing a large (median) gain of 10 dB in PSNR.*

Large-scale Trace-driven Simulations

Since it is hard to scale a testbed running netMIMO on WARP to larger topologies, we now evaluate the complete TRINITY solution along with its clustering and scheduling

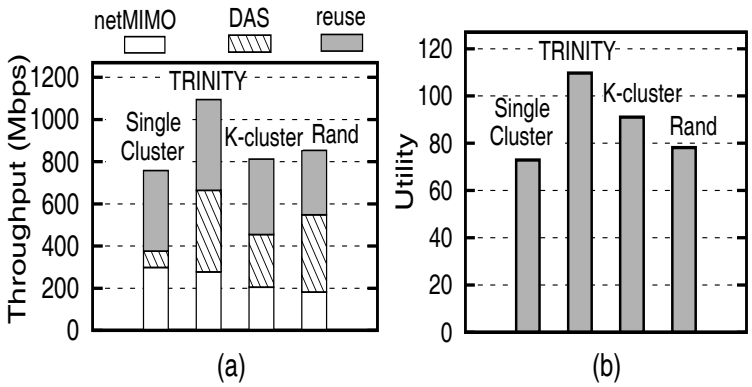


Figure 3.34: Trace-driven simulation results.

component using trace-driven simulations. To create traces for a realistic network topology, we move the WARP nodes to various locations of an office building, and collect the per-subcarrier channel state information (both magnitude and phase distortion) between nearby transmitter/receiver pairs. These WARP nodes together form a network of 20 APs and 30 clients. Each AP has 2 to 6 neighboring APs within its sensing range, and serves 1 to 3 clients within its communication range. Among the clients, there are 8 reuse, 10 netMIMO and 12 DAS users. The channel state information and interference relation between neighboring nodes are then fed into a simulator running TRINITY.

We evaluate TRINITY against its variants, where either its clustering or scheduling component is replaced by one of the following schemes: (i) Single Cluster, which aggregates all netMIMO or DAS APs into a single cluster; (ii) K-Cluster, which greedily clusters nearby K transmitters (K is fixed to 2 in our experiments); (iii) Rand, which replaces TRINITY's proportionally fair scheduler with a scheduler that allows an AP to randomly selects one client in each category and serve it. In TRINITY, the maximum cluster size is default to 4.

Figure. 3.34(a) shows that Single Cluster achieves the highest throughput performance for netMIMO users owing to its global synchronization and data sharing across all netMIMO APs. By splitting APs into small clusters of practical size, TRINITY suffers only a small loss in performance. This is because not all APs interfere with each other in a large network, and TRINITY can leverage spatial reuse between remote APs in different clusters. The K-cluster scheme does not account for the topology factor in its clustering process, *i.e.*, weak/strong interference between nearby APs, and thus it loses 26.1% throughput compared to TRINITY.

For DAS users, Single Cluster achieves the lowest performance, as it completely sacrifices spatial reuse for diversity, causing substantial throughput loss when there are multiple DAS users spread over a large area. TRINITY and K-cluster strike a good balance between diversity (needed for mobility) and spatial reuse. For reuse clients, all the schemes achieve a comparable level of performance. Finally, TRINITY's gain of 28.2% over Rand indicates the importance of scheduling, especially for netMIMO users, where scheduling is strongly coupled with mutual interference between netMIMO users.

Similar inferences can also be made with respect to the utility results in Figure. 3.34(b), where proportional fairness is targeted. These results clearly highlight the need for an efficient clustering and scheduling component in TRINITY.

3.8 Discussions

Complexity and Overhead: The careful design of individual components make TRINITY a light-weight framework for determining the appropriate transmission strategies for users. Its categorization module is seamlessly integrated with the transmission process and can operate even without sensor hints, namely with just SNR and rate estimates (from netMIMO clients) that are readily available. It requires only as much overhead as any closed-loop transmission scheme like beamforming or netMIMO. Further, by identifying the right set of netMIMO clients, it saves on the complexity and feedback overhead incurred in applying netMIMO generically to all clients. Also, by restricting netMIMO to clusters, its complexity and overhead is further reduced. TRINITY also employs simple but efficient algorithms for multiplexing, clustering and scheduling. The one component, where TRINITY incurs slightly more overhead is in its frame transmissions, where multiple clients are multiplexed (OFDMA) unlike in OFDM. However, as discussed in Section 3.5.4, such overhead is not specific to TRINITY and is considered to be a small price to pay for the frequency diversity and power pooling benefits delivered by OFDMA.

Rate Adaptation: Rate adaptation is performed once a transmission strategy is decided for a user. For netMIMO, transmission rates (rate adaptation) are implicitly determined as part of the pre-coding process once CSI is known. For Reuse and DAS, where CSI is not employed, any of the conventional rate adaptation schemes can be employed.

Scheduled vs. Random Access: While the Reuse strategy in TRINITY can employ random access distributed schemes as in 802.11, netMIMO and DAS by definition

would need synchronous operation among cooperating transmitters. However, by realizing them only within clusters, allows TRINITY to still leverage random access schemes (instead of scheduled) for inter-cluster medium access.

Uplink: TRINITY's downlink solution would apply to uplink as well. However, realizing OFDMA on the uplink requires sub-carrier level synchronization between clients being multiplexed simultaneously, which is challenging although do-able. Hence, TRINITY is currently limited to FDM only on downlink, while employing TDM on uplink.

Small Cell Networks: While TRINITY is presented in the context of enterprise wireless networks, it is equally applicable to upcoming, outdoor small cell (LTE, WiMAX) networks as well. Indeed, the problem of mobility is exacerbated in outdoor small-cell networks, wherein the potential benefits of TRINITY would be even more pronounced.

3.9 Conclusions

With the growing user (device) heterogeneity in wireless networks, we envision to identify and optimize transmission strategies that can cater to users of various profiles effectively. We believe such an approach is critical to improving user quality of experience in next generation wireless networks. Through real world experiments, we show that no single strategy can cater to all users and the best-performing strategy varies from one profile to another. Building on our insights, we present the design and evaluation of a practical system *TRINITY*, whose elements are geared towards addressing the challenges in realizing our vision.

Chapter 4

FluidNet: A Flexible Cloud-based Radio Access Network for Small Cells

4.1 Introduction

Mobile network operators are facing the pressure to increase the capacity and coverage of their radio access networks to meet the exponential growth in data traffic demand [54]. While leveraging the increased spatial reuse from smaller cells is a promising direction, every new cell adds to the capital and operational expenses borne by the operators. To address this problem, cloud-based radio access network (C-RAN) architectures have been considered by several operators [55] and service providers [56] as a cost-efficient way of realizing small cells. Unlike typical RANs where the baseband units (BBUs) and the radio units are situated together, the C-RAN concept (depicted in Fig. 4.1) migrates the BBUs to a datacenter (i.e., the BBU pool) hosting high performance general purpose and DSP processors, while providing high-bandwidth optical transport to the remote antennas called remote radio heads (RRHs). We define the high-bandwidth optical transport that carries the cellular signals between the BBUs and the RRHs to be the *front-haul* part of the network, whose bandwidth requirements could be significantly higher (tens of Gbps) than that of the backhaul depending on the nature of the signals (digital/analog, layer 1/2) carried [55]. The decoupling of the BBUs and radio units in a C-RAN allows for sophisti-

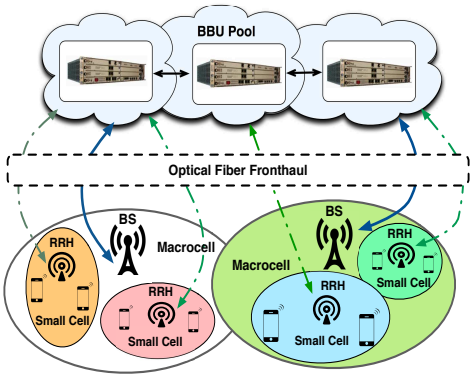


Figure 4.1: C-RAN Architecture.

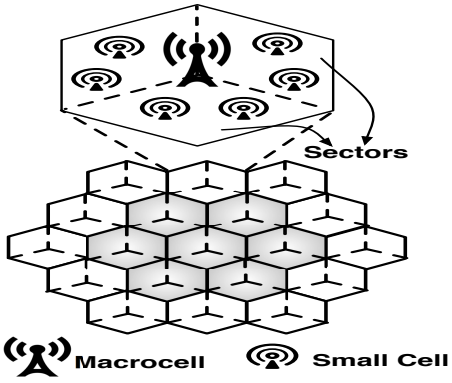


Figure 4.2: Network Deployment.

cated centralized techniques for interference management, where the BBUs in the pool can seamlessly cooperate to improve the RAN capacity. In addition, the deployment of radio units is made light-weight and can be realized in a fast and scalable manner for small cells (other benefits of C-RAN are detailed in [55]).

In this work, we argue that the front-haul that is unique to a C-RAN has a critical role in delivering its performance and cost benefits. We note that although the BBUs are decoupled from the RRHs in terms of physical placement, there exists a one-to-one logical mapping between BBUs and RRHs in that one BBU is assigned to generate (receive) a signal (e.g., LTE or WiMAX frame) to (from) an RRH (although the mapping can change over time). This one-to-one mapping allows for generating a distinct frame for each small cell (deployed in the form of a RRH), which is key for enhancing the network capacity via techniques such as dynamic fractional frequency reuse (dynamic FFR [57]) or coordinated multi-point transmissions (e.g., LTE CoMP [58]). We contend that this notion of a fixed, one-to-one mapping is not optimal in a practical cellular network deployment for two reasons.

RAN Performance: First, these techniques primarily apply to static users. The mobile users will have to bear frequent handoffs (exacerbated by smaller cells) and the associated performance penalties. In addition, tracking a mobile user's location and channel may be difficult for such techniques. In fact for mobile clients, a traditional distributed antenna system (DAS [59]) is arguably better suited. In a DAS setting, the same signal (carrying the user's data) is transmitted simultaneously by multiple small cells to provide

coverage benefits (which in turn reduces handoffs) and diversity gain. DAS can be realized by changing the one-to-one to a one-to-many logical mapping in the C-RAN front-haul.

BBU Energy Consumption: Second, the one-to-one mapping requires several BBUs to be active and generating frames, which consumes energy in the BBU pool. However, the enhanced capacity of techniques such as [57, 58] may not be needed in all parts of the network or at all times (e.g., 50% of cells carry 5% of net traffic [60]). When the traffic load is low in a region (e.g., coverage area of multiple small cell RRHs), a single BBU may suffice to serve the offered load (via a DAS mapping). This in turn reduces the number of BBUs and hence the compute resources (e.g., CPU cores, DSPs), thereby allowing energy savings in the BBU pool.

Given the above observations, we envision a C-RAN architecture with a novel, flexible front-haul that supports one-to-one as well as one-to-many logical mappings between BBUs and RRHs. Our vision is to utilize this architecture *to address the traffic needs of users (static and mobile) while leveraging the energy savings made possible by the traffic load heterogeneity (i.e., temporal and spatial load variations in the network).*

Towards realizing this vision, we present *FluidNet* - a flexible C-RAN system for small cells that houses an intelligent controller in the BBU pool, which dynamically reconfigures the front-haul (at coarse time scales) based on network feedback to cater effectively to both heterogeneous user and traffic profiles. This allows *FluidNet* to maximize the amount of traffic demand satisfied on the RAN for both static and mobile users, while at the same time optimizing the compute resource usage in the BBU pool. Briefly, *FluidNet* adopts a two-step, scalable approach: based on spatial traffic distribution and demand from users, *FluidNet* first determines the optimal combination of configurations (one-to-one and one-to-many, i.e., DAS and FFR strategies) needed to support the traffic demand from a set (termed *sector*) of small cells. Then, it employs a novel and efficient algorithm (with an approximation factor of $\frac{3}{2}$) to consolidate (cluster) the configurations of multiple sectors in the network to further reduce the compute resource usage without compromising on the traffic demand satisfied. *FluidNet* is both standards and technology agnostic. It allows for desirable features such as co-existence of multiple mobile operators and technologies (LTE, WiMAX, WiFi) in the same C-RAN, while employing different front-haul configurations tailored to each of their respective traffic.

We prototype *FluidNet* on a small-scale WiMAX C-RAN testbed with 6 BBUs and 6 RRHs, employing radio-over-fiber (RoF) as the front-haul. With *FluidNet* 's al-

gorithms, the logical BBU-RRH configurations are determined and executed on the fly. Real-world experiments with COTS WiMAX clients show that featuring flexible front-haul configurations and hence strategies, allows *FluidNet* to provide a 50% improvement in traffic demand satisfaction, while also reducing the compute resource usage in the BBU pool by 50% compared to baseline DAS and FFR strategies. Complementary, standards-calibrated (3GPP) simulations for large networks show that the clustering component in *FluidNet* helps further reduce the compute resource usage by 50% during low traffic load periods. Our contributions are as follows:

- We propose *FluidNet* - a light-weight, scalable framework to determine the optimal use of strategies (DAS, FFR) to cater to dynamic user and traffic profiles, while realizing them through appropriate configurations that help minimize compute resource usage in the BBU pool.
- We design efficient algorithms with performance guarantees in determining the appropriate configurations.
- We build a small-scale C-RAN system with 6 BBUs-RRHs; prototype *FluidNet* on it; and conduct over-the-air experiments, complemented by standards-calibrated large-scale simulations to demonstrate its feasibility and benefits.

4.2 Background

4.2.1 C-RAN Primer and Related Work

The C-RAN architecture, depicted in Fig. 4.1, includes three components: (i) remote radio heads (RRH), (ii) pool of baseband units (BBUs), and (iii) the front-haul (optical fiber based transport network).

RRHs: These are simple, light-weight radio units with antennas. Several proposals have focused on making RRHs power-efficient and scalable (e.g., [56, 61]) to support multiple bands and technologies (e.g., 3G, 4G).

BBU Pool: This helps migrate bulk of the base station (BS) processing of a large set of cells to a datacenter [55], allowing for easier realization of interference (e.g., CoMP [58, 62], HetNet [63]) and mobility management solutions.

On the energy front, [64, 65, 66] have looked at the benefits of switching off entire macrocell BSs based on prevailing traffic conditions. Moving the processing to a central entity in C-RAN allows for fine-grained use of resources in the pool and hence better energy savings (evaluated in Section 4.7). Further, these savings can be obtained without having to switch off an entire BS (allowing RRHs to be ON) and hence sacrificing performance or coverage.

For the BBU pool, there are several proposals for the use of heterogeneous platforms consisting of general-purpose processors as well as DSPs for compute-intensive baseband functions [67, 68]. Recently, [69] focused on assigning processor cores in a homogeneous platform to different BBUs in the pool, to meet latency requirements. Being complementary to [69], we focus on optimizing the use of BBUs themselves, which has an impact not only on compute resource usage in the BBU pool (especially in a heterogeneous platform) but also on RAN performance.

Front-haul: Optical fiber with wavelength multiplexing serves as the front-haul and distributes signals from the BBU pool to the RRHs either as (i) digitized radio signals over CPRI (common public radio interface) [70], or (ii) analog radio signals via radio-over-fiber (RoF) [52]. While CPRI is more robust than RoF over long distances, it requires more transport bandwidth. Optical front-haul is already used in several DAS deployments [59]. Recently, [71] articulated the need for a re-configurable front-haul in a C-RAN, but did not offer a solution. Our focus is to design and build a dynamically re-configurable front-haul along with the intelligence to adaptively determine the appropriate configurations.

4.2.2 Overview of Strategies

Fractional Frequency Reuse (FFR): FFR is the mechanism for radio resource management (RRM) in cellular networks, whereby inter-cell interference is addressed. Unlike WiFi, the synchronous operation of downlink (BS-MS) and uplink (MS-BS) transmissions across cells requires transmissions to be intelligently scheduled to manage interference. In the popular 1-3 FFR scheme for macrocell networks, the spectrum is divided into four fixed-size bands. One band is used by all the cell-interior clients (in each cell), who do not see interference due to the close proximity to their BS, while the other three bands are used (by cell-exterior clients) in an orthogonal manner between the three sectors (Fig. 4.2) of a cell to mitigate interference with sectors of adjacent cells. Thus, while the band used

by cell-interior clients is reused in each cell, the reuse of the other three bands are subject to the spatial reuse possible. Recently, dynamic FFR approaches [57] have been proposed specifically for small cells, and determine the number and size of bands to be used by each small cell only based on the aggregate traffic demand from its cell-interior and cell-exterior clients; they allow for better spectral utilization and do not rely on planned sectorization (unlike macrocells). Note that the FFR schemes only determine the set of spectral resources assigned to cells - scheduling of clients within those resources is done by each cell locally (based on per-client feedback) to leverage multi-user diversity.

We adopt [57] for FFR in *FluidNet*, although other FFR schemes can also be easily used. While point-to-point MIMO is automatically incorporated in FFR, other cooperative techniques such as multi-user MIMO and co-ordinated multi-point transmissions (CoMP) can also be applied under FFR.

Distributed Antenna Systems (DAS): In DAS, a common signal from a single source is delivered to multiple RRHs and transmitted simultaneously. This provides larger coverage and has been adopted by operators for both indoors and outdoors [59]. Unlike FFR that is focused on capacity, increasing the foot-print of the signal across multiple transmit points (small cells) under-utilizes the spectrum in DAS without scope for any spectral reuse.

Relation between Strategies and Configurations: Since interfering cells will be operating on potentially different spectral bands in FFR, different frames (with specific preamble, control, etc.) have to be generated for each cell, thereby requiring a one-to-one logical mapping (configuration) between a BBU and an RRH. This is the conventional mapping considered in C-RAN currently. However, in DAS, a single frame is transmitted by multiple RRHs, which in turn can be accomplished using a single BBU, thereby requiring a one-to-many mapping.

4.3 Motivation and Model

4.3.1 Motivation for a Reconfigurable Front-haul

With the help of a simple experiment conducted on a WiMAX C-RAN testbed (details in Section 4.6), we now motivate why a one-to-one signal mapping between BBUs

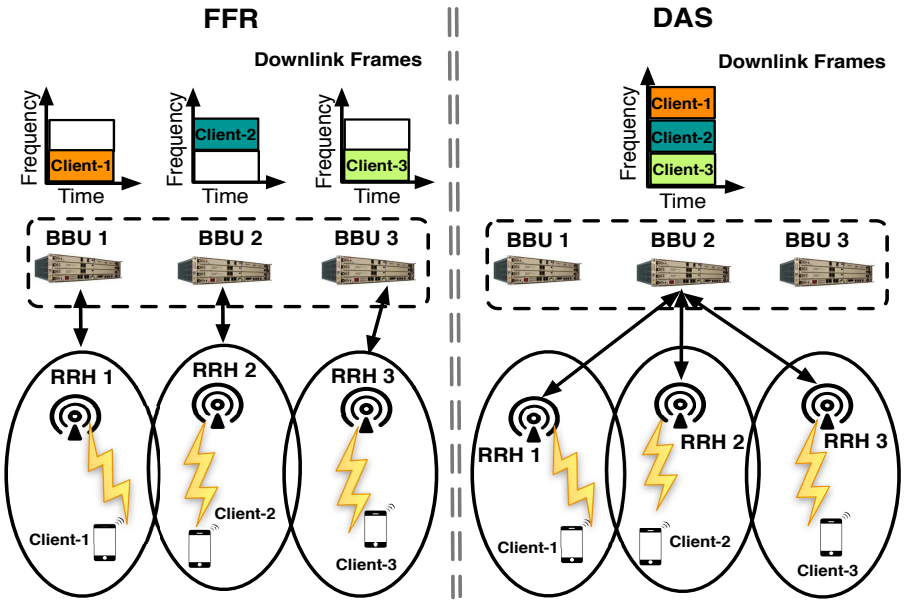
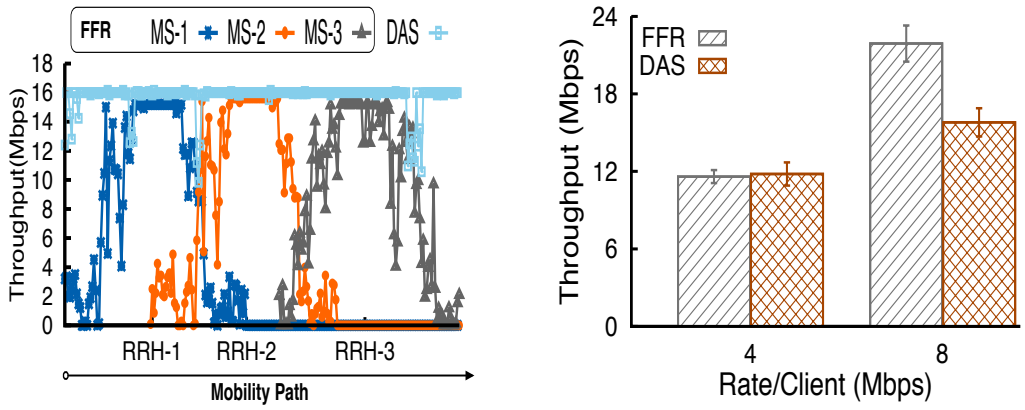


Figure 4.3: 3 BBU-RRH setup for DAS vs FFR.

and RRHs is highly sub-optimal. Consider a system with 3 BBUs and 3 RRHs, serving three clients as shown in Fig. 4.3, where each RRH interferes with its neighbor's client.

1. **Traffic Heterogeneity:** Consider a scenario, where the clients are static, but their data rate varies (see Fig. 4.4(b)). When the total rate (e.g., 8 Mbps per client = 24 Mbps) exceeds the max. data rate supported by all the sub-channels in one frame ($\approx$ 16 Mbps in our testbed), the increased capacity with FFR (by reusing orthogonal half of sub-channels as in Fig. 4.3) is essential to meet the traffic demand, while DAS is limited to one frame's capacity of 16 Mbps. On the other hand, at low load (e.g., 4 Mbps per client), DAS's capacity is sufficient to serve the clients with just one BBU, allowing the other two BBUs to be off. This is unlike in FFR, where all the BBUs have to be active to generate different frames to the RRHs; it lowers compute resource usage and thus enables significant energy savings in the BBU pool.

2. **User Heterogeneity:** Now, let all the three clients be mobile, moving between the 3 RRHs. Catering to the mobile clients through dynamic FFR from individual RRHs is very challenging for multiple reasons - (a) with small cells, there are frequent handoffs, whose associated latency has an adverse impact on throughput, (b) there is increased signaling load on the front-haul and the mobile core network due to frequent



(a) (b)

Figure 4.4: (a) Effect of mobility. (b) Effect of traffic load.

handoffs, (c) it may be hard to track the mobile user to specific small cells to efficiently apply dynamic FFR and leverage reuse. Even notwithstanding such drawbacks and assuming ideal handoffs for FFR, Fig. 4.4(a) clearly shows DAS' ability to deliver consistent coverage and performance for mobile users. Note that while relegating the mobile user to the macrocell is one option, DAS is ideal for such mobile users, since it achieves a coverage similar to a macrocell, while also increasing the link capacity (through shorter links and diversity gain, see Fig. 4.3).

Thus, while dynamic FFR is best suited for static users in high traffic load conditions, employing DAS also has benefits both from RAN performance (for mobile traffic) as well as BBU resource usage (for low traffic) perspectives. Given this, it is imperative for the front-haul to be re-configurable to realize flexible combinations of one-to-one and one-to-many BBU-RRH mappings.

4.3.2 Problem Definition

Network Model

Given that small cells have to co-exist with macrocells, we consider a large number of small cells to be deployed as an under-lay to an operator's macrocell network (Fig. 4.2). Since a macrocell will interfere with the small cells, thereby limiting their spatial reuse ability, two kinds of RRM solutions have been considered in literature: (i) macrocells and small cells operate on different carrier frequencies, and (ii) both use the same frequency

but orthogonalize their radio resources at coarse time scales. We consider the former model and focus entirely on the downlink operation of small cells for ease of exposition. However, our solutions are equally applicable to the latter model and to uplink as well. Further, while small cells themselves could be deployed in an un-planned manner, we leverage the over-lay of macrocells and borrow the notion of logical sectors (from macrocells) to refer to the location of small cells.

