Optimal Data Scheduling of Clients Serviced using Beamforming Antennas in Mobile Scenarios

by

Daniel T. Bennett

M.S., Electrical Engineering,

University of Colorado, Boulder, 2004

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical, Computer and Energy Engineering

2012

This thesis entitled:
Optimal Data Scheduling of Clients Serviced using Beamforming Antennas in Mobile Scenarios
written by Daniel T. Bennett
has been approved for the Department of Electrical, Computer and Energy Engineering

Timothy X Brown

Kenneth Baker

Dirk Grunwald

Youjian Liu

François Meyer

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Bennett, Daniel T. (Ph.D., Electrical Engineering)

Optimal Data Scheduling of Clients Serviced using Beamforming Antennas in Mobile Scenarios

Thesis directed by Professor Timothy X Brown

The use of beamforming antennas has received significant attention over the last decade. I consider beamforming applied to dynamic operations such as networked UAV hubs which interconnect with users on the ground. The key problem involves understanding how to optimally manage the users' data requirements while considering mobility and a dynamic radio environment serviced by one or more hubs with beamforming antenna capability.

In this work I break the problem down into scheduling, tracking and ultimately execution. I develop a regularized linear programming based scheduling algorithm along with developing a very efficient scheduling with uncertainty receding horizon based relaxation and implement them along with a capacity tracking estimation algorithm. Finally I show the results of successfully implementing this system in hardware using Fidelity Comtech's Phocus Array FCI-3100X.

This implementation shows that the problem overview presented in this work provides a solid basis and defines the key components needed for a reliable electronic beamforming antenna system able to successfully service dispersed users in a mobile environment. It also shows the tools developed, refined, and integrated with respect to tracking, scheduling, and practical modifications.

Dedication

This thesis is dedicated to my wife Jenny and my kids Hannah, Laurel, and Grace and for the sacrifices they have made the last eighteen years and continue to make as an Army family. Thanks be to God for them and all the gifts in my life.

Acknowledgements

First and foremost I would like to thank Professor Tim Brown. This work would not have been possible, especially given my tight timeline, without the freedom, guidance, time and resources he provided. I would also like to thank Professors Ken Baker, Harvey Gates, Dirk Grunwald, Eugene Liu and François Meyer for not only taking the time to serve on my committee but also for the instruction, guidance, and feedback they have provided spanning over the last ten years.

I would like to thank my fellow friends and office mates for the lunches and chances to exchange ideas and challenges the last three years. These include Juan Ramirez, Sears Merritt, Prasanna Madhusudhanan, Ben Pearre, Vamshidhar Dantu (who also helped introduce me to Linux), Abishek Chandrasekaran, and Shahjehan Hakim. Furthermore thanks to Maciej Stachura, Jack Elston, Eric Anderson and Caleb Phillips for letting me tag along on some of their flight tests (Maciej/Jack) and for help on random questions.

A special thanks goes out to Joe Carey, Bob Weaver, Russ Brinkman, and especially Alan Schmitz at Fidelity Comtech for their advice, collaboration and help along the way.

Finally I would like to thank the Department of Electrical Engineering and Computer Science Faculty at the United States Military Academy for putting their faith in me to put this all together.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Introduction to Problem

1.1.1 Communicating Wirelessly

“The ability to communicate with people on the move has evolved remarkably since Guglielmo Marconi first demonstrated radio’s ability to provide continuous contact with ships sailing the English channel. That was in 1897 and since then new wireless communications methods and services have been enthusiastically adopted by people throughout the world[68].” Although the true inventor of radio is debatable, the benefit of wireless communications is undeniable. The ability to be able to communicate with someone or to be able to cause an action at a distance without being physically connected via a wire or some other means is now taken for granted. In Marconi’s time it involved one user at a specified location using a spark gap transmitter and an antenna of a specified length, according to Marconi’s law [29], sending a signal out over the air in all directions. A mobile user some distance away was able to pick up this signal by using an antenna with the exact same length and therefore the same frequency. Now wireless communications is a part of everyone’s daily life from checking email on a wireless connection at home, opening the garage door, following the driving instructions provided by a GPS device, listening to the radio, talking via Bluetooth on a 4G smartphone, and so on. All of these applications display the ability to communicate in some form and for varying reasons and benefits without having to be connected via a wire. Communicating wirelessly also has well known challenges and uncertainties involved in utilizing the radio environment. In a wired environment, the key limitations are based on the characteristics of the wire (or channel) and the terminal devices that are attached. In a wireless environment there are many more factors that can affect this open radio frequency channel. Configurations for communicating wirelessly are permutations of the same typical wired network topologies (see 3.1 for a discussion). Perhaps the most common are a series of interconnected stars or hub and spoke (also knows as infrastructure) configurations, mesh (also known as ad hoc) configurations, or some combination thereof. A simple example of the series of star topologies would be cell phone towers and the respective clients receiving service from them at any given location and point in time. A mesh could be any random grouping of users communicating via a shared channel within the reach of the capability provided by their respective antenna and transceiver combination. An example would be kids playing a multi-player game on handheld devices with built in WiFi capability.

1.1.2 Developing Scenarios

This work will focus more on a hub and spoke type scenario, versus a mesh, or ad hoc, type scenario. Both will be discussed more in Chapter 3. In brief, the hub and spoke scenario is where all traffic from one user intended for another user passes through a central hub. The scenarios discussed are as shown in Table 1.1. Initially we will assume that the hub is static and the radio environment also remains static. A simple scenario is as shown in Figure 1.1. As long as the client and the hub are in the omni-directional range (represented, for simplicity by the circle), they should be able to communicate with each other.

Table 1.1: Mobility Scenarios and Environment

Hub	Client	Environment	Scenario
Static	Static	Static	Static
Static	Static	Dynamic	Mobile
Static	Mobile	Static	Mobile
Static	Mobile	Dynamic	Mobile
Mobile	Static	Static	Mobile
Mobile	Static	Dynamic	Mobile
Mobile	Mobile	Static	Mobile
Mobile	Mobile	Dynamic	Mobile

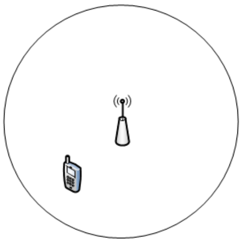


Figure 1.1: Omni Hub with static Omni Client

This scenario can become more complex if that user is mobile as shown in Figure 1.2. However, the service of the client remains fairly simple as long as the client stays in the range of the omnidirectional antenna and his speed is manageable, which will be discussed more later.

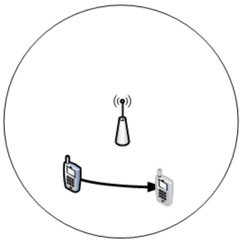


Figure 1.2: Omni Hub with mobile Omni Client

It is important to note that mobility is a relative term. The client could be mobile with respect to the hub as a result of the client's movements but it could also exhibit the same relative movement as a result of the hub moving and the client being static. Furthermore, dynamics in the radio environment such as multi-path could also make it appear that the hub or the client is mobile. For instance, the movement of a scatterer between the hub and the client could cause a reflected or diffracted version of the signal appear stronger, giving the impression of movement. Therefore, the relative movement is a result of the client, the hub, the radio environment or any permuations thereof. Without loss of generality we will express movement, for now, in terms of the client relative to the hub.

The next scenario is multiple clients being serviced within the omni-directional range of the hub as shown in Figure 1.3. As long as each of the clients stays within range then they should continue to be able to be serviced by the hub. Interference can now become a factor by having more than one client communicating via the hub. We consider time division techniques that coordinate between the users and the hub to communicate only one at a time, except in broadcast or multi-cast type scenarios where the same traffic is sent to multiple users at a time.

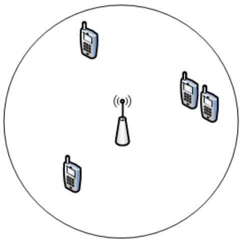


Figure 1.3: Omni Hub with static Omni Clients

Extending the scenario to mobile clients, as long as the mobile users stay within the range of the hub's omni-directional antenna then the challenge remains the same as in the previous configuration. See Figure 1.4. If a client moves out of range then service to them is lost. If a client moves into the area then the association process (depending on the system/protocol parameters) can take place to provide the user service. The challenges remain, as discussed earlier, as to how to schedule the traffic given multiple users and how to prevent them from interfering with each other during their data traffic exchanges via the hub.

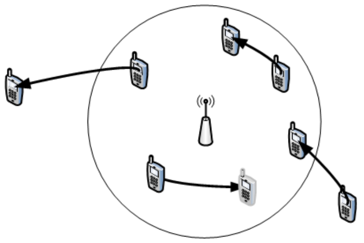


Figure 1.4: Omni Hub with Mobile Omni Clients

Overall, having the hub location use a purely omni-directional configuration is a simpler process and the main challenge is in coordinating the separate transmissions while mitigating interference between users.

A key limitation is the limited range caused by using an omni-directional antenna. An omni-directional antenna can be a vertical dipole antenna that emits radiation in a donut shape in all directions, as in Figure 1.5. As an example, the transmit power of a currently available household/commercial wireless 802.11b/g router is 19.1 dBm with an omni-directional vertical dipole antenna gain of 2.1 dBi. Assume that the wireless client has a receiver sensitivity of -71 dBm at 54 Mbps. Table 1.2 shows an example of the range that can be achieved[1] given the type of antenna being used at the transmitter (XMIT) and receiver (RCVR) with their respective gains.

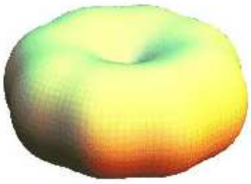


Figure 1.5: Example Omni-Directional Radiation Pattern

Table 1.2: Transmit/Receiver Antenna Type vs. Range Scenarios

XMIT/RCVR Antenna	XMIT Gain	RCVR Gain	Range
Isotropic	0 dBi	0 dBi	320m
Dipole	2.1 dBi	2.1 dBi	400m
Directional/Dipole	15 dBi	2.1 dBi	2300m
Directional	15 dBi	15 dBi	10000m

These calculations are very liberal estimates that assume sufficient Fresnel zone clearance for a single point to point link with no scattering, interference, fading, etc. They also assume that the Signal-to-Noise Ratio (SNR) is at an acceptable level to maintain that data rate. Nevertheless, this highlights that using an omni-directional configuration is simple but inefficient. The energy is focused in all directions and a significant amount of that energy is wasted (since the respective user that the communication exchange is intended for is off in one specific direction). Furthermore, since the energy is transmitted in all directions then it creates more interference for other users located within that footprint, versus the user that the traffic was intended for. More example scenarios will be visited later to consider a directional configuration.

The scenario is complicated further with a single static client that can no longer be reached

[1] based on the Friis equation[68]: $P_{RX}(dBm) = P_{TX}(dBm) + G_{TX}(dBi) + G_{RX}(dBi) - 20log_{10}d(km) - 20log_{10}f(MHz) - 32.45$ for $f = 2412$MHz, $P_{TX} = 19.1$dBm, and $P_{RX} = -71$dBm

with the hub's previous omni-directional configuration. If power is limited, the hub can either use a lower data rate or a directional antenna. Assuming a directional antenna needs to be used, see Figure 1.6. The challenge now includes knowing what direction (or pattern) to use to point to a client such that it can communicate. We mention direction here for simplicity but it could also be any arbitrary pattern, with no single specific direction, with which the client could be reached. In fact, the distinction between the different types of directional (beamforming) antennas are explained by [1] and given in Chapter 3. The equipment used in this work is the switched-beam type. It is able to form a beam pattern in a particular direction by adjusting the respective antenna weights of the antennas in its array. A directional antenna offers the main benefit of increased range. It also contributes to less interference for surrounding users since the transmission can be focused only in a general or specific direction. Applying the same scenario as mentioned before but now assuming that the hub can transmit in a particular direction with a gain of 15 dBi, then the maximum distance ends up being about 2.3 kilometers or an increase by a factor of 5.8.

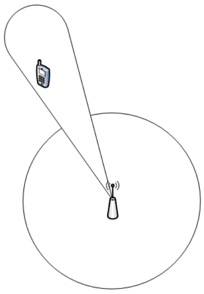


Figure 1.6: Directional Hub with Static Omni Client

Now extend the scenario further to multiple clients with omni-directional antennas and where the hub needs to use directional antennas in order to reach each of them. See Figure 1.7. The challenge now grows to knowing which beam to use for which client. Then an efficient scheduling mechanism needs to be developed to coordinate what direction to point and when, such that the client(s) in each of the respective directions can communicate. Also, if multiple users are located within the same beam footprint then their communication exchanges and possible interference issues need to be coordinated and/or accounted for.

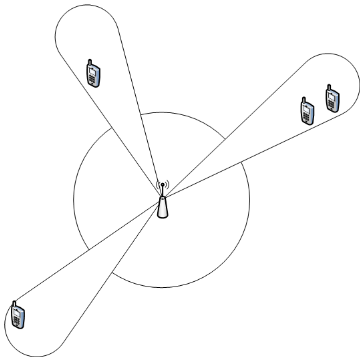


Figure 1.7: Directional Hub with Multiple Static Omni Clients

This scenario is complicated more if the users are mobile with respect to the hub. Now, in addition to the previous requirements, there would have to be the ability to track clients. This involves updating which beam is best in order to communicate with a particular client as the relative location changes over time. It also involves accounting for new users that move within range of one of the beams and dropping those users that move outside the range of any of the beams. See Figure 1.8.

Before proceeding we further classify the types of problems addressed in this work. See Table 1.3. One user in a static scenario is trivial as already discussed. The case of two or more static users has been confronted extensively in both wired and wireless scenarios. Many of the wireless type scenarios will be discussed in related work. To recall, however, one of the key contributions of wireless is to allow for mobility. The effect of mobility can be a limiting factor of a client's communication's capacity from a tracking perspective and other factors discussed in Chapter 3 such as doppler shift. Fast mobility, as shown here, refers to a scenario that would make the use of a steerable beam infeasible. Its delineation with respect to slow mobility will become more apparent throughout this work. The infeasibility comes as a result of increased speed causing the

Table 1.3: Scheduling Scenarios - Mobility vs. User Density

Density	**Static**	**Slow Mobile**	**Fast Mobile**
1	Trivial	Tracking	Fast Tracking
2+	Data Scheduling	Tracking, Scheduling	TBD

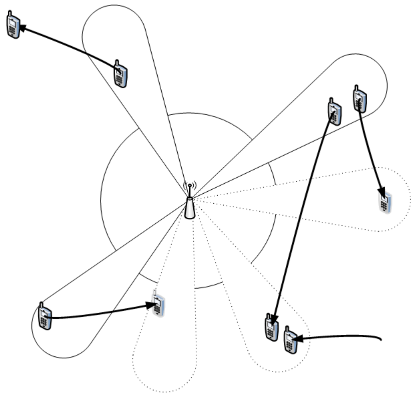


Figure 1.8: Directional Hub with Multiple Mobile Omni Clients

need for continuous tracking, as discussed, and therefore taking away from data exchange. This would become more and more of a problem with additional users and having to track them as well along with coordinating their data exchanges. This situation is also impractical in the sense that a user with too much speed could very quickly move out of the range of the antenna. For these and other reasons, this work will focus on the slow mobility scenario.

1.2 Confronting the Scenario - The Challenge

The current scenario then is as presented in Figure 1.8. It is necessary to be able to track the users and know which direction (pattern) to use for them initially and then as they move about. Once it is known where they are at, then time needs to be allocated/scheduled to allow each of the users to exchange data separately and to minimize the amount of interference that takes place between them. A basic method might be a sequence of tracking and allowing for data exchange. During the initial tracking the hub could cycle through all of its beam patterns and determine what clients can be reached via each beam. During subsequent trackings then the hub could cycle through those beams used for clients already identified. It can then determine if a client can still be reached with that particular beam or if another beam provides a better (if at all) signal. During the data exchange phase the hub could cycle through all the beams that it knows it can reach clients with and allow each of the clients that can be accessed from a particular beam an opportunity to exchange data. Once all users reached through all the respective beams have been provided an opportunity to exchange their data then the hub could go back to tracking old or new users again in a similar manner.

The challenges, for what appears as a simple logical sequence of events are many. The bootstrapping and tracking down the clients initially is difficult because the hub does not necessarily know what and where any clients are and the clients don't know the same regarding a hub. A system needs to be set up that allows a rendezvous between a user and the hub to occur in order for them to find each other. The hub arbitrarily steering a beam in a particular direction does not mean it automatically tracks a client unless some type of signaling takes place to let a client know about the hub. In turn, this would need to elicit a response in a timely manner from the client that can be heard from the hub to let it know a client is there. A timely response is required such that the hub doesn't move on to another direction before it hears that a client can be reached in that direction. Once users have been found then they can be tracked. Tracking users simply by searching through many or all the beams every time can be very inefficient. This process can be simplified by fine tuning what beam is best to use within the vicinity of the last beam used for that particular client. This is similar to the dithering method used to track satellites[23]. Variations as to the dynamics of tracking and changing beams for a particular client will be caused by such factors as the speed and trajectory of the client. As an example, if the client is close to the hub but moving rapidly and in a circular pattern around the hub then the beam needed to be used for it will change rapidly. This is of course assuming we are using patterns with specific directions as we have illustrated up to this point. The client could be moving constantly perpendicular to the direction of the beam pointing towards it. If a user is far away, while moving with the same characteristics, it will take longer for it to require a different beam in order to communicate with it. In Figure 1.9 it can be seen that the inner client and the outer client can both move the same distance in a set amount of time but the changes in beam needed to reach the inner client change much more than the outer client. Furthermore, issues such as scattering and multi-path will become factors. If a client moves behind an object then the best signal received at the hub, if any, would

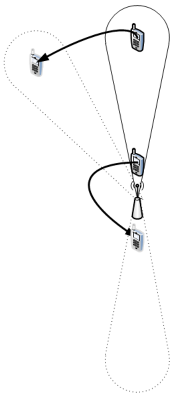


Figure 1.9: Different Relative Movements of Clients vs. Distance away from Hub

likely be a reflected signal coming from a different direction that is not necessarily the straight line direction between the hub and the client. At one point the best beam used would be the beam whose trajectory is closest to the direction pointing directly at the client. Then it could drastically change if the best beam is now a multi-path beam received as a result of scattering or reflection. It could then change back drastically if it moves out from behind the object. Issues such as this could be very disruptive to any tracking algorithm being used. The bottom line is that different clients will likely have different factors associated with them and could require varying means in which to track them better.

Finally, allowing each of the clients the ability to exchange data factors in many difficulties. Users will have varying data exchange requirements. As an example, some cell phone users might only need to send periodic text messages while others prefer streaming movies. Furthermore, the radio environment for each of these users will be different as briefly discussed in tracking. When it comes to time needed to send larger packets of data then the effects of the radio environment will play much more of a role. Factors such as noise, interference, fading, multi-path, doppler shift, etc. will affect the user's ability to reliably transmit at a particular rate at a particular point in time depending on its distance from the hub, the amount of interferers and scatterers in its vicinity, etc. Additionally, the difficulty arises of how much time is allocated and how is it coordinated in tracking and allowing for data exchange for all users individually and collectively. Additional difficulty is included if the hub can only receive or transmit one packet at a time. Many of the variables could be set as provided parameters however the majority of the environmental factors are random and outside the control of the scenario being accounted for. Such are the key challenges confronted in these scenarios.

1.3 Example Real Scenarios

Unmanned Aerial Vehicles (UAVs) provide intelligence and perspective for applications ranging from military operations, tornado chasing, fire-fighting, search-and-rescue to seaborne operations (e.g. counter-pirating), etc. In all of these cases the UAV has a command Radio Frequency (RF) control channel responsible to direct its movements as well as the ability to send back imagery and data from the platform depending on its mission. Therefore it is possible that this channel (or some other communications capability) can serve an additional role of providing a tethered data resource for clients on the ground in order to provide them not only sensor data generated by the platform (e.g. imagery) but also a gateway to data via the command and control or tethered location. For instance, in tornado chasing, a UAV platform, who's command RF channel is bandlimited, can provide (push) imagery and additional vortex characteristic data via an external wireless connection.

In military applications, a large UAV platform that has an external satellite connection could extend service wirelessly to clients on the ground. In addition to the UAV's "eye-in-the-sky" perspective, it could also be used as a relay (or flying hub) between multiple clients dispersed within view of the UAV and provide them various external data resources to keep them connected with different echelons or external entities. See Figure 1.10 which illustrates this example. A ground control station (GCS) is added which is used to monitor the UAV locally and take-over control of the UAV upon takeoff and landing as the UAV is normally flown/controlled from a sanctuary location via the satellite connection while in mid-air.

A similar situation exists in a seaborne operation where pirates have commandeered a ship. The command center that has deployed the UAV can use the UAV to monitor the situation but

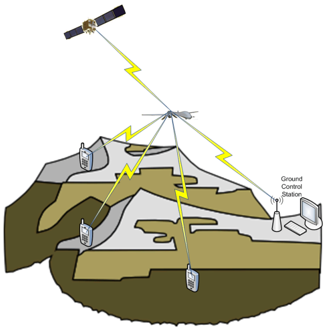


Figure 1.10: UAV as a Flying Hub

also teams that have been deployed out at various locations can have access to the UAVs resources, as well as those possibly tethered resources in order to maximize communications, control, coordinations and synchronization of the operation. With a separate wireless capability these dispersed units could have a device as simple as a smartphone to help facilitate the execution of the operation.

Given these examples we offer a more specific typical scenario of a UAV circling around an objective area as shown in Figure 1.11. Clients that might be accessing the UAV will likely be in the form of soldiers and/or vehicles operating in and around the objective area. For self-preservation the UAV will maintain a safe horizontal distance away from the objective as well. Therefore the majority, if not all, of the clients being supported will be located within the circle as shown. In other cases the UAV might operate in a figure 8 pattern with its boundaries being the edges of the circle. However, the assumption is that all users located within that circular area around the objective are within range of being supported by the UAV and one of its directional antennas at some point during its orbit. Such scenarios offer the general problem for which we will be addressing

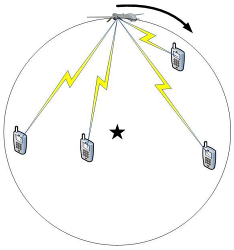


Figure 1.11: UAV Circling an Objective

in this work. This chapter provides a general overview of the problem. The next chapter gets more in depth in formulating the problem this work addresses as well as presenting the research question and the methodology that will be followed.

Chapter 2

Problem Formulation

2.1 Overview

Following from Chapter 1, this work addresses a general slow mobile wireless hub and spoke scenario where the hub's antenna has electronic switched beam antenna capability and the clients have only a fixed antenna capability. The eventual goal is to implement this on a hardware radio system described in Chapter 3.

With respect to the inputs, the hub communicates with up to n clients over period, T, using any of m possible beam patterns. Though we talk about “beams” and “directions” for conceptual simplicity, in fact each beam pattern can have an arbitrary but fixed shape that may or may not be well characterized. However, without loss of generality we will generally refer to it as beamforming throughout the rest of this work unless otherwise noted. Each client, j, is assumed to have a single arbitrary but fixed beam pattern and has a demand in the form of an average “requested” data transfer rate equal to λ_j. We do not directly model any of the client beam patterns. At a given point in time, s, beam i can transfer to client j a capacity $c_j^i(s)$. Given that this is a mobile environment and given a dynamic radio environment then $c_j^i(s)$ will change over time for a particular client for a given beam. For simplicity we do not consider which direction (to or from the hub) and only consider the amount of data that can be transferred.[1] Also, we will initially discuss a single hub and subsequently generalize to multiple hubs.

Looking at Figure 1.10, the general problem is to find an efficient way for each hub to service

[1] The direction can be incorporated by treating each direction as a separate client.

its clients successfully. More specifically, we want to be able to transfer the data for each client. We consider several challenges to making this work. The first challenge is that there are mn possible beam and client combinations (over which their data could be transmitted) and it is prohibitive to have to track each of these combinations. Thus, at any given time we may only have estimates of the capacity of the beam i client j combination denoted $\hat{c}_j^i(s)$.

We formulate the general problem around Figure 2.1. The rest of this chapter will step through each of the elements of this diagram.

2.2 Initialize

At startup the hub has little or no information about the capacity of each beam to each client over time, $c_j^i(s)$. So the first problem is to set initial estimates, $\hat{c}_j^i(s)$. This could simply be $\hat{c}_j^i(s) = 0$ but other estimates are possible. The next step is tracking where we refine these estimates.

2.3 Tracking

See "TRACKING" in Figure 2.1. Tracking could be done continuously or after waiting a certain amount of time. However, tracking continuously is infeasible due to the overhead required in comparing signal quality values and depends on the method being used. It also leaves no time for possibly multiple clients to exchange data.

An appropriate tracked beam pattern should be one that provides sufficient signal quality at both ends to allow the hub and the client to reliably communicate. Ideally, a beam pattern that provides the strongest signal would be preferred. Given our configuration we assume by reciprocity that the beam pattern that provides the strongest receive value is ideally the one that provides the strongest transmit value (receive value for the distant end) as well. Given a discrete set of beam patterns then a client will likely remain able to be reached for some time under a particular beam pattern and is one of the primary reasons for which we will break the period up into discrete intervals later on.

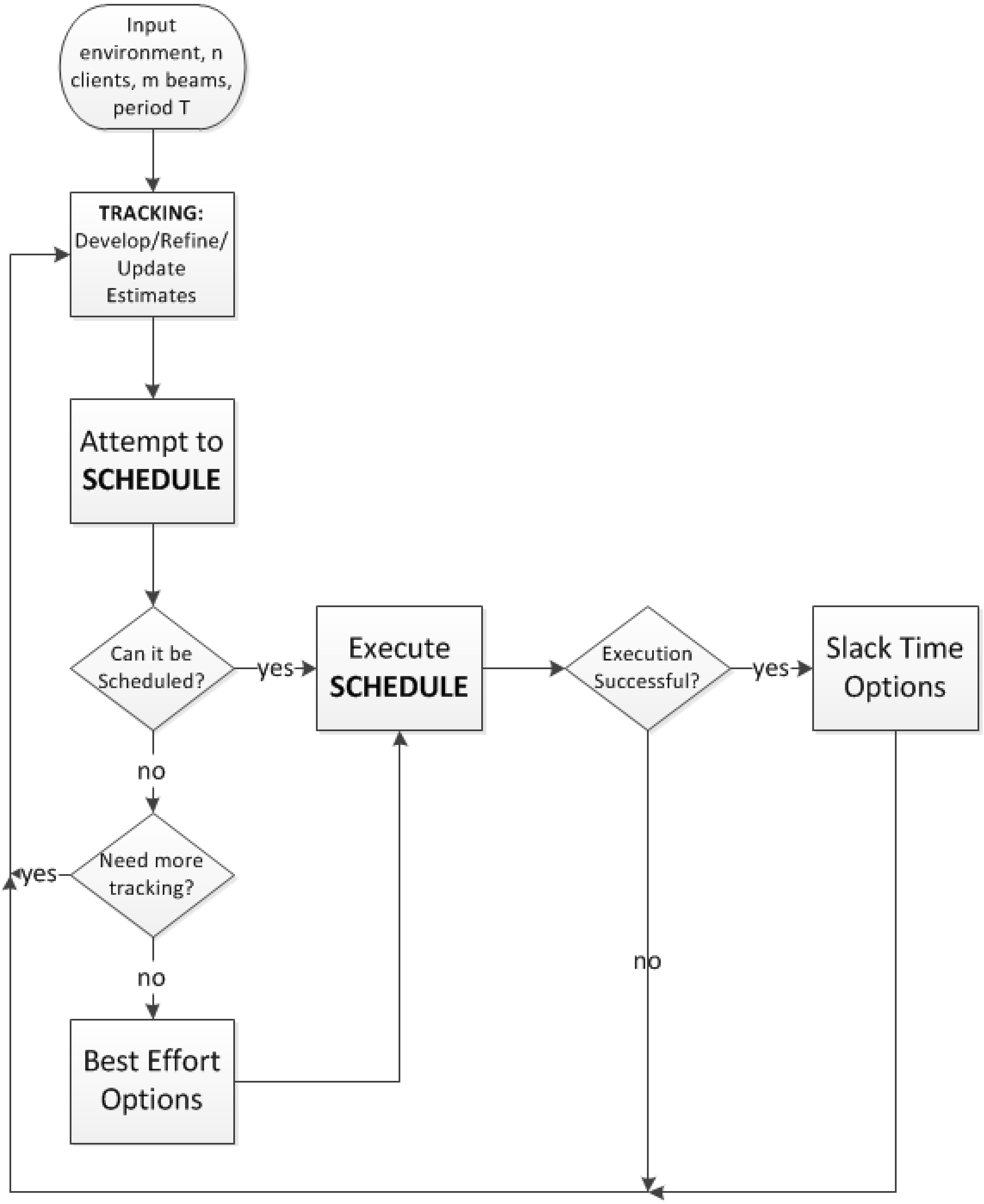


Figure 2.1: Problem Overview

The purpose of tracking is to develop capacity estimates that can be used to schedule client communication. If successful then the quality and quantity of tracking is adequate given those criteria.

2.4 Schedule

See “Attempt to Schedule” and “Can it be scheduled” in Figure 2.1. Data can be transferred using only one beam and client combination at a time. However, the time within a period, T, can be divided among different beam and client combinations. Let $p_j^i(s) = 1$ iff client j communicates using beam i with the hub at time s. For each s there is at most one (i, j) pair such that $p_j^i(s) = 1$. Since only one client can communicate with a beam at a time then we subsequently say, over the total period, T, that $\int\limits_0^T \sum\limits_{j=1}^n \sum\limits_{i=1}^m p_j^i(s)ds \leq T$. Furthermore, $\int\limits_0^T \sum\limits_{i=1}^m p_j^i(s)c_j^i(s)ds$ is the total data transferred to client j over period T. Since data is bursty and may momentarily exceed the channel capacity, we consider a planning period over period T, as given. If the average data transfer rate for client j is λ_j, as given earlier, then the goal is to transfer $T\lambda_j$ for each client j over the period T.

We assume that T is sufficiently large to assist in scheduling. For instance, in the orbit of a UAV above ground clients, the UAV may only be able to communicate with each client at certain points in its orbit. In this case, T would be sufficient to include one UAV orbit.

If a schedule can be developed to meet the goal as mentioned then we proceed to “Execute Schedule” as given in Figure 2.1. If not then we proceed to determine if we “Need more tracking”. However, before proceeding note three aspects to scheduling.

2.4.1 Overhead

While ongoing traffic transfers will keep the capacity of some beam and client pairs up to date, other, potentially better, combinations may go unexplored unless time is used for tracking. In addition to this tracking specific overhead, there can be other unmodeled overhead in order to coordinate client and hub activities or protocol specific. As a result, outside of scheduling traffic, the scheduler should reserve time for the tracker to update alternate beam and client pair capacity estimates and to help account for additional overhead. This reserved time not used for transferring data is a part of what will be called “slack” time. The slack time should be maximized by the scheduler.

2.4.2 Beam Switching

We assume that propagation delay is negligible but the beamforming antenna requires time to switch between beams. This is a non-linear factor as it takes a fixed amount of time to switch beams each time no matter how much time is spent on the pattern. The antenna used in this work, described in Chapter 3, requires $\approx 100\mu s$ to switch beams. Schedules that minimize beam switches will help reduce switching overhead, which helps increase throughput.

2.4.3 Scheduling with Uncertainty

Throughout the scheduling process, a key problem is obtaining the current possible rates and offered loads for clients as well as future estimates of what these rates and loads might be in the future. We may rely on tracking and also data exchanges to help obtain this information but tracking continuously or exchanging data with all of the users simultaneously is infeasible, nor is being able to forecast perfectly what those values will be into the future. This therefore will transform our scheduling into a scheduling with uncertainty problem.

2.5 Execution

If we can schedule in Figure 2.1 we proceed to “Execute Schedule.” At this point a schedule has been developed and the hub follows the schedule as given and considering any hardware, protocol characteristics, etc. depending on the scenario. During this process it is assumed that during the data exchanges the system will be able to gain opportunistic feedback that might be used towards updating capacity estimates due to discrepancies, hardware failures, etc. Note that if a receding horizon controller is used, only the initial period $T' < T$ of the schedule is executed before reassessing the schedule.

2.6 Post Execution

See "Execution Successful?" and "Slack Time Options" in Figure 2.1. Once the schedule has been executed it will have either failed to meet the goal within period T or it will have been successful with some remaining slack time. If it fails then one or more of the capacity estimates was wrong during execution (whether a wrong estimate, equipment failure, etc.) and ideally it will have gained opportunistic feedback to determine why it failed. It then proceeds back to repeating the process for the next period T with this additional feedback to help update the next set of estimates. If successful then the system has additional slack time available to account for possibly additional overhead as mentioned above. This could also include tracking new users, maintenance aspects, etc. Following this slack time the system would then proceed back to repeating the process for the next period T, with the additional feedback it was able to gain from execution and possibly during the slack time as well.

2.7 Unscheduled

See "Need more tracking" and "Best Effort Options" in Figure 2.1. If a schedule cannot be developed then the system should determine if more tracking is needed to gain a better estimate for one or more users. If so then it goes back to tracking and could explore better tracking options to develop better estimates that might allow a schedule to be developed. Or, the system could come to the determination that a successful schedule is not feasible given the set of estimates for the respective clients and their demands. At that point the process could decide what are the best effort options. These could be to prefer higher rate users (essentially dropping lower rate users), prioritizing users, etc. Once making this determination then it proceeds to execute a schedule based on these updated parameters.

2.8 Research Question

This work addresses a situation where a hub is wirelessly servicing users in a dispersed mobile infrastructure based scenario as introduced in Chapter 1 and presented up to this point. We seek to understand how to optimally manage the clients' data requirements while considering mobility and a dynamic radio environment using a hub with a beamforming antenna capability.

2.9 Methodology and Contributions

Figure 2.1 does not present a solution but simply breaks the question down into separate manageable components (smaller problems) while presenting the criteria necessary in order to be able to proceed from one component to the next. These components can be collectively separated into scheduling, tracking and execution.

For scheduling, in Chapter 4 we consider what Kleinrock states in [47], "The basic performance parameters of any resource sharing system include the following: 1. the system response time or delay, 2. the throughput, 3. the resource capacity, [and] 4. the resource utilization." We take a linear programming approach to optimally scheduling client specific traffic in a mobile hub and spoke scenario using a beamforming antenna. We use it to maximize client throughput while considering delay constraints as applicable. We show how to incorporate l_1 norm regularization to account for non-linear factors such as beam-switching while still maintaining optimality and maximizing slack time. We then optimize the beam-switching sequence using a shortest path first approach to further mitigate the inter-switch delay and associated protocol specific overhead and help create additional slack time. We illustrate that this system can easily be extended to a networked system of multiple hubs. Finally, we present what might happen in a situation where a schedule cannot be developed, either due to not enough information being available or not enough resources being available to meet the demand. These factors are largely scenario dependent. Throughout the scheduling process, a key problem is obtaining the current possible rates and offered loads for clients as well as future estimates of what these rates and loads might be in the future. This transforms our scheduling into a scheduling with uncertainty problem as discussed in Chapter 5.

We find in our work that scheduling with uncertainty is a significant factor and requires its own chapter, Chapter 5 - Scheduling with Uncertainty. In this chapter we show a simple receding horizon type approach to address the uncertainty associated with scheduling clients data exchanges due to their mobility. We show that by using a receding horizon approach with a linear program that aggregates future time periods into a small number of intervals not only reduces the complexity of the scheduling but also allows us to use less precise future information. We found that good performance is possible in the simplest case that only considers the current time interval and an aggregate of all future bins. This greatly simplifies the linear program (and scheduling more generally) presented in Chapter 4. Further, it suggests that gross estimates of performance and traffic into the future are sufficient to schedule. Though simple, this approach does significantly better than a simple greedy approach which looks only at current information. These results suggest that we can efficiently schedule communication traffic well using only imprecise information. Though our simulations used aggregates of precise future information, we expect the performance to be similar with estimated future information using for example, average trajectory lengths and loads measured as the scenario progresses. These results can be extended to more complicated military type scenarios such as the UAV problem, convoy operations, logistics site operations, as well as this work. We then proceed to tracking for which to develop those capacity estimates for scheduling client communication given these relaxed requirements.

For tracking, in Chapter 6 we find that it is more scenario dependent and build off concepts presented in related work and in preliminary field experiments. Our main method is similar to the dithering method that is used to track satellites, see [23]. The criteria for tracking, as discussed, should allow for a schedule to be developed to meet the clients' needs. Our implementation does that and if not then it conducts steps to further refine the tracking that was done. This includes a dithering track or a full track to regain a client's "bearings," or find additional clients. Our tracking mechanism also uses the data exchanges to obtain additional information opportunistically in order to gain better estimates. Chapter 7 further expands on our tracking mechanism as used in a full implementation.

Chapter 7, Prototype Implementation, shows how all of these areas are integrated in a full prototype implementation of a solution to the problem. Our implementation shows that the problem overview as provided by Figure 2.1 provides a solid basis and defines the key components needed for a reliable electronic beamforming antenna system able to successfully service dispersed users in a mobile environment. This implementation also shows the tools that we have developed, refined, and integrated with respect to tracking, scheduling, and practical modifications.

Finally, Chapter 8 presents a final conclusion to this work and what future work exists.

Chapter 3

Background and Related Work

3.1 Background

This chapter is a reference to understanding the subject matter in this work.

Section 3.1 covers background material in 802.11 (Section 3.1.1), the pros and cons of using a directional versus an omni-directional antenna (Section 3.1.2), basic resource allocation concepts (Section 3.1.3), queueing theory references with respect to this work (Section 3.1.4), reference for analysis of prior work and in proper conduct of experimentation (Section 3.1.5), and finally background information on Fidelity Comtech's Phocus Array FCI-3100X used in this work (Section 3.1.6). Any of these sections can be skipped at the reader's discretion.

Section 3.2 starts off with a summary of prior work and how it relates to this work. The reader should consider reviewing this, up to Section 3.2.1, at a minimum. A thorough analysis of prior work is then presented from Section 3.2.1 until the end of the chapter and can be skipped as desired.

3.1.1 Basic Relevant 802.11 Information

The information in this subsection is taken largely from [31][74]. This work is a general wireless problem but given the equipment we have, will be implemented and tested using the 802.11 protocol. The key benefits of wireless are the mobility and significant amount of flexibility it provides in enabling users to accomplish many varying tasks. The key benefit of 802.11 in particular is its intended pervasiveness and the low cost of equipment. This is mostly due to it operating in the unlicensed S-Band ISM. Specifically, this consists of up to 14 channels varying from 20-22 MHz of available bandwidth depending on the modulation scheme used. The center frequencies range from 2.412 GHz to 2.484 GHz and separated by 5 MHz each (except Channel 14), see Figure 3.1.

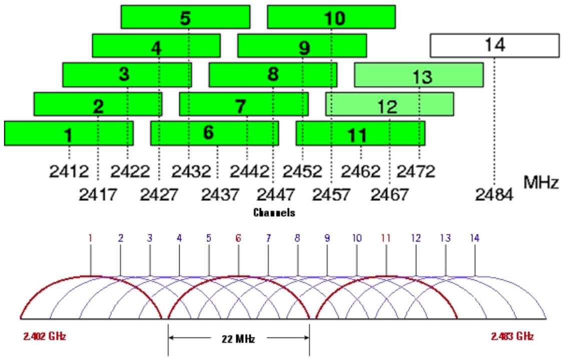


Figure 3.1: 802.11 channels

In this work 802.11b and 802.11g are used in particular. 802.11b uses Direct Sequence Spread Spectrum (DSSS) with Complimentary Code Keying (CCK) or Packet Binary Convolutional Coding (PBCC) coding schemes with data rate specifications as shown in Table 3.1. As shown, it utilizes four different data rates from 1 to 11 Mbps provided via different modulation variations and code lengths. Figure 3.2 then shows the Spectral Mask used by the 802.11b DSSS configuration. As can be seen there is only a required attenuation starting at $f_c \pm 11MHz$ of -30 dBr and then at $f_c \pm 22MHz$ of a total of -50 dBr. It can also be seen that there is no attenuation at the center frequencies for the possibly two channels to the left or two channels to the right of a particular channel.

Table 3.1: Date Rate Specifications for IEEE 802.11b

Date Rate (Mbps)	**Code Length**	**Modulation**	**Symbol Rate**	**Bits/Symbol**
1	11 (Barker Seq)	BPSK	1 Msps	1
2	11 (Barker Seq)	QPSK	1 Msps	2
5.5	8 (CCK)	QPSK	1.375 Msps	4
11	8 (CCK)	QPSK	1.375 Msps	8

802.11g is backwards compatible with 802.11b while also using Orthogonal Frequency Division Multiplexing. It has the additional data rate specifications as shown in Table 3.2. With OFDM

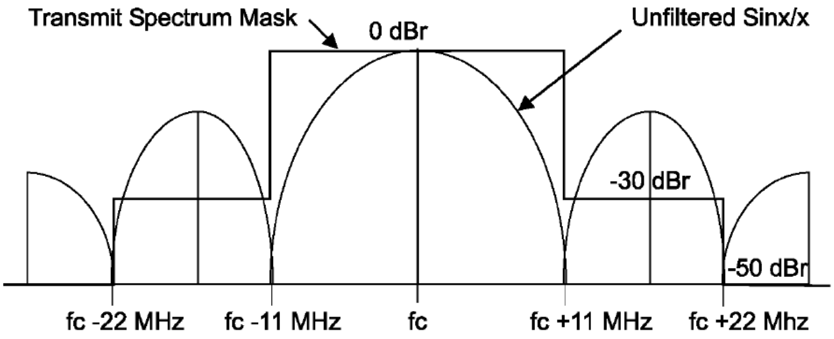


Figure 3.2: 802.11 DSSS Scheme Spectral Mask

Table 3.2: Date Rate Specifications for IEEE 802.11g

Data Rate (Mbps)	**Modulation**	**Coding Rate**	**Coded bit per-subcarrier**	**Coded bits per OFDM symbol**	**Data bits per OFDM symbol**
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	16-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

and the various modulation constellations, it has a range of 8 different possible data rates from 6 to 54 Mbps. Figure 3.3 then shows the respective 802.11 OFDM Spectral Mask. It is somewhat different from the one used for DSSS. As shown there is only a required attenuation starting at $f_c \pm 9MHz$ down to -20 dBr at $f_c \pm 11MHz$. It then drops to a total of -28 dBr at $f_c \pm 20MHz$ and then down to -40 dBr at $f_c \pm 30MHz$. It shows that it also has minimal attenuation between adjacent channels.

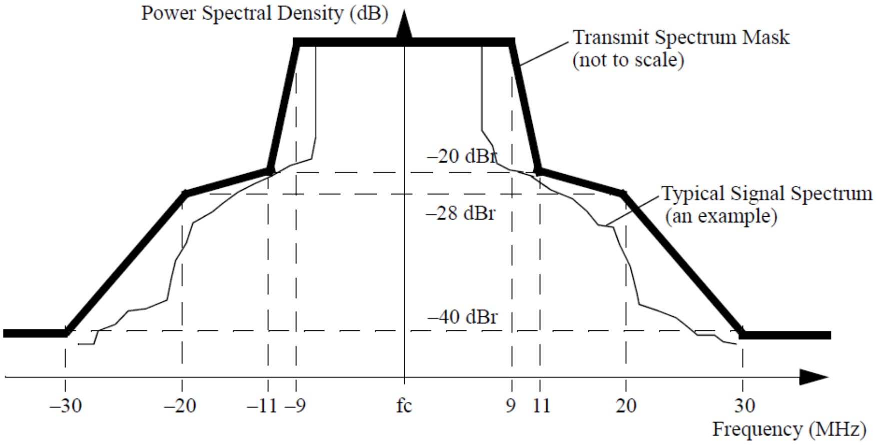


Figure 3.3: 802.11 OFDM Scheme Spectral Mask

Overall, given both the DSSS and the OFDM masks, there is quite a bit of overlap between adjacent channels. Looking at Channel 6, there is less than negligible overlap out to Channel 2 on one side and Channel 10 on the other side given the DSSS mask. The OFDM mask is even less restrictive, as seen. Therefore, a key limitation of 802.11 is from a bandwidth perspective and the amount of contention (resulting in interference) not only within a particular channel but also across adjacent channels. Another key limitation, as via any wireless medium, is from a security perspective where anybody can listen in on any channel at any time and hear ongoing conversations as they wish unless some form of mitigation (i.e. encryption) is used. The security aspects always have to be taken into account but will be discussed minimally going forward within this work. Non malicious factors regarding congestion, contention, and interference will however, play a large role in the problem that this work is addressing.

There are two types of 802.11 networks or basic service sets. The Independent Basic Service Set (IBSS) is more commonly called an *ad hoc network* and the more common Infrastructure BSS is the default network type unless specified otherwise. *Ad hoc* refers to any two or more stations setting up a network amongst each other. It typically extends out in a mesh type configuration. In *infrastructure* mode you have an access point (AP) that serves as a central point, or hub as we've been referring to, for which all communication coming from or going to an individual client is passed through. This work is using an infrastructure based scenario.

The following discusses the basics of an Infrastructure BSS network from set-up to the tools used to deal with contention and congestion when there exists multiple users located in the basic service area of the basic service set.

Access points (hubs) distributed throughout an area broadcast a beacon frame every 100 Time Units (TU - typically 1 ms per TU), by default, in order to announce their presence. The beacon frame consists of a MAC header, the body, and a frame check sequence. Included in the body is a timestamp whereby listening or associated stations can synchronize their clocks. Also included is the capability information regarding the access point/device/network. It announces the type of network such as **ad hoc** or **infrastructure**. Also included is its Service Set Identifier (SSID - a name that identifies a particular wireless LAN), what rates it supports (as to whether 802.11b, g, n, etc.) and other parameters. Key to the beacon frame broadcast is that it is specifically a half-duplex transmission and requires no response from either unassociated (not connected) users or associated users. Similarly, clients can send probe request frames when they are either searching for an access point or require information from an access point. The access point sends a requisite response frame that includes the same basic capability information as contained in the beacon frame. Once a client has identified an AP, either through a beacon or a probe response, and it wishes to connect to it then it sends a simple authentication request that the AP responds to in turn by default. The client then sends an association request frame. This allows the access point to allocate the necessary resources before sending a response frame.

After a client is associated, it can then initiate data transmissions to either another directly connected (to the AP) client or to a client somewhere else in the network, assuming the access point is part of a network extension.

802.11 traffic is not coordinated or scheduled and therefore is subject to contention as a result of multiple users trying to contend for the same channel during the same time period and therefore likely interfering with each other. The 802.11 standard defines two forms of medium access to help account for this called distributed coordination function (DCF) and point coordination function (PCF). DCF is mandatory and based on the CSMA/CA (carrier sense multiple access with collision avoidance) protocol.

Using DCF, 802.11 stations contend for access and attempt to send frames when there is no other station transmitting. Before it accesses the medium, the MAC Layer checks the value of its network allocation vector (NAV). The NAV is a counter maintained by each station that lists the amount of time needed by the previous frame to be sent. The NAV must be zero before a station can attempt to send a frame. Once the NAV is zero the transmitter then waits a DCF interframe space (DIFS) before transmission can begin. Prior to transmitting a frame, a station calculates the amount of time necessary to send the frame based on the frame's length and data rate. The station places a value representing this time in the duration field in the header of the frame. When stations receive the frame, they examine this duration field value and use it as the basis for setting their corresponding NAVs. This process reserves the medium for the sending station.

An important aspect of the DCF is a random backoff timer that a station uses if it detects a busy medium. If the channel is in use, the station must wait until the transmission is complete, then wait the DIFS (Note that if the previous transmission had errors then it waits a variable Extended IFS (EIFS) depending on the length of the frame), and then waits a random exponential backoff before it transmits. The first backoff is uniformly distributed between 1 to 31 slots. If the transmission is not successful then it waits again for any ongoing transmission to end, the DIFS period again, and then a random backoff between 1 to 63. The next attempt would be 127 and so on with each $2^{a+4} - 1$ (a referring to the number of the attempt). Once a successful transmission occurs then the backoff goes back to the base value of 31. This exponential backoff procedure helps to ensure that multiple stations wanting to send data don't immediately transmit at the same time once the last transmission has completed. The backoff timer significantly reduces the number of collisions and corresponding retransmissions, especially when the number of active users increases. The problem with it is that if an individual user needs to send data then there is a chance that it can continue to randomly pick a later slot and continue to lose out on sending its data before others, if at all. Another problem is that in a time of high contention, it doesn't necessarily make sense to always go back to the original backoff range upon one packet being sent through successfully. These issues will be discussed more in related work. Also, note that for high priority transmissions such as RTS/CTS, ACKs etc., the requirement for waiting is a Short Interframe Space (SIFS) instead of the longer DIFS.

Since this is a radio-based LAN utilizing transceivers then most transmitting stations can't listen for collisions while sending data (unless using more advanced equipment). Therefore the receiving station needs to send an acknowledgement (ACK) if it detects no errors in the received frame. If the sending station doesn't receive an ACK after a specified period of time (a set parameter), then it will assume that there was a collision (or RF interference) and retransmit the frame.

For supporting time-bounded delivery of data frames, the 802.11 standard defines the optional point coordination function (PCF) where the access point grants access to an individual station to the medium by polling the station during the contention free period. Stations can't transmit frames unless the access point polls them first. The period of time for PCF-based data traffic (if enabled) occurs alternately between contention (DCF) periods. The access point polls stations according to a polling list, then switches to a contention period when stations use DCF. This process enables support for both synchronous (i.e., video applications) and asynchronous (i.e. e-mail and web browsing applications) modes of operation. No known wireless access points on the market, however, implement PCF.

Additional key control frames are used to help mitigate contention and to facilitate data exchange, including the optional use of Request-to-Send (RTS) and Clear-to-Send (CTS) frames.

In order to reduce collisions, a transmitter wishing to initiate a data exchange can send out an RTS frame to the intended receiver. This request identifies the sender and desired receiver as well as providing the total time needed for handshakes and to send the respective data. The receiver acknowledges this, if applicable, with a Clear-to-Send (CTS). The CTS updates the time needed for the exchange. Other clients in the vicinity that hear it then populate their NAV with the backoff data obtained from the RTS/CTS frames and back off accordingly in order to not interrupt this data exchange. Many times RTS frames are used in conjunction with the sending of larger frames in order to minimize the cost of a collision and having to re-send a short RTS rather than the larger frame. In that case an RTS threshold can be set in which if a frame is larger than that threshold then an RTS will be sent automatically prior and a CTS frame sent in return. If the frame is smaller than the threshold then the normal DCF sequence is followed.

3.1.2 Use of Omni vs Directional Antennas

A limitation of the standard 802.11 scenario is that most devices operate with an omni-directional antenna. Although cheap and simple to use, an omni-directional antenna such as a $\frac{1}{2}$-wave dipole antenna has a radiation pattern similar to that shown in Figure 1.5. The Effective Isotropic Radiated Power (EIRP) is limited to 100 mW by the FCC, for 802.11. Assuming a 1 kilometer link, the Free Space Path Loss at 2.412 GHz is approximately 89.64 dB.

$$FSPL(dB) \cong 20log_{10}(1.00) + 20log_{10}(2412) + 33.45 = 89.64dB$$

This assumes sufficient Fresnel zone clearance as well as no scattering or interferers, as presented earlier. To continue with these assumptions, a typical wireless client has about 15 dBm of transmit power and a receiver sensitivity of -94 dBm at the lowest transmitted rate of 1 Mbps for 802.11b. The scenario also assumes that the lowest Signal-to-Noise Ratio (SNR) acceptable to receive at 1 Mbps is 4 dB. The received power at 1 kilometer given this scenario is -70.44 dBm.

$$RcvrPower(dBm) = 15.0dBm + 2.1dBi(TxGain) + 2.1dBi(RxGain) - 89.64dB = -70.44dBm$$

This is better than the receiver sensitivity for 1 Mbps and the maximum channel noise in this case (assuming the 4 dB minimum) is a reasonable -74.44dBm. Nevertheless, these calculations are based on an ideal scenario for a single point to point link with no obstructions or scatterers, interferers, fading, etc. A typical real scenario limits the range comfortably to about 35 meters indoors (for 802.11b, g) and about 100 meters outdoors, although up to about 250 meters is usually possible. The energy is focused in all directions, see Figure 1.5 again, and so a significant amount of that energy is wasted, as discussed in Chapter 2. For the same reason it can create a lot of interference for all users within that transmitting radius. Once used, all clients within that radius are aware (assuming they have heard the exchange) of ongoing communications but all additional communications exchanges in that area are delayed until the data exchange is completed, which significantly decreases the overall throughput of all users within that area.

If the data exchange can be focused in the general direction of the client that it is intended for then the majority of the energy can be focused in that direction and therefore the range is extended. Such is a key benefit of the use of beamforming antennas. This minimizes the amount of interference that takes place in the circular area around a client because the energy transmitted is concentrated along a particular path (or in a more specific direction) between the receiver and transmitter versus in all directions. This characteristic then allows for greater spatial reuse since the transmission can be focused in one direction (pattern coverage area) and so other transmissions can take place in the same area as long as they aren't within the path of the other transmission. Some basic modifications to the 802.11 protocol have been proposed to account for this spatial reuse as a result of beamforming antennas [40]. These modifications refer to what is called Directional MAC or DMAC. This includes the Directional NAV (DNAV). It is the same as the NAV but with respect to a particular direction. In other words if client a is aware of a directional communication session about to take place adjacent to it between clients b and c along a particular vector then it can update its DNAV to allow for a necessary back-off to keep from transmitting in those particular directions. The benefit is that it can still transmit in other directions as necessary given that it doesn't interfere with those identified directions. Using the scenario above, the FSPL over 1 kilometer was 89.64dB. Using one of this array's directional beams with the provided specifications (to include the same transmit power) the received power (for a typical omni client) at 1 kilometer away is now approximately -57.54 dBm.

$$P_{RCV}(dBm) = 15.0dBm + 15dBi(TxGain) + 2.1dBi(RxGain) - 89.64dB = -57.54dBm$$

This is 12.9 dB greater than the previous scenario of a typical omni client broadcasting to another omni client. If that typical client was a typical commercial wireless access point (approximately 20 dBm transmit power) and using the same gain then the received power would be 17.9 dB more. Clearly the beamforming capability provides an advantage. Currently there are three main types of configurable antennas [1]. They are:

- Switched-beam antenna(s): An antenna or set of antennas providing a finite set of gain patterns from which the user can select one at a time.
- Steerable antenna: An antenna having a pattern that is fixed except for rotation, and can be rotated continuously in the azimuth and/or elevation planes. (An antenna that can be rotated only in discrete increments is effectively a switched-beam antenna).
- Beamforming antenna: An antenna having a pattern that can be varied continuously in real-time to optimize some signal property. Especially an antenna that uses pilot tones to maximize the Signal to Interference Ratios (SIR) for one or more pre-determined stations.

It is necessary to be able to make the distinction when discussing the related work later. However, with respect to this work we will continue to generally use beamforming, unless specified otherwise. As shown in [76], we can express the antenna radiation pattern for a K-element array as

$$A(k) = a_0 exp^{jk\phi_0} + a_1 exp^{jk\phi_1} + ... + a_{K-1} exp^{jk\phi_{K-1}} \tag{3.1}$$

Therefore each a_i and $k\phi_i$ refer to the respective magnitude and phase applied to the $i + 1^{th}$ antenna element and collectively represent the complex weight applied to the antenna elements overall. Consider Figure 3.10 and the beam with a centered direction of 247.5° for a switched beam configuration. If we consider that as the main lobe and express that using Equation 3.1 for a given fixed k and then a particular set of a_i and ϕ_i values, then to obtain the beam centered at 315° it simply requires an adjustment of the set of a_i and ϕ_i values applied. Therefore, given the equation for the main lobe beam for a K-element array, we can express the parameters for any of the other default beams in terms of a set of a_i and ϕ_i values. This representation will come in useful when we consider multi-casting using multi-lobe patterns, discussed towards the end of this chapter.