Objective

Recall from Section 4.3.1 that while FFR (one-to-one configuration) supports the maximum amount of traffic through reuse, it does not save on computing resources in the BBU pool. On the other hand, while DAS (one-to-many configuration) minimizes the resource usage and caters to mobile traffic, it under-utilizes the spectrum. By appropriately employing FFR and DAS in combination in different parts of the network, *FluidNet* 's goal is to strike a fine balance between them. Specifically, subject to the primary requirement of supporting as much traffic (D) as the optimal configuration (D_{OPT}), *FluidNet* strives to minimize the corresponding amount of compute resources needed in the BBU pool (resource usage RU, defined in Sec. 4.4) for the purpose.

$$\min_{\Gamma} RU_{\Gamma}, \text{ subject to } D \geq \lambda \cdot D_{OPT} \tag{4.1}$$

where Γ represents a possible configuration, and λ is the fraction of (optimum) traffic demand that must be satisfied (e.g., $\lambda = 0.99$). The optimal configuration would depend on the relative composition of mobile and static traffic and their priorities ($D_{OPT} = D_{FFR}$ when there is only static traffic demand). We assume mobile traffic to be prioritized over static traffic, albeit other models are also possible. Also note that minimization of compute resource consumption is only subject to satisfying as much of the traffic demand as possible and does not come at the expense of the latter.

BBU Usage as a resource metric: The main components of energy consumption in a traditional base station (BS) are those of air conditioning ($\approx$ 2 KW) and the BS equipment itself ($\approx$ 0.7 KW) [55]. A C-RAN system helps towards both these components by not only simplifying the cell site to a RRH (eliminating the need for air conditioning), but also consolidating the BS processing in the BBU pool. With respect to the latter component, reducing the number of BBU units and hence the frames that need to be processed, has a

direct impact on energy consumption for two reasons. (1) BBU processing involves layer 1 (framing, FFT/IFFT, decoding, etc.), layer 2 (HARQ, resource/QoS scheduling, etc.) and layer 3 (connection management) functions. While layer 3 and part of layer 2 can be handled by generic processors, some of the time-sensitive layer 2 (resource scheduling) and layer 1 (framing, FFT/IFFT, decoding) functions are typically handled by dedicated DSPs for each BBU. (2) When DAS is employed, the traffic demand of multiple cells is handled without any spectral reuse. Hence, while the (traffic) load-dependent processing component is limited to that needed to handle the total number of slots (e.g., resource blocks in LTE) in a single frame, the basic processing component (FFT/IFFT) scales with the number of cells (frames) and soon dominates the former (see [69] for realistic values). Note that optimizing the BBU usage is complementary to assigning compute resources (e.g., GPPs) to the BBUs themselves, for which solutions such as [69] can be leveraged.

4.4 Design Elements in FluidNet

We motivate *FluidNet* 's design by addressing key aspects relevant to the operation of transmission strategies and its impact on the compute resource usage in the BBU pool.

4.4.1 Granularity and Choice of Configurations

A strategy (configuration) is applied to a set of small cells. In macrocells, each sector has its own cell ID and is the smallest granularity for RRM operations. Given this, *FluidNet* adopts *sector* (referring to set of small cells located within the logical sector) to be the minimum granularity for configurations.

Depending on the user and traffic profiles in a sector, one has to determine the appropriate transmission strategy: DAS or FFR. However, picking either DAS or FFR in *isolation* often results in in-sufficient or spare spectral resources respectively, in handling the offered traffic load. Hence, *FluidNet* employs a flexible *combination* of DAS and FFR (called *hybrid* configurations) in each sector. It devotes the right fraction of spectral resources between the two configurations, thereby supporting the offered traffic load with the least possible use of BBU resources.

4.4.2 Realization of Hybrid Configurations

Since two configurations cannot co-exist in the same time-frequency resource, hybrid configurations have to be multiplexed either in time or frequency. If multiplexed in time, a hybrid configuration can be realized at the granularity of an epoch spanning several super-frames (10 ms each in LTE), where a contiguous subset of the sub-frames (1 ms each) operate in a DAS configuration, while the rest operate in FFR. If multiplexed in frequency, the operator's spectrum can be divided into coarse spectral blocks (separate carriers in a multi-carrier scenario such as LTE-advanced; e.g., similar to orthogonal channels in WiFi), which are then split between the two configurations (see Fig.4.5). The fraction of carriers allocated to the configurations is such that the traffic load is satisfied with the least possible use of BBU resources. Since a DAS configuration minimizes the use of BBU resources but supports the least amount of traffic, this is equivalent to finding the largest allocation to the DAS configuration that is capable of sustaining the offered load.

Note that, frequency-multiplexing allows appropriate number of BBU resources to be assigned to each carrier (based on the configuration using it), which do not have to be changed unless the hybrid configuration itself is updated (which happens at coarse time scales; order of minutes). This is unlike time-multiplexing, where the assignment of BBU resources has to be re-mapped even within a hybrid configuration, i.e. switches between DAS and FFR (granularity of super-frames - tens of ms). Although feasible, the time scales of the latter may limit the potential for resource and energy savings in the BBU pool. Hence, *FluidNet* adopts multiplexing configurations in the frequency domain.

4.4.3 Clustering for Reduced Resource Usage

In regions of the network with low traffic load, it is possible to support the traffic demand from multiple sectors jointly with a single DAS configuration. While aggregating such sectors reduces the compute resource usage in the BBU pool, it must be done in a scalable manner. *FluidNet* proposes a novel clustering mechanism for this purpose.

To capture the BBU resource usage for a hybrid configuration in a sector, we define the resource usage metric, RU:

$$RU(b_i, n_i) = b_i \cdot 1 + (B - b_i) \cdot n_i \tag{4.2}$$

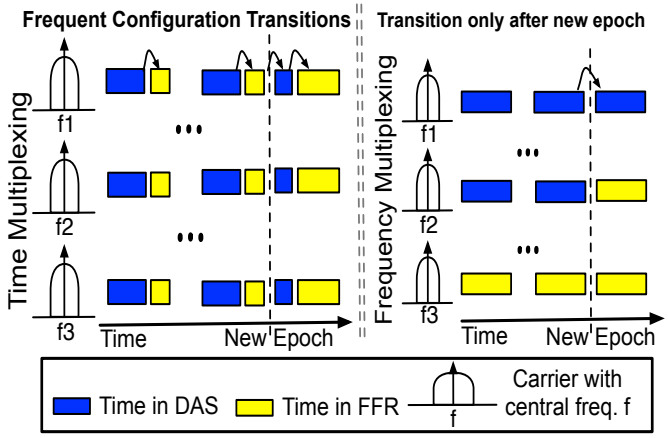


Figure 4.5: Realizing Hybrid Configurations.

where, n_i is the number of small cells in sector i and b_i, the number of carriers (out of B total) allocated to its DAS configuration. In every carrier, the number of BBU units needed for DAS is one, while it is equal to the number of small cells (n) for FFR. Thus, *RU* captures the effective number of BBU units needed to support the offered load on the given spectral resources (OFDMA resources in B carriers).

Using the RU metric, *FluidNet* employs a scalable algorithm (details in Section 4.5) that clusters two neighboring sectors (i and j) at a time, until either their net offered load cannot be supported or the RU of the resulting cluster ($i \cup j$) cannot be improved, i.e.,

$$RU(b_{i\cup j}, n_i + n_j) > RU(b_i, n_i) + RU(b_j, n_j) \tag{4.3}$$

where $b_{i\cup j}$ captures the new split of carriers between DAS and FFR in the cluster). While applying DAS to serve user traffic on $b_{i\cup j}$ resources is straight-forward (shared between users without any reuse), dynamic FFR now has to be applied on $B - b_{i\cup j}$ carriers for a larger number of cells ($n_i + n_j$). The latter, being a non-trivial RRM process, could become computationally intensive as the size of the cluster increases. Hence, for large clusters, *FluidNet* can run its FFR solution seperately in each cluster's constituent sectors (for scalability), albeit on the same set of $B - b_{i\cup j}$ carriers.

4.4.4 Handling User Mobility

So far we had assumed that the offered traffic load in a sector or cluster can be scheduled on any of the carriers operating on either DAS or FFR. Recall that for mobile (mainly vehicular) users, a DAS configuration is essential not just for reducing compute resource usage but even for performance. Identifying such mobile users can be done in many ways (e.g., mobile operator maintains user's mobility state). Then the offered traffic load from vehicular users can be isolated from the rest of the traffic and scheduled on resources supporting the DAS configuration. Hence, the net traffic load from mobile users in a sector or cluster would place a constraint on the minimum number of carriers that need to be allocated to its DAS configuration. Subject to this constraint, the rest of the operations (resource allocation, multiplexing, clustering, etc.) are performed as mentioned above.

4.4.5 Handling Interference across Sectors

Since FFR is executed at the sector granularity for scalability, interference is managed only between cells within a sector. The conventional (simple) solution to handle interference across sectors (or clusters) is to consider all external interference as noise. A more sophisticated approach is to make implicit provisions in the transmission strategy of a sector for alleviating interference across sectors (and hence clusters) without any coordination. Recall that, in a carrier allocated to FFR, only a subset of the sub-channels (called resource blocks in LTE) are used by any of the cells in the sector to account for intra-sector interference (e.g., cells 1 and 2 in Fig. 4.6(a)). When coordination across sectors is allowed, these sub-channels would be further chosen so as to avoid interference between sectors. However, in the absence of any coordination (for scalability), the sub-carriers constituting the sub-channels in the carrier can be *permuted differently* across sectors. While this does not provide the same performance as performing FFR over the interfering sectors jointly, it does provide an interference averaging (alleviating) effect (cells 1 and 3 in Fig. 4.6(a)). Note that, this is not possible when operating in DAS, where all sub-channels in the carrier are used in every interfering sector.

FluidNet determines the sector-exterior traffic that is prone to interference from neighboring sectors and operates it in an FFR configuration to alleviate interference. Hence,

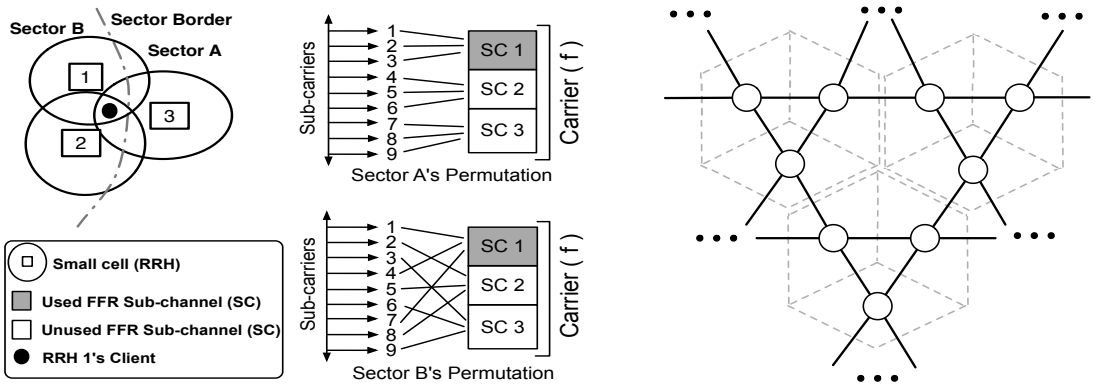


Figure 4.6: (a) Handling Inter-sector Interference. (b) Sector graph.

similar to the minimum set of carriers needed for DAS (for mobile traffic), *FluidNet* reserves a minimum set of carriers for FFR to accommodate sector-exterior traffic.

4.5 Algorithms in FluidNet

4.5.1 Overview of Solution

The sequence of operations in *FluidNet* for every epoch (spanning several minutes) is as follows.

Step 1: For every sector, obtain the aggregate traffic demand (over the previous epoch) from each of its small cells. Determine the minimum set of carriers needed for the DAS and FFR configurations based on traffic demand from mobile and sector-exterior traffic respectively.

Step 2: Determine the optimal multiplexing (in frequency) of DAS and FFR configurations for each sector. This would automatically classify the appropriate traffic that needs to be scheduled on a particular configuration. Based on the resulting allocation of carriers to the configurations, determine the RU metric for the sector.

Step 3: Cluster sectors two at a time based on their RU metric until either their net offered load cannot be supported or the RU of the resulting cluster cannot be improved.

Step 4: For each cell in the cluster, apply the configurations on their allocated carriers as

determined by the cluster's RU metric and assign respective traffic to carriers allocated to their appropriate configurations.

We now describe each of the steps in detail.

4.5.2 Estimation of Radio Resource Demand

Each small cell maintains an estimate of the aggregate traffic demand from its users in the current epoch (of length T s). Given a traffic demand ($d_{c,u}$ in bits) from a user u in cell c, this is translated to the corresponding radio resource demand per sub-frame (i.e. OFDMA resource slots/ms). For this, the average MCS (modulation and coding rate, $\ell_{c,u}$) used to serve the user in the epoch is kept track of and used to obtain the radio resource demand per sub-frame as $r_{c,u} = \frac{d_{c,u}}{T \cdot 1000 \cdot \ell_{c,u}}$ slots. Each cell (c) classifies its net user traffic demand d_c as either mobile or non-mobile. The non-mobile category is further classified as cell-exterior or cell-interior traffic (for FFR purposes) based on presence or absence of interference respectively from neighboring small cells. At the end of the epoch, every cell (c) then provides 3 parameters as input to the central controller: aggregate radio resource demand from mobile ($d_{c,mob} = \sum_{i \in mob} r_{c,i}$), cell-interior ($d_{c,int} = \sum_{i \in int} r_{c,i}$) and cell-exterior ($d_{c,ext} = \sum_{i \in ext} r_{c,i}$) traffic. Note that with centralized processing in C-RAN, there is no associated feedback overhead in providing this information.

Each sector (j) then further aggregates the radio resource demands from mobile traffic in each of its small cells ($D_{j,mob} = \sum_{c \in j} d_{c,mob}$). The minimum radio resource demand needed for its DAS configuration is then the smallest number of carriers needed to satisfy the net mobile traffic demand, i.e, $b_{DAS} = \min_{b \cdot M \geq D_{j,mob}} b$, where M is the number of OFDMA resource slots on each carrier. Similarly, to determine the minimum radio resource demand for FFR, it aggregates the cell-exterior traffic from all its small cells that are on the edge of the sector ($D_{j,ext} = \sum_{c \in edge(j)} d_{c,ext}$), scales them by $\alpha = 0.25$, and obtains $b_{FFR} = \min_{b \cdot M \geq \alpha D_{j,ext}} b$. Note that approximately only half of the cell-exterior traffic of the sector-edge cells will be vulnerable to other small cells from neighboring sectors. Further, every alternate sector-edge cell would be able to reuse the radio resources in the sector. Accounting for both these aspects, reduces the radio resource demand approximately by a quarter that is captured by α.

Remarks: (1) Since *aggregate* traffic demand from a sector of small cells changes slowly at coarse time scales, determining configurations for every epoch (order of minutes) based on

the aggregate demand is both appropriate and robust. For the same reason, it also suffices to estimate the *approximate* radio resource demand from sector exterior traffic. (2) *FluidNet* requires only one parameter (mobile traffic demand) from each small cell in addition to those already required by FFR schemes (i.e. cell-interior and cell-exterior traffic demands). However, one can eliminate the former and simplify *FluidNet* 's design by not catering to mobile and sector-exterior traffic separately (i.e. $b_{DAS} = b_{FFR} = 0$).

4.5.3 Optimal Sector Configuration

With the estimates of aggregate radio resource demands, *FluidNet* determines the optimal split of carriers between DAS and FFR configurations in a sector (j) as follows. With b_{DAS} and b_{FFR} serving as the minimum number of carriers needed for the DAS and FFR configurations, *FluidNet* uses an iterative approach (Algorithm 4) to determine the optimal split $(b_j, B - b_j)$ by starting with b_{FFR} as the minimum set of carriers needed for FFR and allowing it to expand till the radio resource demand can be satisfied or if the limit of $B - b_{DAS}$ carriers is reached. Since mobile and sector-exterior traffic demands are already accounted for, to check if net radio resource demand can be met, *FluidNet* essentially needs to check only if the remaining resource demand $(\sum_{c \notin edge(j)} d_{c,ext} + \sum_{c \in j} d_{c,int})$ can be accommodated by the current split (say $b, B - b$) in the iteration, with $b - b_{DAS}$ and $B - b - b_{FFR}$ carriers in DAS and FFR configurations respectively. Note that this would involve running an FFR scheme on $B - b - b_{FFR}$ carriers first (step 3), wherein to maximize the amount of traffic demand satisfied through FFR, the cell-interior traffic $(\sum_{c \in j} d_{c,int})$ that provides maximum spatial reuse is assigned to FFR prior to the cell-exterior traffic $(\sum_{c \notin edge(j)} d_{c,ext})$. The remaining traffic demand $(D - D_{FFR})$ is then scheduled through DAS on the $b - b_{DAS}$ carriers (step 4).

If the total number of carriers B is small, then a simple, sequential iteration (with increments of one carrier) would suffice. However, if B is large, then the FFR operation in each iteration could be computationally expensive. In this case, *FluidNet* employs binary search, where the split is moved to the left if the traffic demand cannot be met (steps 8-9), and moved to the right if spare resource slots (f_{DAS}, normalized to total # resource slots in a carrier M) remain in DAS configuration after demand satisfaction (steps 5-6). It converges at the split (say b_j), where the number of carriers allocated to FFR cannot be further reduced, while still satisfying the demand. Binary search reduces the number of

iterations and hence FFR operations from linear ($O(B)$) to logarithmic ($O(\log_2(B))$). After convergence, the RU of the sector is computed using Eqn. 4.2 as $RU(b_j, n_j)$.

Algorithm 4 Optimal Configuration for Sector j

Initialize $b_{low} = b_{DAS}$, $b_{high} = B - b_{FFR}$, $b = b_{high}$, $D = \sum_{c \notin edge(j)} d_{c,ext} + \sum_{c \in j} d_{c,int}$
while $b_{high} \neq b_{low}$ **do**
  $(f_{FFR}, D_{FFR}) =$ Schedule_FFR$(B - b_{FFR} - b, D)$
  $(f_{DAS}, D_{DAS}) =$ Schedule_DAS$(b - b_{DAS}, D - D_{FFR})$
  if $f_{DAS} > 0$ **then**
    $b_{low} \leftarrow b$; $b \leftarrow \frac{b + b_{high}}{2}$; $b_{cur} \leftarrow b$
  else
    if $D - D_{FFR} - D_{DAS} > 0$ **then**
      $b_{high} \leftarrow b$; $b \leftarrow \frac{b + b_{low}}{2}$
    end if
  end if
end while
$b_j \leftarrow b_{cur}$

In addition to RU, every sector keeps track of two metrics: spare radio resources (β_j) and reuse factor (r_j) in the sector (for use in clustering). Note that since minimum set of carriers are determined for FFR configuration, spare resource slots, if any, will appear only in the DAS configuration. This is normalized to the total number of slots (M) in each carrier to yield β_j. Similarly, reuse factor determines the number of actual resource slots needed to support the traffic demand in the sector (and captures the average reuse resulting from FFR): $r_j = \frac{\sum_{c \in j} d_{c,mob} + d_{c,ext} + d_{c,int}}{(B - \beta_j)M}$.

Theorem 1. FluidNet *'s iterative scheme converges to the optimal split of carriers between FFR and DAS configurations in each sector w.r.t. the objective in Eqn. 4.1.*

Proof. Recall that any FFR scheme operates in two segments: one, where all cells in the sector reuse the radio resources and the other, where cells share the radio resources (similar to DAS). Thus, due to the nature of the strategies, if y and z resource slots are needed to satisfy x units of traffic in the DAS and FFR configurations respectively, then we have $y \geq z$ always. Given this, it follows that there exists a unique optimal split, where traffic demand (assuming feasible) is satisfied with the least amount of FFR radio resources. In other words, any split to the left would result in spare radio resources in DAS, while any split to the right would not satisfy the traffic demand. Using this inference in its iteration, allows *FluidNet* to converge to the optimal split.

4.5.4 Properties of RU Metric

We present properties of the RU metric that are relevant for clustering. For ease of exposition, we do not consider mobile traffic in the discussions.

Property 4.5.1. *When two sectors i, j are clustered, the split of carriers in the resulting cluster has to be the minimum of those in the constituent sectors ($b_{i\cup j} = \min\{b_i, b_j\}$) to maximize RU.*

Proof. Let $b_i \leq b_j$ without loss of generality. Now, $b_{i\cup j}$ cannot be greater than b_i since this would require some FFR traffic in sector i to be moved to the DAS configuration. However, since b_i was optimal in sector i to begin with, this would in turn require more radio resources in the DAS configuration that what is currently available, and is hence not possible.

On the other hand, if $b_{i\cup j} = b_i - \delta$, then

$$\begin{aligned} RU(b_i - \delta, n_i + n_j) &= (b_i - \delta) + (B - b_i + \delta)(n_i + n_j) \\ \text{while, } \; RU(b_i, n_i) + RU(b_j, n_j) &= b_i + (B - b_i)n_i + b_j \\ &\quad +(B - b_j)n_j \end{aligned} \tag{4.4}$$

Now, the improvement (reduction) in RU is,

$$\begin{aligned} \Delta &= RU(b_i, n_i) + RU(b_j, n_j) - RU(b_i - \delta, n_i + n_j) \\ &= b_j + (B - b_j)n_j - \delta(n_i + n_j - 1) \end{aligned} \tag{4.5}$$

Since Δ is maximum when $\delta = 0$, we have the desired result.

Property 4.5.2. *RU metric does not satisfy the "local" property, i.e. if clustering sectors i, j, k improves the RU, then this does not mean that clustering a subset of its constituent sectors also improves RU.*

$$\begin{aligned} RU(b_{i\cup j\cup k}, n_i + n_j + n_k) &\leq \sum_{\ell=\{i,j,k\}} RU(b_\ell, n_\ell) \\ ! \Rightarrow RU(b_{i\cup j}, n_i + n_j) &\leq RU(b_i, n_i) + RU(b_j, n_j) \end{aligned}$$

Proof. It is easy to create a simple example that satisfies the above inequalities. Consider the optimal split of carriers for the 3 sectors i, j, k as $b, b+\delta, b+2\delta$ respectively and let the number of cells in each of them be n. Now, if δ is chosen such that $\frac{b}{2(n-1)} \leq \delta \leq \frac{2b}{3(n-1)}$, and applying property 4.5.1, the above statement can be easily shown to be satisfied.

Property 4.5.3. *To cluster sectors i and j (with say $b_i \leq b_j$), we need all of the following to be satisfied.*

1. *Both sectors must have spare radio resources in the DAS configuration, i.e. $\beta_i < b_i$ and $\beta_j < b_j$.*
2. *The aggregate traffic from the DAS and FFR configurations of the two sectors must be satisfied by the new split of carriers in the cluster. Equivalently, $b_j - r_j(b_j - b_i) \leq \beta_i + \beta_j$.*
3. *The RU of the resulting cluster must be improved. Equivalently, $b_j \leq \frac{n_j}{n_j - 1} b_i$.*

Proof. When sectors are clustered, the main change is a *common* split of carriers ($\min\{b_i, b_j\}$), whose radio resources must satisfy the traffic from constituent sectors. Since FFR traffic from sectors already account for inter-sector interference, they can continue to leverage reuse within their respective sectors and hence easily be accommodated. However, the DAS traffic from the sectors have to be merged and share radio resources in the new DAS configuration. The first and second constraints in this property provide the necessary and sufficient conditions respectively towards ensuring this. With the common split being $\min\{b_i, b_j\}$, it is easy to see that to accommodate more DAS traffic from either of the sectors, it is necessary for both the sectors to have spare DAS radio resources to begin with. However, to ensure that the entire DAS traffic from both the sectors can be accommodated, we need $(b_i - \beta_i) + (b_j - \beta_j) \leq b_i$. When the sector with the larger DAS split (b_j) reduces to the smaller split of b_i in the cluster, this would provide more radio resources ($b_j - b_i$ carriers) towards FFR in sector j. This can in turn be used to offload some of its DAS traffic. Also, due to reuse in FFR, for a given set of radio resources in FFR, more resource demand ($(b_j - b_i)r_j$, i.e. scaled by reuse factor r_j) can be offloaded from the DAS configuration. Hence, it suffices if,

$$\begin{aligned} (b_i - \beta_i) + (b_j - \beta_j) - (b_j - b_i)r_j &\leq b_i \\ \Rightarrow b_j - r_j(b_j - b_i) &\leq \beta_i + \beta_j \end{aligned} \tag{4.6}$$

In addition to traffic demand satisfaction, we need to ensure that the resulting RU of the cluster is improved, i.e.

$$\begin{aligned} RU(\min\{b_i, b_j\}, n_i + n_j) &\leq RU(b_i, n_i) + RU(b_j, n_j) \\ \text{This reduces to, } b_j &\leq \frac{n_j}{n_j - 1} b_i \end{aligned} \tag{4.7}$$

4.5.5 Clustering of Sectors

Based on the above established properties, *FluidNet* designs a light-weight clustering algorithm (Algorithm 5) to improve the RU of configurations applied in the network. Representing as a graph $G = (V, E)$, each sector forms a vertex in the graph, while an edge $e = (u, v)$ exists between two vertices (u and v) if the corresponding sectors are adjacent (Step 1). Each edge e carries a weight (w_e), which evaluates property 4.5.3 in identifying if the corresponding sectors u and v can be clustered, and if so assigns the resulting RU of the cluster as its weight ($w_e = RU(\min\{b_u, b_v\}, n_u + n_v)$). If however, clustering is not feasible, then this is denoted by $w_e = \infty$ (Step 2).

Algorithm 5 Clustering of Sectors

1: Construct Sector Graph: $G = (V, E)$, $V = \{\text{sectors}\}$, $E = \{e = (u, v)\} : v = N(u)$
2: $w_e = RU(\min\{b_u, b_v\}, n_u + n_v)$ if Property 4.5.3 is satisfied; and $w_e = \infty$ otherwise
3: Let $G' = (V', E')$; initialize $V' = V$, $E' = E$
4: **while** (1) **do**
5: Pick $u = Rand(V')$
6: Select $v^* = \arg\min_{v:e=(u,v)\in E'} w_e$
7: **if** $v^* \neq \emptyset$ **then**
8: Contract (u, v^*) in V', i.e. $(u, v^*) \rightarrow u'$
9: Add edges in E', $(u', v) : (u, v) \in E$ or $(v^*, v) \in E$
10: Update edge weights in E' $w_{e'}$, $\forall e' = (u', v) : v \in N(u')$ &$v \in V'$
11: **else**
12: Exit
13: **end if**
14: **end while**
15: Output clustered graph $G' = (V', E')$

With the above weighted graph, *FluidNet* clusters sectors through a graph coarsening approach. At each step, it picks a random vertex u (Step 5), then selects the

neighboring vertex v (Step 6) that when clustered together minimizes the resulting RU ($v = \arg\min_{e \in E'} w_e$, where $e = (u, v)$). It then contracts u and v, along with edges between them to a new clustered node u' (Steps 7-9). Weights of edges incident on u and v are updated after the contraction (Step 10). The process is repeated until no more clustering is possible. Each vertex in the final graph ($v \in V'$) represents the clustering of sectors in the network for improved RU (Step 15). Further, the RU of each clustered node, represents the common split of carriers between the DAS and FFR configurations for all sectors in that cluster.

Recall that RU does not satisfy the local property (property 4.5.2). Hence, while local clustering schemes are light-weight and scalable, they might miss out on potential clusters that improve the RU. To reduce the impact of such sub-optimality, *FluidNet* leverages the structure of the sector graph as follows. The logical 3-sector operation of macrocell networks results in a graph that has only cliques of size 3 and cycles of size 6 (see Fig.4.6(b)). This special form of G is called a "sector graph". Hence, *FluidNet* includes the following optimization, where in addition to computing the weight of each edge, it also computes the weight of each clique ($w_{(u,v,w)} = RU(\min\{b_u, b_v, b_w\}, n_u + n_v + n_w)$). Hence, it first starts contracting (clustering) all possible cliques ($\frac{|V|}{3}$ in number) before moving to the contraction of edges. This would help improve RU from potential 3-sector (clique) clusters, which would not otherwise result from their constituent 2-sector (edge) clusters.

As with most clustering problems, it can be shown that the problem of finding the network-wide configuration with the smallest RU is NP-hard. We have the following performance guarantee for *FluidNet* .

Theorem 2. FluidNet *'s algorithms yield network-wide transmission configurations with a RU that is within a factor of* $\frac{3}{2}$ *and* 2 *from the optimal for sector and general graphs respectively.*

Proof. Since a sector is the smallest granularity for operation of configurations, *FluidNet* 's optimality at the sector level (Theorem 1) indicates that its sub-optimality is essentially contributed by its clustering component. In bounding the latter, consider the optimal network-wide configuration to cluster the nodes of the sector graph into P disjoint clusters

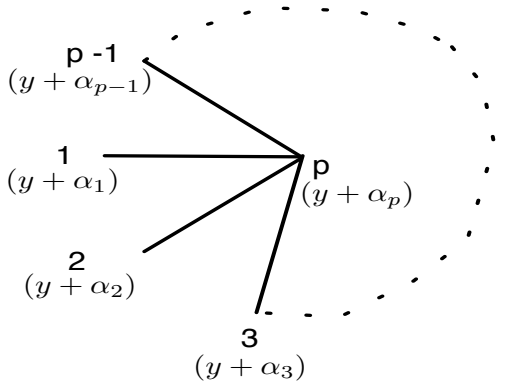


Figure 4.7: Worst Case Sector Toplogy.

($\mathcal{C}_q$, $q = [1, P]$). The optimal resource usage (RU) of the network is then,

$$RU_{opt} = \sum_{q=1}^{P} RU_{\mathcal{C}_q} \tag{4.8}$$

where $RU_{\mathcal{C}_q}$ denotes the RU of cluster $\mathcal{C}_q$. Now, if we bound the RU sub-optimality in each of the clusters, then this would automatically bound the sub-optimality of our algorithm.

Worst-case Topology: Consider p sectors that are clustered by the optimal solution, i.e. cluster $\mathcal{C}_q$. By definition of optimality, clustering properties 4.5.3 must be satisfied. If $RU(b_i, n_i)$ represents the RU of the individual sectors (i) in the cluster ($\mathcal{C}_q$), then let $y = \min_i b_i$, $j = \arg\min_i b_i$ and $\alpha_i = b_i - y$. To understand the worst-case performance for our algorithm, let us compare the RU of the cluster with that of the non-clustered solution.

$$\begin{aligned}
&RU_{\mathcal{C}_q} = y + (B - y) \sum_i n_i \\
&\sum_i RU(b_i, n_i) = y + (B - y) n_j + \sum_{i \neq j} ((y + \alpha_i) + (B - y - \alpha_i) n_i) \\
&\sum_i RU(b_i, n_i) - RU_{\mathcal{C}_q} = \sum_{i \neq j} \Delta_i = \sum_{i \neq j} (y - \alpha_i (n_i - 1))
\end{aligned} \tag{4.9}$$

From the above equation 4.9, it can be seen that depending on $\alpha_i(n_i - 1)$, each sector contributes positively ($\Delta_i \geq 0$, reduces RU) or negatively ($\Delta_i \leq 0$, increases RU) when clustered. Hence, the worst case for our algorithm would be when (i) none of the p sectors

can be clustered, and (ii) the number of clusters that contribute positively is maximized. This is achieved with the sector topology in Fig. 4.7. Here, *wlog*, let the pth sector have the largest split of carriers ($y + \alpha_p$) among all p sectors. Now, from the topology, it can be seen that all the sectors contribute more to (reducing) RU than the central pth sector. By allowing all the $p-1$ sectors to contribute positively to RU but the pth sector to contribute negatively to RU, will prevent any sectors from being clustered in our algorithm and hence provides us with the worst case performance. Note that, we need at least one negatively contributing sector since otherwise, our algorithm would be able to cluster some of the sectors as well.

General Graphs: Using this topology, we can now bound the worst-case performance of our algorithm as follows. Consider the increased RU of the configuration produced by our algorithm as a fraction L of that of the optimal configuration. From Eqn. 4.9, we have,

$$L = \frac{(p-1)y - \sum_{i \neq j} \alpha_i (n_i - 1)}{y + (B-y)\sum_i n_i} \tag{4.10}$$

For none of the sectors to be capable of clustering in our algorithm, we need,

$$\begin{aligned} RU(\min\{y, y+\alpha_p\}, n_j + n_p) &> RU(y, n_j) + RU(y+\alpha_p, n_p) \\ \Rightarrow \alpha_p(n_p - 1) &> y \end{aligned} \tag{4.11}$$

Substituting Eqn. 4.11 back into Eqn. 4.10, we have,

$$L \leq \frac{(p-2)y - \sum_{i \neq j,p} \alpha_i (n_i - 1)}{y + (B-y)\sum_i n_i} \leq \frac{(p-2)y}{(B-y)(\sum_i n_i - 1)} \tag{4.12}$$

Further, any split of carriers must be less than the total number of carriers, i.e. $y + \alpha_p \leq B$. Combining this with Eqn. 4.11, we have $\frac{y}{n_p - 1} < \alpha_p \leq B - y$ and hence $\frac{y}{B-y} \leq n_p - 1$. Substituting back into Eqn. 4.12, we have,

$$L \leq \frac{(p-2)(n_p - 1)}{(\sum_i n_i - 1)} \tag{4.13}$$

The approximation factor for the algorithm is then given by,

$$A = 1 + L \leq 1 + \frac{(p-2)(n_p - 1)}{(\sum_i n_i - 1)} \tag{4.14}$$

With the same number of cells in each sector $n_i = n$, we have $A \leq 1 + \frac{(p-2)(n-1)}{(pn-1)} \leq 2$.

Sector Graphs: In particular sector graphs, as captured in Fig. 4.6(b) involve only cliques of size at most three. Hence, for any given cluster $\mathcal{C}_q$, it can be easily seen that the resulting sub-graph will consist of a combination of only the following three structures: 3-clique, line, and cycle. If the optimal cluster contains a 3-clique, since any sector can be clustered with any other sector in the clique, then at least one pair of sectors must be capable of contraction (clustering) in our algorithm. Otherwise, it can be easily shown that the optimal solution would not contain the 3-clique. In the case of line and cycle structures, it is possible that no sector can be clustered. This would be the case when alternate sectors in the structure contribute negatively to clustering, thereby preventing any sectors from being clustered. Hence, irrespective of which structure or combination of structures is considered, it can be seen that at least $\frac{p-1}{2}$ sectors out of p in the cluster will have to contribute negatively to prevent any clustering in our algorithm. Now, applying Eqn. 4.11 for all such sectors in Eqn. 4.10, results in

$$A \leq 1 + \frac{(p-1)(n_p - 1)}{2(\sum_i n_i - 1)} \tag{4.15}$$

With the same number of cells in each sector $n_i = n$, we have $A \leq 1 + \frac{(p-1)(n-1)}{2(pn-1)} \leq \frac{3}{2}$.

4.5.6 Scalable Realization

While carriers assigned to DAS and FFR (say $(b', B - b')$) in a cluster are fixed for an epoch and determined by the cluster's resulting RU (computed based on aggregate radio resource demands from previous epoch), DAS and FFR strategies are applied to appropriate incoming traffic demand at finer time scales (order of seconds) during the epoch. Further, the DAS traffic of all the constituent sectors simply share the radio resources through a common DAS configuration on b' carriers. However, the FFR for the constituent sectors is executed individually within each sector (and not jointly), albeit on the same set of $B - b'$ carriers. This keeps the complexity of running FFR schemes low (restricted to cells in a sector). Not running FFR jointly across all sectors in the cluster will result in inter-sector interference. However, this does not hurt the estimated RU of the cluster since it is implicitly incorporated in the RU of the constituent sectors prior to clustering. Further, adopting a two-step approach - first determining the RU-optimal DAS-FFR configuration in

each sector, then improving RU of the network by clustering sectors through a light-weight process, forms the key in ensuring scalability of operations in *FluidNet* .

4.6 Prototype of FluidNet

4.6.1 Architecture

The core intelligence of *FluidNet* resides in the central processing entity managing the BBU pool, which consists of two key components.

1. **Resource Manager:** The resource manager is responsible for two key functionalities: (i) determining the appropriate number of BBU units (using *FluidNet* 's algorithms) needed to generate distinct frames and how these frames from BBUs are mapped to specific RRHs, and (ii) assigning compute resources (DSPs, cores, etc.) to each BBU unit. *FluidNet* focuses on the former functionality and is complementary to the processor scheduling problem addressed by studies with the latter functionality [69].

2. **Switching Element:** While the resource manager determines the logical mapping of BBU signals to RRHs, the switching element is responsible for realizing these mappings. Since some BBU frames are sent to multiple RRHs (as in DAS), while other frames are sent individually to specific cells (as in dynamic FFR), the switching element allows for both unicast and multicast switching. Based on the configuration determined by the resource manager on a given carrier, the switch module activates the appropriate set of output ports for an incoming BBU signal depending on the intended set of recipient RRHs. Since a BBU pool may potentially serve tens to hundreds of small cell RRHs, to ensure scalability, the switching fabric may be composed of multiple smaller-size switches (as opposed to one big switch). The size of the switches may be chosen to tradeoff the level of multicasting capability (e.g., for DAS) with cost.

4.6.2 Implementation

We have built a full-fledged, small-scale C-RAN testbed, capable of over-the-air transmissions. Given that LTE requires licensed spectrum, our set-up is currently based on WiMAX (with an experimental license). However, both LTE and WiMAX being OFDMA-

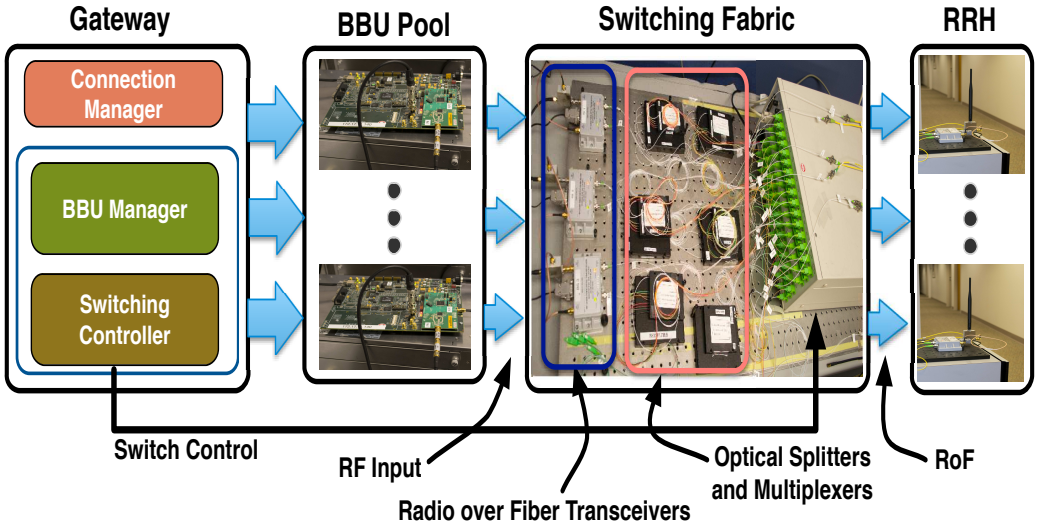


Figure 4.8: Testbed components of *FluidNet* .

based, our testbed suffices to demonstrate the proposed concepts in *FluidNet* that are equally applicable to LTE as well. Our testbed is depicted in Fig. 4.8.

BBU Pool, clients and gateway: Since our focus is on the front-haul configuration, we consider six WiMAX BSs (from PicoChip [72]) directly as our BBUs. We use netbooks with USB WiMAX dongles as the clients. *FluidNet* 's algorithms to determine configurations, are implemented in the WiMAX gateway, whose primary role is to manage the traffic flows from/to the clients. In our set-up, a single gateway is instrumented to manage all the 6 BBUs and their clients. The gateway also hosts the controller to instruct the switch for mapping of BBU signals to RRHs. We implement the controller using LabVIEW and communicate the desired configurations to the switch via serial port (RS232).

Radio-over-Fiber: Ideally, baseband signals should be transported in the digital domain between BBU pool and RRHs to allow for scalable, low-latency switching between configurations. However, the lack of commercially available products to manipulate the baseband signals between BBU pool and RRHs in the digital domain (over CPRI), has prompted us to pick an alternate design, wherein we employ analog RF signal transmission based on radio over fiber (RoF) techniques. With latencies of about 5 μs/Km over the fiber, we have verified that RoF can retain the signal synchronization between RRHs as

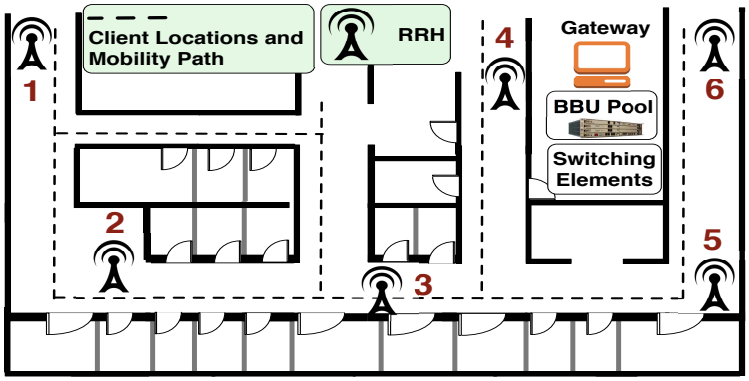


Figure 4.9: Testbed deployment

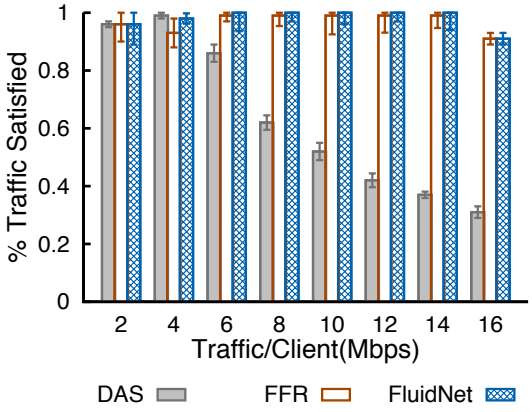


Figure 4.10: Traffic satisfaction with variable traffic demand.

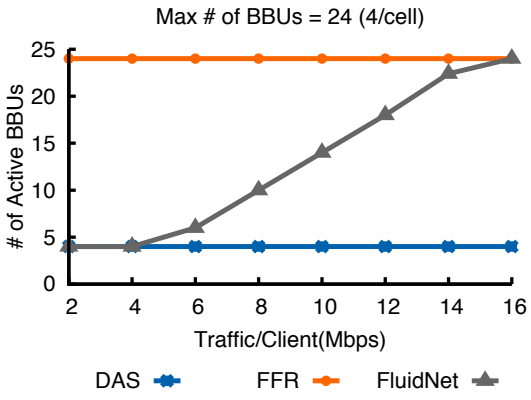


Figure 4.11: Energy efficiency with variable traffic demand.

well as the timing constraint between downlink and uplink signals for reasonable distances of around 10 Km between the BBU pool and RRHs. With RoF, the modulated RF analog signal from a BBU is converted into an optical carrier using a COTS optical transceiver, and delivered to RRHs on a single mode optical fiber.

RRHs: Since all the signal processing (even modulation and RF up/down-conversion) is done at the BBU pool, our RRH design is simple and consists of an optical transceiver attached to an antenna. The optical wavelengths (carrying multiple RF signals) are photo-detected and converted back to the RF domain (for over-the-air transmission) by the optical

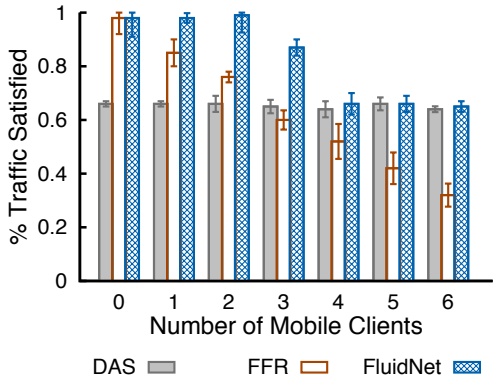


Figure 4.12: Mobile: Traffic satisfaction.

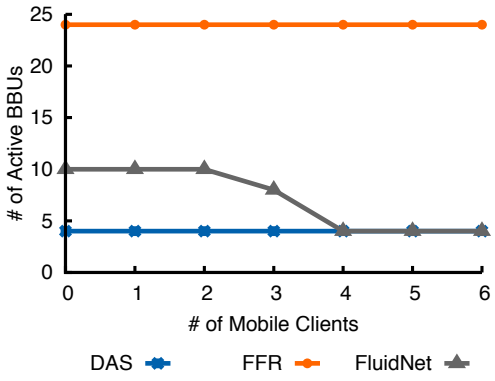


Figure 4.13: Mobile: Energy efficiency.

transceiver. On the uplink (from RRHs to BBUs), the operations are similar but in reverse order.

Switching Element: Since BBU signals are carried as analog RoF, to realize various configurations, we enable switching in the optical domain, which is controlled from the gateway. Since our optical switch supports only one-to-one switching, we enable flexible switching (one-to-one and one-to-many) indirectly by using optical splitters and multiplexers with CWDM. While the latency in switching between configurations is negligible if implemented in the digital domain, it could be appreciable in the optical domain depending on the sophistication of the switch. With our inexpensive optical switch that reconfigures individual port switches, this could amount to 1 s. This is still acceptable if hybrid (DAS, FFR) configurations are realized in the frequency domain (across spectral carriers), where they need to be changed only with appreciable load changes at the granularity of several seconds or minutes.

The gateway controls the optical switch to turn on or off each independent path from each BBU to any RRH to create various configurations. Since each switch in our testbed is limited to supporting all configurations in a set-up with at most 4 BBUs and 4 RRHs, we employ two such switches jointly to serve our 6 BBU-RRH system.

4.7 Performance Evaluation

4.7.1 Prototype Evaluation

Set-up

Testbed: Our testbed consists of six small cell RRHs deployed in an indoor office setting, driven by six physical BBUs co-located in a single room through optical fiber (see Fig. 4.9). There are six clients, each associated to one of the six cells. All the small cells are assumed to be in a single sector of a macrocell. The BBUs can generate WiMAX RF signals over two 10 MHz bands: at 2.59 GHz and 2.61 GHz, for which an experimental FCC license has been acquired to conduct over-the-air transmissions. Hence, we consider four spectral blocks (i.e., carriers), each with 5 MHz bandwidth to realize hybrid configurations. Since our BBUs are BSs themselves, we can operate a BS and hence an RRH on only one carrier at any given time. Due to this technical difficulty, we run DAS and FFR configurations sequentially on the appropriate blocks to realize the hybrid configuration for the sector. This would equivalently amount to 4 logical BBUs (one per carrier) per small cell and hence a maximum of 24 logical BBUs in the system.

Strategies and Metrics: We evaluate *FluidNet* against both the DAS scheme (labeled "DAS") and an FFR scheme (labeled "FFR") for baseline comparison (we consider other baselines in simulations). For FFR, our topology allows each small cell to operate on half the set of sub-channels, while being orthogonal to those of its neighbors. In DAS, a single BBU frame serves all the RRHs and clients. Traffic loads (2 - 16 Mbps) and profile (static, mobile) of clients are the parameters studied. The maximum net throughput that can be delivered in a WiMAX frame (at 64 QAM) in our set-up is around 16 Mbps for 10 MHz bandwidth. Each experiment takes 180 seconds and is repeated multiple times with varying client locations. Impact of rate adaptation is isolated by picking the MCS that delivers maximum throughput for a client (we try all MCSs). The fraction of the offered load supported and the effective number of BBU units consumed in the process are the metrics of evaluation.

Impact of Traffic Heterogeneity

With six static clients, we study the percentage of average traffic satisfied and the number of BBUs required by each scheme with varying per-client traffic demand in Figs.

4.10 and 4.11, respectively. With high load, FFR is essential to support the traffic demand, while DAS can support only a third of the demand (Fig. 4.10). When the load is low, DAS is sufficient and activates only a sixth of the BBUs required by FFR (Fig. 4.11). While *FluidNet* blends the best of DAS and FFR under extreme load conditions, its benefits are more pronounced in the intermediate regime (e.g., 10 Mbps demand per-client), where it outperforms both DAS and FFR. By employing hybrid configurations and adapting them to traffic profiles, *FluidNet* sustains twice as much traffic as DAS and requires only half the BBUs activated by FFR.