Along with the benefit of greater range, less local interference and better spatial re-use comes some additional challenges with using beamforming antennas as mentioned earlier. For example, what is the client's location in order to be able to track them as communication exchanges are ongoing, particularly if they are out of omni-directional range but within the range of one of the directional beams? It is necessary to know which beam to use. Consider a scenario as given in Figure 3.4. In the ideal situation the beam pattern to use would be beam pattern 16. In some instances a multi-path radio environment due to scatterers, interference, etc. might cause a different beam pattern to be more ideal for the same scenario. This would be the result of another direction or angle of arrival for the respective client's signal being stronger and therefore needing a different beam pattern. This method of beam determination is referred to as Angle of Arrival (AOA) or Direction of Arrival (DOA) in the related work discussed later. AOA typically accounts for reflection, refraction,

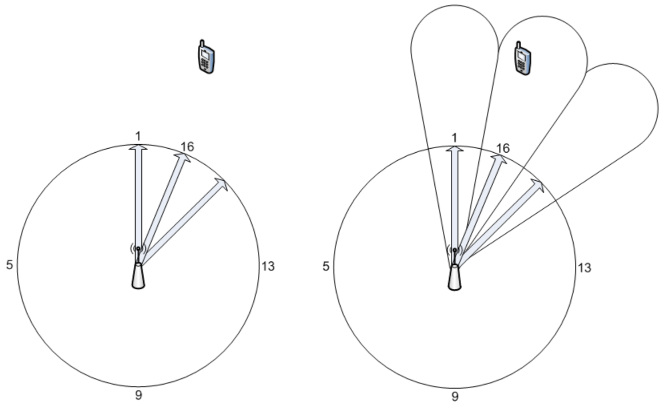


Figure 3.4: Tracking a Client

and other multi-path related phenomena that result in a straight line or, “as the crow flies”, not necessarily being the best direction. Also, a beamforming antenna doesn’t necessarily mitigate interference but merely changes the characteristics of it. The interference changes from being in all directions to being in a particular beamform coverage area. Nevertheless, as can be seen in [36], the authors develop an analytical approach to evaluate the single-hop performance of the IEEE 802.11 DCF based MAC algorithm combined with the use of directional transmitting and receiving beams at each mobile unit. Their results show that with adequate tracking, the directional CSMA/CA system can provide a significant upgrade of network performance when the beamwidth is properly selected. Namely, the best results were obtained with the smallest beamwidths, down to 30° in their work.

However, use of beamforming antennas with 802.11, or using DMAC, can also cause variations of the hidden terminal issue due to both asymmetry in gain and due to unheard RTS/CTS. The issue of deafness is also very prevalent as a result of using beamforming antennas. Basic Hidden Node problems can exist in all 802.11 scenarios (whether omni or beamforming). See Figure 3.5. An access point exists between Node 1 and Node 2. The access point and Node 1 can hear each other and the access point and Node 2 can hear each other but Node 1 and Node 2 cannot hear each other. It is possible that both Node 1 and Node 2 send packets to the access point that arrive simultaneously and collide. The access point would not be able to make sense of anything while Node 1 and Node 2 would not understand that there was a problem. Variations of this crop up

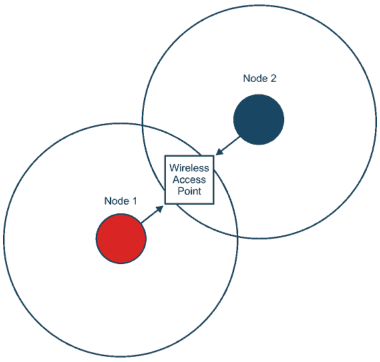


Figure 3.5: Basic Hidden Node

more extensively in the beamforming sense due to asymmetry in gain, see Figure 3.6, and also unheard RTS/CTS, see Figure 3.7. In Figure 3.6, node C has initiated communications with Node B. Node A sits idle omni-directionally and therefore is unaware of the communication session. Node C sends data while Node A tries to send an RTS to node B and therefore causes a collision. This is called a hidden terminal issue due to asymmetry in gain. In Figure 3.7, node A on the left has pointed towards node D. Node A does not hear the RTS/CTS exchange during this time and finishes his exchange. After his exchange, on the right, he points towards node C and tries to initiate a conversation, therefore interfering with the exchange between B and C. This is called a hidden terminal due to unheard RTS/CTS. These variations will be discussed more in related work as to how they might become a problem, depending on the method being used, and how they might be overcome.

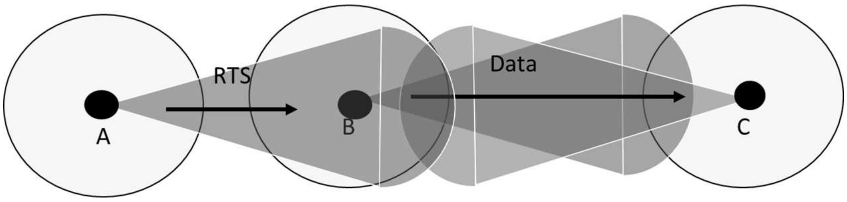


Figure 3.6: Hidden Terminal due to Asymmetry in Gain

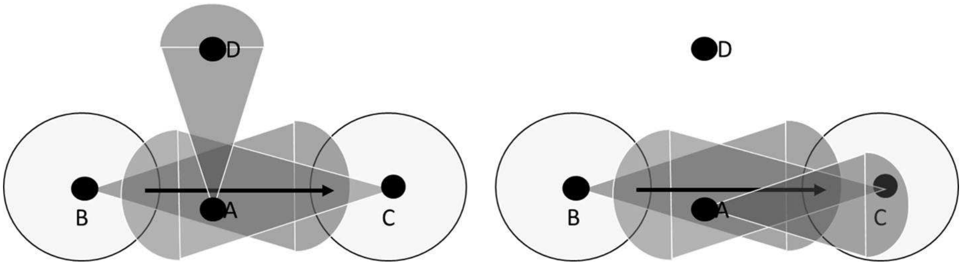


Figure 3.7: Hidden Terminal due to Unheard RTS/CTS

Deafness exists mostly in a directional configuration. See Figure 3.8[15]. There is a communication session going on between node A and node B in one direction. Node C in another direction is unaware of this ongoing session and transmits an RTS to the access point. Node A from either not being able to hear or due to the ongoing session does not reply to Node C. Node

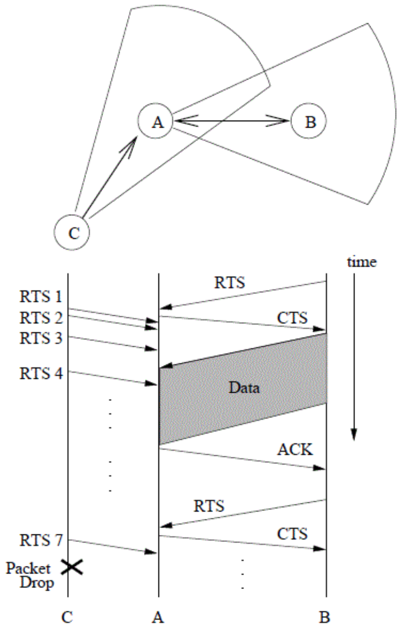


Figure 3.8: A Deafness and Unfairness Illustration

C assumes contention and might back off before trying again. Node C could continue to try and try until it eventually assumes that node A is no longer available. Node C is deaf to the ongoing data exchange that was going on at the access point. Variations of how the deafness issue crops up using various beamforming techniques, and how it can be overcome, will be discussed similarly to the hidden terminal issue in related work.

3.1.3 Basic Resource Allocation Concepts

Fairness is an issue in allocating resources. The landmark paper by Jain et al., [39], discusses the basics of allocating resources (network) in a resource limited scenario. The key contribution of this paper, [39], is to break down the problem of fairness into two parts: selection of appropriate allocation metric, and a quantification of equality. Possible allocation metrics include: response time, response time per hop, throughput, throughput times hops, power, and fraction of demand. They proposed a formula for quantifying the equality. The fairness function has many desirable properties which other ones do not satisfy. It is independent of scale of the allocation metric. It is bounded between 0 and 1 so that it can be meaningfully expressed as a percentage. Finally, it is continuous so that any change in allocation changes the fairness also. Essentially, if the variable being used to express a particular parameter for a user is x_i, then the fairness is

$$f(x) = \frac{[\sum\limits_{i=1}^{n} x_i]^2}{\sum\limits_{i=1}^{n} x_i^2}; x_i \geq 0.$$

The fairness function applies to any system with shared resources, therefore the discussion is independent of any particular application. In particular, they have derived a lower bound for the fairness of computer networks with window flow control. This provides a background understanding and fundamental way of measuring a key parameter, fairness, that ties into the following work.

A follow-on paper by Jain and Chiu, [14], provides a theoretical background understanding of how to deal with contention in order to mitigate congestion for wireless networks. Congestion avoidance (CA) works proactively to try and prevent congestion and congestion control (CC) is how a system reacts to a congestive state. The authors argue that the key component of any CA scheme is the algorithm or control function used by the users to increase or decrease their load (window or rate). They evaluate each user's change in load, or control based on four different control functions: Multiplicative Increase/Multiplicative Decrease, Additive Increase/Additive Decrease, Additive Increase/Multiplicative Decrease and Multiplicative Increase/Additive Decrease. They evaluate the effectiveness of these controls based on efficiency, fairness, distributedness, and convergence time (size of oscillations). The authors prove that for feasibility and in proving the optimal convergence to efficiency and the optimal convergence to fairness then the increase policy should be strictly additive and the decrease policy should be strictly multiplicative.

3.1.4 Queueing Theory

We initially take that the problem posed by this work could be included in the class of what is called an M/G/1 type vacation model with nonexhaustive service [46][47]. An M/G/1 queue refers to a queue with Poisson distributed arrivals (M) and generically (independent and identically distributed) serviced (G) from a single server (1). We can then categorize our service and vacation model as given in [78].

With regards to this work, a particular user reached via a particular pattern has some data to send. The server might service him to a certain extent but then takes a vacation for the purposes of possibly tracking or servicing other customers. The nonexhaustive characteristic is for the situation where the amount of data that a particular user has to send might require too much time to be done in the time, or whatever parameter used, that a given schedule has allotted that user. Although the reasons for this vacation are not typical (e.g. maintenance, deriving and disseminating schedule, etc.) for this type of model, the idea is the same.

Some work has been done in [85] where they study the effective throughput and delay performance in wireless scheduling (in an omni-directional and mobility non-specific scenario) by explicitly considering complexity through a delay model. Complexity largely refers to the overhead associated with developing a schedule and communicating or disseminating it to all of the clients. The time needed to do this is the vacation. Their problem and subsequent approaches can be extended to the work presented here.

3.1.5 Inherent Challenges in Analysis of Prior Work and in Conduct of Experimentation

In the paper by Kurkowski et al.[53], the authors argue how simulation is the research tool of choice for wireless network studies but credibility of the results have decreased based on their survey of the MobiHoc Proceedings from 2000-2005. The authors present that valid simulation results should meet four areas of credibility. These are:

- Repeatable.
- Unbiased - must not be specific to the scenario used in the experiment.
- Rigorous - must truly exercise the aspect of the Mobile Ad Hoc Network (MANET) being studied.
- Statistically sound - execution and analysis must be based on mathematical principles.

Their results illustrate that of the 151 papers reviewed, the vast majority do not meet one or more of these areas. The common pitfalls fell into four main categories:

- Simulation Setup. This was the most often skipped or overlooked category.
- Simulation Execution.
- Output Analysis.
- Publishing.

In the end the authors concluded that less than 15 percent of the simulations run were repeatable. Only 7 percent addressed initialization bias and none addressed any Pseudo-Random Noise Generator (PRNG) details and therefore over 90 percent of the papers' studies and experiments could be biased. Taking from the topics discussed in output analysis, it appears that only about 12 percent of the papers' results are based on sound statistical techniques. Therefore it remains difficult to analyze, infer, and synthesize results from past work in order to try and move forward and build upon them. It also highlights the importance of moving forward with quality, well thought out and controlled simulations and experiments that are communicated well.

In [59], the authors present a simple analytical model for IEEE 802.11 based wireless networks that effectively captures the important network performance characteristics such as throughput, delay and fairness. The paper by Buettner et al., [10], provides some interesting results to consider for experimentation involving dynamically steerable phased array systems. They describe the tools, methodology and metrics used to systematically evaluate topology formation algorithms using a dynamically steerable phased array system. Two resulting observations: (1) the environment heavily influences the structure of how antenna patterns interact. Techniques which depend on predictable null and lobe effects may work in some environments, but are likely to fail in others. (2) environmental variability in the long term is much greater than in the short term. Experimental trials should be short, so that the different cases can be examined under comparatively consistent circumstances. Experiments should also be repeated over a longer term (and in different locations) to verify that results hold over a range of conditions.

3.1.6 Phocus Background

For the purpose of this work the phased array antenna hub with beamforming capability used is Fidelity Comtech's (www.Fidelity-Comtech.com) Phocus Array FCI-3100X. See Figure 3.9 for a picture of the array. The majority of the details here are taken directly from Fidelity Comtech's Phocus Array System Manual [18]. As can be seen, it is an eight element uniform circular phased array 802.11b/g antenna unit. The array allows for its antenna radiation patterns to be electronically shaped (based on preset antenna weight configurations) and steered. It's EIRP is an FCC acceptable 42 dBm (but capable of 45 dBm). In its standard configuration it can radiate either omni-directionally or in 16 uniform directions from 0 to 360 degrees and each separated by 22.5 degrees with 43 degrees 3 dB (half-power) beamwidths and 35 degrees vertical beamwidth. Examples

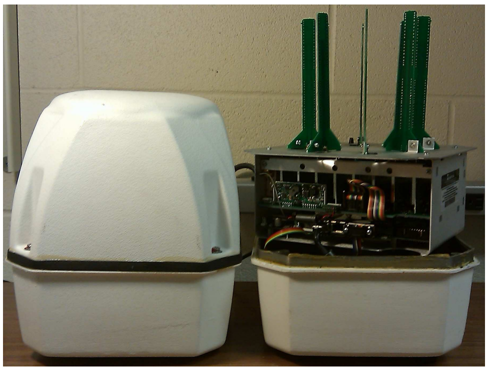


Figure 3.9: Phocus Array FCI-3100X both enclosed and opened

of these patterns, both directional and omni-directional, can be seen in Figure 3.10. Using one of the standard configuration patterns the antenna gain is a maximum 15 dBi. The beam direction can be changed (by specification) in less than 100 μs. Figure 3.11 shows a block diagram of the key components of the Phocus Array System. Each of the individual antenna elements are connected to a specific T/R (transmit/receive) module that configures that antenna's particular output (amplitude, frequency and phase) in conjunction with the collective beam shaping and steering. They then tie into an 8 : 1 splitter running into the IEEE 802.11b/g radio module wireless LAN (WLAN) card. The array is then powered via a Power Over Ethernet (POE) injector which is connected to a power supply and the injector also serves as its external network connection gateway. Its typical receiver sensitivity for 802.11b is $-96dBm$ at 1 Mbps and $-92dBm$ at 11 Mbps and for 802.11g is $-92dBm$ at 6 Mbps and $-76dBm$ at 54 Mbps. Note that this is based on the internal radio cards and does not include the receive signal gain via the beamformer. For these, the gain is typically $9dBi$ for omni-directional and $15dBi$ for a standard configuration pattern, as mentioned above.

3.2 Prior Work

Much work has been done over the years with respect to beamforming wireless resource sharing methods. A large portion of work in this field has been utilizing/modifying Directional Medium Access Control (DMAC) techniques. Much of the related past work with Directional MAC (DMAC) protocols involves modification of the Request to Send and Clear to Send (RTS/CTS)

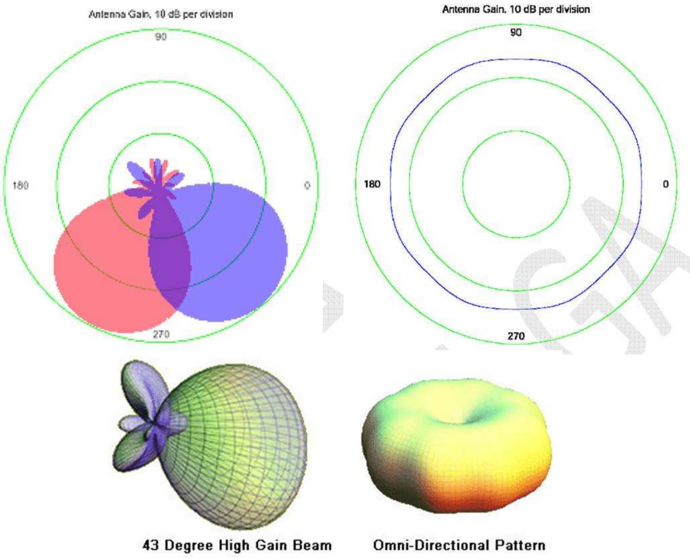


Figure 3.10: Example Standard Directional and Omni-Directional Beam Patterns [18]

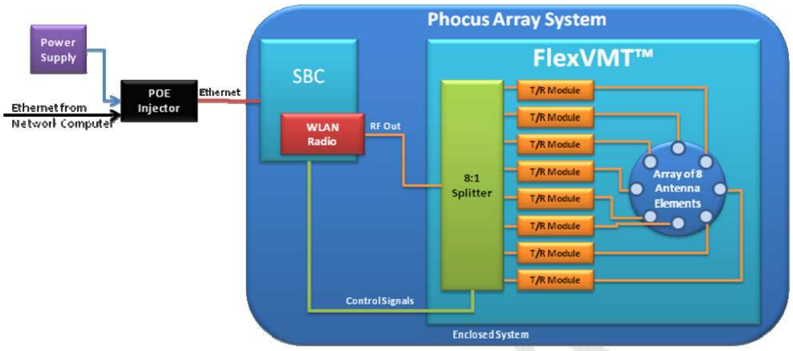


Figure 3.11: Phocus Array System Block Diagram [18]

mechanism as discussed earlier. In fact, in [21], they break up the DMAC protocols. The optional RTS/CTS mechanism is a key component of the random access methods aside from the Carrier Sense Multiple Access/Collision Avoidance mechanisms used by default in 802.11. Prior work involving modifications of RTS and/or CTS schemes will be discussed beginning in Section 3.2.1. For simplicity the following abbreviations apply:

- DMAC - Directional Medium Access Control.
- ORTS or OCTS - An RTS or CTS, respectively, packet sent omni-directionally.
- DRTS or DCTS - An RTS or CTS, respectively, packet sent directionally.
- ODRTS - An RTS packet sent omni-directionally or directionally.
- CRTS or CCTS - Refers to sending RTS or CTS, respectively, packets while cycling (circularly) through the beam directions.
- MRTS - Refers to sending multi-hop RTS packets.

Tone based mechanisms make up a small population along with various other methods in the random access category. They will be discussed in Section 3.2.8.

DMAC Directional TDMA (Scheduling) techniques, discussed in Section 3.2.9, primarily involve Time Division Multiple Access scheduling algorithms/techniques as referred to in this work. Other practical implementation techniques that generally apply to a different type scenario but offer insights are included in Section 3.2.10.

Another class of works that are categorized within the Operations Research field start in Section 3.2.11. They don't address the mobility (e.g. tracking) aspects as much but offer insights into the coordinating or scheduling of data exchanges, using beamforming antennas on many instances. We present game theory approaches in Section 3.2.12 that discuss ways in which a network situation similar to this form can be driven to a stable Nash Equilibrium or a Stackelberg Equilibrium via the use of beamforming antennas in the latter case. Section 3.2.13 offers related work using

Convex Optimization type approaches which also incorporates less complex linear programming approaches. Multicasting in Section 3.2.14 is concerned with sending an amount of data out to clients as quickly as possible. The problem is different from ours in that we are more concerned with individual data exchanges back and forth. However, they are using a hub and spoke type infrastructure while also incorporating convex optimization with linear programming and sub-optimal relaxed algorithms in their techniques. Scheduling with uncertainty in Section 3.2.16 deals with the difficulty in creating a schedule due to channel dynamics, as discussed in 3.2.15 with Channel Characterization, and user dynamics (mobility, offered load, possible throughput rates, etc.). The tracking of users creates a version of a multiarmed bandit problem and there are various approaches to dealing with uncertainty in scheduling.

The following Table 3.3 summarizes the related work that offers techniques to solving areas tied to our research question: Beamforming, Mobility (Tracking and Testing), and Scheduling.

The range column is for the purposes of beamforming where the work doesn't assume that their method will succeed without using beamforming - i.e. the limited range of a straight omnidirectional configuration.

Tracking refers to methods that make a conscious effort to account for the user's mobility given knowing their initial location (either by being given it or determining it) or the method that they are using will inherently track them for a slow mobile scenario as we have discussed. Examples of the latter (inherent case) are the CRTS or CCTS schemes which by sending an RTS/CTS in all directions will track them indirectly. It also includes techniques that make the assumption that the antenna automatically knows the angle of arrival (as discussed earlier) via reception, and therefore the pattern needed to reach a particular client. However, it should be noted that automatically knowing the angle of arrival off of one transmission, is not a trivial problem.

The mobility test column then identifies methods that provide test results of their system via simulation or field experiments in a mobile scenario. In fact, regarding mobility, Dai's paper, [21], discusses maintaining the beamforming antennas functioning in high mobility environments as a, "hard nut". They argue that it is difficult to infer the effects of mobility in the majority of the works mentioned in their 2006 overview. As was discussed in Kurkowski's paper earlier, only the implementation of the protocols in realistic mobile scenarios can verify the effectiveness.

Finally, the scheduling column refers to methods that schedule future data exchanges. Aside from it being the approach that this work will use, it then rules out random access methods, for example, and their inherent unfairness. It also mitigates any hidden terminal and deafness issues prevalent in the majority of non-scheduled methods.

Additional works exist in the sections below but mainly serve as background or further insights into understanding the problem or techniques within that category.

3.2.1 DMACORTSOCTS

In [62], they argue that the main problem of using beamforming antennas in such networks is due to the dynamic nature caused by the frequent node movements. This gives rise, as mentioned, to locating and tracking users while they are trying to access the channel randomly. They propose a MAC protocol that uses beamforming antennas in an ad hoc network where the mobile nodes do not have any location information. They apply their configuration to an ad hoc scenario but it can be applied easily to an infrastructure based scenario. The idea is to send an RTS packet out in all directions and subsequently the receiver sends out a CTS packet in all directions. The receiver, for which the traffic is intended for, notes the direction from which it received the RTS packet with the maximum power. In a similar manner, the sender estimates by the received CTS packet what the best direction is to use in order to exchange traffic with the receiver. Being able to do this by receiving one packet requires the antenna to have more advanced electronics in order to be able to do the real-time signal processing necessary to determine the strongest direction of arrival. They then transmit and receive data and ACKs on the respective beam directions that they've established. Their simulation results show that by using their protocol, with 4 directional antennas per node, the average throughput (vs. offered load) in the network can be improved 2-3 times over the traditional CSMA/CA with RTS/CTS and omni-directional antennas. Under mobility, the performance is minimally affected at speeds up to 3 m/s. Higher speeds were not tested.

Table 3.3: Related Work Method Summary

Work	**Range**	**Tracking**	**Mobility Test**	**Scheduling**
[62]		x		
[3]		x	x	
[48]		x		
[12]	x			
[45]	x	x		
[77]	x	x	x	
[49]	x	x		
[61]	x	x		
[16]	x			
[38]	x			
[22]	x			
[71]	x			
[73]	x			
[72]	x			
[15]	x			
[40]	x	x	x	x
[88]	x	x	x	x
[89]	x	x	x	x
[4]	x	x		x
[32]	x			x
[64]	x	x	x	
[54]	x			
[67]	x			
[55]	x			x
[63]	x	x		
[1]	x			x
[11]	x			x
[37]	x			x
[43]	x	x		x-
[76]	x			
[87]	x			
[86]	x			
[24]		x		
[42]	x	x	x	
[70]		x		
[58]				x
[52]	x			x-

In [3], they argue that in order to fully exploit the beamforming capability, everyone should know the direction of ongoing communications. They propose an adaptive MAC protocol. Users maintain an Angle Signal to Interference and Noise Ratio (SINR) table and an appropriate mechanism for null steering. The idea is a modification of the last method. A sender initially sends an RTS frame. Listening nodes hearing that then populate a table based on the Angle SINR, which is what direction they receive the strongest signal from for that particular user. A similar thing happens at the sender when the receiver sends back the CTS. Then the sender sends back an omni-directional SDC (start data communications) packet. Other nodes in the neighborhood can then issue separate RTS and CTS frames. They simply steer their nulls in the direction of ongoing communications and wait in omni-directional mode to receive RTS/CTS exchanges from unaware nodes. Their simulation results involve networks with 50 mobile hosts, operating at a transmission range of 200 to 350 meters. The speed of movement of an individual node ranges from 5-15 meters/second and they utilize the Random Waypoint method. They then test the improvement in wireless medium utilization by taking 20 node pairs at random. They find that with an increase in transmission range the number of simultaneous node pair communication sessions decreases more drastically in the omni-directional case (versus beamformed). This is expected given the spatial reuse that directional provides. They then evaluate the improvement in one-hop communication efficiency and similarly find that as the number of one hop events per minute increases, the beamforming configuration performs much better. This is expected given the increased range that beamforming provides.

The problem with the techniques that initiate a communication session by starting with sending omni-directional RTS and omni-directional CTS frames is that they do not take full advantage of the additional gain and range that beamforming antennas can provide and of the additional spatial re-use in the local vicinity. Also, both methods require more advanced electronics that can provide the signal processing necessary to determine the angle of arrival of the strongest signal, especially based off of one transmission. Equally, regarding Bandyopadhyay's method, additional capability would be needed in order to steer nulls in an arbitrary direction while communicating in others. Bandyopadhyay's method also doesn't consider informing users, in the direction of the null, of an imminent communication session. This could present a hidden terminal problem. Regarding their testing with respect to mobility, it appears it doesn't play a major role in the results. This could be attributed to the fact that omni-directional exchanges are used and therefore they don't have to determine which beam direction to use at any point in time, assuming the client stays within range. Also, only throughput is tested in Nasipuri's work without considering possible additional constraints with respect to delay. Only medium utilization is directly tested in Bandyopadhyay's work which should have the expected results shown whether static or mobile.

3.2.2 DMACODRTSOCTS

The paper by Ko et al., [48], discusses two basic schemes. The first scheme makes a basic assumption of knowing the neighbors' locations. To initiate a session, a sender sends a DRTS packet in the known direction of the receiver. The receiver then sends an omni-directional CTS packet which also provides the location information of the sender and the receiver for the benefit of other listening nodes. In the second scheme, the main difference is that the sender can decide whether to send an ORTS or a DRTS packet. If none of its antennas are blocked then it will send an ORTS packet where otherwise it will send a DRTS in the direction of the receiver, unless that direction is blocked. Antennas become blocked via hearing RTS or CTS frames being sent that identify imminent or ongoing communications in a respective direction. In either case the CTS frame is sent omni-directionally, as mentioned. Regarding node mobility the authors go on to present at the end that the method can be adjusted. When nodes are mobile then they acknowledge that the directional information used to send a DRTS can become stale and therefore it might be necessary to resort to using an ORTS frame instead. In this case, for mobility, the scheme essentially becomes a DMACORTSOCTS scheme as discussed earlier.

Their results show that these techniques perform better than 802.11 in both cases. Similar to the last section, setting up a data exchange is limited to the omni-directional range because although an RTS packet might be sent directionally, the CTS packets are still sent back omni-directionally. In the first scheme, sending a DRTS could allow for greater spatial re-use but could also introduce hidden terminal issues with respect to the sender and clients in the sender's vicinity that are outside the range of the receiver's subsequent CTS packet. The second scheme can counter this but then has the same issues as the ORTS and OCTS schemes discussed prior. The paper provides no testing results that account for mobility.

3.2.3 DMACDRTSDCTS

A key paper in this category is by Cartigny et al., [12]. It introduces a fairly complex method involving a localized algorithm where each node needs to know only geographic position of itself and its neighbors and stores it in a relative neighbor graph (RNG). Once each node receives a message for the first time from one of its neighbors then it rebroadcasts the information using its beamforming antennas in the direction of the neighbors that it has already identified. This information is in the form of distance and direction estimated by signal strength and delay (using smart antennas) or as determined via GPS data. Directional antennas are then used for the rebroadcasts, as mentioned, and data exchange. Additionally, adjustments are made to transmit with the minimum amount of power necessary to reach a particular receiver based off of the knowledge of distance in order to maximize energy efficiency. In fact their results show a 50 percent reduction in energy consumption versus other schemes.

The main problem with this scheme is the complexity and overhead. This includes being able to automatically determine the received direction of the strongest signal, as mentioned before; estimating distance based on received signal and delay; and then estimating the minimal amount of power necessary in order to reach a particular receiver. With regards to overhead, the authors cite that based off of their extensive measurements, the average degree of RNG is about 2.5 and on average only about 1.5 broadcasts are sent by each node to its neighbors. Also, this scenario is not very agile in a mobile environment and in being able to determine the location of users with respect to them moving around. As users move then additional location information needs to be transmitted so that everyone is aware where everyone else is at before initiating a directional session. This generates much more overhead and becomes less and less efficient as the mobility of a network increases. Also a straight DRTS/DCTS scheme is going to be very prone to hidden terminal and deafness issues. Listeners might be updated on where other clients are at but might not be aware of initiated or ongoing communications sessions. They conclude that their method is “well adapted for ad-hoc networks with significant node mobility” but their simulations are all static based.