Impact of User Heterogeneity

We vary the number of mobile clients in a six client scenario, with each client's traffic fixed at 8 Mbps. To eliminate the adverse impact of handoffs in FFR (triggers, delays, etc.), we move a mobile client at pedestrian speed only in the vicinity of its RRH (sample path in Fig. 4.9). In contrast, seamless coverage and lack of handovers, allow a client to be moved in all deployment areas with DAS and *FluidNet* . Hence, the results in Figs. 4.12 and 4.13 are optimistic for FFR. We see that with increasing fraction of mobile traffic, FFR's performance degrades and ends up being much worse than that of DAS (Fig. 4.12). We observed that, even without handovers, when a client moves away from its RRH, its link deteriorates and faces high interference from the control region of frames of neighboring RRHs in FFR (only data part of the frame is protected in FFR). While DAS's coverage provides consistent link quality, it under-utilizes the spectrum when mobile traffic is low. *FluidNet* strikes a fine balance between the two configurations to support as much as 50% more traffic, while incurring a BBU energy consumption that is only slightly more than that of DAS.

Adaptation to Network Dynamics

We now evaluate *FluidNet* 's adaptability to network dynamics. We start with six static clients, each with a 8 Mbps traffic load. Two events are triggered, one at 40 seconds into the experiment and another at 80 seconds. In the first event, four of the clients become mobile. Then at the 80 second mark, one of the mobile clients becomes static again and the remaining mobile clients reduce their rate to 4 Mbps. From Fig. 4.14, we see that *FluidNet* tracks FFR performance initially (albeit at less number of BBUs activated),

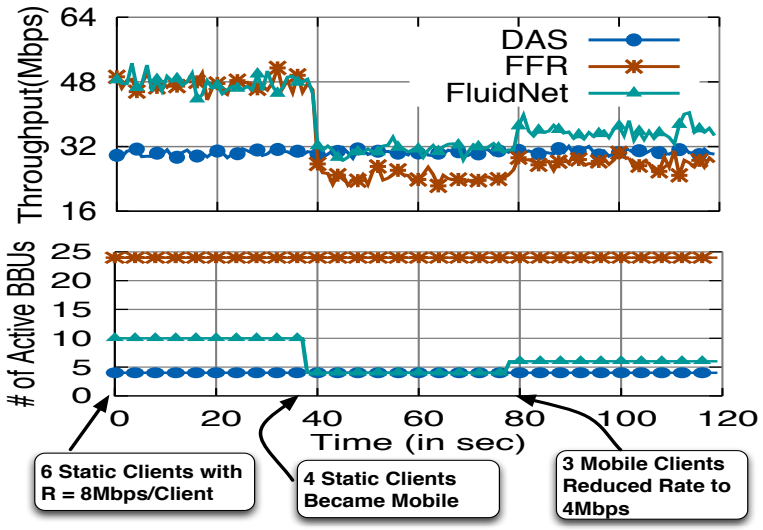


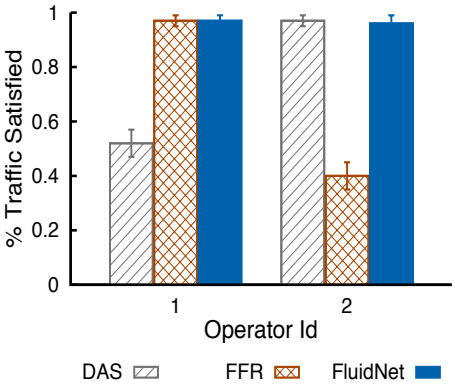
Figure 4.14: Network Dynamics.

when there are more static clients inducing a high traffic load. When a majority of the traffic demand becomes mobile at the first event, unlike FFR that suffers in performance, *FluidNet* immediately (but for a short transition delay) adapts its configuration to track DAS performance that is optimal for the updated network conditions. Similarly, when the traffic load of static clients starts to dominate, while still involving mobile clients at the second event, *FluidNet* employs a hybrid configuration to sustain a higher traffic load compared to both DAS and FFR, while incurring a BBU usage comparable to DAS. This clearly indicates *FluidNet* 's ability to effectively adapt its configurations to varying network conditions.

Multi Operator/Technology Customization

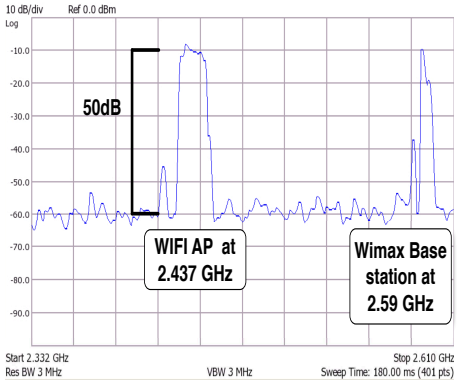
One of *FluidNet*'s key features is its ability to allow for multiple operators to customize the configurations needed to serve their respective clients simultaneously. To illustrate this, we design an experiment with three BBUs and three RRHs. There are two operators, one operating at 2.59 GHz and the other at 2.61 GHz, each with 10 MHz bandwidth. Both operators share the same set of three RRHs to cater to three clients each simultaneously. While all clients for operator 1 are static and impose a net rate requirement

(a) Signal Spectrum



(b) Traffic Satisfaction

Figure 4.15: Two Operators.



(a) Signal Spectrum.

(b) Traffic Satisfaction.

Figure 4.16: WiFi + WiMAX

of 21 Mbps, those for operator 2 are all mobile with a net rate requirement of 12 Mbps. To check transmission feasibility on our front-haul over longer distances, the fiber between BBUs and RRHs is made to be a 10 Km fiber spindle. Fig. 4.15(a) presents the spectral graph from one of the RRHs captured using a spectrum analyzer. It is clearly seen that both the operators are able to co-exist simultaneously on the same front-haul without any interference to each other's RF signal. Furthermore, this is achieved over a large distance of 10 Km, which demonstrates feasibility for an outdoor cellular deployment. Also, Fig. 4.15(b) shows that *FluidNet* tailors the right configuration for each operator to provide maximum satisfaction of traffic demand.

This is also evident from Figs. 4.16(a) and 4.16(b), where a single operator uses two different access technologies (WiFi and WiMAX) to serve five clients (each with 10 Mbps traffic rate) through 3 RRHs. Two of the clients on WiFi (2.43 GHz) are static and associated to two of the RRHs, while the other three are on WiMAX (2.59 GHz) and mobile. It is interesting to see that *FluidNet* is capable of simultaneously supporting an asynchronous (WiFi; one-to-one for CSMA) and synchronous (WiMAX; one-to-many for DAS) access technology for the same operator. *FluidNet*'s support for multiple operators and technologies are very useful features in a C-RAN, given the growing popularity of RAN-sharing and dual carrier small cells (for WiFi offload).

4.7.2 Simulation

Set-up: We use a 3GPP-calibrated system simulator to create a outdoor heterogeneous cellular network, with 19 macrocell sites (each has three sectors) and ten small cells per sector. Thus, the network has a total of 627 cells (57 macro + 570 small) based on the scenarios defined in 3GPP 36.814 [73]. We distribute 3600 small cell clients according to the '4b' distribution [73]. We assume that the macrocells and their clients use pre-determined spectral resources orthogonal to the ones used by the small cells and their clients, and thus ignore the interference from/to the macrocell network.

To generate traffic demands, we resort to emulating a typical operational day in outdoor cellular networks. Since we do not have access to such operator data (and public data does not exist to the best of our knowledge), we use the reported peak hour distribution from [64] as follows. We mark each sector (and the small cells in it) as either "business" or "residential". As seen in Fig. 4.2, we geographically determine that the central, shaded sectors are business sectors (there are a total of 21 such sectors with 210 small cells in them) and peripheral sectors are residential sectors (36 of them exist). The small cells in a business sector hit their peak loads between 10 a.m. and 4 p.m. and residential cells have peak hours between 4 p.m. and 8 p.m. The traffic outside the peak hours is chosen such that there is a gradual increase until the peak interval and a decrease after that.

We compare *FluidNet* against three other schemes. The first (labeled "FFR") is a pure FFR solution running with a fixed cluster size corresponding to a macrocell (3 sectors = 30 small cells). The second (labeled "DAS") is a pure DAS solution with opportunistic clustering. When the total load of neighboring sectors is less than a frame's worth of

resources (i.e., the max. capacity of DAS), they are merged in a DAS cluster and thus served by one BBU. The third (labeled "GRID") is reported in [64] and addresses energy consumption by turning small cells off during non-peak periods.

Traffic Heterogeneity: We first simulate a network where no clients are mobile. Each result is the average of five different runs with randomly selected traffic demands from clients, subject to the spatio-temporal traffic distribution.

Figs. 4.17(a) and 4.17(b) plot the traffic satisfaction ratio and the energy consumption (RU), respectively. We first see that *FluidNet* has a competitive traffic satisfaction ratio with FFR (is only 3% worse on average). The slight reduction is because FFR explicitly accounts for inter-sector interference by considering a cluster size of three sectors. In contrast, *FluidNet* applies FFR at a granularity of one sector and resorts to resource permutations to address inter-sector interference in a scalable manner. We also see that while having a competitive traffic ratio, *FluidNet* is much more (*3x* on average) energy efficient than FFR. DAS, albeit the most energy efficient strategy, suffers from lack of spatial reuse and hence satisfies only 65% of the traffic on average.

When compared with GRID, while the fraction of traffic satisfied does not differ considerably, *FluidNet* activates *2.2x* less BBUs than GRID. This is due to the fact that while energy savings from BS-switching approaches such as GRID are inherently limited based on physical proximity of cells, *FluidNet* can cluster arbitrarily large numbers of cells to yield more energy savings. This is exemplified in Fig. 4.17(e) where we plot the temporal progression of clusters in *FluidNet*; clusters (color-coded) are seen to shift spatially from residential areas in the morning to business areas in the evening. White (uncolored) sectors are clusters of size one (i.e., cannot be merged with other sectors due to high traffic load). To closely look at clustering in *FluidNet* during non-peak hours, we compare *FluidNet* with and without the clustering component (the latter called "*FluidNet-NC*"). As seen in Fig. 4.17(c), even without its clustering component *FluidNet* outperforms GRID. Further, while *FluidNet* -NC requires 80 BBUs on average, *FluidNet* requires only 43 BBUs, resulting in much lower energy consumption. This shows that clustering is critical in realizing high energy savings.

In summary, *FluidNet* effectively exploits the spatial and temporal load asymmetry in the network and yields more energy savings than state-of-the-art solutions while satisfying a high fraction of the traffic demand.

User Heterogeneity: We now evaluate *FluidNet* with vehicular mobility. Here, we take the peak traffic hour of the day (4 p.m.) and investigate the traffic satisfaction ratio (averaged over 5 runs) with varying percentage of mobile clients. Each client moves at 60 miles per hour, only within its sector. From Fig. 4.17(d) we see that DAS performance is not affected by mobility since it results in a uniform signal quality for mobile clients; the network capacity is unchanged. With FFR, performance degrades as we increase the percentage of mobile clients (due to handovers and degraded SNR). With *FluidNet* , increasing number of mobile clients results in more carriers being allocated for DAS. While associating mobile traffic with DAS is beneficial in most of the cases, it can lead to lower performance (compared to FFR) when *all* the traffic is mobile. Ideally, one would need to identify the tradeoff between DAS (uniform per-client SNR but no spatial reuse) and FFR (degraded client SNR but high spatial reuse) for mobile traffic, and make careful decisions.

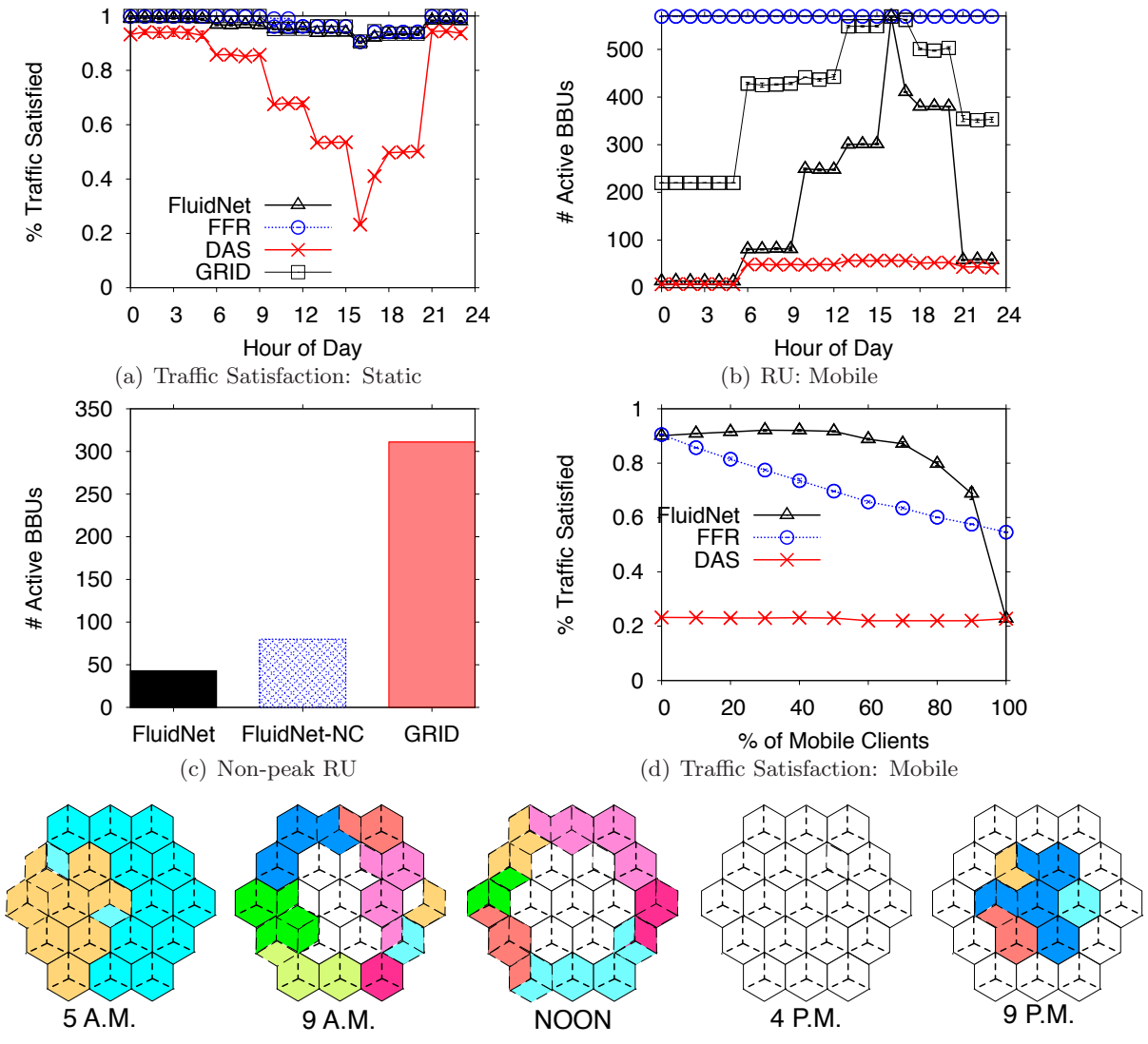
4.8 Discussions and Remarks

We presented *FluidNet* - a framework for dynamically re-configuring the front-haul of a C-RAN to meet the dual objective of improved RAN performance with reduced resource usage in the BBU pool. Our evaluations show promising benefits towards these goals. Going forward, we would like to consider the following.

Applicability to other C-RAN Models: Since *FluidNet* focuses on *logical* front-haul configurations, it can work with any front-haul (e.g., microwave wireless) as long as the latter can support the data rates needed for transport of BBU signals. Similarly, it also applies in a partially-centralized C-RAN model [55], where more processing is entrusted to the RRHs to reduce the load on the front-haul. However, the energy savings in this model needs to be investigated.

Co-existence with Carrier Aggregation: LTE-advanced systems will support multiple component carriers and carrier aggregation. Carrier split for configurations in *FluidNet* can be realized much more easily with multiple component carriers. However, the interaction of FFR and DAS with joint scheduling on multiple carriers needs further study.

Migrating to Digital Front-Haul Transmissions: Instead of using RF over Fiber, we would like to migrate our BBUs to those that provide access to digital I-Q streams that can be transported over CPRI. This would allow for scalable realization of our configurations in the digital domain.



(a) Traffic Satisfaction: Static

(b) RU: Mobile

(c) Non-peak RU

(d) Traffic Satisfaction: Mobile

(e) Temporal Progression of Clusters

Figure 4.17: *FluidNet* has comparable traffic satisfaction ratio to FFR (a), and is *3x* and *2.2x* more energy efficient than FFR and GRID respectively.

Chapter 5

FlexiWeb: Network-Aware Compaction for Accelerating Mobile Web Transfers

5.1 Introduction

The ill-effects of slow websites, especially when browsing using a cellular network, are well-documented. Recent surveys suggest that two-thirds of users encounter slow websites every week [74], and that 49% of such users abandon a site or switch to a competitor upon experiencing large delays [74]. To address this, several systems augment mobile web browsing with support from middleboxes or proxies in the cloud (e.g., [75], [76], [77], [78], [79], [80]); many of these systems are widely used today, such as Opera Mini [78], Amazon Silk [79] and Chrome beta [80]. These cloud-supported mobile browsing options can potentially reduce download times, device energy consumption, and data usage costs.

Network conditions should dictate whether or not a middlebox should be used for performing content compression: The aforementioned middleboxes primarily compress content to be delivered to wireless clients based on the common belief that compression reduces the volume of bytes downloaded, and thus, improves performance. Unfortunately, this is not always the case. As our first contribution, we conduct an extensive measurement study which shows that, when network conditions are good, there could in fact be an *increase* in page load times (by as much as 28%) due to the use of middleboxes

(we refer to this as `Proxy Assisted` browsing). The problem especially occurs if the web page to be downloaded only contains small objects (no high quality images or large scripts). The primary reason for this degradation is that diverting content to a middlebox causes additional delays due to longer routes and/or processing at the middlebox; these delays offset the gains achieved due to compression in some cases, and thus, worsen page load times. On the other hand, when network conditions are bad, `Proxy Assisted` browsing decreases page load times compared to `Conventional` browsing by 32%. This suggests that the decision on whether or not to use middlebox support for mobile web transfers must be made based on (i) network conditions and (ii) object sizes.

Content transformations ought to be network-aware: Today, most middleboxes for the mobile web apply the same compression to all content. Insufficient compaction could potentially lead to large page load times when network conditions are poor. If users and web providers are willing to compromise on content quality for better page load times, one could enable middleboxes to perform different degrees of content compaction based on network conditions. Specifically, this is possible in the case of images if the network conditions are bad. For example, when the quality of the client's network connection is very poor, the middlebox could even take extreme steps, such as transform a color image to a gray scale version, to ensure reasonable load times.

Challenges: Realizing the above vision towards network-aware, dynamic usage of a middlebox for accelerating mobile web transfers is associated with two primary challenges: (1) For every object, the decision as to whether to fetch it directly or via the proxy depends on the object's size, but that is typically known only after fetching the object. Issuing a HTTP HEAD request just to query the webserver hosting the object for its size will add significant overhead. (2) The extent to which any particular object on a page should be compressed depends on the sizes of other objects on the page. However, given how the process of loading a web page works, the objects on a web page are only revealed iteratively as the page load proceeds.

Design and implementation of FlexiWeb: As our primary contribution, we design and prototype the FlexiWeb framework, which addresses the above challenges, towards providing dynamic, network-aware middlebox support for the mobile web. FlexiWeb has a novel design that encompasses three key elements. First, it uses an empirically derived model that determines, given an object's size and the network conditions, whether

the object should be fetched via a proxy[1] or not. Second, FlexiWeb incorporates a classifier that predicts an object's size, based primarily on features derived from the object's URL. Third, FlexiWeb performs network-aware data compaction at the proxy, wherein appropriate transformations are applied to images based on the bandwidth available between the client and the proxy; other objects (e.g., Javascripts) are compressed as in conventional middleboxes. For this, FlexiWeb uses an online algorithm that selects near-optimal compaction levels while attempting to keep page load times within a latency target dictated by user tolerance (shown to be 2–5 seconds by user studies [81]). We cast this algorithm as a utility maximization subject to latency constraints.

Flexibility: While performing content transformation, it is important to ensure that web browsing sessions do ensure a good Quality-of-Experience (QoE). While transformations reduce delays, they must still cater to a user's or a web provider's requirements in terms of quality (e.g., a user may not want gray scale transformations). Hence, we allow users or web providers to set a minimum quality limit either via browser settings or via metadata embedded in a web page's HTML; this sets the maximum allowable compression allowed on each object in a web page.

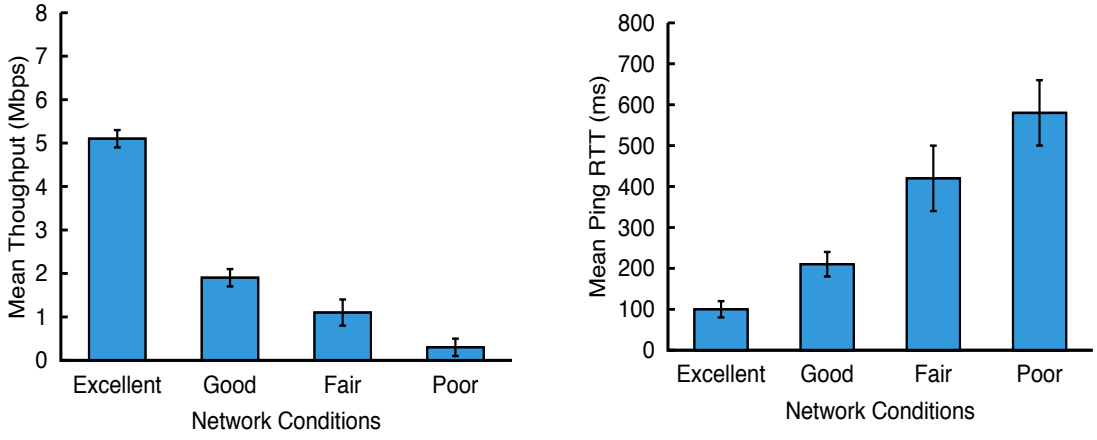
Evaluations: Finally, we perform extensive evaluations of FlexiWeb both via in-house emulations and real-world experiments using AT&T and T-Mobile's 4G networks in both static and mobile settings. Our experiments demonstrate that FlexiWeb reduces page load times by up to 35–42% as compared to both `Proxy Assisted` and `Conventional` mobile web browsing.

5.2 To proxy or not to proxy

We begin with a measurement study to understand the implications of using a middlebox for compressing mobile web content. We first describe our setup for the measurements and then discuss the results.

Client side setup: We set up multiple rooted mobile devices (HTC One phones with Android 4.3) running the open source web browser, Chromium, for Android. A Python

[1]We use the terms proxy and middlebox interchangeably.



(a) (b)

Figure 5.1: (a) Average values of Throughput (b) Average values of RTT between the client and the proxy server in different cellular network conditions

Table 5.1: Network Conditions (Note that MAC layer retransmissions limit error rates even under poor conditions [1])

Network Conditions	**RTT(ms)**	**Throughput(Mbps)**	**Loss Rate(%)**
Excellent	100	5	0.006
Good	200	2	0.006
Fair	400	1	0.04
Poor	600	0.3	0.1

program controls the browser via Chromedriver using RemoteWebDriver. All experiments are performed with a cold browser cache. We use "page load times" as the primary metric to capture performance. The page load time is defined as the time it takes for the browser to download and process all the objects associated with a web page. Most browsers fire a Javascript event (onLoad()) when the page is loaded. Chrome's remote debugging interface provides us the time taken to download each object in a web page.

Server side setup: To ensure that a phone retrieved the same content each time it fetched a particular URL (in repeated experiments), we captured Alexa's top 500 websites (we believe these represent typically downloaded web pages) and replayed the content using the Web Page Replay tool [82]. All content was cached on our Web Page Replay server. Our server has 16 CPU cores and 64 GB of memory and is hosted on our campus network.