The letter by Khalid et al., [45], points out that many methods rely on the RTS/CTS combinations, as discussed up to this point, while updating the DNAV. In situations of high mobility and narrow beam-width the coverage becomes too short (depending on location, velocity, beam-width, and distance between transmitter and receiver) and causes frequent DNAV updates. They propose an Adaptive Directional MAC (ADMAC) protocol that incorporates an intelligent neighbor discovery mechanism. It estimates where the client will be next based on the last sector (beam pattern with respect to a location), last known transmitter-receiver distance d, elapsed time since last update μ, relative velocity ν, and the beam-width α. They estimate d via the knowledge of transmitter power, pathloss model, shadow fading, and received power. They also point out that most of the research done in directional MAC with neighbor discovery either does not consider mobility or requires frequent updates or polling mechanisms. The authors state that each node is equipped with a switched beam antenna and employs the AoA estimation for sector finding, as mentioned before. Inter-beam switching time is negligible and nodes only use sector location rather than specific coordinates. When idle, it receives omni-directionally but all RTS, CTS, Data and ACKs are sent via beamforming. The basis for their estimation for neighbor discovery (tracking) is illustrated by the scenarios given in Figure 3.12. Take the last location and assuming the location is within the middle of a sector (beam pattern) as shown and then determine what scenario(s), see Figure 3.12, might it be categorized by estimating the distance from the client. Then, within a given time and for a given velocity, the client will move uniformly within the circles shown for the applicable scenario. Using this then the search span can be estimated in order to track the client again as shown in Figure 3.13. They cite that given their method, numerical results confirm better

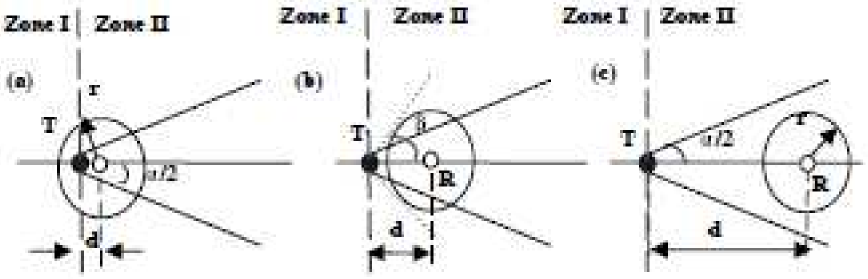


Figure 3.12: ADMAC Tracking Scenarios given distance from client

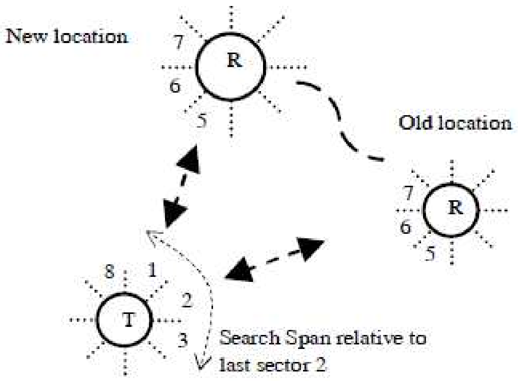


Figure 3.13: ADMAC Search Span

average throughput performance (anywhere from 40-400 percent) over Last Sector and Random Sector based DMAC methods, particularly for smaller beamwidth configurations and lower node velocities. Although it is not clear as to the characteristics of the mobility of the nodes in terms of trajectory, unlike many other schemes or particularly any other DRTS/DCTS schemes, this one directly accounts for mobility. It requires the additional capability of being able to do the AoA estimation and estimate the distance via understanding the path loss model, fading characteristics, etc. As with the other DRTS/DCTS scheme, no steps are taken to account for possibly prevalent, depending on the node density, hidden terminal and deafness issues.

3.2.4 DMACODRTSDCTS

The paper by Takai et al., [77], was written in 2002 and introduced some fairly unique new concepts at the time. Their work makes some modifications to the basic carrier sensing mechanism used by 802.11. It employs three new basic concepts: Angle of Arrival (AoA) Caching, beam locking and unlocking, and Directional Network Allocation Vector. The nodes use their AoA information when sending an RTS frame via beamforming. If no information is available or if it fails to elicit a response after four tries then the frame is sent omni-directionally. A client will try and send an RTS frame a total of seven times by default before abandoning the exchange. Upon receiving an RTS frame then the receiver adapts its beam direction to lock in the strongest one and uses it for the CTS response. Similarly the sender then locks its beam direction based on the received CTS frame. The pair utilizes their respective beams until an ACK frame is received for the respective conversation, thereby ending the conversation and unlocking the beam pair used. A Directional Network Allocation Vector (DNAV) is used versus the NAV discussed in Section 3.1.1 to provide the benefit of spatial re-use. Their simulations involved one hundred nodes randomly placed over a 1500 by 1500 meter flat piece of land. Among other variables, their mobility scenario is with respect to the random waypoint model moving at 10 m/s. They also test no-mobility (static) scenarios. Employing their method proved interoperable with the basic 802.11 omni-directional DCF MAC scheme and showed an improvement by a factor of 3-4 times for a 100 node ad hoc network over a purely omni-directional set-up. It steadily improved from a purely omni-directional configuration up to a fully Directional Virtual Carrier Sensing (DVCS) employed scheme. The mobility scenarios followed the same trend but the peak throughputs were at least 50 percent less than the static scenarios. They also found that employing physical carrier sensing alleviates the effects of accumulated interference due to many concurrent beamformed transmissions in some, namely mobile, situations.

This is a fairly robust technique. It maximizes use of the beamforming capability, if possible, and the increased range and spatial reuse that it provides while also highlighting the difficulty added by incorporating mobility. To be able to fully employ this scheme would require the additional capability needed to determine the best direction when receiving an RTS or CTS, based off of the reception of just one transmission. This technique could also be prone to hidden terminal and deafness issues based on its preferred configuration, RTS and CTS frames being sent via beamforming. However, the ideas presented in this paper with AoA caching and the use of DNAV remain common themes throughout many subsequent writings. Furthermore, its ability to integrate into existing methods on a scalable basis is very advantageous.

3.2.5 DMACCRTSDCTS

In [49] by Korakis et al., they present one of the first techniques that employs a fully beamformed configuration. In this method a sender sends an RTS frame circularly in all directions for an intended receiver. The sender only needs to be aware of the receiver but doesn't necessarily need to know the receiver's location. The neighbors, if they have traffic to send, can make a determination as to whether they need to defer transmitting altogether or in a particular direction, assuming the direction of the receiver can be inferred, in order to not destroy the imminent or ongoing transmission. In this way, there is a strong decrease in the hidden terminal problem. The CTS is returned via beamforming and neighbor DNAV tables are updated with respect to the RTS and/or CTS that they have received. If they wish to initiate a session then they avoid certain directions based off of the information contained in their DNAV table. The key disadvantage of this technique is the overhead associated with sending out the same RTS frame multiple times. It would also require changes to be made with respect to DIFS, SIFS, the associated timeouts and backoffs in order to account for the sequencing of the consecutive RTS frames, the returned CTS, and the subsequent data exchange. Although the method acknowledges that users could be mobile, the sender doesn't really care given. He he sends out an RTS in every direction and so wherever the receiver is at, at that point in time, then uses a capability of determining the angle of arrival to know which direction to point back. Mobility is not addressed in the simulation results. In fact, the results don't compare the method against any other techniques besides showing some alleviation of the hidden terminal issue due to the RTS frame being sent in all directions. Therefore, hidden terminal and deafness issues are minimized with respect to the sender (sending an RTS in all directions) but it can still be a factor from the receiver's perspective (his local area away from the transmitter) because the CTS packet is sent only once and in one particular direction.

3.2.6 DMACCRTSCCTS

The method presented by Na et al., [61], is essentially the same as that by Korakis et al. except now the CTS frame is sent circularly as well, versus just the RTS frame. This is meant to resolve inherent hidden terminal and deafness issues as discussed regarding the Korakis method but at the cost of the additional overhead and delays created by sending the multiple CTS frames. Results do not compare against any other scheme but simply highlight increased spatial reuse. Mobility is not addressed at all.

3.2.7 DMACMRTSDCTS

Choudhury and collaborators in [16] present an ORTS and a DCTS scheme similar to what was discussed above and also a unique Multi-hop RTS (MRTS) and DCTS scheme. This newer method focuses on using multi-hop RTSs to establish links between distant nodes, and then transmit CTS packets, data, and the ACK over a single hop by both clients using their beamforming capabilities. See Figure 3.14. Node A wants to establish a link with node F but can only do so via

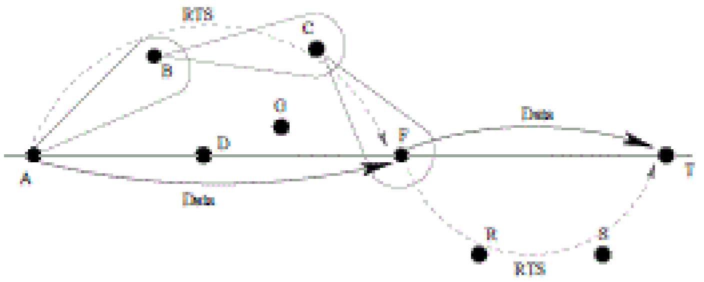


Figure 3.14: Example MRTS/DCTS Scenario

directional-directional (DD). It sends an RTS in that direction to reserve the channel and inform users in that direction about an impending session. It also sends a forwarding RTS to what is termed a directional-omni (DO) neighbor (Node B in this case) that is the first step in the DO route to node F. By default, nodes listen in omni-directional mode and transmit in beamforming mode. Each of the nodes in that route forwards the RTS packet until it reaches F and provides it with the necessary information to send back a beamformed CTS to A. While waiting, Node A will beamform in the direction of F and wait in anticipation of the CTS. If the CTS does not come back within a CTS-timeout designated duration, then A goes back and initiates re-transmission. The time for which A waits for the CTS is calculated as the time required for the forwarding RTS packet to reach F (over the specified route) plus the turn-around time for F to send the CTS back. Since intermediate nodes do not backoff while forwarding RTS packets, the CTS timeout duration can be calculated accurately. This method establishes a method to coordinate beamformed to beamformed communications but is very complicated as a result and requires a significant amount of overhead. The assumption is made that the upper layers come up with a DO neighbor route to a DD neighbor and then the necessary transmit/receive parameters necessary to establish that link. As mentioned earlier, it is also assumed that the radio is automatically able to determine the strongest angle of arrival of the signal in one iteration. Furthermore, their method does not account for node mobility. Their results suggest that MMAC outperforms DMAC since it utilizes the longest possible links between node pairs. It requires that only RTSs travel on the DO-links and data can be transmitted on the longer DD-links. This enables MMAC to use fewer hops in several instances. However, the higher failure probability in transmitting the multi-hop RTS packet, when using MMAC, increases the latency of packet delivery due to frequent time-outs and retransmissions. This partially offsets the advantage of utilizing DD-links when using MMAC. Therefore, the performance of MMAC (in terms of end-to-end delay) is only slightly better overall in comparison to DMAC. The authors offer no results with respect to how either method performs with mobile nodes however it appears that it would increase the higher failure probability even more. Given the assumptions named above this is not an ideal protocol for a location unaware, mobile environment and especially using beamforming antennas that are not necessarily smart (i.e. able to immediately determine AoA).

3.2.8 Tone-based Directional MAC Protocols

The scheme discussed in [38] is essentially a modification of the DRTS and DCTS scheme. They also make the assumption that the antenna is automatically able to determine the direction providing the strongest signal based off of one transmission. Their concern is hidden terminal issues caused by colliding or unheard RTS and/or CTS frames. Their rationale for the former is somewhat confusing since if there is a collision with an RTS frame then the sender will send another one due to the receiver not replying with a CTS frame. Furthermore, if there is a collision with the CTS frame then the sender won't send its data. An unheard RTS and/or CTS frame by other nodes is possible. In order to correct it they modify it via the Dual Busy Tone Multiple Access (DBTMA) in [22]. This scheme splits a channel into two sub-channels for data and control respectively. A transmit busy tone and a receive busy tone are assigned two separate single frequencies in the control channel. The authors make use of this system, similar to sending a DRTS and a DCTS, by the sender transmitting a transmit busy tone and the receiver transmitting a receive busy tone respectively during the duration of the transmission. Therefore, those that couldn't hear the DRTS/DCTS, despite being in the same line of transmission direction, should at least be able to hear the tones at any point during the transmission. Despite clearing up this particular issue, the same problems with hidden terminal and deafness still exist that were mentioned in the DRTS/DCTS schemes earlier. Their results suggest that their protocol performs better than the standard 802.11 protocol but appear to be based on a completely static scenario with no consideration for mobility.

The protocol discussed in [71] is a slotted MAC system based on the slotted ALOHA protocol. The protocol, named DOA-MAC (Direction of Arrival - MAC), is broken into three minislots and works as follows:

(1) The first minislot is called the DOA-minislot and it is here that a node identifies the angular direction of all transmitters that it can hear. All transmitters transmit a simple tone during the DOA-minislot towards their intended receivers. The signal received at some receiver is thus the complex sum of all of these tones. The receiver runs a DOA algorithm (varying ones exist) to determine the angular direction of each of the transmitters and the received power from each transmitter.

(2) Once a receiver determines the DOA of all transmitters it can hear, it forms its directed beam towards the one that has the maximum power and forms nulls in all the other identified directions.

(3) The second (and largest) minislot is the packet transmission slot and it is here that the packets are transmitted. After the receiver has formed its beam and nulls, it receives the packet from the transmitter. After receiving the packet, it looks at the header and rejects the packet if it was not the intended destination.

(4) The last minislot is the ACK slot where the receiver transmits an ACK using the already formed beam to the sender (if the packet was not rejected and correctly received).

(5) When a transmitter does not receive an ACK, it retransmits the packet at a later time (as in slotted ALOHA).

This technique does not use the RTS/CTS mechanism at all but makes use of these tones to help deal with the hidden terminal issue.

Once again this technique involves the added complexity of the antenna being able to automatically determine the direction providing the strongest signal from a particular user but also the added complexity of breaking down the sum of multiple transmitted signals in its direction and inferring the direction of the strongest signal. This is a complex and processing intensive determination. This method seems to make the assumption that since a signal is the strongest from a particular direction then a user in that direction must be trying to transmit to it. Various scenarios exist where this would not necessarily be true. Furthermore, the authors offer no consideration of how their method performs under mobility but do suggest that it does perform 2-4 times better than the method offered by Choudhury et al. in the last section, [16]. Interestingly enough none of the simulations compare their technique against any other techniques to illustrate this.

A follow on paper, [73], is a modification of their DOA-MAC technique except they now distinguish the sender and receiver tones from the simple tone for both. Unlike DOA-MAC this method is unslotted versus slotted. In this work the authors address some of the issues discussed with their DOA-MAC scheme. The intuition behind the receiver beamforming in the direction of the maximum signal is because it is, they argue, the intended recipient for the packet with high probability because of the directivity of the antenna. They acknowledge that in some cases the receiver incorrectly beamforms towards another because its signal is stronger than the other. An optimization they implement is a single entry cache scheme which works as follows: If a node beamforms incorrectly in a given timeslot then it remembers that direction in a single-entry cache. In the next slot if the maximum signal strength is again in the direction recorded in the single entry cache then the node ignores that direction and beamforms towards the second strongest signal. If the node receives a packet correctly, it does not change the cache. If it receives a packet incorrectly, it updates the cache with this new direction. If there is no packet in a slot from the direction recorded in the cache, the cache is reset. The authors acknowledge that they make no attempt to combat the hidden terminal problems and do not maintain the NAV which can lead to deafness issues as well. They mention that the steering of nulls in the unused directions and the use of Forward Error Correction (FEC) takes care of some of the hidden terminal issues and they are trading simplicity for the resulting loss of performance. Nevertheless, the deafness issue ends up being aggravated. The authors mention their method performs 3-9 times better than standard 802.11b in their OPNET and MATLAB simulations. This was helped by an increase of antenna elements to 16 which increases the spatial re-use. Also, this method is effectively an omni-directional scheme in its standard configuration and therefore limited in range and the authors offer no consideration with respect to mobility.

In [72], the authors summarize their DOA-MAC and Tone Based MAC Protocol schemes and re-name them Smart-Aloha and Smart-802.11b respectively. They provide further illustration that their schemes perform much better than 802.11b for aggregate throughput versus sending rate in a couple of scenarios but still offer no results with respect to its performance with mobile nodes.

Choudhury et al. in [15] evaluates the impact of deafness on wireless medium access control. They propose a tone-based directional MAC protocol (ToneDMAC) that essentially operates as regular DMAC where all RTS, CTS, Data and ACK frames are sent via beamforming. However, to alleviate the common deafness issue, once two nodes, A and B, have exchanged their Data and ACK frames then they switch back to omni-directional and exchange a series of tones. The common channel is broken up into two sub-channels where data is sent on one and the tones are sent on the other. These tones serve as indicators that nodes A and B were recently engaged in a communication session. A neighboring node, C, unable to communicate with A immediately prior, can utilize the tone from A as an indication of recent deafness. Realizing deafness, C can suitably modify its retransmission strategy. In a given area, these tones are unique to individual clients and the higher layers for each client (with respect to communication exchange layers - e.g. OSI) are used to correlate each tone to a particular user. Also unlike DMAC, ToneDMAC requires a node to switch back to the omni-directional mode while performing the backoff countdown, if it senses contention. While backing off in the omni-directional mode, a node senses the channel as busy only if a signal arrives from the direction in which the node intends to transmit. However, if an RTS or CTS arrives from other directions, a node will be capable of receiving them. This mitigates the "deadlock" problem arising from directional backoff.

Huang's method assumes multiple transceivers capable of transmitting data packets as well as busy tones via beamforming. Moreover, the protocol suffers from the problem of deafness as mentioned earlier. ToneDMAC splits up the single control channel for signaling while maintaining the need for only a single transceiver. Tones are assigned on that sub-channel and need not be transmitted in parallel with data packets. In this respect, tones are different from "busy tones" that are typically transmitted when a tone is "busy" with ongoing transmission/reception.

The authors clearly identify unfairness as a direct result of deafness. When multiple nodes attempt to communicate with node A, the node that wins channel contention retains the privilege to access the channel for a long time. Although the receiver remains busy almost all the time, the transmitter nodes experience short-term unfairness. An example of how deafness leads to this unfairness is shown in Figure 3.8. B submits an RTS to A and then A replies with a CTS to initiate a beamformed communication session between them. Immediately after the CTS is sent back, node C sends a beamformed RTS to A as well. At that point A is beamformed towards B and is "deaf" to $C's$ request. C, not receiving a CTS in return, assumes contention and backs off. It continues to try with its back-off window growing exponentially after each failure to receive a CTS in response. At some point the session between A and B ends, see the ACK. B still has more data to send. He has an "unfair" advantage of starting at a base contention window versus node C having an exponentially larger one and therefore easily and unfairly grabs the channel again. Node C eventually reaches the default of 7 RTS requests before assuming Node A no longer exists and drops the packet.

The authors show that ToneDMAC overcomes this and channel access is performed with reasonable fairness and lower backoff values while outperforming DMAC as well in terms of aggregate throughput (particularly over multi-hop networks) and lower packet drop rate (namely when channel contention is high). End-to-end delay is also lower with ToneDMAC. Since ToneDMAC uses omni-directional backoff, it alleviates the "deadlock" problem discussed earlier. When TCP traffic is used, the overall performance benefits are even greater. The authors attribute this improvement of ToneDMAC over DMAC to the two main modifications of (1) notification from tones and (2) omni-directional backoff.

The technique does improve on DMAC with regards to overcoming deafness issues. Nevertheless, with all these tone based schemes, there is overhead associated with using a separate (or sub) channel for sending these tones. Furthermore, clients need to wait until after a conversation is over to listen for these tones, if applicable, and then determine if deafness took place previously. If they determine it did take place then they must analyze the tones to determine the applicable user that was deaf in the previous conversation. This technique also appears to assume static nodes and doesn't show how this system performs in a mobile environment. The authors do offer the following however, "While proposing the ToneDMAC protocol, we modified IEEE 802.11 to suit a beamforming antenna system at the physical layer. However, it is not clear that modifying 802.11 is optimal in terms of performance. MAC protocols, designed specifically for beamforming antennas, may prove to be more efficient. For example, it is unclear whether CSMA/CA protocols are appropriate when using beamforming antennas. Time division multiple access (TDMA) schemes might prove to be more effective. Even if CSMA/CA principles are used, it is unclear whether RTS/CTS exchanges (as in 802.11) are necessary – with narrow beamwidths, bandwidth wastage due to RTS/CTS's might exceed the gains from channel reservation. Directional carrier-sense is another mechanism that might not be meaningful when using directional antennas."

3.2.9 DMAC Directional TDMA Techniques

Jakllari et al. propose PMAC in [40] which incorporates a mechanism for neighbor discovery and a scheduling based medium sharing that allows for exclusive beamforming transmissions and receptions. They validly argue that previous works don't address full exploitation of beamformed transmissions and locating and tracking neighbors under mobility. Their method tracks neighbors using DOA (Direction of Arrival) or AOA. It has a frame structure consisting of a search state, a polling state and a data transfer state. During the search state they identify new users and establish a rendezvous time during the polling state. During the polling state, a similar rendezvous is established for the exchange of data, whether necessary or not, during the data transfer state. Their studies determined that eight (8) was an acceptable neighbor count. The frame size is adjustable based on mobility parameters. Their method is one of the most robust techniques that considers all factors of tracking[1] and data exchange and is one of the very few papers closest to this work. The technique does introduce quite a bit of overhead with the addition of a polling phase to searching and data exchange phases. This state also serves to keep each node "continuously" (periodically) aware of its neighbors' locations. They use the random waypoint as their mobility model with speeds of 2 m/s in a pedestrian environment and 10 m/s in a vehicular environment. They test their neighbor discovery in a star topology with 8 uniformly placed nodes around a center node. They determine that if the Search Segment Length (SSL) is 20 then it takes 5 frames on average to find all of its neighbors with a 90˙ antenna pattern and 10.5 frames for a 60˙ antenna pattern. This amount of time is quite significant but they do mention that this is only needed during the initialization phase in order to track down all of the clients. Note that their polling phase serves equivalently as a tracking phase, with regards to this work, since during this phase they update the beam to be used for a particular client. As with previous schemes, this assumes that the antenna has the advanced capability to know the strongest AOA given receipt of one transmission. Regarding throughput, they compare PMAC against 802.11 and the CRTS scheme by Korakis discussed earlier. They mention that they chose to compare it against the CRTS scheme mentioned because it is the only method that integrates node discovery and that it outperforms previous schemes. For these experiments they place 50 nodes randomly in a 500m by 500m flat terrain with EACH (ad hoc based) node deploying an electronically steerable antenna with a 45˙ beam. They run the simulation for a static scenario and a mobile scenario in which the clients are moving at 10 m/s. The results for the static scenario are given in the paper and the results for the mobile scenario are as shown in Figure 3.15 Their results show that PMAC clearly outperforms both the CRTS scheme and especially 802.11. CRTS performs well but suffers some asymmetry in gain at heavy loads due to the CTS frame only being sent via beamforming. The results also illustrate the small drop in performance for PMAC under mobility due to their discovery and tracking mechanism. The authors also analyze the performance of their protocol with respect to the channel utilization ratio (CUR). This allows them to find the operational "sweet spot" of PMAC

[1] although they assume automatic AoA as discussed

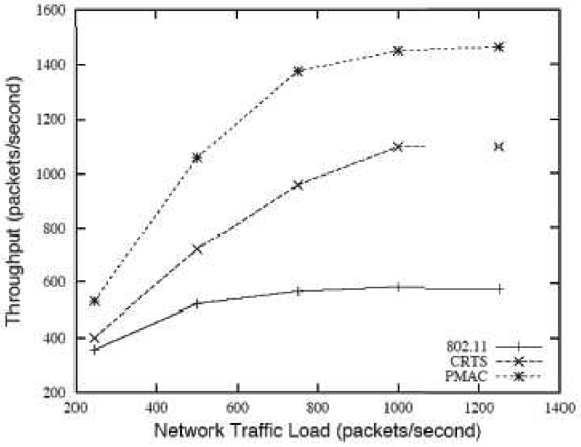


Figure 3.15: PMAC - Throughput vs. Traffic Load for Mobile (10m/s) Scenario

given that the frame size is dictated by mobility. The results are as expected. In considering per node channel utilization ratio versus network traffic load (packets/second) for a fixed frame size then the more static the scenario and the smaller the beamwidth, the better the performance. In the case of the static scenario, the only limitation is with regards to the inherent overhead of the protocol. The higher speed situation results in losses due to the overhead and also having to find lost clients over again due to the higher mobility. The smaller beamwidth obviously results in less interference and a higher data rate per exchange. Further results illustrate that at 2 m/s the CUR is better with a larger frame size compared to a smaller frame size at 10 m/s. This is expected since the higher mobility requires keeping track of the clients more often. A tradeoff exists between having enough overhead associated with the tracking of clients but not too much to take away from the data exchange as expected. In terms of fairness the authors, who have seen the CRTS scheme as the next best, argue that it being a random access scheme with exponential back-offs then it inherently has low fairness versus their scheme which is almost an entirely pure TDMA scheme (which would have a fairness index of 1). Their results illustrate this point with PMAC having a fairness index of almost 1 and CRTS operating at just under .6. The benefit of it being a TDMA scheme also minimizes hidden terminal and deafness issues.

Zhang's papers, [88][89], argue the lack of specific reservation/scheduling algorithms and that all previous works assume some type of omni-directional reception at some stage of the algorithm. This limits the effective range of the system to the omni-directional case as discussed previously. Their TDMA based system is very similar to that presented by Jakllari. They offer that their system has four advantages that include: 1) assumes fully beamformed transmission/reception; 2) is distributed, relies on local information only; 3) allocates slots to different links dynamically based on demand; 4) power control is easily carried out during neighbor discovery, reservations, and during the data transmission period. As mentioned, this technique is very similar to that offered by Jakllari but considers power control along with their 3-way handshake coordination method in the neighbor discovery phase. It also introduces some useful concepts related to determining the parameters needed for the searching, polling, and data exchange schedule, both individually and collectively. For their scenario they show that the total number of mini-slots required for a complete scan to find all N users is $\lceil log_2(N) \cdot L \rceil$, where N is the total number of users in the neighborhood and L is the number of beam directions needed to cover the space. F is the total number of mini-slots in an entire frame, n is the number of searching slots and r is the number of reservation slots as determined by the user. The amount of time it takes to conduct a complete scan is $\lceil log_2(N) \rceil \cdot L \cdot \frac{F}{n}$. The length of one frame can be one second. In their scenario, for a beam-width of $5^{\cdot}$, it takes about 300μs to finish the 3-way handshake during the searching phase. This includes the beam switching time, guard time and message lengths. So, the mini-slot length should be about 300μs. If one slot consists of 10 mini-slots then one slot length will be $3ms$. One frame includes 333 slots. Say $n = 10 \cdot 10 = 100$ and $r = 20 \cdot 10 = 200$. Then $100 \cdot .3ms = 30ms$ per frame(second) is used for neighbor discovery and another $60ms$ per frame is for reservation. That gives a 9 percent overhead. In their scenario it takes 1050 beams to finish one complete hemisphere (horizontal and vertical). If $N = 128$ then it will take $1050 \cdot \frac{7}{100} = 70$ seconds to complete an entire scan. A new node will be discovered on average in about half a minute and 70 seconds maximum. Since $N = 128$, any two-node pair can make a reservation every 2 frames (2 seconds) at maximum and within one second on average. If reservations can only be made during neighbor discovery then the waiting time for making new reservations would be too long and hence why the second sub-frame (reservation) is introduced. Simulation considers 14 nodes initially uniformly distribured within a 200m x 200m area and where all nodes can reach each other in one hop (with omni-directional). The mobility factor consists of each node using the random waypoint technique with speeds up to 100 miles/hour. The overhead is 9 percent they show (via simulation) that the technique can achieve up to a factor of 6 improvement over 802.11 that can be mostly attributed to the increased spatial reuse.

Bao and Garcia-Luna-Aceves introduce a distributed receiver-oriented multiple access (ROMA) channel access scheduling protocol for ad hoc networks with beamforming antennas in [4]. Each of these nodes' antennas can form multiple beams and commence several simultaneous communication sessions. Unlike random access schemes that use on-demand handshakes or signal scanning to resolve communication targets, ROMA determines a number of links for activation in every time slot using only two-hop topology information. The writers claim that ROMA shows superior performance over the best-known polynomial time approximation algorithms (UxDMA) for scheduling in ad hoc networks in terms of the network throughput and packet delay. Their proposed neighbor protocol uses an allocated random access section to send signals to track neighbor positions for ROMA. The neighbor protocol exchanges neighbor information to synchronize topology information within two hops of each node. They claim that the ability of ROMA to achieve collision freedom for channel access using only two-hop topology information is more efficient than in UxDMA with respect to the control overhead incurred by the two approaches.