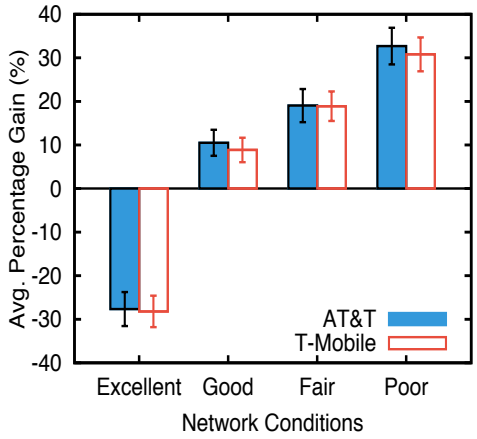


Figure 5.2: Gains from using a commercial compression proxy (Google compression proxy) in downloading Alexa's top 500 web pages under different client network conditions. Gains are measured in comparison to a `Conventional` browser.

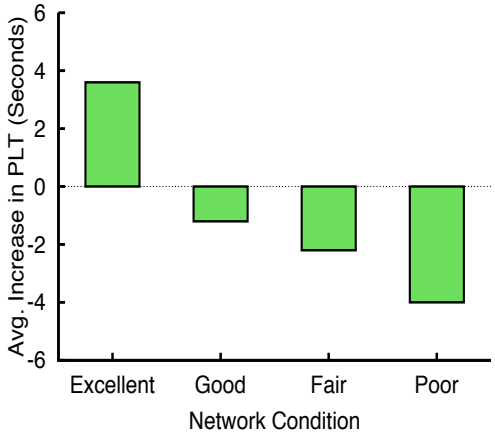


Figure 5.3: Average increase in Page Load Time (PLT) in seconds from using a commercial compression proxy (Google compression proxy) in comparison to a `Conventional` browser.

Network conditions: We operate over the 4G networks of two major US cellular providers, AT&T and T-Mobile. To examine the influence of different network conditions, for each provider, we carefully choose four locations where the RTT and throughput values were similar to those listed in Table 5.1. We choose these locations after monitoring the network conditions by running a long term (2 day) experiment to ensure the stability of the conditions (they stayed more or less the same) as shown in Figures 5.1(a) and 5.1(b).

Data collection: For each page downloaded, we record the browser-reported page load time, as well as the load time for each individual resource on the page. In addition, we run `tcpdump` on the phone to capture a trace of all network traffic during the page load. In each run, we first load a page by downloading the resources on a page directly from the webservers hosting them, and then via a compression proxy, back-to-back. We perform 10 trials for each experiment.

Today's compression proxies can increase mobile web page load times: First we perform experiments over both AT&T and T-Mobile's networks, with currently deployed commercial compression proxies. Google's Chrome mobile browser comes with the `Proxy assist` option. Upon enabling this, the browser redirects all the requests to Google's compression proxy. Using this feature, we perform experiments with a *live version* (clients

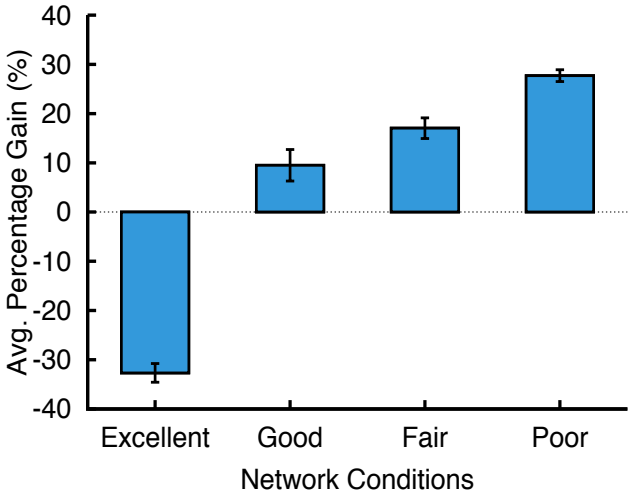


Figure 5.4: Gains from using a `Proxy Assisted` browser (using our own proxy) in downloading Alexa's top 500 web pages under different network conditions. Gains are measured in comparison to a `Conventional` browser.

download the webpages directly from the source web sites as opposed to downloading a replay from our web server) of Alexa's top 500 websites.

Figure 5.2 shows the percentage gain in performance in different network conditions, when using Google's proxy in comparison to a `Conventional` browser (where all content is retrieved from the source). We see that the use of the proxy provides an average gain of about 32% in bad network conditions. In excellent network conditions, the `Conventional` browser (no proxy) outperforms the `Proxy assisted` browser by about 28%. Figure 5.3 shows the average gain in page load times in seconds. We see that the proxy can increase the page load time by ≈ 4 seconds. According to prior studies [83] a 1 second delay could potentially result in a nett loss of 2.5 million dollars in sales for an e-commence website. It is also seen that a 4 second delay can cause up to 25% increase in page abandonment. These results clearly demonstrate that the use of a proxy can significantly hurt performance when network conditions are excellent; however, as conditions degrade, compression at a proxy can provide significant benefits.

An in-depth study: To get a further understanding of the implications of using a compression proxy, we set up our own proxy (to emulate the behavior of Google's proxy) and conduct more in depth studies.

Compression proxy setup: We set up the compression proxy on Amazon EC2 to have a controlled environment. The proxy is located in northern California, relatively close to the geographical region of the client. We use Google's open-source compression proxy module called PageSpeed, which we setup as a forward proxy [84]. We use the optimization strategies [85] recommended for reducing page load times. These include combining and minifying JavaScript (JS) and Cascading Style Sheets (CSS) files, inlining small resources, and others. We configure PageSpeed to dynamically optimize images by removing unused metadata, resizing images to specified dimensions, and re-encoding images to the WebP format (which requires fewer bytes than other popular formats such as JPEG and PNG).

Figure 5.4 presents the average gain in page load time with the compression proxy as compared to `Conventional` browsing (or direct downloads). We see that the results are very consistent with what was observed with the real-world compression proxies.

The question that we then seek to answer is: *"why does performance degrade in excellent network conditions due to the use of a proxy?"* Via a careful study, we find that this is primarily due to penalties associated with loading content via a compression proxy: (i) circuitous routing between the client and web servers via the middlebox, and/or (ii) processing delays at the middlebox.

Penalties due to route stretch: Zarifis et al [1] provide a detailed analysis of path inflation in mobile networks and suggest that factors that primarily contribute to the path inflation are diversions due to (i) ingress points and (ii) peering points. With regards to the former factor, the device's carrier network may not have ingress points to the Internet in the device's area. With regards to the latter, the carrier network may connect with a web service provider at a peering point or Internet exchange point (IXP), which may be at a location distant from both the mobile device and the web provider.

An analysis of our traceroute data indicates that, in the presence of a proxy, traversal of multiple peering points (because of three different networks viz., the carrier network, the network hosting the proxy and the network hosting the web server, instead of two) is likely to lead to longer paths. Indeed, we observed that end-to-end routes in some cases were inflated by up to 8 hops due to the use of the proxy, which resulted in increases in RTTs by up to 45ms.

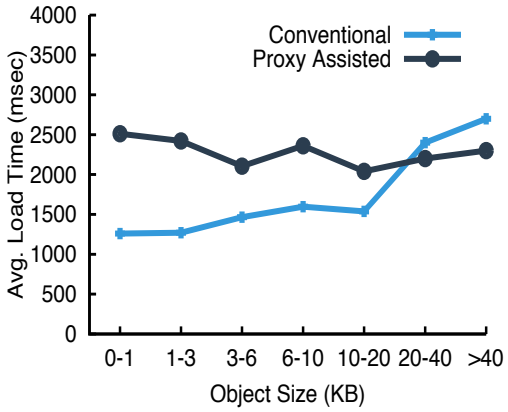


Figure 5.5: Download times of objects under Excellent network conditions

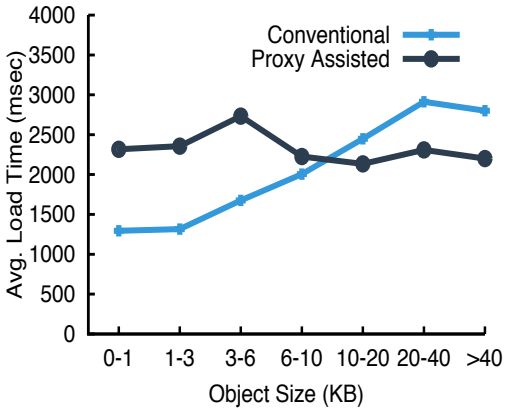


Figure 5.6: Download times of objects under Good network conditions

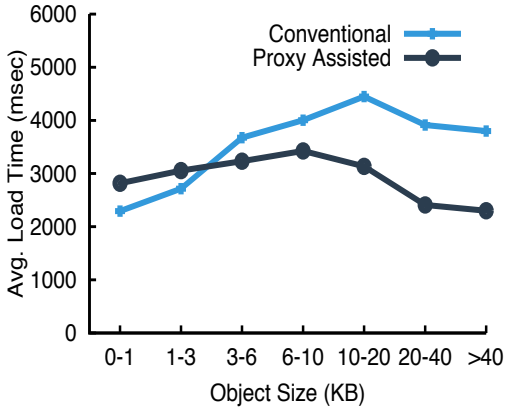


Figure 5.7: Download times of objects under Fair network conditions

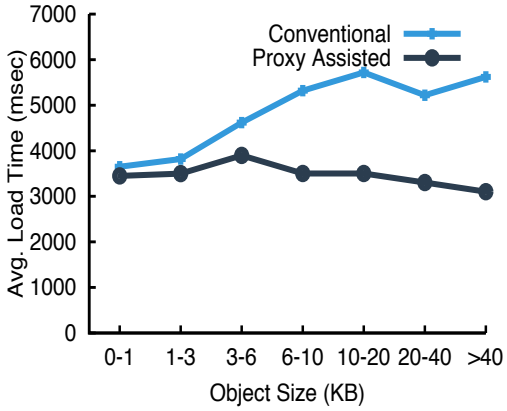


Figure 5.8: Download times of objects under Poor network conditions

For example, when we fetched cnn.com without using the proxy, from a location in Southern California, the request for object `http://i.cdn.turner.com/cnn/.e/img/4.0/logos/city_of_tomorrow_bw.png` traversed through AT&T's core mobile network to a CDN (content distribution network) server which was geographically proximate to the client (near Los Angeles). We also found that ingress points for AT&T's mobile core network were in the same geographical location. When fetching the same object via the compression proxy, the request went to the proxy server located in northern California and the request for the object was forwarded to the CDN located near the proxy server (northern California).

Table 5.2: Mapping that dictates when to fetch an object directly from the source server and when to fetch it via the proxy

Network Condition	**0–1 KB**	**1–3 KB**	**3–6 KB**	**6–10 KB**	**10–20 KB**	**20–40 KB**	**$\geq$ 40 KB**
Excellent	Direct	Direct	Direct	Direct	Direct	Proxy	Proxy
Good	Direct	Direct	Direct	Direct	Proxy	Proxy	Proxy
Fair	Direct	Direct	Proxy	Proxy	Proxy	Proxy	Proxy
Poor	Proxy	Proxy	Proxy	Proxy	Proxy	Proxy	Proxy

We observe a similar effect when fetching the same object via Google's compression proxy; the request was sent to Google's compression proxy located in northern California while the closest CDN server was in region Southern California (Los Angeles).

We wish to point out here that the cellular providers also deploy (transparent) proxies in their core network. According to [86] however, only Sprint performs any form of content rewriting. Other providers (T-Mobile, ATT and Verizon) primarily perform delayed handshaking, connection persistence and redirection using such a transparent proxy.

Penalties due to processing overhead: We find that the processing overhead to convert an image to WebP format (the image transformation performed by Google's compression proxy) ranges from 10ms to 30ms per image, depending on the image's size; if a web page contains a large number of objects, the total processing delay can be significant. In general, when compared to JPEG, the encoding speed for WebP is $\approx 10X$ slower (and the decoding is $\approx 1.4X$ slower) but it does provide $\approx$ a 30% gain in terms of a size reduction, on average [87]. A commercial proxy can reduce the processing times by using advanced caching techniques (to amortize processing costs across users) but websites are increasingly personalizing their content, e.g., amazon.com shows different items to different users based on their browsing patterns. This, in turn, results in the fetching of different objects for different users.

Impact of object size: To summarize, our evaluations indicate that neither the `Conventional` nor `Proxy Assisted` approach is the winner in all scenarios (in terms of network conditions). To further understand the implications of using `Proxy assisted` browsers, we next analyze the data collected for both types of browsing to understand the impact of network conditions on load times for objects of different sizes. We partition objects into 7 bins: 0–1 KB, 1–3 KB, 3–6 KB, 6–10 KB, 10–20 KB, 20–40 KB, and $\geq$ 40 KB. In Figure **??**, we compare the average load times computed over all objects in each category, with both browsing strategies in different network conditions. We see that when

the conditions are good, using the proxy is only beneficial for objects larger than 30 KB. This suggests that the delays due to indirection via the proxy outweigh the benefits of data compaction for small objects. As network conditions degrade, fetching compressed objects via the proxy improves object download times as compared to fetching them uncompressed directly from the source. In the extreme case, we see that the proxy's compression enables better download times for objects of all sizes in poor network conditions.

Takeaways: In summary, our measurement studies suggest the following. In very good/excellent network conditions, it is best if only larger sized objects ($>$ 30 KB) are diverted to a compaction middlebox during a web page download. Under fair to poor network conditions, one should retrieve all content indirectly via the middlebox. While TRINITY addresses other key challenges as well, these takeaways form the basis for middlebox usage in our framework.

5.3 Design of FlexiWeb

In this section, we describe the design of the FlexiWeb framework. As discussed earlier, FlexiWeb seeks to reduce page load times while browsing the web on mobile devices. The key property of FlexiWeb is that it is adaptive and achieves its goal under all network conditions.

Overview: To improve the performance vs. experience tradeoff on the mobile web, our over-arching goal is to enable mobile clients to dynamically select which resources on a web page to download via a proxy and to enable the proxy to determine how it should transform every resource that it relays. This goal imposes three logical challenges:

- How can a client account for the characteristics of the objects on a page and its network conditions to determine what to fetch directly from the source web servers and which objects to fetch via the proxy?
- Since the determination of whether to fetch an object via the proxy or not depends on the object's size, how can we predict an object's size before fetching it?
- To ensure that all objects on a page are delivered to a client within a target load time, the manner in which the proxy transforms any particular object on a page must depend on the sizes of other objects on the page. How can the proxy do so, given that all the objects on a page are only revealed iteratively during the page load process?

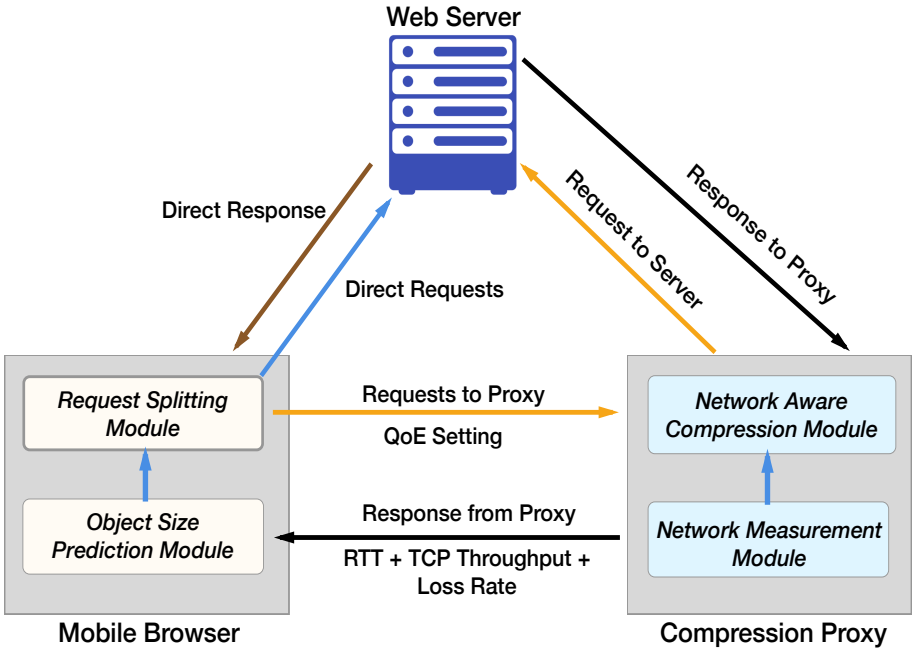


Figure 5.9: Overview of FlexiWeb's architecture

Our architecture of TRINITY towards tackling these challenges is shown in Figure 5.9. The *Network Measurement component* at the proxy measures the network's characteristics via a series of measurements. The measurements made by FlexiWeb do not consume excessive resources on the client side in terms of bandwidth or energy. These measurements are then fed back to the mobile browser. The mobile browser contains an *Object size prediction module* which estimates object sizes in a web page that it seeks to download. This module essentially uses (i) measured distributions to determine what objects (in terms of sizes) to fetch via the proxy after transformation, and what objects to fetch directly from the source and (ii) an incremental learning algorithm to predict object sizes. The *Request splitting module* then fetches each object, either directly from a web server or via the proxy; large objects are always fetched via the proxy, while the decision on "from where to fetch a small object?" is made based on network conditions (this part of FlexiWeb uses the inferences from our measurements in Section 5.2).

The *Network Aware Compression module* at the proxy fetches the objects from their sources. Note here that different objects could be potentially fetched from different web servers (e.g., ads may be fetched from a different domain than the one in which the web page being fetched is hosted). Since not all objects are fetched simultaneously, the

proxy has to decide on the level of compression (transformation) to be performed on each object as it is fetched, with the objective of trying to keep the total page load time within a target. The proxy meets this goal by using a novel algorithm to dynamically choose the compression level for each object (images specifically) prior to sending it to the mobile client. Specifically, it performs an online optimization to determine object transformations that allow the download to fit within a target budget page load time; it reverts to a minimal transformation (maximum compression) when the optimization formulation is infeasible.

Handling Videos: Videos in a webpage are downloaded as an image (snapshot from the video) when a page is loaded by the browser. If a user clicks on the play button of the video it starts downloading/streaming the video from a server. Video streaming is not considered to be part of the webpage download [88] and hence, we do not address videos explicitly in this work.

In the rest of this section, we elaborate on the different functionalities of FlexiWeb.

5.3.1 Splitting requests

Current proxy-assisted browsers either fetch all or none of the objects in a web page, via a middlebox. However, as seen earlier, this model can hurt page load times when network conditions are excellent. To minimize the page load times given the conditions, FlexiWeb seeks to identify which objects should be diverted to the proxy and which ones should not. Based on our measurements in Section 5.2, we create a mapping between how objects should be retrieved based on their sizes and current network conditions; Table 5.2 depicts this mapping. The *Request Splitting Module* uses this mapping to dynamically send the request for an object either directly to the web server or to the proxy.

Dynamically selecting how to fetch an object based on its size is challenging because any object's size is not readily available to the client. After fetching the main HTML file (e.g., index.html) of a web page, current browsers parse the file to determine the next object that is to be fetched in order to render the web page. At this point, the browser only knows the object's URL; the object size information is yet unknown. While the browser could try to determine an object's size by issuing a HTTP HEAD request [89] to the web server hosting the object, this would add significant overhead since it imposes an additional round trip of wide-area communication for every object.

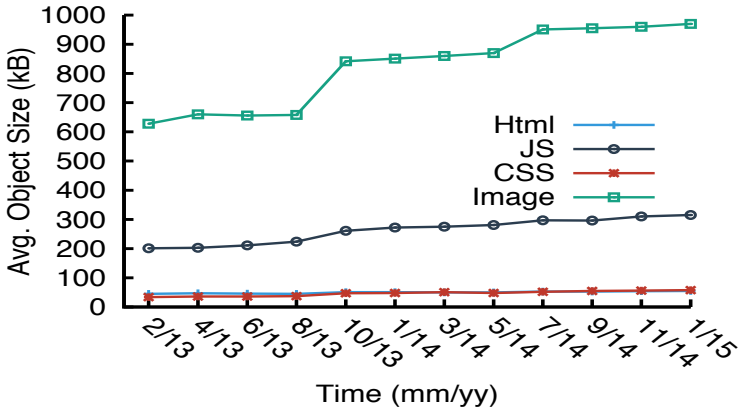


Figure 5.10: Average object size of top 4000 Alexa web pages over last 2 years

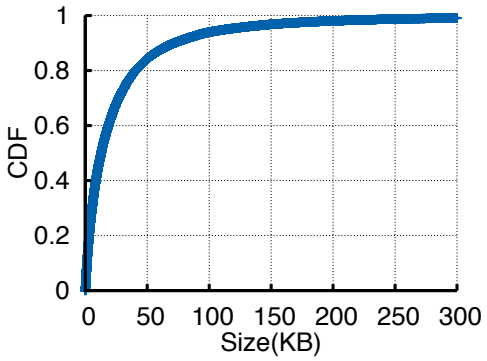


Figure 5.11: Sizes of all objects in top 4000 Alexa Web pages.

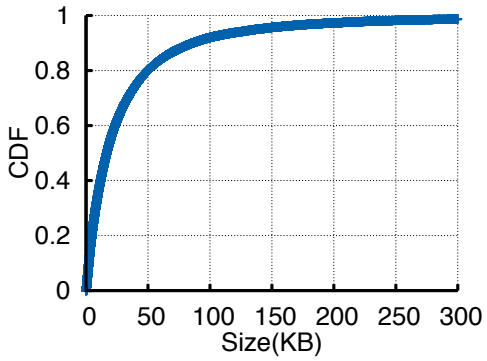


Figure 5.12: Sizes of image objects in top 4000 Alexa Web pages.

5.3.2 Predicting object sizes

To predict the size of an object given only its URL, we rely on learning and applying a predictive classifier. Based on the objects that it fetches over time the proxy continually builds this classifier, and periodically sends its prediction model to the client. The data from Alexa's top 4000 web pages gathered by http archive [88], shows that the average size of web page objects changes only once every few months (as shown in Figure 5.10). Thus, it suffices that the update frequency of the prediction model is set to (say) every few months.

Training the model: To construct a prediction model, a dataset to train the model is required. The middlebox records the URLs of the objects that it fetches and their

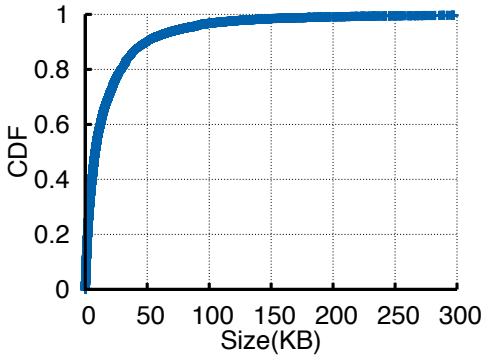


Figure 5.13: Sizes of CSS, JS, and HTML objects in top 4000 Alexa Web pages.

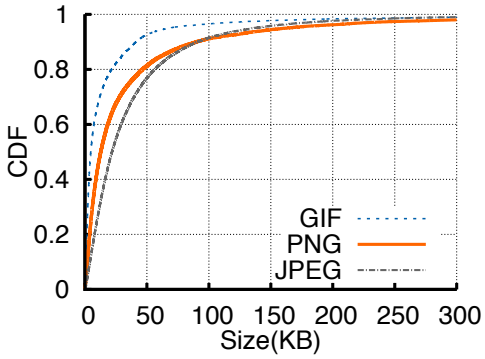


Figure 5.14: Sizes of different types of images in top 4000 Alexa Web pages.

corresponding sizes, and it feeds this information as input to a machine learning algorithm to predict the sizes of objects not seen thus far. Figure **??**(a) depicts the distribution of object sizes in such a sample dataset, which comprises all objects seen on Alexa's top 4000 websites. We see that over 80% of objects are smaller than 50 KB. This is also true in the case of images (Figure **??**(b)). Figure **??**(c) shows that objects with text content (JavaScript, HTML, and CSS) are even smaller. We find that images are one of the biggest contributors to large web page sizes, and hence, we looked at the distribution of the sizes of different image types. Figure **??**(d) shows that GIF images are much smaller in size than PNGs and JPEGs.

Rather than attempt to predict the precise size of every object, which is likely an impossible task, we leverage the fact that we only need to map every object to one of the bins in Table 5.2. Therefore, the problem at hand is to accurately predict the size range given an object's URL.

Extracting features: To perform this classification of object URLs to size ranges, we extract various features from every URL. First, we extract a "bag of words" [90] from the host name in the URL. For example, the host name in the URL `http://i.cdn.turner.com/cnn/.e/img/3.0/global/icons/gallery_icon2.png` yields the following bag of words: {i, cdn, turner, com}. In addition to features extracted from the host name, we include features from the URN and the type of content. Hence, the feature set for the above URL would be domain = {i, cdn, turner, com}, urn = {cnn, .e, img, 3.0, global, icons, gallery_icon2.png} and type = png.

Classifier: The problem of classifying a URL into one of the size ranges is a multi-class classification problem, i.e., it is a classification task with more than two classes. Many multi-class classification algorithms make the assumption that each sample is assigned to one and only one label; in our case, this means that an object's URL can only be "classified" into one size range.

The random forest learning algorithm [91], which is known to perform well for multi-class classification problems can be utilized at the proxy for our purposes. It uses multiple decision trees and the final decision is made based on an aggregation of different decision outcomes from the individual trees. The main parameters of the random forest algorithm are the maximum depth of each (individual) tree, the maximum number of trees, and the maximum number of features used to train the model; details can be found in [91].

Execution: In summary, the object size prediction component of FlexiWeb executes as follows. First, the proxy labels every URL in the training dataset by assigning each object to the appropriate class based on its size. Next, the proxy uses the random forest learning algorithm to generate a classifier model based on the extracted features from the labeled dataset. The client browser then uses this model to predict the size of the object associated with each request, and subsequently (in conjunction with network condition assessments), it makes a decision on whether to retrieve the object via the proxy or directly from the source.

5.3.3 Assessing Network Conditions

An assessment of network conditions by the *Network measurement module* at the proxy serves two purposes. First, it enables the *Request splitting module* at the client to determine whether to retrieve each object from the source directly or via the proxy. Second, it also enables the proxy to make an informed decision on the extent to which it should compress each object. To assess network conditions, the *Network measurement module* at the proxy tracks the RTT, the loss rate, and the TCP throughput to the client.

A simple way to accurately estimate network conditions is to perform measurements in the background; the client and the proxy would exchange packets periodically to infer network conditions. However, such an approach will be prohibitive in terms of bandwidth and energy overheads on the client side. In FlexiWeb, to enable accurate estimation of the average RTT and throughput, the client browser always downloads the main HTML

file for every web page via the proxy. The main HTML file is typically sufficiently large for the purposes of TRINITY's estimation of network conditions, since the HTML's transfer requires the transmission of tens of packets. This also solves the issue of cold start, when the client browser is sending a page request for the first time.

From the HTML's transfer, the *Network measurement module* obtains a sequence of RTT samples, and uses the median sample value to estimate the RTT to the client; we borrow this approach from [92]. Since web page loads last only tens of seconds, these RTT measurements suffice in predicting subsequent data transfer delays to the specific client.

The proxy sends this "network condition report" back to the client in the object response headers of the requests; this enables the client to determine the network conditions (excellent, good, fair or poor as in Table 5.1) and how to fetch other objects on the web page (using Table 5.2).

5.3.4 Network Aware Compression

Current practices: Current proxy-assisted browsers transform content in a pre-determined manner. For example, scripts, CSS files, and other text-based content are minified and zipped before they are sent to the mobile client [93]. On the other hand, proxies convert images to a fixed image format with a fixed compression ratio [85]. For example, Google compression proxy's recommended setting is to transform images to Google's *WebP* image format with a pre-defined quality; WebP is known to have much better performance in comparison to other image formats (e.g., the average WebP file size is 25–34% smaller compared to the JPEG file size with equivalent image quality).

Problem formulation: In contrast to the status quo, we seek to adaptively transform any web page's content in a manner that enables us to deliver the page within the user's attention span, irrespective of the client's network conditions; a user's attention span has been shown to be in the 2 to 5 second range previously [81].

To deliver a web page's content within the user's attention span, in this work, we focus only on *adaptively* transforming images on the page. Since images make up around 65% of the bytes on the average web page [88], transformation of images (to either significantly compressed versions or lower qualities) can significantly reduce page load times. Text and Javascript are compressed as with traditional proxy assisted browsers (to static extents). The adaptive transformation of images is based on assessed network conditions. If network

conditions are good very little compression is invoked; on the other hand, if conditions are poor, higher levels of compression (including a lowering of the image quality) are in play. Our focus here is primarily to achieve the right balance between page load time and user experience. Reducing the download time of an image by applying an appropriate transformation is important, but at the same time, quality or utility of the transformed image is also important. To achieve the right balance, we associate each transformed version of an image with a cost and a utility.

Cost: We define the cost of a transformation as the time taken to download the transformed image from the proxy to the client. Since TCP throughput can be modeled as $\frac{MSS}{RTT \times \sqrt{p}}$ [94] (where MSS is the maximum segment size, RTT is the round-trip time between the proxy and the client, and P is the packet loss rate), we compute the download time for an image of size S bytes as $\frac{S \times RTT \times \sqrt{P}}{MSS}$; the proxy measures the RTT and loss rate using the technique described earlier. Here, the size S corresponds to the *transformed size* of the image. Thus, the higher the degree of compression due to the transformation, the lower will be the cost.

Utility: We use the Peak signal-to-noise ratio (PSNR) as the utility resulting from a transformation. PSNR is the most widely used metric to quantify the quality or utility of images. It essentially captures the relative quality between two images and is defined as:

$$PSNR = 10log\left(\frac{MAX^2}{MSE}\right),$$

where MAX is the maximum possible pixel value (defined next) and MSE is the mean-squared error given by:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2.$$

In the above, the two images are of size $m \times n$ and $I(i,j)$ and $K(i,j)$ are the pixel values in the two images at position (i,j). The pixel values essentially reflect the intensity level in each pixel; thus, if the intensity is represented using 8 bits, then MAX will be 255. For color images (with three RGB values per pixel), note that the MSE is the average over all RGB values. With higher levels of compression or transformation, MSE increases and thus, PSNR decreases.

The proxy calculates the PSNR (utility) with each transformation with respect to the original image. Clearly, the more aggressive the transformation, the lower will be the utility of the image.

Cost vs. utility trade-off: As evident from the above discussion, a higher degree of compression reduces both utility and cost; an attempt to either decrease cost or increase utility adversely affects the other. Thus, though FlexiWeb should ideally maximize utility and minimize cost, both objectives cannot be realized simultaneously. Instead, it seeks to do the best it can in reducing cost and increasing utility as follows.

For tractability, we consider a limited number of transformation choices, and the proxy chooses from one of these options depending on network conditions. The set of transformations range from high utility and high cost to low utility and low cost: WebP with 85% quality, WebP with 65% quality, WebP with 45% quality, WebP with 25% quality, WebP with 5% quality, and a gray scale version of the image. As one might expect, the lowest quality image (gray scale) has the least associated cost but also offers the least utility; the highest quality image provides the highest utility but also incurs the highest cost.

Processing times: In the above discussion, we characterize the cost associated with a transformation to be the time taken to download that specific transformed version of an image. In reality, there is an additional cost due to processing the image. We find that regardless of which transformation we perform on each specific object, the processing delay does not change by much. We perform all of the transformations on 1000 images from Alexa's top 500 webpages and we find that the processing times are almost identical; in the worst case, the difference in processing time is about 8 % for a very few objects (we omit results due to space constraints). Thus, the processing cost does not influence the choice of transformation and is hence ignored here.

Compressing objects based on network conditions: Utilizing the cost vs. utility characterization of web content as above, we cast the proxy's task of performing network aware compression as follows.

Suppose that the client needs to download a total of N images in sequence over the duration of a web page download, and that it has a total page load time budget of B seconds (within which the page should be ideally downloaded). For every image object i, let us assume that the proxy can apply M_i different transformations. The question then is "which transformation should the proxy apply on each image ?" The overall goal here is to maximize the sum of utilities (i.e., PSNR values) of all the selected versions for

the N images, subject to the constraints that (i) exactly one version for each requested image must be selected, and (ii) the total cost of all the selected versions must be within budget B (within the desired page download time). Note that maximizing the sum of the PSNR (chosen as the utility) values corresponds to a minimization of the sum of expected distortion over the images in the web page; the lower this value, the better the quality of the downloaded content and thus, user experience.

The above optimization can be formulated as follows:

$$maximize_{x_{ij} \in \{0,1\}} \quad \sum_{i=1}^{N} \sum_{j=1}^{M_i} u_{ij} x_{ij} \tag{5.1}$$

$$\text{subject to} \quad \sum_{j=1}^{M_i} x_{ij} = 1, \forall i$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M_i} c_{ij} x_{ij} \leq B,$$

Here, the indicator variable x_{ij} is 1 if version j of image i is selected, and 0 otherwise. u_{ij} and c_{ij} are the utility and cost, respectively, associated with version j of image i.

The transformation selection problem as formulated in Equation 5.1 maps to the Multi-Choice Knapsack Problem (MCKP). The time budget is the size of the knapsack, and the goal is to fill in objects (images) such that the total utility of the objects in the knapsack is maximized. However, in our case, a client's requests for objects are made online, i.e., the middlebox does not have a priori knowledge of future image requests since the objects are dynamically fetched. Thus, we need an online algorithm for solving the MCKP problem.

The online MCKP problem is well studied. Competitive online algorithms have been proposed [95] with the relaxed constraint $\sum_{j=1}^{M} x_{ij} \leq 1$ for all i, i.e., the algorithm need not to choose a version for each image (image can also be discarded). We modify the relaxed online MCKP algorithm designed by Zhou et al. [95] for solving our problem. Specifically, (i) we do not allow for the discarding of images and (ii) we modify the time budget dynamically to account for time spent either due to transfer of other content (e.g., compressed Javascripts or text) or inactivity due to waiting for objects. However, these uncertainties can cause us to exceed the original target time budget. Thus, our modified

goal is to try and stay within the budget, but if we are to exceed it, to do so to the minimum extent possible.

Now, the online transformation selection problem 5.1 can be transformed into the following relaxed problem with the inequality constraints shown; if an online algorithm for the relaxed problem has competitive ratio C (the performance of the algorithm w.r.t the optimal), then, so does the original problem in Equation 5.1.

$$maximize_{x_{ij} \in \{0,1\}} \quad \sum_{i=1}^{N} \sum_{j=1}^{M_i} u'_{ij} x_{ij} \tag{5.2}$$

$$\text{subject to} \quad \sum_{j=1}^{M_i} x_{ij} = 1, \forall i$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M_i} c'_{ij} x_{ij} \leq B,$$

In the above formulation, the key idea is to subtract the utility and cost of the cheapest transformation of each image from other transformations of the same image, and use these *modified* utilities u' and costs c' as the new representations associated with the corresponding transformations. If the relaxed problem does not choose a transformation, in our case we default to the cheapest transformation. The reader is referred to [96] for the proof of the above claim.

The algorithm provides a competitive ratio of $log(\frac{U'}{L'}) + 1$, where the utility to cost ratio of the transformations are upper and lower bounded by U' and L', respectively. Zhou et al. also assume that all transformed versions are much smaller in size than the knapsack, which is true of objects on a web page (each image is transferred in a time $<< B$).

Algorithm 6 presents our approach to modifying and using Zhou et al.'s solution to our problem of selecting the appropriate transformations of images at the FlexiWeb proxy. The intuition behind the algorithm is as follows. We only want to pick image transformations that have a sufficiently high utility to cost ratio, i.e., ones that satisfy a certain efficiency threshold. As we spend more budget, this threshold should increase (in other words, the cost for transferring objects later in the page load should decrease). This efficiency threshold is captured by a function $\phi(b(t), B) = (\frac{U}{L})^{\frac{b(t)}{B}} \frac{L}{e}$, where $b(t)$ is the amount

Algorithm 6 Online selection of best image transformations

Initialize: $b = 0$, Q (QoE level)
Event: Image i requested at time t_i
if $Q \leftarrow auto$ **then**
  $M_i \leftarrow M_i$
else
  if $Q \in \{1, 2, 3, 4, 5\}$ **then**
    $M_i \leftarrow M_i - Q$, by removing the lower Q transformations
  end if
end if
for $j = 1 \rightarrow M_i$ **do**
  $u'_{ij} \leftarrow u_{ij} - u_{i1}$
  $c'_{ij} \leftarrow c_{ij} - ci1$
end for
$transformations \leftarrow \{j | \frac{u'_{ij}}{c'_{ij}} \geq \phi(b, B)\}$
if *transformations* is not empty **then**
  $J \leftarrow \arg\max_j \{u'_{ij} | j \in transformations\}$
  if $b + c'_{ij} \leq B$ **then**
    $selectedTransform \leftarrow J$
  else
    $selectedTransform \leftarrow 1$
  end if
else
  $selectedTransform \leftarrow 1$
end if
Deliver transformation *selectedTransform*
if time $t_i \geq b$ **then**
  $b \leftarrow t_i + c'_{i,selectedTransform}$
else
  $b \leftarrow b + c'_{i,selectedTransform}$
end if

of the budget spent at time t.[2] $\phi(b(t), B)$ increases with $\frac{b(t)}{B}$, the fraction of the knapsack filled at time t. Therefore, as time progresses, fewer transformations satisfy the efficiency condition.

Referring to Algorithm 6, the proxy first initializes the current usage b to 0. Then, the transformations $u_{ij} \longrightarrow u'_{ij}$ and $c_{ij} \longrightarrow c'_{ij}$, are performed. The proxy then considers the set of transformations that have a utility-to-cost ratio that is higher than $\phi(b, B)$. It chooses the transformation that has the maximum utility and fits within the budget; otherwise (i.e., if none fit within the budget), it chooses the cheapest transformation. At the end of each step (i.e., after an image is transformed and sent to the client), the proxy updates b with the cost of the selected transformation. The current budget usage is updated

[2]We perform a one time estimation of U and L offline, using a training set of images.

after the selection of a transformation. In case of inactive time periods (because of waiting for objects) an increase in knapsack usage is applied (lines 19-22 in Algorithm 6). Whenever static compression is applied to objects such as JS, CSS etc., we again apply an increase in knapsack usage (as with inactivity periods). If the target budget is exceeded, then the middlebox simply performs the cheapest transformation on each image, prior to sending it to the client. Note here that we simply subtract an object's cost from the budget, despite the client fetching multiple objects on a page in parallel, because we consider bandwidth to be the bottleneck between the client and the proxy.

QoE vs. page load latency: While transforming content, it is important to ensure that web browsing has a good, associated Quality-of-Experience (QoE). While transformations reduce delays, they must still cater to a user's or a web provider's requirements in terms of quality (e.g., a user may not want gray scale transformations). Hence, users or web providers can set a minimum quality limit either via browser settings or via meta-data embedded in a web page's HTML; this sets the maximum allowable compression level (QoE level) allowed on each object in a web page. The QoE level is set to an auto mode by default. In this mode, all possible (M_i) transformations are considered (when applying Algorithm 1). Instead of the auto mode, the user or the provider can specify a QoE level from 1 to 5. If QoE level 1 is chosen, Algorithm 1 can use $M_i - 1$ transformations; the the lowest quality transformation is left out. If a QoE level of 5 is chosen, Algorithm 1 can only apply a single transformation, viz., the one that performs the minimum compression and retains the highest quality (WebP with 85% quality). Note that the higher the level of QoE chosen, the larger could be the deviation (increase) with respect to a desired page load time. At this time, FlexiWeb chooses the higher of the QoE levels specified by the user and the provider.

5.4 Implementation and Setup

In this section, we first describe our implementation; subsequently we elaborate on our experimental setup.

5.4.1 Implementation of FlexiWeb

Client side implementation: We modify Google's Chromium 38.0.2125.50, an open source Android browser to implement the client side modules of FlexiWeb. Specifically,

we make changes to the *url_request* module in Chromium to 1) determine whether or not to go through the proxy, for each object, and 2) to predict the size of any object based on its URL. We also modify Chromium to store network measurements in the browser cache after retrieving them from the HTTP request response header sent by the proxy. For each web page, the request for the main HTML file always goes through the compression proxy; this allows the proxy to gather accurate network condition estimates (RTT and throughput). These estimates are returned to the client in a custom field in the HTTP response (sent by the proxy). The estimates stay valid for 15 seconds, and if no object is fetched via the proxy by then, the client sends the immediately following request via the proxy. Based on an object's size and the current network conditions (stored in the browser), our modified version of Chromium chooses between retrieving the object via a proxy or directly from the source.

Proxy side implementation: We configure the compression proxy with 4GB memory and 2.40 GHz Intel Core 2 CPU, running Ubuntu 12.04. It is implemented using Google's mod_pagespeed-1.8.31.4 module running on top of an Apache 2.2 web server. TCP CUBIC is used. The Apache web server runs in the forward proxy mode. All the requests pass through mod_pagespeed module before content is compressed. We implement the passive RTT, throughput and loss rate estimation scheme described in [94] as a kernel module; the estimated values are exposed using the Linux socket interface, to other programs and modules. Specifically, these values are used by the network aware compression module to calculate the cost of a transformation. We implement the network-aware compression module by modifying the mod_pagespeed's image transformation module to invoke network-aware compression (using Algorithm 1) instead of static compression. To report the network conditions to the client, we implement an Apache module to control and modify the HTTP response headers. This Apache module reads the current RTT, loss rate and throughput values and adds them to every out going response to the client using the previously discussed custom header field.

5.4.2 Experimental Setup

Scenarios for evaluation: We have three different scenarios in which we evaluate FlexiWeb.

Controlled settings: To reduce variability and control network conditions, we tethered our client phone to a laptop using a USB connection and applied traffic shaping to the tethered connection using Dummynet [97]. We emulated a 4G network with an uplink bandwidth of 1 Mbps [98] and with varying downlink bandwidths shown in Table 5.1.

An automated script was used to change the values of the link bandwidth, the loss rate and RTT for every test case.

Static clients on cellular networks: To evaluate FlexiWeb on real cellular networks, we use AT&T and T-Mobile's 4G network connections. To create different network conditions we run experiments at the same locations that we used to gather our measurements in Section 5.2. Due to space constraints, we only present results from AT & T's network except in some sample cases.

Mobile scenarios: Finally, we also evaluate FlexiWeb in mobile scenarios. We choose around 20 paths in two US metropolitan areas in two states with a mix of local streets and Interstate highways. Our speeds on interstate highways were around 65 miles per hour while on city streets our speeds were 30-40 miles per hour. We omit the exact details of the paths to preserve anonymity.

Web pages requested: We use the top 500 web sites visited by mobile users to run our tests (the top Alexa sites). These have a good mix of news websites, online shopping and auction sites as well as professionally developed websites of large corporations. These websites contain anywhere from 5 to 323 objects. The objects in these sites were spread across 3 to 84 domains. Each web site had HTML pages, Javascript objects, CSS and images. Due to experimentation constraints (time), the experimental results reported for mobile scenarios are only using the top 50 web pages.

Execution: We setup 2 rooted mobile devices (HTC One phones with Android 4.3) running Chromium (with our modifications) for Android. We generated a random order in which to visit the web sites and used that same order across all experiments. The period between website requests was set to 60 seconds both to allow for websites to load completely, and to reflect a nominal think time that users typically indulge in between requests. If the web page took a much shorter time to load, the system was idle until the 60 second window elapsed. We used the "page load time" as the primary metric of performance. We alternated our test runs between "Direct", "Compression Proxy" and "FlexiWeb" to ensure that temporal factors did not affect our results. With Direct, requests are directly sent to the source web server. With Compression proxy, all the requests are sent to a proxy

with static compression settings where images are transformed to WebP with a quality of 75% (this is what is done with commercial proxies today [99] to achieve some compression without compromising the quality significantly). We ran each experiment multiple times at different times during the day.

5.5 Evaluation of FlexiWeb

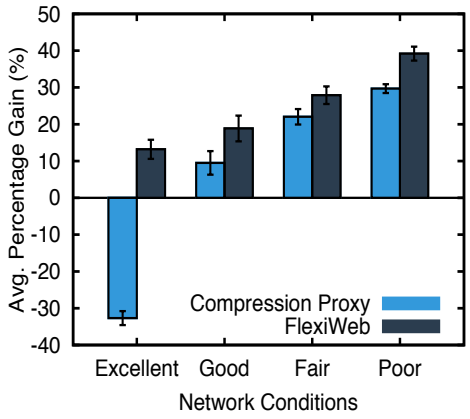


Figure 5.15: Performance gains with FlexiWeb

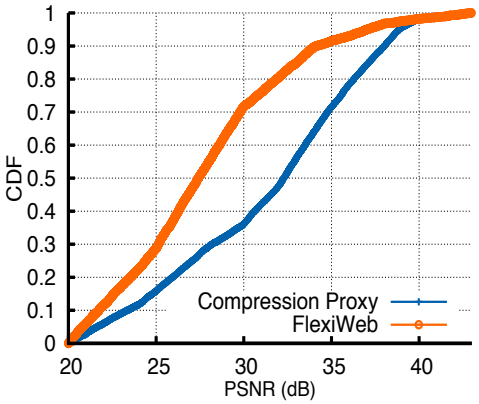


Figure 5.16: PSNR of transformed images

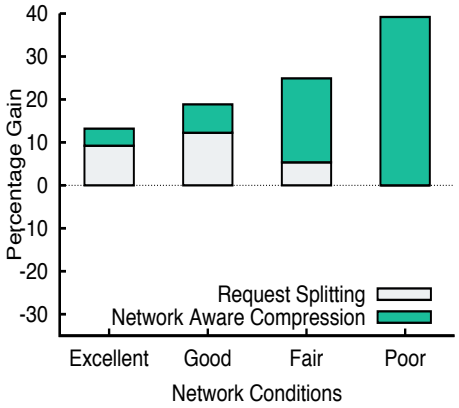


Figure 5.17: Impact of request splitting and network aware compression on FlexiWeb's gains

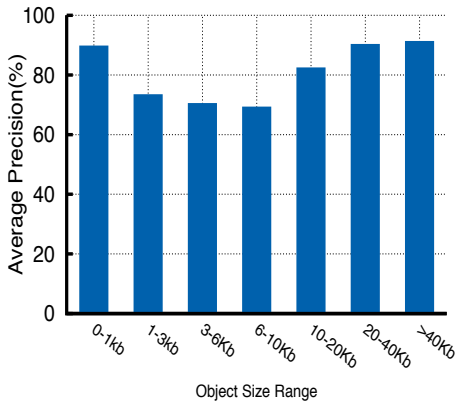


Figure 5.18: Precision in predicting object sizes

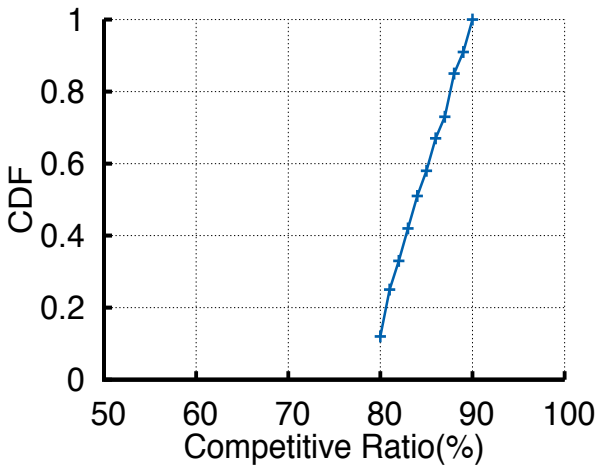


Figure 5.19: Competitive analysis of online MCKP

We evaluate FlexiWeb extensively, in each of the scenarios described in Section 5.4, and discuss the results below.

Performance of FlexiWeb in controlled settings: We first evaluate FlexiWeb in the controlled network settings listed in Table 5.1. We calculate the percentage gains relative to the direct scheme (no proxy) for both (i) FlexiWeb and (ii) compression proxy based browsing. Figure 5.15 shows the average percentage gains in page load times under different network conditions. The results, shown in Figure 5.15, indicate that the compression proxy degrades the performance in terms of page load times, by up to 32% as compared to direct browsing in excellent network conditions (also seen earlier). As network conditions degrade, the compression proxy outperforms direct browsing; in all other settings the reduction in object sizes (from compression) outweigh the latency overheads of longer routes and processing. We see that FlexiWeb outperforms compression proxy based browsing by up to 45% in excellent network conditions. FlexiWeb significantly gains from sending requests for small objects directly to the web server; it thereby avoids the latency overheads incurred in retrieving such objects via the proxy. FlexiWeb still downloads the large objects via the proxy, but after performing network-aware compression (large objects benefit from compression). As network conditions deteriorate, the performance gains with FlexiWeb in comparison to compression proxy based browsing diminish (only about 10 %). This is because the gains due to compression now increase and almost all objects are now

retrieved via the proxy; FlexiWeb outperforms the compression based proxy here, mainly due to network-aware compression.