This method is fairly robust but the use of multiple beams and conducting several communication sessions at once is outside the capability of the majority of equipment used and outside the scope of this problem. Furthermore, the neighbor protocol relies on the exchange of information via omni-directional means which hampers the capability of the system overall. They acknowledge that their simulation results only consider static networks in order to know beforehand the two-hop neighbor information or the entire topology.

In the paper by Grimaud et al., [32], the authors desire to develop an efficient scheduling algorithm that considers throughput and latency as optimization criteria. They argue that the time it takes to electronically switch beams is not trivial and a key part of maximizing throughput, while minimizing delay, is to consider how much time is spent on a beam before switching to a different one. They compared First-in First-out (FIFO) to Round Robin (RR - alternate beams based on directions of users) and determined that FIFO performs better on average. They introduce the Temporized Beam Window (TBW) parameter which represents the time window during which they stay on a beam selected by the FIFO algorithm. The FIFO TBW scheduler alternately selects a beam and sends packets on this beam until the beam buffer is empty or when the time represented by TBW is elapsed. Simulations show that the FIFO TBW algorithm performs better than both FIFO and RR. The next step is to optimize the length of the TBW in order to satisfy the criteria of maximizing throughput and minimizing latency. The real time scheduling problem is NP-complete.

It is therefore necessary to use simulation to try and arrive at optimal results. So, optimizing the occurrences of the time-delay switch is achieved by staying for an optimized duration, depending on the saturation of the bandwidth. Good results are obtained using their method which reduces the wasted bandwidth of the antenna by an average factor of 5 for a TBW=0.01s.

The results of this paper are fairly intuitive. The TBW will be a dynamic parameter that will change based on the number, density, type, location, etc. of users along with other environmental factors. Most papers assume the time to switch beams is negligible, however, in general it should be considered with respect to the characteristics of the network's users and environment before being ignored as their results show here. Unfortunately the authors don't appear to consider the effects of mobility with regards to their work.

3.2.10 Other Practical Implementation Techniques

The paper by Navda et al., [64], provides some interesting techniques related to probing in an infrastructure mode where the hubs (access points) are static and the clients (spokes) are dynamic. Additionally, their studies are done with the same beamforming wireless hub as used in this work. Their technique, Mobisteer, operates in two modes. In cached mode, radio survey data is collected during idle periods when a vehicle is not communicating with fixed infrastructure. A geocoded RF signature database is created and maintained for frequently driven routes. This database is used to drive an algorithm that generates a trace of how beams should be steered and handoffs initiated as the car moves along the known route. In online, or dynamic mode, the client scans the environment in all beams and channels using active probing and chooses the best beam and hub combination depending on SNR values of probe response frames received (ranked in a table). The client scans with active probes and takes into consideration that 90 percent of access points (hub for WiFi scenario) utilize channels 1, 6 and 11. Depending on the hub (access point) identified using either the cached or active scanning mode, the client then connects (associates) to it until d consecutive packet drops. It then goes to the next hub in the table until none are left and then it goes to a probing phase automatically. The size of the table and threshold of number of drops are parameters that can be varied. The amount of probing time necessary is the penalty a user experiences for inefficiencies in the system.

Authors identified that steering clearly outperforms an omni-directional configuration by a factor of 2-4. It also showed an improvement in connectivity duration by greater than a factor of 2 and an average increase in SNR by about 15 dB. They further identified about a 39 percent decrease in packets received during online mode versus cached mode and cached mode did better than online mode in general by 50 percent. This is expected due in large part to the time spent probing while online and the real-time resources saved by processing a database of the steering and handoff plan prior to execution from the cached results versus determining it in real time. In this work, the cached method is not an option.

The authors of [54] propose a MAC protocol exploiting Space Division Multiple Access (SDMA). This method can improve the throughput of bottleneck nodes by synchronizing the packet receptions from other nodes. The scheme assumes that each node has only one adaptive array antenna system. A receiver-initiated approach is proposed to achieve the time synchronization for receptions. The receiver polls all its neighbors by sending an omni-directional Ready-to- Receive (RTR) packet periodically. The intended senders will reply with a beamformed RTS packet, with beamformed CTS/DATA/ACK sequences following. Since the RTR packet is larger than the typical control packet size, it may result in extra overhead. From the simulation results, it is observed that when the load is low, the RTR-based protocol performs even worse than IEEE 802.11b with omni-directional antennas. When the bottleneck of transmission is very noticeable, this protocol performs much better than IEEE 802.11. The authors don't appear to consider the effects of mobility in their work.

The work of [67] discusses a robust system involving clients communicating via beamforming with multi-lobe beam shaping and rate adaptation to a diverse set of base stations. Their experiments have all been analytically and simulation based but demonstrate the feasibility and show that throughput is maximized as a specific combination of rate and beam-width. They show that by individually controlling either by itself can lead to local maximas, thus making the case for joint adaptation of these parameters. This seems to support similar findings by [1] discussed earlier and by [84] to be mentioned later.

Lee and Chung's work, in [55], is an interesting proposal of a new channel access mechanism, and RTS aggregation. RTS aggregation allows the processing of multiple requests in a single duty cycle. It helps alleviate high latency problems caused by duty cycle ops. They argue that a key problem of DMAC is complexity introduced by time synchronization and staggered scheduling. In this method, the client listens and then sends an RTS as appropriate. If other users have data intended for the same receiver then they add their information and forward an aggregated RTS. Through the aggregated RTS packet, the receiver can decide the transmission order and time for each sender node. The receiver then makes the data transmission schedule for the multiple RTS nodes and sends the CTS packet with embedded information regarding the data transmission schedule. The nodes that receive the CTS packet from the receiver node start their data transmission at each scheduled time. Their results show that it provides similar energy efficiency, throughput improvement and latency reduction compared to Short Preamble MAC. As the number of nodes and requests increase then RA-MAC does better than Short Preamble MAC regarding latency. It also does better regarding packet interval to throughput and average transmission rate per unit time. The authors don't consider mobility in their work. By including mobility then the hub would either have to know the updated location of the relayed (with respect to the RTS) client or the location of the client would have to be included into the relayed RTS. This could significantly increase the amount of overhead and if the updated location information is relayed then it could become stale before it reaches the hub. Many factors to consider regarding this technique and mobility.

3.2.11 Operations Research

From [81], "During World War II, British military leaders asked scientists and engineers to analyze several military problems: the deployment of radar and the management of convoy, bombing, antisubmarine, and mining operations. The application of mathematics and the scientific method to military operations was called operations research. Today, the term *operations research* means a scientific approach to decision making, which seeks to determine how best to design and operate a system, usually under conditions requiring the allocation of scarce resources." At this point in our look at related work we start to narrow in on work that takes the same type of approach we will in solving our problem. Operations Research consists of many different areas, of which we'll highlight with the following subsections and related work that applies with respect to our problem.

3.2.12 Game Theory

The paper [13], by Chen et al., discusses how contention resolution is usually achieved via persistence and backoff. They mention how the Distributed Coordination Function (DCF), see Section 3.1.1, uses exponential backoff and a binary contention signal (packet collision or packet sent successfully). They argue how in high load situations using DCF, or the majority of current schemes, the result is excessive collisions and low throughput because setting to the base contention window initially or after a successful transmission is too drastic. They also offer a similar argument to Choudhury that short term unfairness exists due to oscillations in the contention window but it is unavoidable due to this binary (collision or success) mechanism. Their intent is to stabilize the network into a steady state with target fairness and high efficiency. The authors discuss a method where access and contention is treated as a random access game. A node's strategy is its channel access probability and its payoff is the gain from access and the cost of collisions. A node estimates its conditional collision probability by observing consecutive idle slots between transmissions (assuming a single cell - where all users can hear all other users) and adjusts its channel access probability accordingly. The method adapts to this continuous feedback signal (vice binary) and tries to keep a fixed channel access probability (persistence or equivalent contention window size) specified by the Nash equilibrium of the random access game. It adjusts via a gradient play where adjustment is small when the current state is close to equilibrium and it is large otherwise, unlike DCF. So, it achieves better contention control and short-term fairness as well as decoupling contention control from handling packet losses. Several parameters can be manipulated.

- Some include duration of idle slot, maximum channel access probability (which affects the number of equilibria).

- System designer can specify a set of weights according to the levels of Quality of Service (QoS) he/she wants to provide and each node will choose a weight that corresponds to the specific level of service he/she desires.

- Dynamic properties affect the stability and responsiveness. Step-size affects convergence. In practice, it is typical to choose a constant step-size for all nodes. Additional properties include the number of transmissions used for each node before updating access probability and the size of the contention windows which affects convergence and accuracy.

Finally, in their case study they achieved optimal throughput with low collisions and good short-term fairness and it can provide flexible service differentiation among wireless nodes. Now obviously this work doesn't specifically address the use of beamforming antennas and mobility but its approach could be incorporated as part of the data scheduling.

The work by Park et al. in, [66], and Nasr et al. in, [63], continues on with the game theory approach as put forth by Chen et al. In the standard Wireless LAN MAC layer standard the thought was that all transmitters would follow the standard rules. However, a selfish transmitter can violate the rules in order to increase throughput. A game theoretic analysis for this problem has been introduced, as mentioned. With respect to an infrastructure (hub and spoke) based scenario, an equilibrium can be reached if the hub forces a pricing technique or a bargain is reached by the transmitters. The authors argue that control-theoretic approaches, as discussed by Chen et al., assume that users are obedient. Their concern is about selfish behavior and to start from a utility function. A user's utility increases from being able to successfully send packets (throughput) and is decreased from the cost of having packets collide. The idea then is to use their utility function to affect them via an intervention scheme which can lead to a distributed algorithm to achieve a desired operating point. By formulating the medium access problem as a non-cooperative game, they show the following:

- Nash Equilibria are inefficient and/or unfair. Their idea is to transform it into a Stackelberg game in which any feasible outcome with independent transmission probabilities can be achieved as a Stackelberg equilibrium.
- A particular form called total relative deviation (TRD) based intervention is constructed and used to achieve the latter.
- The additional amount of information flows are minimal and can be further reduced without large losses in efficiency.

They transform the game into the Stackelberg contention game by introducting an intervening manager (the hub or access point) which can implement any transmission probability profile as a Stackelberg equilibrium using a class of intervention functions. The authors argue that Nash Equilibrium payoff profiles are inefficient or unfair. The probability that a particular user transmits, $p_i = 1$, and therefore extending to all users transmitting is a weakly dominant strategy and is the most likely Nash equilibrium. This likely leads to a network collapse. Their argument is to transform the contention game. The strategy of the manager is an intervention function $g(p)$ where g is the level of his intervention (the result of his intervention affects all users) when the users choose p. The transformed game is a Stackelberg contention game because the manager chooses the strategy before the users make their decisions - leader and followers. So (g, p) is a Stackelberg equilibrium if p is a Nash equilibrium of g and $g(p) = 0$, or essentially the manager doesn't have to intervene at all. So a symbiotic relationship exists here. The TRD works as follows.

- The manager sets a target probability (probability of transmitting), p_t. The users choose their p_i so that the TRD of p_i from p_t is ≤ 0. If it is larger than 0 then the manager will respond to a 1 unit (unit is a manager defined parameter) increase in p_i by increasing his p by $1/p_i$ until the TRD reaches 1. The manager determines the degree of punishment based on a target transmission probability profile. If he wants a user to transmit with a low probability then the punishment is large. The authors prove that this system will converge to a Stackelberg equilibrium. In a general non-cooperative game the users need to know, or predict correctly the strategy profile of others in order to find the best response strategy. There are a few requirements for the Stackelberg Contention Game with TRD-based intervention and Stackelberg Equilibrium.

  ∗ Requirement M (as they define it) is that once users choose their transmission probabilities, then the manager observes the users' strategy profiles.

  ∗ Requirement U then is that user i knows g and p_{-i} (collective probability of a transmission besides him) when it chooses its transmission probability.

  ∗ A user observing that $p_0 = 0$, probability that the manager intervenes is zero, confirms its belief that the other users are playing the recommended strategies and so it has no reason to deviate.

- The users acquire knowledge about g in one of three ways:

  ∗ Known Protocol.

  ∗ Announcement.

  ∗ Learning.

- If users are obedient, the manager can use centralized control by communicating a target p_i to user i. Additional communication and estimation overhead required for the Stackelberg equilibrium can be considered as a cost incurred to deal with the selfish behavior of users, or to provide incentives for users to follow the target p_i.

Their results are broken up into two categories as follows:

- Homogeneous Users.

  ∗ p_i is 1/n for all n users.

  ∗ The system utilization converges to $1/e$ (36.8 percent) as n goes to infinity. This coincides with the maximal throughput of a slotted Aloha system with Poisson arrivals and an infinite number of users. Here users maintain selfish behavior and no feedback information on channel state.

- Heterogeneous Users.

  ∗ 3 scenarios: Assign higher p to higher valued user, assign the same p to all users despite value, assign the same payoff to all users which implies that a lower value gets a higher p and a higher value gets a lower p.

  ∗ The results show a trade-off between efficiency (sum of the payoffs) and equity exists when users are heterogeneous.

  ∗ A higher aggregate payoff is achieved when users with high valuations are given priority and it limits access by users with low valuations.

  ∗ Overall it increases variations in individual payoffs.

So the basis for the method presented in the first paper (Park and van der Schaar) is that the network manager acts as a game leader, enabling selfish users to reach a Stackelberg equilibrium. The drawbacks are that the leader punishes the whole network if any transmitter floods the network, the Stackelberg equilibrium is not unique, and the manager cannot force users to reach equilibrium if they have large deviations. The authors mention that previous techniques assume a single antenna and that some papers considered multiple antennas. None of the previous papers considered contention based protocols with multiple antennas where you can only receive one packet at a time. This is important since the new generation of WLANs is based on the 802.11n multiple antenna standard. The new standard's MAC protocol will be a contention based protocol and its multiple antenna capability allows beamformed signal transmission and reception through beam forming. In the latter paper, [63], they use game theory again to show the existence and uniqueness of a Stackelberg equilibrium when different selfish single antennas access the same multi-antenna hub. This is based off of the same contention based protocol as discussed in the earlier paper. Each transmitter tries to maximize its own utility. The hub tries to force accessing with certain probabilities but can now use antennas to block deviant transmitters. This method achieves Stackelberg equilibrium and is simple to implement. In the Passive AP scenario, as they call it, the receiver does not use a multiple antenna capability. The system reaches a Nash equilibrium and, as discussed earlier, users face a “tragedy of commons”. In the Active AP scenario they change the radiation pattern by changing the antenna weights with time to enable reception from any transmitter or block him. They determine that a Stackelberg equilibrium can be reached when using the multi-antenna capability. When user i sends packets with target p_i then the AP will not block any of the packets. It becomes in the self-interest of the user to transmit with that target probability. At the Stackelberg equilibrium, users should not have any incentive for individual deviation from the equilibrium point.

In Figure 3.16 the authors show the feasible utilities User 1 and User 2 for both Passive AP and Active AP scenarios. For the Nash Equilibrium game, essentially the utility can only be maximized for one user. The Pareto boundary is at the Stackelberg equilibria or where the sum of the user probabilities equals the sum of the target probabilities. This is similar to the efficiency line as discussed earlier in 3.1.3. Inside that line, the sum of the user probabilities is below the target and so one or more users can increase their transmission probability. Outside the line, one or more users is exceeding the target probability and will be blocked. A representative value is shown in

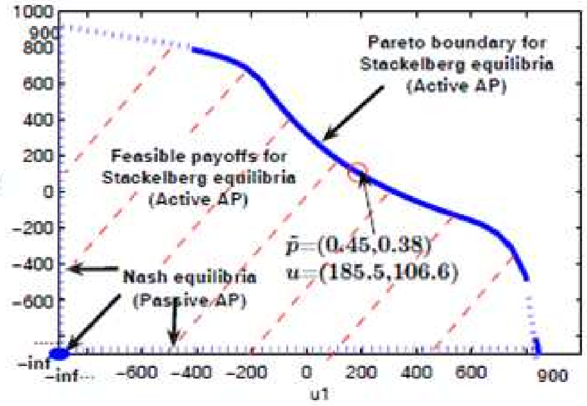


Figure 3.16: Feasible Utilities for User 1 and User 2 for Passive and Active Scenarios

the diagram illustrating example probabilities with their respective utility values.

Chen's work above applied more to improving the random access characteristics of 802.11 given their inherent limitations. Park et al. and Nasr et al. provide a valid argument that Chen's recommendations will lead to network collapse but by modifying them by using a central controller (hub) subject to rules and the tool of a beamforming antenna then you can force a Stackelberg equilibrium to occur. These results and insight can be extended to our implementation via the general use of convex optimization. Also, consider a game managed by the hub and adopting a TDMA scheme. You define a proportion of time for a client to transmit over a given interval, it becomes a probability that a user will transmit at some point along that interval. Then along with the ability to steer beams you are similarly forcing this Stackelberg equilibrium to occur as they have done.

3.2.13 Convex Optimization

A broad encompassing area of Operations Research is convex optimization, see [9]. Boyd et al. cites that much has been written and much is known regarding linear programming and least squares problems but, "...convex optimization problems (beyond least-squares and linear programs) are more prevalent in practice than previously thought....There are great advantages to recognizing or formulating a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved, very reliably and efficiently,..."

Examples include works that formulate these type of network flow type problems in the form of convex problems that can be optimized in real time. Kelly's substantial work on the topic has extended as far back as 1998. For instance, in [44], he confronts the issue of how bandwidth should be shared between competing streams of elastic traffic. Elastic referring to where a user can adjust his data transfer rate based on available bandwidth. An optimization framework leads to breaking down the problem into separate problems between the respective users and the network. They show that two classes of rate control algorithms are naturally associated with the objective functions appearing in the respective primal and dual formulation of the network's problems. As a result, the algorithms provide natural implementations of proportionally fair pricing.

In [65], they take the convex optimization approach in order to optimize aspects of a network or, “network utility,” in the form of offering a tutorial paper. They cite that an understanding of the decomposability structures in network utility maximization is key to both resource allocation and functionality allocation. It helps to obtain the most appropriate distributed algorithm for a given network resource allocation problem and to be able to measure it against alternatives of a modularized network design. Decomposition theory provides the mathematical language to build a foundation for the design of modularized and distributed control of networks. They review the basics of convexity, Lagrange duality, distributed subgradient method, Jacobi and GaussSeidel iterations, and implication of different time scales of variable updates. They introduce primal, dual, indirect, partial, and hierarchical decompositions, focusing on network utility maximization problem formulations and the meanings of primal and dual decompositions in terms of network architectures. They present examples on: systematic search for alternative decompositions; decoupling techniques for coupled objective functions; and decoupling techniques for coupled constraint sets that are not readily decomposable.

In [57], the authors offer a tutorial as well on developments, circa 2006, in optimization based approaches for resource allocation problems in wireless systems. They discuss results in the area of opportunistic (channel-aware) scheduling for cellular (single-hop) networks, similar to this work in practice, where easily implementable policies are shown to optimize system performance. As part of their analysis they show that a “clean-slate” optimization-based approach to the multihop resource allocation problem naturally results in a “loosely coupled” cross-layer solution. So the algorithms obtained map to different layers of the protocol stack, and are coupled through a limited amount of information being passed back and forth. They determine that the optimal scheduling component at the MAC layer is very complicated and therefore needs a simpler distributed solution. They show how to use simpler, although possibly imperfect, scheduling in the cross-layer framework and describe recently developed distributed algorithms along these lines.

Anderson in his thesis, [1], uses convex optimization with Dantzig Wolfe Decomposition techniques with respect to optimized TDMA scheduling and beam selection and control in a wireless ad hoc scenario. Unlike [11] which mentions that a "column generating approach allows us to separate the routing[2] and scheduling problem...," he argues that these aspects should be a coupled process. Scheduling decisions should not be made without considering the radio environment from a noise and interference perspective that will affect the beam selection being used and vice versa. He goes on to develop an algorithm based on the, "joint beam steering and scheduling problem for spatial reuse TDMA," that achieves up to a factor of 6 speed increase over TDMA in experiments performed. Such a system hasn't been developed in reality to this point but it illustrates the necessity of optimally integrating the layer 1 and layer 2 aspects together versus separately.

Yee, in [84], provides insights into analysis of optimization designs (schemes) for beam forming 802.11 WLANs in an ad hoc scenario. He points out that optimizing the total throughput is an important challenge. He continues that what has usually been done in the past is to address factors such as channel assignment or transmit power control and test it across various environments, topologies, and traffic rates. He argues that this method provides only a limited context for finding the best combination and ordering of these independently derived and tested methods. The dissertation presents a characterization of the combinatorial and ordering effects of algorithms from five optimization domains: channel assignment, association control, transmit power control, bit-rate adaptation and beam form selection. His method is executed via simulation and administers these optimization configurations. The aggregate throughput results are processed by a decision tree to classify optimization configurations into top and bottom tiers using pairwise ordering and algorithm selection attributes. These results show that the relative ranking of an optimization domain is dependent upon the combination and ordering upon which it is applied. Also, the ordering of a set of algorithms is as significant to final performance as the combination selected. It shows the importance of identifying beforehand what the key criteria (or optimization parameters) are to be analyzed in order to determine success. It also shows the necessity of showing due diligence in the conduct of experiments and variation of combinations of parameters used in order to arrive at

[2] routing referring to beam selection in this regard

a confident result.

Linear Programming is one of the key subsets of convex optimization where the objective function along with its constraints are not only convex but also linear. Basic linear programming along with numerous examples of applications and extension to network flow problems are explained in the seminal work by Chvátal in [17]. Boyd in [9] also goes into some detail. Its basic format is as follows.

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & a_k^T \geq b_i, \ i \in \{1, \cdots, m\}. \end{array}$$

Where vectors c, $a_1, \cdots, a_m \in \mathbf{R}^n$ and the scalars $b_1, \cdots, b_m \in \mathbf{R}$ are parameters that specify the objective and constraint functions. These references describe many of the methods that exist in solving these type of problems from Dantzig's simplex method to branch and bound, cutting-planes and the now more popular interior point method. The complexity can vary but in practice is $O(n^2 m)$ where n represents the number of variables in x and m the number of constraints. As Boyd [9] explains though, these algorithms are quite reliable and, "can easily solve problems with hundreds of variables and thousands of constraints on a small desktop computer, in a matter of seconds. If the problem is sparse, or has some other exploitable structure, we can often solve problems with tens or hundreds of thousands of variables and constraints." Solving a linear program is a fairly mature technology. The challenge becomes being able to transform an otherwise convex problem into a linear program. Boyd also provides theory behind tools such as the l_1 norm-penalty or regularization techniques in Chapter 6 of [9] that can be used to make a problem sparse and subsequently to prefer a solution that includes as much slack as possible. Theory behind the basis for using regularization for feature selection in linear programming problems is explained in depth by Yao and Lee in [83].

Huang et al. in [37] formulates the maximum flow problem using multipath routing subject to interference as a Linear Programming (LP) problem for multi-hop wireless ad hoc networks using beamforming antennas. The problem is different from the traditional maximum flow problem because of the interference constraints. They mention, It can be solved by a centralized algorithm at an omniscient base station. This is feasible because a base station is usually available for commanding and data collection. Typically, the base station has greater computation capacity and higher energy level, thus it is able to carry out complex computing. Although the centralized LP solution gives the optimal multipath flow, it has the inherent and common disadvantages of all centralized algorithms of it not being scalable to the network size and cannot quickly adapt to changes in link condition and topology. The computation time shows that the computation load increases steeply as the network size increases. So developing a distributed algorithm for large scale ad hoc networks, which is jointly routing and scheduling, is their future work. Nevertheless, this work directly highlights the benefits of using linear programming for these types of problems. With regards to the complexity issues and this work, it is distributed whereby each hub is servicing local users using local information which mitigates the issues mentioned.

3.2.14 Multicasting

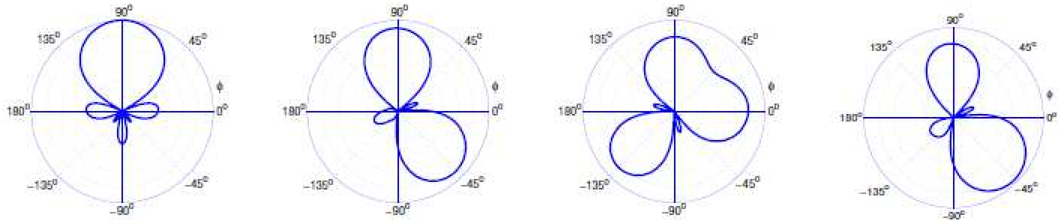
Jorswieck et al. in [43] analyze opportunistic beamforming with finite number of single-antenna users under the constraint that the feedback overhead from the mobiles to the base is constant. This problem is with regards to a MIMO broadcast channel with partial side information in terms of the feedback from the clients. This feedback is the index of the best beam and the channel norm. This mechanism is similar to the ones presented in Channel Characterization below. At first, we apply majorization theory (see reference for details) to better understand the impact of system parameters on the performance of TDMA-OB (opportunistic beamforming) and SDMA-OB. This is a broadcast problem so the determination is if or when it is better to send out traffic to one user using one beam at a time (TDMA-OB) or to send out to multiple users using multiple beams (or a composite) at a time. They find that the difference between the sum rate of the TDMA-OB using the maximum throughput scheduler increases with increasing spatial correlation at the base station and with more unbalanced user variances. The upper bound on the sum rate performance of the SDMA-OB was shown to behave conversely with respect to the user distribution. The conclusion of this analysis is that for (a small number of) less spread out users and moderate to high SNR, SDMA-OB performs worse than TDMA-OB. One remedy without increasing the feedback is to adapt the number of beams to the number of active users (and their channel statistics). Although we are limited to the TDMA case in our work, their analysis is useful in terms of addressing tracking and subsequent channel estimation analysis.

Sundaresan et al. in [76] consider the problem of efficient link-layer multicasting in wireless networks with switched beamforming antennas. There is an inherent tradeoff between multicasting and beamforming. As discussed, beamforming provides the advantaqe of gain and increased signal strength in the directions the pattern is servicing. The disadvantage is limited broadcast with respect to all directions that an omni-directional multicast would provide. In their work, they design and evaluate an optimal algorithm and an easier greedy algorithm with performance guarantees, which generate and schedule efficient beam patterns for multicast transmissions. These algorithms are built upon the novel usage of composite beams that provide an ability to control the operating point of the tradeoff curve between beamforming and multicasting. This is in contrast to the sub-optimal usage of either purely beamformed or omni-directional beams by existing schemes. These composite beams strike a balance by being beams that could service multiple directions at once through adjustments of the amplitudes or phases as expressed in Equation 3.1. They consider two models for composite beam pattern generation: one in which transmit power is (a) equally split, and (b) asymmetrically split between the constituent main lobes. Consider (a) of Figure 3.17.[3] They refer to that default pattern as $A(k, 0)$. They consider the beam pointing with a center direction of $-45°$ to then be $A(k, 3)$. The intent is to form a composite beam that points in both directions. They do this in the form of creating a beam $B(k)$ such that

$$B(k) = \frac{A(k,0) + A(k,3)}{\sqrt{2}}.$$

This is expressed as (b) in Figure 3.17. The result is the power transmitted with respect to each of the specific beams is then reduced by $\frac{1}{2}$ or 3 dB. Similarly, if we consider (c) of Figure 3.17 then

[3] Figure taken from [76]



(a) 1 lobe at 90° (b) 2 lobes at 90°,−45° (c) 3 lobes at 90°,0°,−135°, (d) Asymmetric power split

Figure 3.17: Multi-lobe Beam Patterns

we have

$$B(k) = \frac{A(k,0) + A(k,2) + A(k,5)}{\sqrt{3}}.$$

Subsequently each beam's power is reduced by 4.7 dB. Nevertheless it is higher than omni-directional and if they are out of reach of omni-directional but within the reach of this composite beam then the benefit is achieved. The Asymmetrically Split beam pattern is shown, by example, in (d) of Figure 3.17. The same idea is applied but in this case

$$B(k) = \frac{1}{\sqrt{3}}A(k,0) + \sqrt{\frac{2}{3}}A(k,3).$$

Assuming you want to service multiple clients at a time, they all don't necessarily have the same signal strength characteristics and therefore a group of users has a bottleneck user that affects all of the users in that group. There is effectively a single RSSI value associated with each group based on that of the bottleneck user. The AP then has to determine an efficient way to, "determine how to partition (group) the beams and make individual transmissions to each of these groups such that the aggregate time consumed to disseminate a given amount of data is minimized." For the equal power technique, they formally state the problem as:

$$\min \sum_{p=1}^{G} \frac{L}{W\log_2(1 + \frac{min_{j:x_{j,p}=1}\{R_j^*\}}{\sum_j x_{j,p}})}$$
$$s.t.\ \sum_p x_{j,p} = 1,\ \forall j \in \mathcal{B};\ x_{j,p} \in \{0,1\}$$

L represents the message size, W is the bandwidth, $x_{j,p}$ is a binary variable of if beam j belongs to partition p, and G is the number of partitions. The outputs then are $x_{j,p}$ and G. They show that this problem can be solved in polynomial time and establish that the partitions required by this problem are contiguous. They then determine that this problem reduces to a sequential partioning problem where given a set of beams ordered with respect to their effective RSSI, then the goal is to partition this ordered set into groups so as to minimize the aggregate delay. They express that this can be formulated with the following integer program:

$$\min \sum_{p=1}^{M} c_p y_p; \;\; s.t. \; \sum_{p=1}^{M} a_{jp} y_p = 1, \;\; \forall j \in [1, |\mathcal{B}|]$$

$$where \; c_p = \frac{L}{W \log_2(1 + \frac{min_j\{a_{jp} R_j^*\}}{\sum\limits_j a_{jp}})}; \;\; a_{jp}, y_p = \{0, 1\}$$

They go on to prove that this linear program relaxation solves the problem optimally. Therefore, given this schedule and the means for which to utilize the prescribed composite beam based on the schedule outcome, an algorithm can be implemented integrating both of them. The problem lies in that this algorithm results in a worst case running time $O(K^7)$ or $O(K^8)$. They develop a greedy algorithm that provides, "near optimal performance" and is $O(K^2)$. The authors then take their solutions and, using their Table 3.4, map the effective SNR from the solution values to discrete rate based on the respective threshold.