Next, we present the PSNR values of the transformed images with FlexiWeb and the traditional compression proxy. Figure 5.16 shows that the average degradation of PSNR using FlexiWeb is only $\approx$ 4 dB in comparison to compression proxy; this is in spite of FlexiWeb performing more aggressive compression during poor network conditions.

How does FlexiWeb provide performance gains?: Next, we examine the gains from the two main functions performed by FlexiWeb i.e., request splitting and network-aware compression. We analyze the web page download traces and separate the objects into those retrieved directly from the source web server and those via the proxy. We measure the gain with respect to each object's load time due to each function, in comparison to the load time of the object when fetched directly from the server. We then calculate the average gain across all objects fetched, with respect to each function. From Figure 5.17, we see that in excellent and good network conditions the gains are primarily due to objects being fetched directly from the source web server; in excellent conditions, almost 70% of the gains are due to this. As network conditions degrade the number of objects fetched via the proxy increases and the gains due to network-aware compression dominate. Since in poor network conditions, FlexiWeb fetches all objects via the proxy, all the gains are due to network-aware compression.

Accuracy of object size predictions: Next, we evaluate the accuracy with which FlexiWeb predicts the size of objects. We use the data collected at the proxy as described in Section 5.3. We use the 10 fold cross validation method [100]. We divide our dataset into two parts: a training set that contains 30% of the dataset, and a test set that contains the remaining 70%. We train our model using the training set and later perform prediction on the remainder (70 %) of the data which was held out. For the purposes of evaluation, we use the *precision criterion* [101], which determines the fraction of records that are correctly classified from among those that that are grouped under a classification. Specifically,

$$Precision(\%) = \left(\frac{TruePositive}{TruePositive + FalsePositive} \right) \times 100.$$

Figure 5.18 shows that the accuracy of predicting the size of objects based on the URL text is quite high. We can predict objects of size 0 to 1 KB and objects larger

than 20 KB with more than 90% accuracy. We can predict other object sizes with at least 70% accuracy. Note that an inaccurate prediction simply causes FlexiWeb to fetch the object sub-optimally (directly instead of via the proxy or vice versa). Thus, while it slightly causes a degradation in FlexiWeb's performance, it does not disrupt the web browsing process. Note here that our prediction has high accuracy because we only seek to predict "the" range into which an object's size falls, and not the object's precise size.

Quantifying the sub-optimality of network-aware compression: For network-aware compression, we use the modified online MCKP algorithm shown in Algorithm 6. To illustrate the difficulty of the online transformation selection problem, consider an end user with a time budget of 5 seconds to download a page. Let the download of version j of image i be represented by a utility-cost pair (u_{ij}, c_{ij}), where u_{ij} is the utility (e.g., PSNR or negative distortion) and c_{ij} is the cost. Suppose the first image has two possible versions with utility-cost pairs (1,1), (2,2), and the second image also has two versions with utility-cost pairs (2,1),(4,2). If the user requests for both the images sequentially, then the optimal (offline) solution is to choose the stream (1,1) followed by (4, 2). This gives a total utility of 5 with a total cost of 3. However, an online transformation selection algorithm must first choose a stream from $(1, 1), (2, 2)$, without knowledge of the subsequent request. Our online MCKP algorithm is likely to choose the transformation (2, 2) followed by (2, 1), resulting in sub-optimal utility.

Here, we seek to quantify the sub optimality of Algorithm 1. We select 100 web pages from the Alexa's top 500 web pages with varied number and sizes of the objects. We first download the pages using a modified version of MCKP, called OPT-MCKP. OPT-MCKP [102] is an offline version of MCKP, where the object requests are known in advance. With this information OPT-MCKP can always choose the right transformation. We repeat the same experiment using our ON-MCKP algorithm (Algorithm 6) . A popular way to evaluate online algorithms is using what is called the "Competitive Ratio" [103]. The competitive ratio of an online algorithm for an optimization problem is simply the ratio between the cost of the solution found by the algorithm and the cost of an optimal solution. Figure 5.19 shows the CDFs of Competitive ratio (converted to percentage) across the 100 pages; we see that 80% of the time, the page load times with ON-MCKP are within 10%-20% of that with OPT-MCKP.

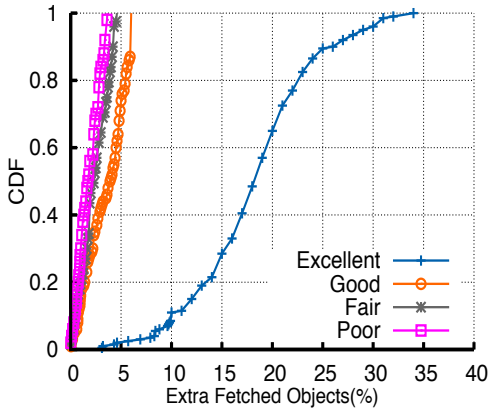


Figure 5.20: Percentage of extra fetched objects in a time budget of 5 seconds

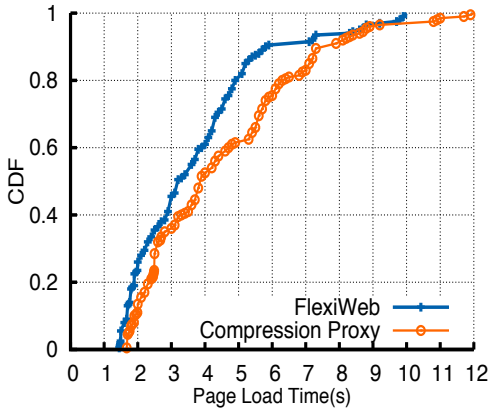


Figure 5.21: Gains in page load times with FlexiWeb and Compression Proxy

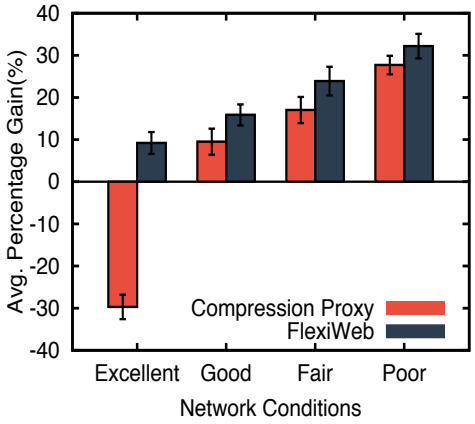


Figure 5.22: Performance of FlexiWeb on T-Mobile's network

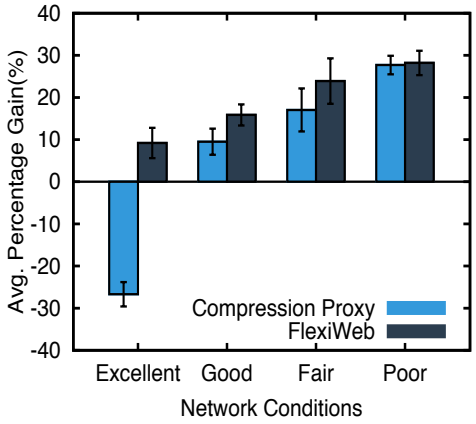


Figure 5.23: Performance of FlexiWeb on AT&T's network

Number of objects and web pages downloaded within a time budget: Next we evaluate FlexiWeb in terms of number of objects retrieved within a specified time budget of 5 seconds. We compare the number of requests satisfied by FlexiWeb in comparison with conventional compression proxy based browsing. Figure 5.20 depicts the CDF of the percentage of additional objects retrieved by FlexiWeb as compared to conventional compression proxy based browsing; we see that FlexiWeb fetches 17% more objects on average, within the same time budget. FlexiWeb achieves this by fetching small objects directly from the web server in excellent and good network conditions; the decrease in per-object download times allow the downloading of more objects within the page load

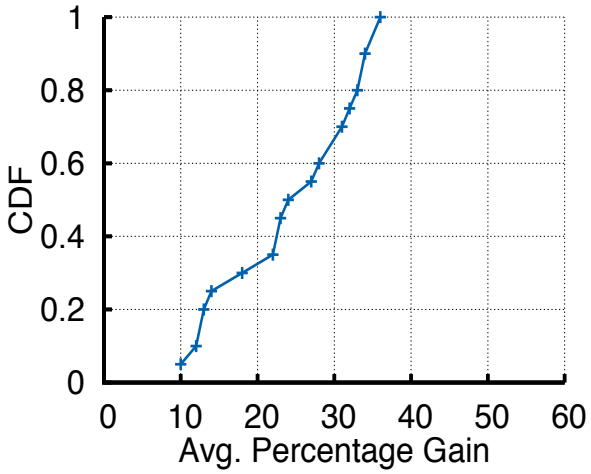


Figure 5.24: Gains with FlexiWeb in mobile settings

time budget. Even in bad network conditions FlexiWeb fetches 1-2% more objects on an average than conventional compression proxy based browsing, but mainly due to network aware compression. An increase in number of additional fetched objects also translates to an increase in the number of web pages downloaded in the target budget. Figure 5.21 shows the CDF of page down load times of web pages using regular compression proxy and FlexiWeb. We see that FlexiWeb can download about 19% more web pages on average, within a target budget of 5 seconds.

Evaluations on AT &T and T-Mobile's networks: We evaluate FlexiWeb on real cellular networks to show that the gains seen in controlled settings also exist in real cases. From Figure 5.22 and 5.23 we see that FlexiWeb outperforms conventional compression proxy based browsing in excellent network conditions (by up to 38% with T-Mobile and 37 % with AT & T) In poor conditions it out performs the latter by up to 6% with T-Mobile and 2 % with AT & T. The gains are in between for other network conditions.

Scenarios with mobility: Next, we seek to evaluate FlexiWeb in scenarios where a passenger accesses web pages while in a moving car. We choose 20 different paths to capture typical scenarios: (i) driving on a highway at high speeds and (ii) driving around in a city at moderate speeds. We used two mobile devices (both HTC One phones) connected to two laptops via USB, one downloading web pages via a conventional compression proxy while the other does so using FlexiWeb. We choose to download Alexa's top 50 web pages for this experiment. Both the devices are connected to AT & T's network. In addition to

page load times we also keep track of signal strengths at both the devices. We start the experiments on both the phones at almost the same time to avoid variations in network conditions. We measure the average percentage gains with FlexiWeb while downloading all the web pages. In Figure 5.24, we plot the CDF of this average percentage gain over all the webpages downloaded on the different paths; we see that 80% of the time, FlexiWeb provides an average percentage gain of 34 % over compression proxy based browsing, in terms of page load times. In 20% of the cases, it only provides 14% gains. Upon closer inspection, we found that some paths were experiencing excellent to good signal strength throughout while others were experiencing low signal strengths much more often than good signal strengths. It is well known that excellent/good signal strength is highly correlated with good network conditions (low RTT and loss rates, high throughputs) while the opposite is true with low/poor signal strength [104]. In excellent/good network conditions FlexiWeb significantly outperformed conventional compression proxy based browsing (since FlexiWeb retrieved most objects in the web pages directly and only few large objects via the proxy). On paths where network conditions were bad, both the conventional compression proxy based browser and FlexiWeb, fetched almost all objects via the proxy. FlexiWeb outperformed conventional compression proxy based browsing because of its network-aware compression module.

5.6 Related Work

Measurement studies: There is prior work on characterizing the properties of web pages, and protocols that impact mobile web browsing. The study in [105] shows that optimization of compute intensive tasks at proxies only offers marginal gains. In [106], the complexity of web sites is studied via browser-based active measurements. Other studies have evaluated the benefits of existing mechanisms for improving mobile web performance. Erman et al. [107] showed that SPDY [108], a recently proposed alternative to HTTP, does not clearly outperform HTTP over cellular networks. They identify the disharmony between TCP and cellular networks as the underlying cause. Similarly, Sivakumar et al. [109] showed that proxy-assisted thin client browsers, such as Amazon Silk [79], do not provide clear benefits in terms of page load time and energy. None of these efforts study the implications of using `Proxy Assisted` mobile web browsing in different network conditions, as we do here.

Client-based solutions: Client-side solutions have been proposed to improve mobile web performance. WebSieve [110] generates mobile-friendly websites from the original desktop versions. Zoomm [111] speeds up web page loads by parallelizing the execution of dynamic components of any web page. Adrenaline [112] parallelizes the fetching of web pages by decomposing existing web pages on the fly into loosely coupled mini pages, and loading mini pages in parallel via separate processes.

User studies: Lymberopoulos et al.'s user study [113] shows that mobile web browsing exhibits a strong spatio-temporal signature, different for every user. Based on this study, they propose a machine learning based model to accurately predict future web accesses for a user, and they use this prediction to prefetch content in a timely manner. Wang et al. [114] show that caching and prefetching provide very limited benefits for mobile web browsing, but speculative loading can decrease page load times by up to 20%.

Unlike FlexiWeb, none of the above approaches consider modifying web clients to dynamically select when to use proxies for data compaction.

Proxy-based solutions: Due to the computation and bandwidth limitations on mobile devices, researchers have proposed to offload various types of functionality from client devices to proxies. For example, Zhao et al. [76] offload execution of dynamic content to the proxy, while Cho et al. [115] propose the delegation of DNS lookups and TCP connection establishment to the proxy. Chava et al. [116] try to reduce the usage cost incurred by the end-user by computing a cost quota for each web request, and having a proxy adapt the web page accordingly. The cost quota for each web request is dynamically calculated based on the pricing plan of the user and her current data usage levels. Recently, Google [85] and Nokia [117] have incorporated data compression proxies in their mobile web browsers; these proxies (hosted in data centers) are expected to reduce cellular data usage and speed up mobile web browsing. Wang et al. [75] propose a framework that allows the execution of "any" portion of the page load process in the cloud (unlike browsers such as Opera Mini that only allow fixed parts to be executed in the cloud).

None of these proxy-based solutions are network-aware. While some of the proprietary solutions are not documented, to our knowledge, all existing systems always fetch content via the proxy, unlike FlexiWeb; further, they apply the same content compression irrespective of network conditions.

5.7 Conclusions

In this paper, we argue based on an in-depth measurement study that always using cloud-based middleboxes to assist mobile browsing can be detrimental to performance in terms of web page download times. Our measurements reveal that the middlebox should be used only when network conditions are bad; otherwise, most objects in the web page should be directly fetched from the source web server. Based on this observation we build FlexiWeb, a framework that supports network-aware middlebox usage. In addition, FlexiWeb also performs dynamic network-aware compression to provide further performance gains. We demonstrate via extensive experiments that FlexiWeb outperforms conventional compression proxy based browsing by decreasing page load times by as much as 42 %, on average.

Bibliography

[1] K. Zarifis, T. Flach, S. Nori, D. Choffnes, R. Govindan, E. Katz-Bassett, Z. M. Mao, and M. Welsh, "Diagnosing Path Inflation of Mobile Client Traffic," in *Passive and Active Measurement Conference (PAM '14)*, March 2014.

[2] [Online]. Available: http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf

[3] D. Tse and P. Vishwanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[4] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and mac protocols," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, ser. MobiCom, 2009.

[5] A. Bhartia, Y.-C. Chen, S. Rallapalli, and L. Qiu, "Harnessing frequency diversity in wi-fi networks," in *Proceedings of the 17th annual international conference on Mobile computing and networking*, ser. MobiCom, 2011.

[6] "Wireless lan medium access control (mac) and physical layer (phy) specifications: High-speed physical layer in the 5 ghz band," *Part 11, Standard ed.,IEEE802.11 Working Group*, 1999.

[7] T. J. Willink and P. H. Wittke, "Optimization and performance evaluation of multi-carrier transmission." *IEEE Transactions on Information Theory*, 1997.

[8] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, ser. MobiCom, 2006.

[9] "Onoe rate control." [Online]. Available: http://bit.ly/1aNyIi4

[10] M. McKinley, K. Remley, M. Myslinski, J. Kenney, D. Schreurs, and B. Nauwelaers, "Evm calculation for broadband modulated signals," *64th ARFTG Conf. Dig*, 2004.

[11] J. C. Bicket, "Bit-rate selection in wireless networks," Master's thesis, MIT, Tech. Rep., 2005.

[12] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom, 2001.

[13] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, ser. MobiCom, 2002.

[14] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys, 2008.

[15] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Accurate: Constellation based rate estimation in wireless networks," in *NSDI*, 2010.

[16] S. Kittipiyakul and T. Javidi, "Subcarrier allocation in ofdma systems: beyond waterfilling," in *Signals, Systems and Computers. Conference Record of the Thirty-Eighth Asilomar Conference on*, 2004.

[17] C. Y. Wong, R. S. Cheng, K. B. L. R. D., R. D. Murch, S. Member, and S. Member, "Multiuser ofdm with adaptive subcarrier, bit, and power allocation," *IEEE Journal on Selected Areas of Communications*, 1999.

[18] H. Ishikawa, M. Fujii, M. Itami, and K. Itoh, "Bi-directional ofdm transmission using adaptive modulation that spreads data symbols," in *Power Line Communications and Its Applications, IEEE International Symposium on*, 2006.

[19] "Evm reporting notes." [Online]. Available: http://bit.ly/19Aoq4q

[20] R. Steel and J. Torrie, *Principles and procedures of statistics*, 1960.

[21] "Practical manufacturing testing of 802.11 ofdm wireless devices." [Online]. Available: http://bit.ly/1fa33PI

[22] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005.

[23] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack problems*. Springer, 2004.

[24] K. Dudzinski and S. Walukiewicz, "Exact methods for the knapsack-problem and its generalizations," *European Journal Of Operational Research*, 1987.

[25] D. Pisinger, "A minimal algorithm for the multiple-choice knapsack problem." *European Journal of Operational Research*, 1994.

[26] E. Balas and E. Zemel, "An algorithm for large zero-one knapsack problems," *Operations Research*, 1980.

[27] *Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, 2009.

[28] J. Mittag, S. Papanastasiou, H. Hartenstein, and E. G. Strom, "Enabling Accurate Cross-Layer PHY/MAC/NET Simulation Studies of Vehicular Communication Networks," *Proceedings of The IEEE*, 2011.

[29] C.-X. Wang, M. Patzold, and Q. Yao, "Stochastic Modeling and Simulation of Frequency-Correlated Wideband Fading Channels," *IEEE Transactions on Vehicular Technology*, 2007.

[30] X. Zhao, J. Kivinen, and P. Vainnikainen, "Tapped delay line channel models at 5.3 GHz in indoor environments," *Composites Part A-applied Science and Manufacturing*, 2000.

[31] H. Rahul, S. S. Kumar, and D. Katabi, "MegaMIMO: Scaling Wireless Capacity with User Demand," in *ACM SIGCOMM*, 2012.

[32] K. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous mimo networks," in *ACM SIGCOMM*, 2011.

[33] L. S. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden, "Improving Wireless Network Performance Using Sensor Hints," in *USENIX NSDI*, 2011.

[34] "Rice University WARP project," http://warp.rice.edu.

[35] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," 1998.

[36] "DAS in Action : Atlanta," http://bit.ly/ACRt4V.

[37] D. Tse and P. Viswanath, "Fundamentals of wireless communications," 2004.

[38] E. Aryafar, M. A. Khojastepour, K. Sundaresan, S. Rangarajan, and E. W. Knightly, "Adam: An adaptive beamforming system for multicasting in wireless lans," in *INFOCOM*, 2012.

[39] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. Voelker, "Sam: enabling practical spatial multiple access in wireless lan," in *ACM MobiCom*, 2009.

[40] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and Experimental Evaluation of Multi-user Beamforming in Wireless LANs," in *ACM MobiCom*, 2010.

[41] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen, "Rate adaptation for 802.11 multiuser mimo networks," in *ACM Mobicom*, 2012.

[42] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, ser. SIGCOMM '09, 2009.

[43] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "Centaur: realizing the full potential of centralized wlans through a hybrid data path," in *ACM MobiCom*, 2009.

[44] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: a distributed wireless architecture for exploiting sender diversity," in *ACM SIGCOMM*, 2010.

[45] M. Maddah-Ali and D. Tse, "Completely stale transmitter channel state information is still very useful," in *Allerton*, 2010.

[46] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *ACM SenSys*, 2010.

[47] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, "Link quality prediction in mesh networks," *Elsevier Comput. Commun.*, 2008.

[48] X. Chen, P. Gangwal, and D. Qiao, "Practical rate adaptation in mobile environments," in *IEEE PERCOM*, ser. PERCOM '09, 2009.

[49] A. Tarighat, M. Sadek, and A. Sayed, "A multi user beamforming scheme for downlink mimo channels based on maximizing signal-to-leakage ratios," in *IEEE ICASSP*, 2005.

[50] K. Jamieson, "The SoftPHY Abstraction: from Packets to Symbols in Wireless Network Design," Ph.D. Thesis, MIT, 2008.

[51] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: practical many-antenna base stations," in *ACM Mobicom*, 2012.

[52] "Wired is the new wireless," http://bit.ly/UaD2AC.

[53] "Xiph.org video media," http://media.xiph.org/video/derf/.

[54] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017," Feb 2013.

[55] C. Mobile, "C-RAN: The Road Towards Green RAN," http://bit.ly/Ya1zuW.

[56] A. Lucent, "lightRadio Network: A New Wireless Experience," http://bit.ly/VpR4Cb.

[57] M. Y. Arslan, J. Yoon, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee, "FERMI: A Femtocell Resource Management System for Interference Mitigation in OFDMA Networks," in *ACM MobiCom*, Sept 2011.

[58] Q. Wang, D. Jiang, J. Jin, G. Liu, Z. Yan, and D. Yang, "Application of BBU+RRU Based Comp System to LTE-Advanced," in *IEEE ICC Workshops*, Jun 2009.

[59] T. D. Forum, www.thedasforum.org/news/.

[60] U. Barth, "How to reduce green house emissions from ICT equipment: Wireless Networks, EARTH research project," in *ETSI Green Agenda*, Nov 2009.

[61] Ericsson, "AIR: Antenna Integrated Radio," http://bit.ly/dLOIAu.

[62] J. Andrews, W. Choi, and R. Heath, "Overcoming Interference in Spatial Multiplexing MIMO Cellular Networks," in *IEEE Wireless Communications*, vol. 14, no. 6, Dec 2007.

[63] H. Li, J. Hajipour, A. Attar, and V. Leung, "Efficient HetNet Implementation Using Broadband Wireless Access With Fiber-Connected Massively Distributed Antennas Architecture," in *IEEE Wireless Communications Magazine*, Jun 2011.

[64] P. Chunyi, L. Suk-Bok, L. Songwu, L. Haiyun, and L. Hewu, "Traffic-Driven Power Saving in Operational 3G Cellular Networks," in *ACM MobiCom*, Sept 2011.

[65] D. Tipper, A. Rezgui, P. Krishnamurthy, and P. Pacharintanakul, "Dimming Cellular Networks," in *IEEE Globecom*, Nov 2010.

[66] M. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal energy savings in cellular access networks," in *IEEE GreenCom*, Jun 2009.

[67] T. Flanagan, "Creating cloud base stations with TI's KeyStone multicore architecture," http://bit.ly/ztLLcz.

[68] Intel, "Heterogeneous Network Solution Brief," http://intel.ly/ZIMRcI.

[69] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, "CloudIQ: A Framework for Processing Base Stations in a Data Center," in *ACM MobiCom*, Aug 2012.

[70] "Common radio public interface," www.cpri.info.

[71] C. Liu, K. Sundaresan, M. Jiang, S. Rangarajan, and G. Chang, "The Case for Reconfigurable Backhaul in Cloud-RAN based Small Cell Networks," in *IEEE Infocom*, 2013.

[72] Picochip, http://www.picochip.com.

[73] G. T. 36.814, "Evolved Universal Terrestrial Radio Access (E-UTRA): Further advancements for E-UTRA physical layer aspects," http://bit.ly/Ysh6Cq.