Table 3.4: SINR-Rate Table (802.11g/b)[76]

SINR (dB)	Rate (Mbps)	SINR (dB)	Rate (Mbps)
$\geq$ 24.56	54	[10.79,17.04)	18
[24.05,24.56)	48	[9.03,10.79)	12
[18.8,24.05)	36	[7.78,9.03)	9
[17.04,18.8)	24	[6.02,7.78)	6
$\geq$ 12	11	[5.03,9.02)	2
[9.02,12)	5.5	[2.01,5.03)	1

They then go on to formulate the respective problem for the asymmetric power model as:

$$\min \sum_{p=1}^{G} \frac{L}{W \log_2(1 + \frac{1}{\sum\limits_j \frac{x_{j,p}}{R_j^*}})}$$

$$s.t. \; \sum_p x_{j,p} = 1, \;\; , \forall j \in \mathcal{B}; x_{j,p} \in \{0, 1\}.$$

They prove that it can also be solved in polynomial time and that its optimal solution does not have any partitions. They similarly apply an algorithm that combines this schedule along with the composite beam former to an overall solution and subsequently develop a mapping method to extend it using discrete rates. The authors go on to highlight some important points as it applies to our work as well. They highlight the fact that there are overlapping beams and it is almost trivial to extend their algorithm to account for this. Considering that they are using the same antenna as used in this work then we will account for that as well. Another key point they mention is with regards to the $100\mu s$ switching delay that is incurred in going from pattern to pattern. They argue that this additional delay will scale with the number of partitions and that is is easy to incorporate into their formulation and solutions by adding a constant cost to each partition cost. The authors evaluate the performance of their proposed algorithms using signal traces from an experimental indoor scenario. Clients are distributed randomly around the AP where they take a subset of them. The AP sends out packets on each of the 8 beams used sequentially at 1 Mbps. The clients measure the RSSI values and report their findings back to the AP. From these results, they develop different topologies and scenarios and run their algorithms based on these scenarios. Given this, they find that beamforming with 8 element arrays can improve multicast delays by about four-fold versus omni-directional. Their greedy solution performs very close to the optimal and outperforms existing schemes by two-fold, with higher gains in user clustered environments. Their asymmetric method solutions improve performance over the equal-powered methods by about 20-30 percent and the asymmetric solutions are less robust when it comes to fading and interference. To address this point, the authors mention that they are operating at effective RSSI values and therefore there is not much margin for accomodating fading losses in most of the clients. They argue that the design of an efficient retransmission scheme is all the more important, however point out that their retransmission scheme requires further investigation along with client mobility.

In [87] and the full paper version, [86] the authors build off of the work done in [76]. Using the equal power model they provide a lower complexity, dynamic-programming based optimal solution for both continuous and discrete rate functions. The complexity of this algorithm is much better at $O(K^2)$ versus $O(K^7)$ as presented in [76]. The majority of their work is, however, regarding the more difficult asymmetric power splitting (ASP) model. They present that for the ASP model there currently exists no results or approximation solutions. They prove that it is NP-hard to have a reasonable approximate solution for a general rate function while showing that for discrete rate functions, the multicast beamforming problem can be converted to a generalized version of the bin-packing problem. This then allows them to use general bin-packing methods to obtain a polynomial time approximation scheme as well as an asymptotic approximation solution. Their experimentation is done similarly based on traces and using the same AP as in [76]. Their evaluations, based on simulations conducted off of these traces, indicate that the proposed algorithms significantly improve the state of the art in literature. The multicast delay reduction for 802.11a and 802.11b is up to 20 and 25 percent respectively, compared to those done in [76] using the ASP model.

The results based on these methods are somewhat difficult to infer as they do not appear to consider mobility and based on measurements from a real-time executed scenario. The results are based on generating simulations and running the algorithms against RSSI traces from clients based off of a single transmission. Simulations are then run against the parameters returned by the algorithms. It does not appear that the time it takes to run the algorithm is even considered as well.

These papers are concerned with only serving static users' multicast (or broadcast) traffic whereby the same information is sent out to everyone as quickly as possible. To reiterate, the problem presented in this work is with respect to a dynamic mobile environment with traffic that is unique to a particular user. Nevertheless, the concepts and background information presented in these works are novel and useful.

3.2.15 Channel Characterization

Much work has been done with regard to channel estimation and subsequent scheduling in terms of being able to service mobile users in a hub-spoke scenario. This is similar to a cellular type scenario. With cellular phone traffic being much more data based then before then the transition to account for these factors are well described in [6], [41]. The general basis for these methods are the clients transmitting back to the hubs the index of the highest data rate that they can reliably receive.

This is done similar to standard training and tracking techniques that include path loss, large scale shadowing (which also accounts for large scale fading), and small scale temporal fading. Small scale is typically modeled as Rayleigh for mobile applications and Ricean for stationary applications. The scheduler then schedules based on the reported data rate requests from the clients and the amount of data that has been sent in the past. Furthermore, diversity exists between overlaps of different hubs and subsequent handoffs as a result of a client possibly being able to be better served by a different hub. This is one key difference from our work. Also, our work has only one transceiver that can only listen or transmit to one user at a time. We take a TDMA approach not only from a layer 2 approach but also in part due to a layer 1 limitation. Nevertheless the techniques provided in these general methods as well as the following more specific methods and channel modeling techniques are considered.

Furthermore, regarding channel characterization another goal of this work is to find an acceptable model that can predict the signal quality for a given user at a point in the future. Hallen et al. in [24] mention, "While many researchers have addressed a related problem of estimation of current fading conditions, prediction of future fading coefficients has not been addressed until recently." They discuss recent proposals to adapt different transmission methods such as modulation, power control, channel coding and antenna diversity to rapidly time variant fading channel conditions. They cite that these proposals cannot be realized without being able to predict the channel coefficients several tens-to-hundreds of symbols ahead in practice. They describe an adaptive long range fading channel prediction algorithm (LRP) and its utilization with adaptive transmission methods. This channel prediction algorithm computes the linear Minimum Mean Squared Error (MMSE) estimates of future fading coefficients using a standard Auto-Regressive (AR) model based on past observations (samples), and subsequent estimates of the current channel. See [75]. This algorithm forecasts future fading signals far due to increased memory span, achieved by using a sufficiently low sampling rate for a given fixed filter size. The samples are taken at least the Nyquist rate given by twice the maximum Doppler frequency, in order to fit within the coherence time (frequency). This sampling rate is therefore much lower than the data rate. However, the predicted samples can then be interpolated to forecast the fading signal at the data rate. The LRP is validated for standard stationary fading models and tested with measured data and with data produced by their realistic physical channel model. This model accounts for the variation of the amplitude, frequency and phase of each reflected component of the fading signal. For mobile situations involving their physical channel model they found that the poles of their AR model are primarily determined by the Doppler shifts and needs to be continuously updated. Therefore, as the Doppler frequency shift increases then the less reliable their model is. They found that in this case reliable prediction ranges are for half a wavelength of less. Nevertheless, both their numerical and simulation results show that long range prediction makes adaptive transmission techniques feasible for mobile radio channels as we expect.

Zhou in [90] cites how adaptive modulation improves the system throughput considerably by matching transmitter parameters to time-varying wireless fading channels. Crucial to this adaptive modulation is the quality of channel state information at the transmitter. They present a Minimum Mean Square Error (MMSE) channel predictor based on pilot symbol assisted modulation (PSAM) for independent and identically distributed Rayleigh fading multiple-input multiple-output Rayleigh fading channels, similar to the approach in [24]. PSAM works similar to standard training and tracking sequences as explained in [68]. Known pilot symbols are periodically inserted at the transmitter. At the receiver, the samples corresponding to the known pilots are extracted, based on which CSI is interpolated using optimal Wiener filtering. Coherent detection is then performed for symbol demodulation. They then analyzed the impact of the channel prediction error on the bit error rate performance of a transmit-beamformer with adaptive modulation that treats the predicted channels as perfect. Their results reveal the critical value of the normalized prediction error, below which the predicted channels can be treated as perfect by the adaptive modulator; otherwise, explicit consideration of the channel imperfection must be accounted for at the transmitter.

Feng et al. in [27] provide statistical models for air-to-ground radio channels in a dense urban environment. Their results show that air-to-ground channels have a much higher line-of-sight (LoS) probability, as expected. They have less non-LoS (NLoS - direct path blocked) probability and less NLoS path loss as compared to LoS channels. They also have less shadowing than terrestrial peer-to-peer (P2P) channels. This highlights the fact that airborne platforms may serve as relaying nodes to extend the range and improve the connectivity between terrestrial ad hoc terminals. Their models are based on hilly terrain, but considering that the airborne height is much larger than the terrain irregularity, they find that terrain obstruction is infrequent. Diffraction loss is mainly caused by the building height above ground level. Therefore, these models are also suitable for flat terrain with similar building clutter. These models can be used for satellites or UAVs with elevation angles greater than 10 degrees. The models can be used at frequencies from 200MHz to 5GHz, and thus serve both civilian and military applications. Additional analysis regarding characterizing the UAV to ground channel for 802.11a based links is provided in [82].

Some work has been done with regards to utilizing UAVs to provide relay services for mobile ad hoc networks with terrestrial communication nodes. Jiang et al. and their work in [42] is similar to this work in that they consider a UAV with a multi-antenna capability flying over a collection of N single antenna mobile ground nodes. Their goal is to control the motion of the UAV so as to optimize the uplink communications performance. They assume having multiple receivers at the UAV for the purposes of multiple users communicating with the UAV simultaneously. The UAV flies with a constant velocity and it adjusts its heading in discrete time steps (assuming a constraint on the maximum turn rate). They assume that the first and second order characteristics of the channel are known to the UAV (e.g., via training data from the ground nodes). They then take the vector that maximizes the SINR ratio depending on the scenario while assuming a correlated Rician fading channel between each user (given the simultaneous transmissions) and the UAV with consideration of large-scale path loss. They determine that since there is little multipath scattering near the UAV then any Rayleigh fading components will experience high spatial correlation at the receive end of the link. Similar to the scenario given above by [43], they consider both SDMA and TDMA approaches (the differences explained above). They adopt a first-order AR model for the dynamics of the ground nodes and assume the nodes provide their location and trajectory at each time step. The UAV then takes this feedback and uses a Kalman Filter to predict their positions at time n + 1. They then determine the UAV's heading based on their estimates of the channel and on the movement of the client based on which heading will minimize the user's mutual interference and provide a maximum sum rate between all of the users or a proportionally fair one. For their simulations the user's transmit power is set to 45dB, UAV altitude is 350m. and is at a speed of 50m/s.

Although the scenario and goals are somewhat different, this method is also very similar. Its insights with regards to channel estimation are fairly standard however they offer an approach to estimating the user's next location given their AR model with a Kalman filter.

3.2.16 Scheduling under Uncertainty

Scheduling with uncertainty has been an area that has plagued military planners throughout history. It is a logistics challenge to be able to forecast what demands will arise and what resources will be available where and at what time. Similar issues arise in wireless communications. In a mobile environment, a channel may have throughput that changes over time, as discussed in the last section, while the traffic to be carried may show up in uncoordinated bursts. Consider the scenario presented earlier where dispersed teams are served by an overhead unmanned aircraft vehicle (UAV) hub as shown in Figure 1.10. As discussed, those teams might receive unique individual data flows in some form via the UAV overhead (aside from general video broadcasts). The data waiting for each user and the data rates may change over time and the UAV radio must schedule when to communicate with which user at which time in order to meet the different users' traffic loads. A key challenge in efficiently scheduling their communications, versus a simple sub-optimal greedy approach, is being able to forecast their load demands and how much time it will take to meet those in terms of their changing channel rate capability. It becomes necessary to be able to track them and forecast how their respective rate might change over time.

The tracking problem transforms this problem into a multiarmed bandit problem with restless bandits but moreover a scheduling with uncertainty type problem. As given by Scala et al. in

[70], the classic multiarmed bandit problem is a special type of stochastic dynamic programming problem where there are N parallel projects, each of which has a finite state space. At each discrete time instance, exactly one project will be active and that project earns a reward related to the current state of the project. If a project is active at state n then it changes state according to a homogeneous transition probability matrix. Inactive projects do not change state. The aim of the problem then is to schedule the projects such that you maximize the sum of the rewards. Somewhat relevant multiarmed bandit scenarios are discussed in [51][50], however the classic multiarmed bandit problem assumes that only active projects (e.g. a task acted upon) change state. In a dynamic scenario, as presented here, this is not the case. Scala et al. [70], with references to [8][80], discusses target tracking with restless bandits. In this case, "all bandit processes evolve at each time instant," which relates to the dynamic scenario presented here. In his scenario he shows a greedy strategy performs better than an alternating strategy. Our work will show that a greedy strategy isn't always preferred.

More generally, key overviews of research on scheduling with uncertainty are presented in [35][56]. Approaches in dealing with uncertainty include: reactive scheduling, scheduling under fuzziness, proactive (robust) scheduling, sensitivity analysis, and stochastic scheduling. Reactive scheduling seeks to repair discrepancies that show up as result of unexpected events while following an original baseline schedule. In our problem, new information arrives every time step and it is easy to modify schedules at every time step. So, the notion of a baseline schedule is less relevant. Fuzzy programming approaches assume little to no data is available to be able to estimate the uncertain parameters. This isn't the case in this scenario in that general parameters are understood or could be gathered, such as the noise environment, average load per user, or mobility patterns. As explained in [5], robust optimization approaches seek to minimize the effect of disruptions, however they can be overly conservative. Techniques have sought modified less conservative approaches by scheduling given "adjustable" and "non-adjustable" variables. The non-adjustable variables are needed as a minimum to develop a schedule but are affected by the adjustable variables that aren't available until realization. The adjustable counterpart can be much less conservative but previous approaches have been shown to be NP-hard [5].

Sensitivity analysis looks at how a given model output is affected by the input parameters. This provides insight into the quality of the model as well as understanding its robustness and reliability. However, as discussed in [56], the combinatorial nature of the scheduling problem poses problems. Stochastic approaches are the most widely used. They seek to maximize the expectation of a certain performance criterion. These approaches are divided into two-stage, multi-stage, or chance-constraint programming based approaches. The two-stage schedule versions schedule first probabilistically. They then follow it up with a recourse decision in the second stage to account for variations experienced due to actual realizations. Multi-stage follows similarly where subsequent decisions are made as they become known. Chance-constraint approaches integrate variability in the scheduling constraints, such as demand uncertainty. Nevertheless, despite the stochastic type approach the resulting deterministic optimization problems are all posed as Mixed Integer Linear Programs (MILP) problems which are considered NP-complete [86].

The deficiencies in these approaches lead us to consider what is called Receding Horizon Control or Model Predictive Control as explained in [58]. In this method, given the future uncertainty, an optimization problem is solved iteratively from time step to time step while accounting for additional information as it becomes available. Although calculations are conducted each time instance the scope of the problem is scaled down significantly as it deals with a smaller planning horizon with a smaller client set versus trying to establish a full schedule over a larger horizon and client set. Also, given the hub and spoke scenario used here, the calculation capability at the hub is more robust than that a standard client might have. As noted in [58], this policy is not an optimal policy, however it is a sophisticated heuristic that delivers good performance while making the problem much more tractable. In our work we seek to combine the simplicity of scheduling over a smaller horizon, while taking advantage of what can be gained by considering a very long, possibly infinite, horizon.

3.2.17 Beam Switching

Grimaud focused on beam switching in [32] in his TDMA schemes presented earlier.

Kumar et al. in [52] show that link scheduling and beam switching are closely related such that they are difficult to separate. As mentioned in Chapter 2, the number of beam switches can unnecessarily increase depending on the order in which the links are scheduled for transmission. Although their techniques are with respect to an STDMA approach, with simultaneous transmissions, they illustrate that by jointly considering beam switching and link scheduling, the number of beam switches can be dramatically decreased.

Chapter 4

Scheduling

4.1 Problem Overview and Formulation

We begin with scheduling, versus tracking, as it is the main focus of this work and sets the requirements for how good the tracking needs to be and subsequently creates the plan for execution. Therefore in this section we address “Attempt to Schedule” and “Can it be scheduled” specifically from Figure 2.1. Towards the end we discuss the unscheduled aspects of “Need more tracking” and “Best Effort Options”.

We first describe and analyze the problem with a single hub by initally repeating many of the points, with slight modifications, made earlier. We will later on extend to a mesh of hubs. We will assume the hub has m different beam patterns, communicates with n clients, and each client has a single arbitrary beam pattern as presented earlier.

Once again, we do not directly model any of the beam patterns. Instead, we consider time broken into a sequence of intervals, so that in interval s beam i can transfer to client j a capacity $c_j^i(s)$. Given that this is a mobile environment and given a dynamic radio environment then $c_j^i(s)$ will change over time for a particular client for a given beam, however within the interval, s, we assume it maintains that value constantly. Furthermore, we do not consider which direction (to or from the hub) and only consider the amount of data that can be transferred.[1]

Data can be transferred using only one beam and client combination at a time. However, the

[1] The direction can be incorporated by treating each direction as a separate client. By treating each direction separately, we are able to incorporate link asymmetry (e.g. different transmit powers) separately.

time in each interval can be divided among different beam and client combinations. Let $p_j^i(s) \in [0,1]$ be the fraction of interval s that the hub communicates with beam i to client j. Since only one beam can be used at a time $\sum_{j=1}^{n} \sum_{i=1}^{m} p_j^i(s) \leq 1$. Further, $\sum_{i=1}^{m} p_j^i(s) c_j^i(s) \cdot \tau$ is the total data transferred to client j in interval s, where τ is the amount of time in the interval.

Looking at Figure 1.10 again, the general problem is to find an efficient way for each hub to service its clients successfully. More specifically, we want to be able to transfer the data for each client. Since data is bursty and may momentarily exceed the channel capacity, we consider a planning period of the next T intervals. If the average data transfer per interval for client j is λ_j then the goal is to transfer $T\lambda_j$ for each client j.

We assume that the number of intervals, T, is sufficiently large to assist in scheduling. For instance, in the orbit of a UAV above ground clients, the UAV may only be able to communicate with each client at certain points in its orbit. In this case, T would be sufficient to include one UAV orbit. The length of the intervals, later denoted as τ, themselves are chosen sufficiently small to capture both fine-grain changes in the capacity and to meet real-time requirements.

The first challenge to making this work is that there are mn possible beam and client combinations and it is prohibitive to have to track each of these combinations. Thus, at any given time we may only have estimates of the capacity of the beam i client j combination denoted $\hat{c}_j^i(s)$.

It is the job of tracking to maintain these estimates. These estimates may come from direct link measurements, indirect measures (e.g. GPS coordinates and orientation of sender and receiver), or estimates from past measurements. In the latter case, we may have a conservative estimate that degrades to zero over time at a rate that is a function of client mobility, radio dynamics, etc. Thus, given the degrading information over time, scheduling over a future planning horizon must tradeoff which user is scheduled in which order to ensure that the needs of each user are met. Chapter 5, Scheduling with Uncertainty, will address this more specifically.

This leads to a second challenge as mentioned earlier. While ongoing traffic transfers will keep the capacity of some beam and client pairs up to date, other, potentially better, combinations may go unexplored. In addition to this tracking specific overhead, there can be other unmodeled overhead in order to coordinate client and hub activities. As a result, outside of scheduling traffic, the scheduler should reserve time for the tracker to update alternate beam and client pair capacity estimates and the other overhead. This reserved time not used for transferring data is denoted as *slack* time.

A third challenge is that the beamforming antenna requires time to switch between beams. As mentioned prevously in Section 2.3.2, two otherwise equal schedules may differ if one cycles through more beams than the other. The scheduler should choose schedules that minimize beam switches to minimize this switching overhead.

A final constraint is that some users' traffic may have a real-time component which requires either specific deadlines or periodic guaranteed transfers.

So, the problem is to schedule the communication over the next T intervals so that (a) each client j transfers an average of λ_j of data with the hub, (b) the slack time is maximized, (c) the number of different beams used in each interval is minimized along with the overall number of beam switches, and (d) real-time constraints are met.

4.2 Methodology and Initial Analysis

4.2.1 Linear Program

This scenario incorporates many questions to be addressed such as how and when to track users, how are they polled for both these and data exchange purposes (coordination), and how do you estimate $c_j^i(s)$. In this chapter we focus on the scheduling aspects involved in meeting the clients' throughput requirements and subject to their delay constraints, as applicable. We present a Linear Program for the scheduling problem as follows:

$$
\begin{aligned}
GIVEN: \ T, n, m, \ \{\lambda_j | j = \{1, ..., n\}\}, \\
\{\hat{c}^i_j(s) | i = \{1, ..., m\}, j = \{1, ..., n\}, s = \{1, ..., T\}\}; \\
OBJECTIVE: \ \max \sum_{s=1}^{T} \sum_{j=1}^{n} \sum_{i=1}^{m} p^i_j(s) \hat{c}^i_j(s) \\
s.t. \\
\sum_{s=1}^{T} \sum_{i=1}^{m} p^i_j(s) \hat{c}^i_j(s) \leq T\lambda_j \ \forall j \\
\sum_{j=1}^{n} \sum_{i=1}^{m} p^i_j(s) \leq 1 \ \forall s \\
p^i_j(s) \geq 0 \ \forall j, i, s
\end{aligned}
$$

τ is not included in the objective throughput equation because it is a constant value, common throughout, and could be normalized. The first constraint in the program makes sure that the clients cannot exceed their offered rate over the length of the period. To account for negative queues or bursts of traffic we refer to our scheduling with uncertainty receding horizon approach presented in Chapter 5. For the purposes of this chapter we can assume an infinite queue or sufficient load carry-over to account for this. Nevertheless, to account for this the first constraint could easily be changed to $\sum_{k=1}^{s} \sum_{i=1}^{m} p^i_j(k) \hat{c}^i_j(k) \leq \sum_{k=1}^{s} \lambda_j(k) \ \forall j, s$. Also of note is that this first constraint uses a less than or equal sign. One might expect a greater than or equal sign here. However a little thought shows that it is the correct direction. First, it is easy to come up with an initial feasible solution: $p^i_j(s) = 0$ for all beams and clients. Second, if there is a solution that carries all of the traffic, then the maximum of the objective will be the desired $\sum_{j=1}^{n} T\lambda_j$. If the opposite direction is used, then the objective will force solutions that use up all the available time and there will be no slack left in the schedule. As noted earlier, slack is necessary for the tracking, overhead, and beam switching.

The second constraint makes sure that only one client can use one beam at a time. The third constraint makes sure that all of the time allocations are non-negative.

This formulation does not specifically include real-time constraints but they are easily added and have been verified through simulation. For instance, if a voice communication with client j

required λ'_j data sent every 20 msec, then the intervals could be set to $\tau = 20$ msec in length and we could add the constraints

$$\sum_{i=1}^{m} p_j^i(s)\hat{c}_j^i(s) \geq \lambda'_j \ \forall s.$$

Finally, we note that though T is the planning horizon a receding horizon controller could be used in which after each $T' < T$ intervals, a new schedule is recalculated with updated data requirements and capacity estimates. This will be addressed in Chapter 5.

4.2.2 Regularized Linear Program

At this point we have developed a schedule that is optimal but we wish to have a system that can be modified to provide slack where possible, to be able to incorporate other features, and to be able to account for non-linear factors such as beam-switching. We turn to use of the l_1 norm penalty or regularization as discussed in [9] and in [30]. We modify our original objective to incorporate a penalty with respect to the number of time sub-intervals included as shown:

$$OBJECTIVE: \ \max \sum_{s=1}^{T} \sum_{j=1}^{n} \sum_{i=1}^{m} [p_j^i(s)c_j^i(s) - p_j^i(s)]$$

subject to the same constraints in the previous linear program. Using this penalty method, as proven in [30], and as shown in our simulation results, the optimal solutions with and without the penalty are the same. However, from the space of optimal solutions, a sparse optimal solution is selected as discussed in [9] since max $T - \sum_{s=1}^{T} \sum_{j=1}^{n} \sum_{i=1}^{m} p_j^i(s)$ occurs when $\sum_{s=1}^{T} \sum_{j=1}^{n} \sum_{i=1}^{m} p_j^i(s)$ is minimum.

4.2.3 Beam Switching

The smart ordering of beams is not trivial and if not done efficiently can result in unnecessary additional beam switches. In [32][52] the authors cite that the switching of beams consumes energy and requires a certain amount of time to stabilize, therefore introducing delays. So, minimizing the number of beam switches improves performance and therefore increases throughput. The first step in our switching method is to sequence all of the users that use a beam in interval s, in succession. Therefore, you only have to use that beam once per interval. Within an interval you have to switch to all of the beams used in that interval before you can proceed to the next interval. Efficiency is gained by utilizing common beams as you go from interval to interval. In other words, if an interval shares a common beam with the next interval then generally you seek to use that beam last in the initial interval such that you use it first in the next interval and don't have to switch beams in going from interval to interval. As you go from interval to interval then it follows that you incur $m(s) - 1$ beam switches within each interval, where $m(s)$ represents the number of beams used in interval s, $m(s) \leq m \ \forall s$. If you don't transition from one interval to the next via a common beam then you incur a switch between intervals. Therefore the number of beam switches used over the entire period becomes $\sum_{s=1}^{T}(m(s) - 1) + \sum_{s=1}^{T-1} I(l(s) \neq e(s+1))$, where $I(\cdot)$ is the indicator function and $l(s)$ is the last, or leaving, beam and $e(s)$ is the first, or entering, beam used in interval s. Therefore the sequence breaks down to determining the transition beams from interval to interval, or the leaving beam from one interval to the entering beam of the next. The goal is to maximize the occurrences of the leaving and entering beams being the same over the course of the entire schedule. Different methods exist to try and determine this. Greedy algorithms, such as what is presented in [52], are less than optimal.

We define a beam matrix as a $T \times m$ matrix, $B = \{b_{si}\}$, where $b_{si} \in \{0, 1\}$ and $b_{si} = 1$ iff beam i is used in interval s. We present an example matrix,

$$B = \begin{vmatrix} 011 \\ 111 \\ 011 \\ 001 \end{vmatrix}.$$

A greedy algorithm would look at the current interval, s, and sequence the last beam for the interval (designated by the same row in B) to be one that matches a beam in the next interval (row), $s + 1$. Using this method we could come up with the following sequence of beams, $b_{13}, b_{12}, b_{22}, b_{21}, b_{23}, b_{33}, b_{32}, b_{43}$. This is a slight abuse of notation given $b_{si} \in \{0, 1\}$ as given above but here we are simply identifying that beam i is being used in time interval s. Within this sequence we have 5 switches. Following the greedy algorithm again we could just as easily obtain

$b_{12}, b_{13}, b_{23}, b_{21}, b_{22}, b_{32}, b_{33}, b_{43}$ or 4 switches. Both proceed in a similar manner but do not consider that the last row causes a bottleneck that affects how the first switch should be handled. An optimal sequence using a binary integer linear program can be determined as well but its complexity is NP-hard [76]. We therefore take another approach. As discussed, the schedule only needs to determine what the entering and leaving beams are per interval. If the interval only has one beam used then it is both the entering and leaving beam. If the interval has more than two beams then the additional beams, beyond the entering and leaving beams, are taken in any order between the entering and leaving beam since they still have to be visited. If there is more than one beam in an interval then the entering and leaving beams have to be different, otherwise you would either be visiting one beam twice in the interval or you didn't visit the other beams in the interval.

Given B and the aforementioned rules that apply in the sequencing of these beams, all of the possible sequences follow a directed graph. A network flow problem can be set-up, see [17][60], by establishing a Weighted Beam Graph, $G = (E, N, W)$, as given in Algorithm 1.

An example is shown in Figure 4.1. By setting up this Weighted Beam Graph (WBG) as

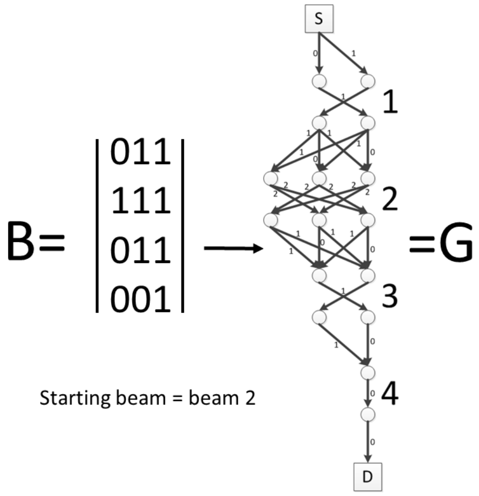


Figure 4.1: Creating a WBG from B

given and according to the rules presented earlier, then by finding the lowest cost path from S to D we will have the most efficient sequence of the entering and leaving beams per row over the course of the entire period. Thus, it is a Shortest Path problem. The WBG as we present it also gives us the

Algorithm 1 Define a Weighted Beam Graph

```
Input: Beam Matrix, $B$
Output: graph $G = (E, N, W)$

DEFINE NODES
1. Let source, $S \in N$, and destination, $D \in N$.
2. For each row in $B$ we create a set of entering nodes, $\{e_{si}\}$, and leaving nodes, $\{l_{si}\}$, where
$e_{si} \in N$ iff $b_{si} = 1$ and $l_{si} \in N$ iff $b_{si} = 1$.
NOTE: if a row has all zeros then we disregard it and schedule using the rows before and after
it independently.

DEFINE DIRECTED EDGES
3. $(S, e_{1i}) \in E$ for each $e_{1i} \in N$.
4. $(l_{Ti}, D) \in E$ for each $l_{Ti} \in N$.
5.
for $S = 1$ to $T$ do
  if $m(s) = 1$ then
    $(e_{si}, l_{si}) \in E$ for the $e_{si}, l_{si} \in N$.
  else
    $(e_{si}, l_{sj}) \in E$ for each $e_{si}, l_{sj} \in N$ and $i \neq j$.
  end if
  $(l_{si}, e_{(s+1)j}) \in E$ for each $l_{si}, e_{(s+1)j} \in N$ and where $s < T$.
end for

DEFINE WEIGHTS
6. for each $e_{1i} \in N$, $W(S, e_{1i}) = \begin{cases} 0 & \text{if } i = \text{starting beam} \\ 1 & \text{otherwise} \end{cases}$.
7. $W(l_{Ti}, D) = 0$ for each $l_{Ti} \in N$.
8. $W(e_{si}, l_{sj}) = m(s) - 1$ for each $e_{si}, l_{sj} \in N$.
9. for each $l_{si}, e_{(s+1)j} \in N$
 $W(l_{si}, e_{(s+1)j}) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases}$.
Return: $G = (E, N, W)$
```

cumulative costs, or beam switches, necessary for the entire period. Once we have the entering and leaving beams, then the unused beams per row can be placed in between the entering and leaving beams in any order. Our optimal beam sequencing algorithm is as presented in Algorithm 2

Algorithm 2 Beam Sequencing Algorithm

```
Input: Beam Matrix, $B$.
Output: sequence $x = (b_{1i}, \ldots, b_{Tj})$.
$G$ is weighted beam graph of $B$.
$x = null$.
1. Let $P = \text{ShortestPath}(G) = (S, e_{1i}, \ldots, l_{Tj}, D)$.
2.
for $s = 1$ to $T$ do
  $x = x + (e_{si} \in P)$.
  if $(e_{si}, l_{sj} \in P)$ and $i \neq j$ then
    for any $(b_{sk} \in B) \neq (e_{si}$ or $l_{sj} \in P)$ then $x = x + b_{sk}$.
    $x = x + (l_{sj} \in P)$
  end if
end for
Return: beam sequence $x = (b_{1i}, \ldots, b_{Tj})$.
```

This method can be solved efficiently using any shortest path first algorithm as mentioned. For the purpose of this work we use Dijkstra's algorithm which has time complexity only $O(\log |N| \cdot |E|)$ with $|N|$ being the number of nodes and $|E|$ the number of edges (connections). In our formulation $|N| = 2(1 + \sum_{s=1}^{T} m(s))$ and the number of edges are $O(\sum_{s=1}^{T} m^2(s))$. The quadratic dependence of the time complexity on the number of beams used in each interval, highlights the need to minimize the number of beams used.

4.2.4 Simulation

We consider a MATLAB simulation based on the 17 beam commercial beam steering antenna [18] described earlier. The 17 beams include the 16 directional beams and an omni-directional beam. Our experimental results in the past show that at a distance, out of reach of the sidelobes, a client can only be reached by at most 2 or 3 of the directional beams and possibly omni-directional. For the purposes of these simulations we will assume a primary and secondary beam for each client along with omni-directional. Therefore for the scenarios presented here we assume that the

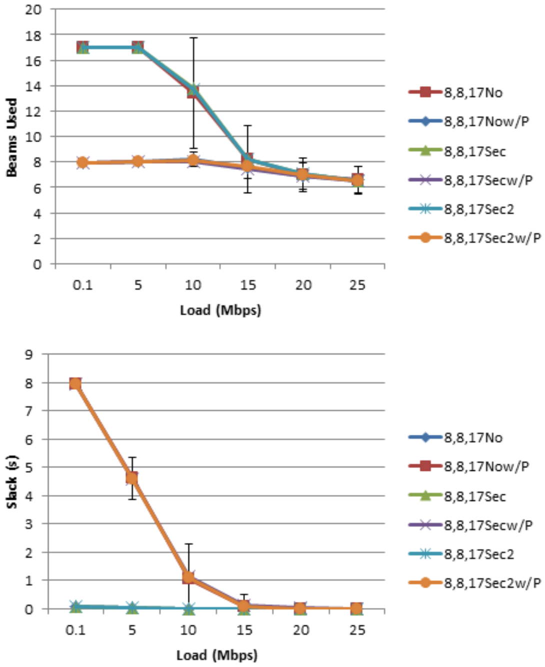


Figure 4.2: beams used vs. offered load(top), slack vs. offered load (bottom)

capacity is mapped to a rate as expressed earlier in Table 3.4. We'll assume a tracking model for the purposes of this chapter. In our receding horizon approach, presented in Chapter 5, the first interval will be based on realizations of observed capacities and subsequent intervals will be based on estimates. The schedule will be re-formulated every interval and therefore the estimates are never used directly, beyond for planning purposes. Without loss of generality for the sake of this chapter, we assume then that the rate of each client's primary beam is uniformly randomly chosen $U(0, 54)$ Mbps (emulating the range for IEEE 802.11g without loss of generality), secondary beam is similarly $U(0, 27)$ Mbps and finally omni-directional is $U(0, 13.5)$ Mbps. In each time interval the rates are chosen independently. The length of a time interval is $\tau = 1$ sec so that an x Mbps rate on beam i to client j in interval s means the $\hat{c}^i_j(s) = x$ Mb.

Figure 4.2 shows results for a collection of similar scenarios involving 8 dispersed users, 8 time intervals, and 17 beams. Scenario $8, 8, 17No$ represents that the clients are all evenly dispersed with no overlap between directional beams. For example it could be expressed as $[P_1|S_1|P_2|S_2|...]$ where each slot represents the sequence of directional beams, P_j represents client j's primary beam and S_j their secondary beam. They all share the omni-directional beam. Scenario $8, 8, 17Sec$ represents that sequentially a client's secondary beam is the primary beam for the next client, $[P_1|S_1P_2|S_2P_3|...]$. Scenario $8, 8, 17Sec2$ then represents that a client's secondary beam is the secondary beam for the next client in four separate but sequential pairs, $[P_1|S_1S_2|P_2|P_3|S_3S_4|...]$. The data transfer requirement for each client is a uniform random variable chosen between 0 and the load value specified in the plot, $\lambda_j = U(0, Load)$ Mbps. Each scenario and load combination is repeated 1000 times and the results averaged where the error bars represent ±1 standard deviation. The load starts out very low, .1 Mbps, such that the system can easily meet the load. In the top graph of Figure 4.2 all three of these scenarios use all 17 beams, exactly. They use all of the allotted time, 8s, as well. They leave no slack despite being able to easily meet the load. At 5 Mbps the system is still able to meet the load with the same results. At 10 Mbps the system is slightly over the threshold by being only able to meet 97% of the load. At this load, there is still no slack but there is high variability in the number of beams used. As the capacity varies, when it is greater than the load, then all the beams are used as before. When capacity is below the load the scheduling is focused on fewer higher capacity beams to carry as much load as possible. In the subsequent situations of 15, 20, and 25 Mbps it continues to use all the time but resorting to using less and less beams with less variability. This is a result of the system becoming more and more overloaded as shown by it only able to support 77%, 60% and 49% of the loads respectively.

We therefore see that the system works as expected when it cannot meet the load. It uses all the allotted time and the number of beams used is limited to the number of higher capacity beams. We also know that it is optimal considering MATLAB's **linprog** solver utilizes the primal-dual method which is proven optimal in [17]. We note that these basic solutions do not account for slack and beam switching. It never has slack and, except where constrained by high loads, it uses all 17 beams.

The results for Scenarios $8, 8, 17Now/P$, $8, 8, 17Secw/P$, and $8, 8, 17Sec2w/P$ include the regularizing penalty of Section 4.2.2 (i.e. w/P). As shown in Figure 4.2, the biggest difference lies where the system can meet the load. Now the same optimal solution is obtained by only using 8 beams and the time is cut down to where slack is maximized. Consequently, as given here, as seen in Figure 4.2, and confirmed throughout our results, we see that the amount of different beams used is minimal.

The next set of simulations highlights the effect from a beginning basic solution to a final solution with penalties and optimal beam switching. In this scenario $T = 20$s, $n = 10$ and $m = 9$. Each user's primary beam is chosen randomly $U(2, 9)$ at a random rate $U(24, 54)$ Mbps. Their secondary beam is directly adjacent to their primary beam at a random rate $U(9, 48)$ Mbps. They all share the omni-directional beam, 1, at a random rate $U(0, 24)$ Mbps. They all share the same fixed load, $\lambda_j = \lambda$ as given in the plotted results shown in Figure 4.3.

These results reinforce what we've seen so far where the basic solution utilizes all the beams and leaves no slack when the system can easily meet the offered load. The solution with penalties uses minimal beams necessary while maximizing the amount of slack. Starting at the knee of the system, or where it can just meet the offered load, $\approx$ 5 Mbps, then the two solutions start to converge to the same results as discussed earlier.

Figure 4.4 then highlights the significance of beam switches and the effect that regularization and optimal ordering of beam switches has on the system. If you consider the basic solution, labeled as $SwOrigMax$ then the maximum number of beam switches would be $T \cdot (m-1) + T$ which in this case is $20 \cdot (9-1) + 20 = 180$. So, when the system can meet the offered load easily then since the system is using every beam during every interval then it results in the maximum number of beam switches, as the results show. $SwPenMax$ represents what would be the maximum possible number of beam switches the regularized solution would use. The simulation takes $\sum\limits_{s=1}^{T}(m(s)-1)$ directly from the results and then adds T to it which would assume that there aren't any common transition beams between any of the intervals, s. Subsequently the theoretical minimum possible value, $SwPenMin$, is T less than the maximum value shown here, or 20 less, which is the case where you have a transition beam between every interval. As shown here the regularized solutions are much better than the original and the optimal beam switching makes even more of an impact, particularly at the lower loads. As the loads increase then all the solutions converge because less and less beams are being used as the system maximizes use of the high rate beams due to the overloading. Nevertheless, the benefit is clearly worthwhile as shown.

4.3 Multiple Hubs

At this point we extend this scheduling to a network of multiple hubs servicing the same set of users. We assume that we have K hubs where B_k represents the set of beams in hub k and $B = \cup_{k=1}^{K} B_k$. We assume that the hubs can establish connectivity amongst themselves and therefore establish an ad hoc network. We also assume that the routing of traffic within this network is handled at a higher level and here we focus on the scheduling of the resultant traffic. For further simplicity we say that N_C represents the set of all clients and N_H is the set of all hubs and $N = N_C \cup N_H$. We can express it with the following LP:

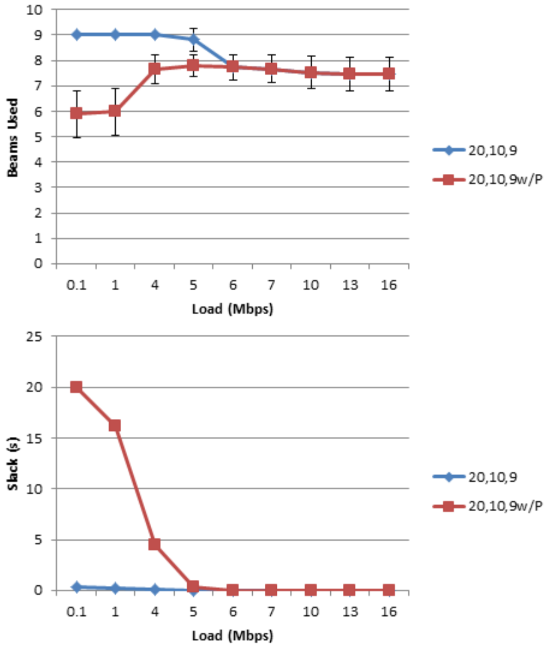


Figure 4.3: Beams used vs. offered load(top), slack vs. offered load (bottom).

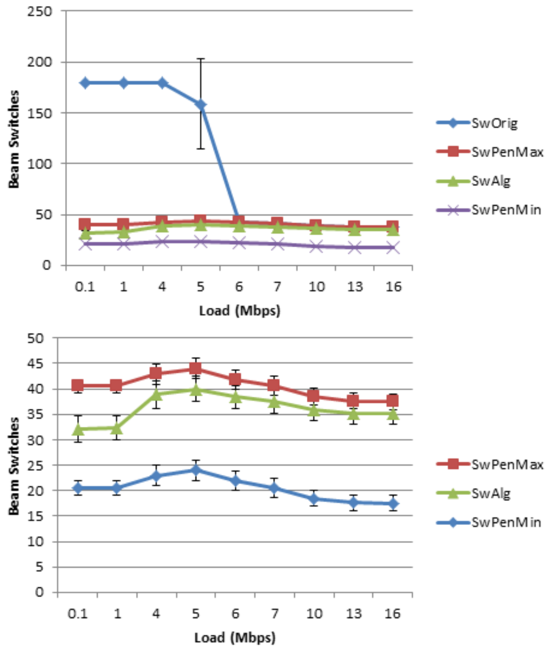


Figure 4.4: Beam switches vs. offered load. Top is all four cases, bottom focuses on the bottom three.

$$
\begin{aligned}
\max \ \sum_{s=1}^{T}\sum_{j\in N}\sum_{i\in B} p_j^i(s)\hat{c}_j^i(s)& \\
s.t.& \\
\sum_{s=1}^{T}\sum_{i\in B} p_j^i(s)\hat{c}_j^i(s) \le T\lambda_j \ \forall j \in N_c& \\
\sum_{s=1}^{T}\sum_{i\in B-B_k} p_k^i(s)\hat{c}_j^i(s) \le T\lambda_k \ \forall k \in N_H& \\
\sum_{j\in\{N-k\}}\sum_{i\in B_k} p_j^i(s) \le 1 \ \forall s,\ k \in N_H& \\
\sum_{i\in B} p_j^i(s) \le 1 \ \forall s,\ j \in \ N_C& \\
\sum_{i\in B-B_k} p_k^i(s) \le 1 \ \forall s,\ \forall k \in \ N_H& \\
p_j^i(s) \ge 0 \ \forall s,\ j \in N,\ i \in B.&
\end{aligned}
$$

The objective is to maximize the throughput across the entire network that includes the hubs themselves. As given, the overall constraint of the amount of traffic (flow) allotted to each user j over the entire period T is limited by their offered rate (first constraint). Similarly the amount of traffic allotted to each hub (via other hubs) over the entire period T is limited by their offered rate as well (second constraint). A hub can only service one client or other hub at a time using one beam at a time (third constraint). A user can only access one beam of one hub at a time (fourth constraint). A hub can only access one beam of one other hub at a time (fifth constraint). Finally, all time allocations must be greater than or equal to zero (sixth constraint). Note that interference between hubs is not addressed and could be solved by using separate frequencies for each hub. The regularizing penalties can also be added. The size of this problem is much larger, although the variables will likely be sparse and we can drive the results to be sparse as well, as shown above by incorporating penalties. In addition, the scheduler in this construct will require situational awareness regarding the parameters from all K hubs. Their interconnections could help facilitate this process.

Figure 4.5 represents simulation results for a multi-hub scenario. In this scenario their are 3 hubs. $T = 20$s, $n = 10$ and $m = 9$ with the conditions as presented in the multiple hub linear program. Once again each user's, and hub's, primary beam is chosen randomly $U(2, 9)$ at a random rate $U(24, 54)$ Mbps from each of the hubs. Their secondary beam is directly adjacent to their primary beam at a random rate $U(9, 48)$ Mbps. They all share the omni-directional on each hub at beam, 1, at a random rate $U(0, 24)$ Mbps. They all share the same fixed load, $\lambda_j = \lambda$ as given in the plotted results shown in Figure 4.5.

These results illustrate the capability to be able to extend this to a multiple hub model. The characteristics are the same as what we've shown so far to include the benefit of penalties and with the hubs able to support much more of a load between them, to include them being added as additional clients. In the same single-hub scenario the point at which the hub was able to support the load with equality was at $\approx$ 5 Mbps. In this scenario it was at $\approx$ 19 Mbps, more than 3 times that for a single hub.

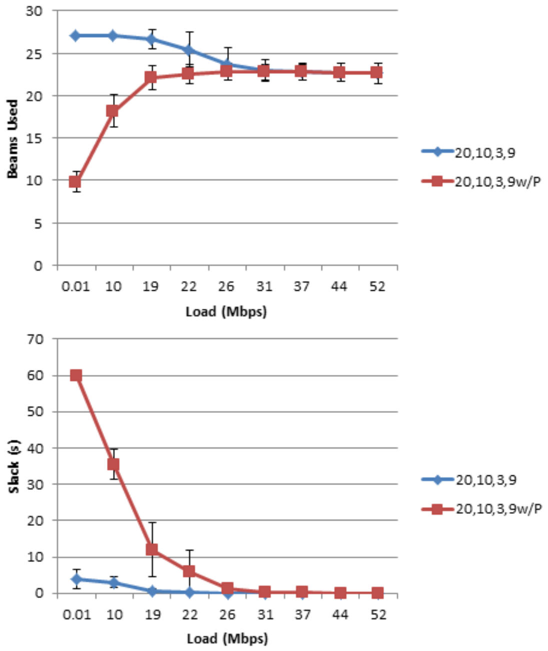


Figure 4.5: Beams used vs. offered load(top), slack vs. offered load (bottom). Beams used and slack is total across all three hubs.

4.4 Best Effort Options

In the case where a scenario cannot be scheduled then how it proceeds will generally be scenario dependent. If it determines that it does not have enough information regarding the clients' numbers, their rates, their offered loads, etc. then a full track might be necessary in order to find these details, or perhaps just a refined track to gain more details. In this case we will discuss these specifics in Chapter 6, Tracking. If these have been performed and no additional tracking will help then it is a matter of having the information but not being able to schedule such that all of the clients' demands are met. In this case we proceed to best effort options. Best effort options will also largely be scenario dependent. As mentioned in Chapter 3, fairness would be a key criteria. This best effort could be fairness based for scheduling in terms of throughput fairness. It could further be utility based where different users might place more or less value, or utility, on throughput. The problem then becomes a utility maximization problem. It could be made to be fair from a utility perspective. It could be variable where more value is placed on throughput received in a short amount of time due to real-time constraints or where higher priority users' utility is higher than others. It becomes a weighted utility maximization then. For instance, in a military type scenario if a schedule cannot be developed given the set of clients with their respective capabilities then users would likely be prioritized based on their importance to the mission. It could also relate to limiting the type of traffic being sent, where utility is assigned to the type of traffic. Or set to simply drop borderline low rate or low utility users. The parameters for scheduling could then be modified given what best effort option is chosen by the system administrator. These modifications might reflect an adjustment to the linear program. Or modifications might lend to a much more complicated integer linear program or perhaps non-linear and relying on bin-packing algorithms or other techniques. In these cases we will seek to relax them to be able to integrate them within the standard linear program construct or account for them as part of slack in the system.

4.5 Scheduling Conclusions

Up to this point we have taken a linear programming approach to optimally schedule client specific traffic in a mobile hub and spoke scenario using a beamforming antenna. We use it to maximize client throughput while considering delay constraints as applicable. We show how to incorporate l_1 norm regularization to account for non-linear factors such as beam-switching while still maintaining optimality. We then optimize the beam-switching sequence using a shortest path first approach to further mitigate the inter-switch delay and associated protocol specific overhead. We illustrate that this system can easily be extended to a networked system of multiple hubs. Finally, we present what might happen in a situation where a schedule cannot be developed, either due to not enough information being available or not enough resources being available to meet the demand. These factors are largely scenario dependent. Throughout the scheduling process, a key problem is obtaining the current possible rates and offered loads for clients as well as future estimates of what these rates and loads might be in the future. This transforms our scheduling into a scheduling with uncertainty problem as discussed in the next chapter.

Chapter 5

Scheduling with Uncertainty

5.1 Overview

As discussed in the last chapter, being able to “Attempt to Schedule” (see Figure 2.1) and determining whether a realistic schedule “Can be scheduled” using the linear programming relies on having accurate capacities, $c_j^i(s)$, and loads, $\lambda_j(s)$, for every time slot over the schedule period. Due to dynamics (channel and user mobility) the throughput changes over time while the traffic to be carried may show up in uncoordinated bursts such as the scenario we’ve discussed from Figure 1.10. We rely on tracking and also data exchanges to help obtain this information but tracking continuously or exchanging data with all of the users simultaneously is infeasible, nor is being able to forecast perfectly what those values will be into the future. This is a multiarmed bandit problem with restless bandits as discussed in Section 3.2.16 and presented by Scala et al. in [70]. As mentioned previously, it changes the scheduling problem into a scheduling with uncertainty, or under uncertainty, problem. Scheduling with uncertainty has been an area that has plagued military planners throughout history. It is a logistics challenge to be able to forecast what demands will arise and what resources will be available where and at what time.

To address the uncertainty associated with scheduling clients’ data exchanges due to channel and user dynamics, we start with a simple traffic intersection scenario. The results however can be extended to more complicated military type scenarios such as the UAV problem, convoy operations, logistics site operations, etc. In particular, we extend to the problem we present in this thesis.

We start with a linear program and perfect future information to create an optimal scheduling baseline [17]. This is computationally expensive and requires precise information far into the future. To simplify the problem and reduce the amount of future information, we introduce a receding horizon scheduler with which a schedule is re-computed in periodic time steps as presented in [58]. In this approach, based on information we have up to the current time, we schedule over a finite future window using a linear program, execute the first step of the schedule, then slide the window forward one step and repeat. The method is non-optimal since it does not include all future information. However, since it recomputes the schedule, as new information on the traffic and channels is gathered, it can get near optimal performance. Though this method has a shorter horizon and so is simpler and requires less future information, it still has a high computational load and requires detailed future information.

To simplify further, we use a receding horizon approach that aggregates future time periods into a small number of intervals. This reduces the complexity of the scheduling and allows us to use less precise future information. We study different permutations of the number of intervals and their size as a function of load. Surprisingly, good performance is possible in the simplest case that only considers the current time interval and an aggregate of all future bins. This greatly simplifies the linear program. Further, it suggests that gross estimates of performance and traffic into the future are sufficient to schedule. Though simple, this approach does significantly better than a simple greedy approach which looks only at current information. These results suggest that we can efficiently schedule communication traffic well using only imprecise information.

5.2 Problem Formulation

We first describe and analyze the problem with a single hub having a single beam, versus the multiple beams we've used previously for the purposes of this analysis. Similarly the hub communicates with n clients, each having a single arbitrary beam pattern. Once again, we consider time broken into a sequence of intervals, so that in interval s the hub can transfer to client j a capacity $c_j(s)$. Given that this is a mobile environment and given a dynamic radio environment then $c_j(s)$ will change over time for a particular client.

As before, data can be transferred between the hub and only one client at a time but the time in each interval can be divided among different clients. Let $p_j(s) \in [0, 1]$ be the fraction of interval s that the hub communicates with client j. Since only one client can communicate at a time $\sum_{j=1}^{n} p_j(s) \leq 1$. Further, $p_j(s)c_j(s)$ is the total data transferred to client j in interval s.

We still consider a planning period of the next T intervals. If the average data transfer per interval for client j is λ_j then the goal is to transfer $T\lambda_j$ for each client j. However, more or less than λ_j can arrive in each interval.

We assume again that the number of intervals, T, is sufficiently large to assist in scheduling. In this case, for the traffic intersection scenario T might be the maximum amount of time it could take for a vehicle to make it through the intersection area. Though we may plan over the interval T, the underlying scenario is continuous and may continue over an infinite horizon. Thus as a minimum the schedule must be updated as we reach the end of our planning horizon. We can update before reaching the planning horizon and in the limit as described below, we can update after each interval.

A key challenge to making this work is that there are n possible clients and although we are now only concerned with a single beam, it is still not possible to know with certainty the throughput in the future. Thus, at any given time we may only have estimates of the capacity that we can communicate with client j denoted $\hat{c}_j(s)$.

It is the job of tracking to maintain these estimates as discussed earlier where these estimates may come from direct link measurements, indirect measures (e.g. GPS coordinates and orientation of sender and receiver), or estimates from past measurements. In the latter case, a conservative estimate may degrade to zero over time at a rate that is a function of client mobility, radio dynamics, etc. Thus, given the degrading information over time and therefore increasing uncertainty, scheduling over a future planning horizon must tradeoff which user is scheduled in which order to ensure that the needs of each user are met.

So, once again, the problem is to schedule the communication over time such that the throughput of each client is maximized up to their respective offered rate.

5.3 Methodology and Initial Analysis

5.3.1 Scheduling with Uncertainty - an Example

A key issue regarding scheduling is the uncertainty with regards to estimating a client's future rate and offered load as mentioned. We therefore consider a simpler model and present how a receding horizon approach can perform with respect to this problem, versus having perfect knowledge and being able to solve optimally. Consider a traffic intersection in an urban city center as shown in Figure 5.1. We assume a time in the middle of the day where traffic is consistently coming and going in all directions and a traffic light is cycling between directions in $60s$ increments while assuming there are no arrow lights. The clients move at $13\frac{m}{s}$ or $\approx 30\frac{miles}{hr}$ and that traffic can be broken down into 12 specific paths as shown in Figure 5.2.

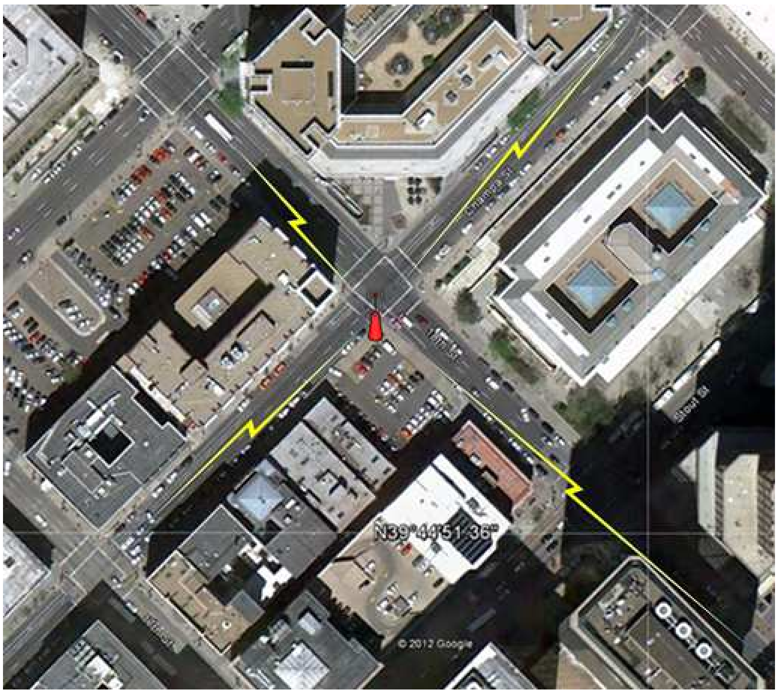


Figure 5.1: Central hub servicing ground clients in urban corridor

We assume that all of these paths are the same length and broken into intervals of $13m$ each as shown in Figure 5.3.