[74] [Online]. Available: http://bit.ly/1sgDfJ1

[75] X. S. Wang, H. Shen, and D. Wetherall, "Accelerating the mobile web with selective offloading," in *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, ser. MCC '13. New York, NY, USA: ACM, 2013, pp. 45–50. [Online]. Available: http://doi.acm.org/10.1145/2491266.2491275

[76] B. Zhao, B. C. Tak, and G. Cao, "Reducing the delay and power consumption of web browsing on smartphones in 3g networks," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011.

[77] K. Matsudaira, "Making the mobile web faster," *Commun. ACM*, 2013.

[78] [Online]. Available: http://www.opera.com/turbo

[79] [Online]. Available: http://docs.aws.amazon.com/silk/latest/developerguide/split-arch.html

[80] [Online]. Available: https://developers.google.com/chrome/mobile/docs/data-compression

[81] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, 1993.

[82] [Online]. Available: http://info.iet.unipi.it/~luigi/dummynet/

[83] [Online]. Available: https://blog.kissmetrics.com/loading-time/?wide=1

[84] [Online]. Available: https://httpd.apache.org/docs/2.0/mod/mod_proxy.html#forwardreverse

[85] [Online]. Available: https://developers.google.com/speed/articles/spdy-for-mobile

[86] X. Xu, Y. Jiang, T. Flach, E. Katz-Bassett, D. Choffnes, and R. Govindan, "Investigating transparent web proxies in cellular networks," in *Technical report 14-944, University of Southern California*. USC, 2014.

[87] [Online]. Available: https://www.igvita.com/2013/03/07/faster-smaller-and-more-beautiful-web-with-webp/

[88] [Online]. Available: http://httparchive.org

[89] [Online]. Available: http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

[90] T. M. Mitchell, *Machine Learning*. McGraw-Hill, Inc., 1997.

[91] R. A. Berk, *Statistical Learning from a Regression Perspective*. Springer, 2008.

[92] J. MOGUL and L. BRAKMO, *Method for dynamically adjusting multimedia content of a web page by a server in accordance to network path characteristics between client and server*. U.S. Patent 6,243,761, 2001.

[93] [Online]. Available: https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/optimize-encoding-and-transfer

[94] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *SIGCOMM Comput. Commun. Rev.*, 1997.

[95] Y. Zhou, D. Chakrabarty, and R. M. Lukose, "Budget constrained bidding in keyword auctions and online knapsack problems," in *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, 2008.

[96] A. G. Jiasi Chen and M. Chiang, "Qava: Quota aware video adaptation technical report," Department of Electrical Engineering Princeton University, Princeton NJ, USA, Tech. Rep., 2012.

[97] [Online]. Available: http://info.iet.unipi.it/~luigi/dummynet/

[98] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. ACM, 2012.

[99] [Online]. Available: https://developers.google.com/speed/pagespeed/module/install

[100] [Online]. Available: http://scikit-learn.org/stable/modules/cross_validation.html

[101] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," Tech. Rep., 2007.

[102] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, Berlin, Germany, 2004.

[103] S. M. LaValle, *Planning Algorithms*, 2006.

[104] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "Concise paper: Tcp performance over mobile networks in high-speed mobility scenarios," *IEEE ICNP*, 2014.

[105] Z. Wang, F. X. Lin, L. Zhong, and M. Chishtie, "Why are web browsers slow on smartphones?" in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '11. New York, NY, USA: ACM, 2011. [Online]. Available: http://doi.acm.org/10.1145/2184489.2184508

[106] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding website complexity: Measurements, metrics, and implications," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 313–328. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068846

[107] J. Erman, V. Gopalakrishnan, R. Jana, and K. K. Ramakrishnan, "Towards a spdy'ier mobile web?" in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 303–314. [Online]. Available: http://doi.acm.org/10.1145/2535372.2535399

[108] [Online]. Available: http://www.chromium.org/spdy/spdy-whitepaper

[109] A. Sivakumar, V. Gopalakrishnan, S. Lee, S. G. Rao, S. Sen, and O. Spatscheck, "Cloud is not a silver bullet: a case study of cloud-based mobile browsing," in *15th Workshop on Mobile Computing Systems and Applications, HotMobile '14, Santa Barbara, CA, USA, February 26-27, 2014*, 2014.

[110] M. Butkiewicz, Z. Wu, S. Li, P. Murali, V. Hristidis, H. V. Madhyastha, and V. Sekar, "Enabling the transition to the mobile web with websieve," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '13. ACM, 2013. [Online]. Available: http://doi.acm.org/10.1145/2444776.2444795

[111] C. Cascaval, S. Fowler, P. Montesinos-Ortego, W. Piekarski, M. Reshadi, B. Robatmili, M. Weber, and V. Bhavsar, "Zoomm: A parallel web browser engine for multicore mobile devices," in *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ser. PPoPP '13. New York, NY, USA: ACM, 2013, pp. 271–280. [Online]. Available: http://doi.acm.org/10.1145/2442516.2442543

[112] H. Mai, S. Tang, S. T. King, C. Cascaval, and P. Montesinos, "A case for parallelizing web pages," in *Proceedings of the 4th USENIX Conference on Hot Topics in Parallelism*, ser. HotPar'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342788.2342790

[113] D. Lymberopoulos, O. Riva, K. Strauss, A. Mittal, and A. Ntoulas, "Pocketweb: Instant web browsing for mobile devices," in *Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XVII. New York, NY, USA: ACM, 2012, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/2150976.2150978

[114] Z. Wang, F. X. Lin, L. Zhong, and M. Chishtie, "How far can client-only solutions go for mobile browser speed?" in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 31–40. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187842

[115] J. Cho, J. Jeong, and E. Seo, "Twob: A two-tier web browser architecture optimized for mobile network," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, ser. MoMM '12. New York, NY, USA: ACM, 2012, pp. 267–270. [Online]. Available: http://doi.acm.org/10.1145/2428955.2429006

[116] S. Chava, R. Ennaji, J. Chen, and L. Subramanian, "Cost-aware mobile web browsing," *Pervasive Computing, IEEE*, vol. 11, no. 3, pp. 34–42, 2012.

[117] [Online]. Available: http://developer.nokia.com/Develop/Series_40/Nokia_Browser_for_Series_40/

[118] "Rice university warp project." [Online]. Available: http://warp.rice.edu

[119] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-Layer Wireless Bit Rate Adaptation," in *ACM SIGCOMM*, 2009.

[120] D. Pisinger. Mcknap algorithm. [Online]. Available: http://www.diku.dk/hjemmesider/ansatte/pisinger/codes.html

[121] "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," in *IEEE Std. 802.11ac Draft 1.0, 2011.*

[122] "Cisco Bring Your Own Device," http://bit.ly/xO4SXj.

[123] "LMR-400 50 ohm coaxial cable," http://amzn.to/MrAdcw.

[124] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J.Sel. A. Commun.*, 2006.

[125] "Optical Zonu: RF over Fiber transceiver," http://www.opticalzonu.com/.

[126] M. Sanchez, T. Giles, and J. Zander, "CSMA/CA with Beam Forming Antennas in Multi-hop Packet Radio Networks," in *Swedish Workshop on Wireless Ad-hoc Networks*, Stockholm, Mar. 2001.

[127] R. Ramanathan, "On the Performance of Ad Hoc Networks with Beamforming Antennas," in *Proc. of ACM MOBIHOC*, Oct. 2001.

[128] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," in *Proceedings of ACM MOBICOM*, Atlanta, sept 2002.

[129] S. Krishnamurthy, A. Acampora, and M. Zorzi, "Polling based Media Access Protocols for Use With Smart Adaptive Array Antennas," in *IEEE Trans. on Networking*, Apr 2001.

[130] S. Roy, D. Saha, S. Bandhyopadhyay, T. Ueda, and S. Tanaka, "A Network-aware MAC and Routing Protocol for Effective Load Balancing in Ad-hoc Wireless Networks with Directional Antenna," in *Proceedings of ACM MOBIHOC*, Jun 2003.

[131] R. R. Choudhury and N. H. Vaidya, "Impact of Directional Antennas on Ad-hoc Routing," in *8th Conference on Personal and Wireless Communcations*, Venice, Sept 2003.

[132] K. Sundaresan, R. Sivakumar, M. Ingram, and T. Chang, "A Fair Medium Access Control Protocol for Ad-hoc Networks with MIMO Links," in *Proceedings of IEEE INFOCOM*, Hong Kong, Mar 2004.

[133] J. H. Winters and M. J. Gans, "The Range Increase of Adaptive Versus Phased Arrays in Mobile Radio Systems," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 2, pp. 353–362, Mar 1999.

[134] J. B. Anderson, "Array Gain and Capacity for Known Random Chnanels with Multiple Element Arrays at Both Ends," *IEEE JSAC*, no. 11, Nov 2000.

[135] S. Ramanathan, "A Unified Framework and Algorithm for Channel Assignment in Wireless Networks," *Wireless Networks Journal*, vol. 9, 1999.

[136] M. M. Carvalho and J. J. Garcia-Luna-Aceves, "A scalable model for channel access protocols in multihop ad hoc networks," in *Proceedings of ACM MOBICOM*, 2004.

[137] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[138] F. Alizadeh-Shabdiz and S. Subramaniam, "Analytical models for single-hop and multi-hop ad hoc networks," in *First International Conference on Broadband Networks*, 2004, pp. 449–458.

[139] K. Medepalli and F. A. Tobagi, "Throughput Analysis of IEEE 802.11 Wireless LANs using an Average Cycle Time Approach," in *IEEE Globecom*, 2005.

[140] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?" in *Proceedings of ACM MOBIHOC*, 2004, pp. 9–19.

[141] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Procedings of ACM MOBICOM*, 2003, pp. 66–80.

[142] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels," *IEEE/ACM Trans. on Networking*, vol. 13, no. 4, pp. 868–880, 2005.

[143] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[144] R. W. H. J. T. Tang, M. Park and S. M. Nettles, "A MIMO-OFDM Transceiver for Ad Hoc Networking," in *Intnl. Workshop on Ad-hoc Wireless Networks*, 2004.

[145] M. Hu and J. Zhang, "MIMO Ad Hoc Networks: Medium Access Control, Satuation Throughput, and Optimal Hop Distance," *Special Issue on Mobile Ad Hoc Networks, Journal of Communications and Networks*, 2004.

[146] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *Proceedings of ACM MOBIHOC*, 2005.

[147] W.-Z. S. Xiang-Yang Li, Kousha Moaveni-Nejad and W. Wang, "Interference-Aware Topology Control for Wireless Sensor Networks," in *Proceedings of IEEE SECON*, 2005.

[148] E. L. Lloyd and S. Ramanathan, "On the complexity of distance-2 coloring," in *Proceedings of ICCI*. IEEE Computer Society, 1992, pp. 71–74.

[149] G. Szekeres and H. S. Wilf, "An inequality for the chromatic number of a graph," *Journal of Combinatorial Theory*, vol. 4, pp. 1–3, 1968.

[150] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in sensor networks: Distributed edge coloring revisited," in *Proceedings of IEEE INFOCOM*, 2005.

[151] Y. J. Zhang and K. B. Letaief, "Adaptive resource allocation for multiaccess MIMO/OFDM systems with matched filtering," *IEEE Transactions on Communications*, vol. 53, no. 11, Nov 2005.

[152] J. Tang and X. Zhang, "Link-adaptation-enhanced dynamic channel allocation for mimo-ofdm wireless networks," in *IEEE WCNC*, Mar 2005.

[153] W. Huang and K. B. Letaief, "A cross-layer resource allocation and scheduling for multiuser space-time block coded mimo/ofdm sytems," in *IEEE ICC*, May 2005.

[154] A. Sang, X. Wang, M. Madihian, and R. Gitlin, "Downlink schedulink schemes in cellular packet data systems of multiple-input multiple-output antennas," in *IEEE Globecom*, Dec 2004.

[155] H. Boche and M. Schubert, "Resource allocation for multi-antenna multi-user systems," *IEEE Transactions on Signal Processing* , vol. 54, no. 6, Jun 2006.

[156] D. Aktas and H. E. Gamal, "Multiuser scheduling for mimo wireless sytems," in *IEEE VTC*, Oct 2003.

[157] M. Airy, S. Shakkotai, and R. W. Heath, "Spatially greedy scheduling in multi-user mimo wireless systems," in *Asilomar Conference on Signals, Systems and Computers*, Nov 2003.

[158] H. Viswanathan and K. Kumaran, "Rate scheduling in multiple antenna downlink wireless systems," *IEEE Transactions on Communication*, vol. 5, no. 4, Apr 2005.

[159] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks - Part I: Theoretical Framework," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, Mar 2005.

[160] Z. Zhang, Y. He, and K. P. Chong, "Oppotunistic downlink scheduling for multiuser ofdm systems," in *IEEE WCNC*, Mar 2005.

[161] N. Damji and T. Le-Ngoc, "Dynamic downlink ofdm resource allocation with interference mitigation and macro diversity for multimedia services in wireless cellular systems," in *IEEE WCNC*, Mar 2005.

[162] Y. J. Zhang and K. B. Letaief, "Cross-layer adaptive resource management for wireless packet networks with OFDM signaling," *IEEE Transactions on Wireless Communications*, vol. under submission, 2005.

[163] P. Liu, R. Berry, and M. L. Honig, "Delay-sensitive packet scheduling in wireless networks," in *IEEE WCNC*, Mar 2003.

[164] S. Shakkottai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real time data in hdr," in *Bell Labs Technical report*, 2000.

[165] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting, "Cdma data qos scheduling on the forward link with variable channel conditions," in *Bell Labs Technical report*, 2000, p. Apr.

[166] V. Huang and W. Zhuang, "QoS-oriented packet scheduling for wireless multimedia CDMA communications," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, Jan-Mar 2004.

[167] H. Jiang and W. Zhuang, "Cross-layer resource allocation for integrated voice/data traffic in wireless cellular networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, Feb 2006.

[168] A. Sang, X. Wang, M. Madihian, and R. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," in *ACM Mobicom*, Oct 2004.

[169] X. Liu, K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, Oct 2001.

[170] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable Scheduling Policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, Apr 2005.

[171] S. Das, H. Viswanathan, and G. Rittenhouse, "Dynamic load balancing through coordinated scheduling in packet data systems," in *IEEE INFOCOM*, Mar 2003.

[172] R. Knopp and P. A. Humblet, "Information capacity and power control in single cell multiuser communications," in *IEEE ICC*, Jun 1995.

[173] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[174] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.

[175] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity (Chapter 11)*. Prentice Hall, 1982.

[176] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bhargavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," Proceedings of ACM MOBICOM, Aug 2000.

[177] R. G. Mukthar, "Qns: Queuing network simulator," in *QNS v0.1*, http://www.cubinlab.ee.mu.oz.au/rgmukht/qns, Nov 2003.

[178] H. Luo and S. Lu, "A New Model for Packet Scheduling in Multihop Wireless Networks," in *Proceedings of ACM MOBICOM*, Aug. 2000.

[179] B. Radunovic and J. L. Boudec, "Rate performance objectives of multi-hop wireless networks," in *IEEE INFOCOM*, Mar 2004.

[180] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. on Information Theory*, vol. 49, pp. 2415–2425, Oct 2003.

[181] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy efficiency of MIMO and cooperative MIMO in sensor networks," *IEEE JSAC*, vol. 22, no. 6, 2004.

[182] H. Ochiai, P. Mitran, and V. Tarokh, "Design and analysis of collaborative diversity protocols for wireless sensor networks," in *IEEE VTC*, 2004.

[183] E. Zimmerman, P. Herhold, and G. Fettweis, "On the performance of cooperative diversity protocols in practical wireless sytems," in *IEEE VTC*, Oct 2003.

[184] X. Wang and K. Kar, "Cross layer rate control for end-end proportional fairness in wireless networks with random access," in *Proc. of ACM MOBIHOC*, May 2005.

[185] K. Sundaresan, H.-Y. Hsieh, and R. Sivakumar, "Ieee 802.11 over multi-hop wireless networks: problems and new perspectives," *Ad-hoc Networks Journal, Elsevier*, Feb 2004.

[186] W. Wang, Y. Wang, W. Song, and X. Li, "Efficient interference-aware link scheduling for wireless networks," in *ACM MOBICOM*, Los Angeles, USA, Sept 2006.

[187] G. Brar, D. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *ACM MOBICOM*, 2006.

[188] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, "Ucan: A unified cellular and ad-hoc network architecture," in *ACM MOBICOM*, Sept 2003.

[189] P. Herhold, W. Rave, and G. Fettweis, "Relaying in cdma networks: pathloss reduction and transmit power savings," in *IEEE VTC*, Apr 2003.

[190] A. So and B. Liang, "Effect of relaying on capacity improvement in wireless local area networks," in *IEEE WCNC*, Mar 2005.

[191] S. Mengesha and H. Karl, "Relay routing and scheduling for capacity improvement in cellular wlans," in *WiOpt*, Mar 2003.

[192] A. So and B. Liang, "An efficient algorithm for the optimal placement of wireless extension points in rectilinear wireless local area networks," in *QShine: International Conference on QoS in Heterogeneous Wired/Wireless Networks*, Aug 2005.

[193] H. Wei, S. Ganguly, and R. Izmailov, "Adhoc relay network planning for improving cellular data coverage," in *IEEE PIMRC*, Sept 2004.

[194] M. Sternad, T. Svensson, and g. Klang, "Winner mac for cellular transmission," in *IEEE VTC*, Sept 2006.

[195] N. Challa and H. Cam, "Cost-aware downlink scheduling of shared channels for cellular networks with relays," in *IEEE International Conference on Performance, Computing, and Communications*, 2004.

[196] M. Herdin, "A chunk based ofdm amplify-and-forward relaying scheme for 4g mobile radio systems," in *IEEE ICC*, Jun 2006.

[197] A. Hottinen and T. Heikkinen, "Subchannel assignment in ofdm relay nodes," in *Proc. of CISS*, Mar 2006.

[198] P802.16j, "Multihop relay specification," in *Amendment to IEEE Std. for LAN/MAN-Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, Mar 2006.

[199] C. Hoymann, P. Dallas, A. Valkanas, A. Gosteau, D. Noguet, and R. Hoshyar, "Flexible relay wireless ofdm-based networks," in *Funded by European Commission*, 2006.

[200] D. C. J. Lee, S. Zheng, and A. Zhang, "A flexible multi-hop frame structure for IEEE 802.16j," *IEEE 802.16 Broadband Wireless Access Working Group*, Nov 2006.

[201] K. Sundaresan, X. Wang, and M. Madihian, "Scheduling aspects in relay-enabled ofdm cellular networks," in *NEC Labs America Technical Report: 2007-L090*, Jun 2007.

[202] H. Viswanathan and S. Mukherjee, "Performance of cellular networks with relays and centralized scheduling," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, Sep 2005.

[203] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company, 1979.

[204] M. Andrews and L. Zhang, "Scheduling algorithms for multi-carrier wireless data systems," in *ACM MOBICOM*, Sept 2007.

[205] L. Fleischer, M. Goemans, V. Mirrokni, and M. Sviridenko, "Tight approximation algorithms for maximum general assignment problems," in *ACM SODA*, 2006, pp. 611–620.

[206] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *ACM MOBICOM*, Sept 2006.

[207] S. Deb, V. Mhatre, and V. Ramaiyan, "Wimax relay networks: opportunistic scheduling to exploit multiuser diversity and frequency selectivity,," in *ACM MOBICOM*, Sep 2008.

[208] K. Sundaresan and S. Rangarajan, "On exploiting diversity and spatial reuse in relay-enabled wireless networks," in *ACM MOBIHOC*, 2008.

[209] H. Won, H. Cai, D. Yun, K. Guo, and A. Netravali, "Multicast scheduling in packet data networks," in *IEEE INFOCOM*, Apr 2007.

[210] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-time video multicast in wimax networks," in *IEEE INFOCOM*, Apr 2008.

[211] K. Sundaresan and S. Rangarajan, "On the coexistence of unicast and multicast traffic in relay-enabled wireless networks," in *IEEE BROADNETS*, Sept 2008.

[212] 3GPP, "Lte release 10 and beyond (lte advanced)," in *RP-090939*, http://www.3gpp.org/LTE-Advanced.

[213] M. Fisher, G. Nemhauser, and G. Wolsey, "An analysis of approximations for maximizing submodular set functions- ii," in *Mathematical Programming Study*, 1978.

[214] R. Wireless, "4g networks will be macro, micro, das and more," Oct 2010, http://www.rcrwireless.com/article/20101007/networks/.

[215] T. . 3GPP, "Service aspects and requirements for network sharing," in *Technical Report*, Dec 2009, http://ftp.3gpp.org/specs/html-info/22951.htm.

[216] ——, "Network sharing; architecture and functional description," in *Technical Report*, Mar 2010, http://ftp.3gpp.org/Specs/html-info/23251.htm.

[217] H. Guan, T. Kolding, and P. Merz, "Discovery of cloud-ran," in *Cloud RAN Workshop*, Apr 2010, http://labs.chinamobile.com/article_download.php?id=63327.

[218] T. . 3GPP, "Evolved universal terrestrial radio access(e-utra); further advancements for e-utra physical layer aspect," in *Release 9, v.9.0.0*, Mar 2010.

[219] M. Huang, K. Kim, M. Jiang, Y. Hsueh, Y. Shao, S. Rangarajan, and T. Wang, "Employing moving chain cells and rof technique for future internet access on the train environment," in *NEC Labs Technical Report*, Oct 2010.

[220] Y.-T. H. et. al., "A Novel Wireless over Fiber Access Architecture Employing Moving Chain Cells and RoF Technique for Broadband Wireless Applications on the Train Environment," in *OFC*, Mar 2011.

[221] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Nvs: A virtualization substrate for wimax networks," in *ACM Mobicom*, Sept 2010.

[222] N. S. Networks, "Liquid Radio: Let Traffic Waves Flow Most Efficiently," in *White Paper*.

[223] 3GPP, "Evolved universal terrestrial radio acces (e-utra); lte physical layer; general description (release 10)," in *TS 36.201*.

[224] K. M. et. al., "Experimental demonstration of mimo rf signal transmission in rof-das over wdm-pon," in *International Topical Meeting on Microwave Photonics*, 2011.

[225] T. T. et. al., "40 km Fiber Transmission of Time Domain Multiplexed MIMO RF Signals for RoF-DAS over WDM-PON," in *Optical Fiber Communication Conference*, 2012.

[226] K. Sundaresan, M. Arslan, S. Singh, S. Rangarajan, and S. Krishnamurthy, "Fluidnet: A flexible cloud-based radio access network for small cells," in *NEC Laboratories Technical Report*, Mar 2013.

[227] Z. N. et. al., "Cell zooming for cost-efficient green cellular networks," in *IEEE Communications Magazine*, Nov 2010.

[228] E. O. et. al., "Towards dynamic energy-efficient operation of cellular network infrastructure," in *IEEE Communications Magazine*, Jun 2011.

[229] [Online]. Available: http://www.forbes.com/sites/louiscolumbus/2012/08/27/how-google-is-driving-mobile-video-market-growth/

[230] [Online]. Available: http://www.usatoday.com/story/tech/2013/06/06/reviewed-smartphones-replace-point-and-shoots/2373375/

[231] [Online]. Available: https://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/

[232] H. Mansour, P. Nasiopoulos, and V. Krishnamurthy, "Real-time joint rate and protection allocation for multi-user scalable video streaming." in *PIMRC*, 2008.

[233] L. Breiman, "Random forests," *Mach. Learn.*, 2001.

[234] J. C. Mogul, L. S. Brakmo, D. E. Lowell, D. Subhraveti, and J. Moore, "Unveiling the transport," *Computer Communication Review*, 2004.

[235] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, 1999.

[236] C. Gianluigi and S. Raimondo, "An innovative algorithm for key frame extraction in video summarization," *Journal of Real-Time Image Processing*, vol. 1, no. 1, pp. 69–88, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11554-006-0001-1

[237] B.-W. Chen, J.-C. Wang, and J.-F. Wang, "A novel video summarization based on mining the story-structure and semantic relations among concept entities," *Multimedia, IEEE Transactions on*, vol. 11, no. 2, pp. 295–312, 2009.