Each end is $143m$ from the middle of the intersection for a total of $286m$ from end to end.

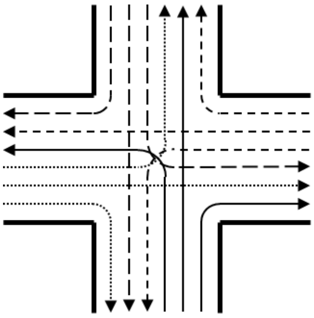


Figure 5.2: Possible Traffic Paths

Therefore, with no light a client could make it across the area in $22s$. The first light change is chosen randomly from between 0 to $60s$ with all subsequent light changes occurring each $60s$.

The time through the intersection depends on a combination of where the user is when the light changes and whether they are going straight or turning right or left. All users stop if they reach the intersection (designated simply as the midpoint) when the light is red. In addition, they wait an additional time depending on when the light turned red and the direction of travel (i.e. straight, right, or left). This represents the time to wait for pedestrians to clear the intersection, for cars ahead of them to start moving, and for them to accelerate.

For a straight client, if the light is red when they reach the intersection and it was red for less than $3s$ before then, then they are considered near the front of the waiting cars. When the light turns green they wait an additional exponential holding time with mean $7s$ and maximum $15s$. If the light was red more than $3s$ before reaching the intersection, they are further back in the line of waiting cars and they wait an exponential holding time with mean $10s$ and maximum $20s$.

For right-handed turns, the light doesn't necessarily affect them as they could turn right on red, but they would have to wait on traffic and pedestrians to clear. Regardless of the light state, right turning cars wait an exponential holding time of mean $10s$ (max $= 20s$). For left-handed turns the turner needs to wait for on-coming cars to clear. The turner always stops at the intersection and waits for an exponential holding time with mean $15s$ (max $= 30s$). If at the end of this time, the light is green then the car proceeds. If it is red, it waits until the light turns green and then

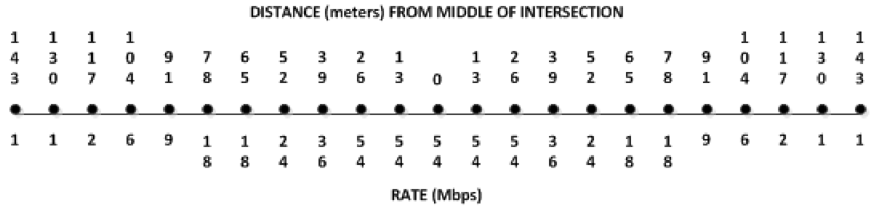


Figure 5.3: Intersection paths distance/rate intervals

waits another exponential holding time with mean $15s$ ($\max = 30s$).

Client cars arrive as a Poisson process at a given rate and are equally likely to enter one of the 12 paths through the intersection. We start the scenario with the intersection empty. We then allow it to fill up for a period of time, before subsequently allowing it to empty. The shortest amount of time a client can spend in the intersection is then $23s$ (we take both ends inclusively) and the longest amount of time would be in a left-turn scenario at $23 + 30 + 60 + 30 = 143s$. Therefore the full time period allows for the amount of time of generating clients and to ensure that the last client generated can spend the maximum amount of time. For instance if we generate clients for $100s$ then the full amount of time must be $\geq 243s$. We assume that each client j generates λT_j traffic while they are in the area where $\lambda_j = \lambda$ is common to all users for simplicity and T_j represents the amount of time that client j spends in the intersection area. The load is bursty. Client j generates T_j messages each of size λ_jMb. Each message is uniformly randomly and independently assigned to one of the T_j intervals. Thus some intervals will have one or more messages that arrive while others will have none. We let $\lambda_j(s)$ be the amount of traffic that arrives for user j in interval s.

To determine the rate $c_j(s)$, we compute the distance from client j to the intersection at time s, and compute the data rate assuming a 802.11g communication system combined with a Walfisch-Ikegami model (WIM) as it has been shown in [20]. The WIM has been shown to be a very accurate model for a dense urban environment. The capacities are calculated using the WIM for each $1s$ interval for each of the distances $143m$, $130m$, etc. down to the intersection itself at $0m$, as shown in Figure 5.3. Before a car enters the intersection area and after it leaves, the rate is 0. As a result, even in the best case, some clients will not be able to carry all their traffic if a large burst arrives in the final seconds before it leaves.

We consider the scheduling based on three situations. The **Prescient Overall (Pres. Over)** scheduling scenario assumes that the scheduler has perfect knowledge of all the clients, the path they take, their rate at each interval, and their offered load at each interval. This will provide a performance upper bound for our subsequent sub-optimal approaches. Given the uncertainty we will also then take a **Greedy Receding Horizon** approach that will send from the user that has the highest rate and has traffic to send. These two approaches serve as our competing performance baselines. The **Prescient Overall** has the advantage of having all information available to it for the entire timeframe and therefore uses a linear program to determine the optimal schedule to determine the most throughput the whole scenario can achieve with respect to the load offered. The **Greedy** approach takes a very simple local optimal approach without considering what might happen in the future.

Finally we will take variations of different receding horizon approaches. At each interval we take knowledge we gain for each user over the last interval. Since we know we are rescheduling after every interval then for future estimates we don't necessarily need to know their specific rate or load interval by interval or individual by individual for that matter. We don't necessarily need to be specific regarding the order in which users proceed or the arrival of their load but simply that their loads arrive at some point and in some amount during the duration of their time in the intersection. Furthermore, we don't know how long a user will stay in the intersection, but we do know that some users will stay longer than the expected duration and some will stay shorter and their rates will be reflected accordingly. We therefore schedule based on bins. In each of these bins we will take either a single realization or varying combinations/aggregates of future estimates. We describe these in detail below. To show the basic approach we first return to the **Prescient Overall** scheduling and its respective linear program.

5.3.2 Linear Program

We present a Linear Program for the scheduling problem for this scenario as follows:

$$GIVEN: \ T, n, \{\lambda_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\}, \{c_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\};$$

$$OBJECTIVE: \ \max \sum_{s=1}^{T} \sum_{j=1}^{n} p_j(s) c_j(s)$$

$$s.t.$$

$$\sum_{k=1}^{s} p_j(k) c_j(k) \leq \sum_{k=1}^{s} \lambda_j(k) \ \forall j, s$$

$$\sum_{j=1}^{n} p_j(s) \leq 1 \ \forall s$$

$$p_j(s) \geq 0 \ \forall j, s$$

The objective seeks to maximize the overall throughput of all the clients over the entire timeframe.

The first constraint in the program makes sure that the clients cannot have throughput up to interval s that exceeds their offered load up to interval s, a slight modification from before that considered the period as a whole. The second constraint makes sure that only one client can communicate at a time.

T serves as the number of intervals for which you want to schedule over. Our key problem though, versus a standard Linear Program (in a case of having prescient knowledge), is that we don't know the capacity $c_j(s)$ or the specific load $\lambda_j(s)$ in future intervals, s; hence the uncertainty. Although the load for client j might average λ_j over a period T, it could come in bursts, continuous, or some combination thereof. Such problems are considered no less than NP-complete [56] and likely NP-hard, [26]. Our approach is to schedule based on the current realization and to form groups of aggregate bins (collections of intervals) for a long period of time out. We will illustrate how a simple scenario of scheduling based on the current instance and aggregates of the remaining T will perform almost optimally and much better than a greedy approach.

Consider our previous linear program. Our aggregate scheduling will be depicted in the form of **R1,RExK,Ragg**. This is interpreted as a bin for the current interval (**R1**), followed by K bins each that aggregate non-overlapping blocks of E slots (**RExK**), followed by one bin that aggregates the remaining $T - E \cdot K - 1$ intervals (**Ragg**). Note that this yields $K + 2$ bins. In the linear program this is equivalent to planning over $T = K + 2$ intervals where for each user, each interval's capacity is the sum of the capacities during that interval and each interval's load is the sum of the loads that arrive during that interval. If there are n_{now} clients that are in the system in the current interval, the other clients will not be scheduled during the current interval. However these clients will influence the scheduling of the n_{now} clients who are present now. Since the details of the other client's future scheduling is not important, they can be combined into a single aggregate other user.

Thus, given **R1,RExK,Ragg**, we can formulate the inputs to the linear program as follows with $T = K + 2$ time intervals and $n = n_{now} + 1$ clients.

$$GIVEN: \; T, n, K, E, \; \{\lambda_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\}, \{c_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\};$$

$$OBJECTIVE: \; \max \sum_{j=1}^{n} [p_j(1)c_j(1) + \sum_{k=1}^{K} (p_j(s = k+1)(\sum_{e=1}^{E} c_j(s = ((k-1)E + e + 1)))) + \\ p_j(s = K+2)(\sum_{s=KE+2}^{T} c_j(s))]$$

We take a 'fisheye' approach. The original constraints still apply. The first instance, $s = 1$ is based on real knowledge of the client's current possible rate and offered load. For subsequent time instances we might not have perfect knowledge, but we likely have fairly good estimates. For those aggregates of fixed size, we take K groups of size E of a client's estimated rate aggregated into one interval. We take the expected load for that same period aggregated in a similar manner. In other words, $\lambda_j(k+1) = \sum_{e=1}^{E} \lambda_j((k-1)E + e + 1)$. Finally for the remaining interval, it is just an aggregate of all remaining rates and respective expected loads. This reduces the complexity of a linear program that normally does not scale well, particularly for longer periods of time.

Furthermore, the load and capacities can be more easily estimated over those long periods of time. If we have a larger number of users and a longer period of time then we can take advantage of the law of large numbers [75] by using expectations for future estimates. By aggregating then, although individual users might deviate, the grouping as a whole will follow more closely to the expected values. By taking a receding horizon approach, a future schedule isn't acted on, beyond the next interval, and so any loss in fidelity due to the aggregation is resolved as the scheduler steps into the future. In our scenario, future loads used in the receding horizon scheduling are based on the client's expected offered load per interval.

To return to our 'fisheye' approach, a specific example with respect to our notation might be **R1,R5x4,Ragg**. This schedule has 6 bins. The 1^{st} bin uses a realization of the current rate for all n clients as well as their current offered load. The next four subsequent bins are each then aggregates of each client's 5 sequential rate and respective offered load estimates. Estimates need to be determined for these rates and loads but since they are aggregated then not as much fidelity is needed for a specific rate/load combination for a specific interval. The final bin is then just an overall aggregate of the remaining rates and respective loads up until time T. We assume for the equation given that T is larger than $1 + KE$. We analyze how this approach performs in a traffic scenario as presented earlier.

5.3.3 Simulation

Given the vehicle intersection mentioned earlier we run sets of simulations for a Poisson arrival rate with mean $4\frac{clients}{sec}$. We allow for clients to arrive for a period of $30s$. This requires a full T of up to $173s$ for all the users to eventually clear out. This simulation is limited by the size of problem that MATLAB could solve for the most complex case of the **Prescient Overall** scheduler. This computes an entire schedule from the intersection area being empty, filling up and subsequently emptying out later.

We run an extensive battery of simulations over increasing λ that includes the **Prescient Overall**, the **Greedy**, and then 'fisheye' versions starting from an $R1, Ragg$ version up to initial tests using interim fisheye sizes of $E = 20$ and up to $K = 6$ groups of them. We also analyze a widening fisheye approach where you take the first realization, the second one is then an aggregate of intervals $2-3$, the third is an aggregate of intervals $4-6$ and so on up to a bin of size 12. These fisheye permutations are shown in Table 5.1 along with the percentage of the offered load that each configuration meets. For space considerations the trailing $Ragg$ for each permutation is omitted. Also $R11+$ in the 3 bin row denotes that permutations from $R1, R11, Ragg$ to $R1, R31, Ragg$ were tested.

Increasing λ by increments of $.1Mbps$ we find that the **Prescient Overall** and the **Greedy** deviate from each other the most where $\lambda = .9Mbps$. We find that at this point the upper bound, using the **Prescient Overall** is 83.96% of the offered load (recall that even under the best situations, not all load can be carried if it arrives in the final seconds before a client leaves the intersection area). The **Greedy** method meets only 70.91% of the offered load. We then determine the best performer for the amount of bins used. We find that the simple $R1, Ragg$ method using only 2 bins can meet 80.53% of the load. For 3 bins $R1, R11, Ragg$ meets 82.02%. For 4 bins, $R1, R8x2, Ragg$ meets 82.62% and for 5 bins $R1, R8x3, Ragg$ meets 82.88% of the load. As the number of bins increase, all of the results remain around the 83% mark with none doing better than 83.08% for 7 bins and none worse than 82.88% for 3 different scenarios. With added bins comes added complexity and a need for more precise estimation with respect to the rates and offered loads. Considering this and the negligible gains beyond using 5 bins we therefore take the binning permutations listed above and see how they perform over the entire range as shown in Figure 5.4 Over the range of results we see in the top plot that all of the 'fisheye' or bin methods generally follow the optimal **Prescient Overall** with a gap existing between them and the **Greedy** just below the $\lambda = 1$ mark. The lower plot detail in the figure shows the disparity more closely. As presented earlier, the largest gap exists around the $\lambda = 0.9Mbps$ mark where the **Greedy** deviates by about 13 percentage points whereas the simple 2 bin $R1, Ragg$ deviates by only about 3.4 percentage points here and never more than 3.6 percentage points.

This result shows that for an easily extended scenario we can run a linear program using two time intervals. The first interval takes realized information for the data rate and offered load as observed every horizon. The second interval simply takes aggregate information for a client's future. Furthermore, the timeframe could be extended out as long as necessary. In this case you might know some users will hit a traffic light and wait an expected amount of time. While other users will not have to wait and pass on through. Similarly loads may be bursty, but over a long period the total load will approach its expected value. As shown in this case, the cost is only about 3.6% at worst. If the scheduler has more accurate estimates of the future, it can narrow the gap with the optimal by using additional bins. However, given these results, the cost/benefit should be considered.

A receding horizon approach incurs more complexity, however, given these aggregate bins, then this complexity is significantly reduced. Although these results were used for a specific traffic scenario, the fact that the **Greedy** approach deviates from the optimal by a significant amount shows the benefit of our approach and how it could easily be extended to similar scheduling with uncertainty type problems.

5.4 Conclusion

In this chapter we have shown a simple receding horizon type approach to address the uncertainty associated with scheduling clients' data exchanges due to their mobility. We show that by using a receding horizon approach with a linear program that aggregates future time periods into a small number of intervals not only reduces the complexity of the scheduling but also allows us to use less precise future information.

We found that good performance is possible in the simplest case that only considers the current time interval and an aggregate of all future bins. This greatly simplifies the linear program (and scheduling more generally) presented in the previous chapter. Further, it suggests that gross estimates of performance and traffic into the future are sufficient to schedule. Though simple, this approach does significantly better than a simple greedy approach which looks only at current information. These results suggest that we can efficiently schedule communication traffic well using only imprecise information. Though our simulations used aggregates of precise future information, we expect the performance to be similar with estimated future information using for example, average trajectory lengths and loads measured as the scenario progresses.

Although applied to a simple traffic intersection these results can be extended to more complicated military type scenarios such as the UAV problem, convoy operations, logistics site operations, as well as this work. We therefore proceed to tracking for which to develop those capacity estimates for scheduling client communication given these relaxed requirements.

Table 5.1: Permutations of Fisheye Methods (M) with Bins Used (each has a trailing *Ragg*) and percentage of offered load met

Bins	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
2	R1,Ragg 80.53										
3	R1x2 80.37	R1,R2 80.71	R1,R3 81.0	R1,R4 81.28	R1,R5 81.54	R1,R6 81.64	R1,R7 81.89	R1,R8 81.96	R1,R9 81.93	R1,R10 81.93	R1,R11+ ≤82.02
4	R1x3 80.55	R1,R2x2 81.07	R1,R3x2 81.62	R1,R4x2 82.16	R1,R5x2 82.4	R1,R6x2 82.5	R1,R7x2 82.6	R1,R8x2 82.62	R1,R9x2 82.38	R1,R10x2 82.43	R1-R3 81.29
5	R1x4 80.79	R1,R2x3 81.69	R1,R3x3 82.25	R1,R4x3 82.58	R1,R5x3 82.71	R1,R6x3 82.71	R1,R7x3 82.81	R1,R8x3 82.88	R1,R9x3 82.74	R1,R10x3 82.43	R1-R4 82.27
6	R1x5 80.96	R1,R2x4 82.18	R1,R3x4 82.52	R1,R4x4 82.68	R1,R5x4 82.91	R1,R6x4 82.96	R1,R7x4 83.06	R1,R8x4 82.06	R1,R9x4 80.88	R1,R10x4 80.8	R1-R5 82.62
7	R1,R2x5 82.64	R1,R3x5 82.63	R1,R4x5 82.85	R1,R5x5 83.08	R1,R6x5 82.91	R1,R7x5 81.28	R1,R8x5 81.04	R1,R9x5 80.86	R1,R10x5 80.81	R1-R6 82.74	
8	R1x7 81.65	R1,R2x6 82.74	R1,R3x6 82.69	R1,R4x6 82.98	R1,R5x6 83.01	R1,R6x6 81.39	R1,R7x6 81.26	R1,R9x6 80.75	R1,R10x6 81.12	R1-R7 83.22	
9	R1x8 81.9	R1,R2x7 82.78	R1,R3x7 82.78	R1,R4x7 83.12	R1,R5x7 81.45	R1,R6x7 81.4	R1-R8 82.08				
10	R1x9 82.11	R1,R2x8 82.77	R1,R3x8 82.83	R1,R4x8 82.41	R1,R5x8 81.44	R1-R9 81.62					
11	R1x10 82.41	R1,R2x9 82.74	R1,R3x9 82.96	R1,R4x9 81.68	R1,R5x9 81.47	R1,R6x9 81.29	R1-R10 81.63				
12	R1x11 82.52	R1,R2x10 82.82	R1,R3x10 82.94	R1,R4x10 81.7	R1,R5x10 81.44	R1,R6x10 81.32					
13	R1x12 82.62	R1,R2x11 82.83	R1,R3x11 82.82	R1,R4x11 81.7							
14	R1x13 82.63	R1,R2x12 82.84	R1,R3x12 81.65	R1,R5x12 81.52							
15	R1,R2x13 82.98	R1,R3x13 81.67									
16	R1x15 82.75	R1,R2x14 83.02	R1,R3x14 81.72								
17	R1,R2x15 82.93	R1,R3x15 81.76	R1,R4x15 81.64								
18	R1x17 82.76	R1,R2x16 82.83	R1,R3x16 81.72								
19	R1x18 82.85										
20	R1x19 82.85	R1,R2x18 81.83									
21	R1x20 82.85										
22	R1x21 82.89	R1,R2x20 81.86	R1,R3x20 81.71								
23	R1x22 82.92	R1,R2x21 81.94									
24	R1x23 82.99	R1,R2x22 81.87									
25	R1x24 83.04										
26	R1x25 83.02	R1,R2x24 81.83									
27	R1x26 83.04	R1,R2x25 82.17									
29	R1x28 83.06	R1,R2x27 81.94									
30	R1x29 83.19										
31	R1x30 83.24										
32	R1,R2x30 81.88										
41	R1x40 83.56										
51	R1x50 83.81										
61	R1x60 83.93										
63	R1x62 83.95										
64	R1x63 83.96										
66	R1x65 83.96										

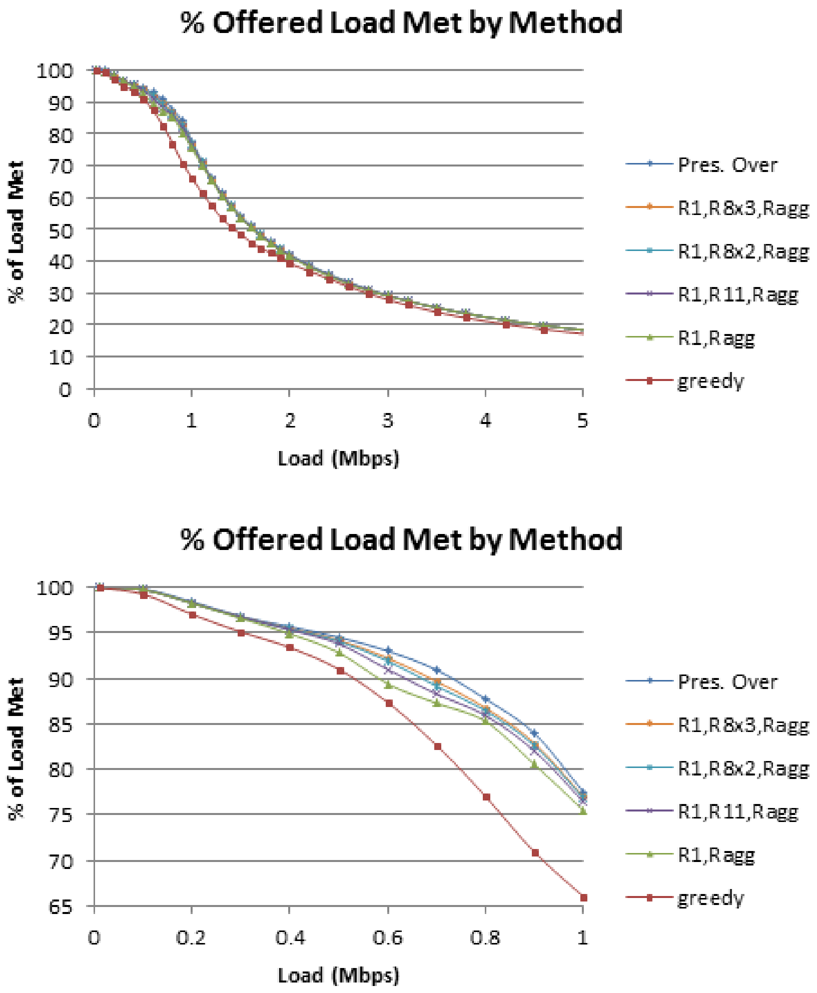


Figure 5.4: Percentage of offered load met for different methods. The bottom graph is a detail of the top graph.

Chapter 6

Tracking

We return to "TRACKING" as shown in Figure 2.1 and what was introduced in Chapter 2. Let $i_j^*(s)$ denote the optimal pattern for client j at time s, i.e.

$$i_j^*(s) = \arg\max_i c_j^i(s).$$

We've discussed that tracking could be done continuously or at discrete time instances but tracking continuously is infeasible due to the overhead required in comparing signal quality values depending on the method being used and it leaves no time for clients to exchange data. The hub is likely servicing more than one client at a time and must divide its tracking efforts between them. As well, the appropriate tracked beam pattern to be used for a particular client is only important during the points in time during which the hub and client are supposed to be communicating.

As mentioned, an appropriate tracked beam pattern should, as a minimum, be one that provides sufficient signal quality (e.g. Received Signal Strength Indicator (RSSI) or Signal-to-Noise Ratio (SNR)) at both ends to allow the hub and the client to reliably exchange information. Ideally, a beam pattern that provides the strongest signal would be preferred. By Shannon's Channel Capacity Theorem [19], using this beam will allow for the client to transmit with the largest data rate possible. Since, with our hardware, transmit and receive frequencies are the same, it can be assumed by reciprocity that the beam pattern that provides the strongest receive value is ideally the one that provides the strongest transmit value (receive value for the distant end) as well.[1] A client

[1] Differences between the radio's receiver and transmitter characteristics could cause some asymmetry in the channel in each direction.

will likely remain able to be reached for some time under a particular beam pattern depending on characteristics such as the transmit power, antenna gain, beamwidth and mobility. These were the primary reasons for which we broke the period up into discrete intervals. Whatever the strategy, we denote by $\hat{c}_j^i(s)$ as the estimated capacity of beam pattern i if used by client j at time s as returned by the tracking strategy.

Tracking is very dependent on the hardware and protocols being used. As presented in Chapter 2, the purpose of tracking is to develop capacity estimates that can be used to schedule client communication. If successful then the quality and quantity of tracking is adequate given those criteria. Therefore considering our use of the Phocus Array FCI-3100X and the 802.11 protocol we discuss the background and subsequently the methods for which we used to successfully track.

6.1 Tracking Implementation

Extensive laboratory and field experiments have been conducted utilizing the Phocus Array. This includes some close collaboration along with field tests with fellow graduate students in Aerospace (flight tests) and with some in Computer Science, see namely [1][25][84]. We first describe our early background tracking tests and then proceed to how we track with respect to our current method as used in our prototype configuraton.

6.1.1 Background Work

Related to this work specifically, simple tracking techniques tested have included:

- Location aware: A client, using a BU-353 GlobalSat® GPS "puck" would run a script that would parse its GPS coordinates from the NMEA strings provided by the puck. It and the hub would already be connected (associated via 802.11 in this configuration) and have an open socket connection. In one configuration the client would forward its GPS coordinates to the hub and the hub, knowing its own static GPS coordinates and orientation would, via a script, automatically beamform towards the client. Similarly, in another technique we modified a client's probe request that would repeatedly advertise its GPS coordinates. This would allow the hub (via scanning through the different beams) to hear and know which beam (based on direction) to use to point towards the client. In yet another configuration the client would know the hub's static GPS coordinates (as well as the hub's orientation) along with its own GPS coordinates and determine the appropriate beam the hub should use to beamform towards it. It would then command beamform the hub over the socket connection. The drawback of these techniques is that they rely on GPS and that the best radio direction is based on geographic direction.

- Location unaware: This assumes that the hub and the client know of each other but don't know if they're in vicinity of one another or their general direction. Once again, given our equipment, we are using 802.11. An 802.11 access point sends out a beacon frame every $100ms$ by default. It does not require a response from clients that can hear it. In our configuration we wish to initially track clients from the set of known n clients and so we want clients to respond when they hear the hub's beacon frame. We set up the hub with a unique SSID. This would then automatically be broadcasted in its beacon frame. The hub would then cycle through all the default beams, waiting a conservative $250ms$ in each beam direction. Therefore the beacon frame would be broadcasted at least twice for every default beam direction. At the same time, each client would then run a script where, using *iwlist*, it would scan the access points in the area. If that unique SSID would show up then, using *iwconfig* it would automatically try and associate. It worked successfully in the lab and then worked out in the field in a configuration similar to the tracking scenario presented next. This technique is lengthy and doesn't necessarily identify the best beam, but it identifies a beam that is good enough to allow the client to be able to associate. In the process the hub and the client have established an initial contact. They also don't rely on external capabilities such as GPS or another separate overhead channel nor rely on the geographically based direction being the ideal direction. For this method then further or refined tracking would be used to find a possibly better beam pattern to use for the client. This successfully implemented technique illustrates that the array could be configured to be able to initially track clients from the set of n clients.

- Full Track and Dithering Track: We proceed to the basis for what is used in our prototype implementation. To find an initial acceptable beam for them the array would then cycle iteratively through all of the default beams. At each beam it would *fping* the IP range where it would broadcast an abbreviated *icmp* request to all of the IP addresses in that range. Each user that successfully receives the message would reply and the array would then note their RSSI value, if received. After cycling through all the beams and having all of the RSSI values for the reached users at each beam direction then the script would identify the initial best beam for each client. This completes what we call a "full track." The array, via the script and in sequence for each client, then compared their current beam's RSSI value versus the value of the beams to the left and the right. This is a "dithering track" similar to what is shown in Figure 3.4. It then updates the best beam (according to the beam with the highest RSSI value) to use for each of them at each iteration. The key advantage of this technique is that this is all performed by the array itself without having an external device controlling it.

6.1.2 Current Implementation

In this previous implementation the tracking was conducted on the array itself, as mentioned. The scheduler relies on the information obtained from tracking. Therefore due to the complexity involved in scheduling, as well as the other tasks presented in Figure 2.1, it became necessary to control the tracking from a laptop connected directly to the hub. The laptop could then directly manage all aspects of the implementation. The entire process would be coded in C and obtaining the RSSI data as well as controlling the beamforming of the array could all be done with the array over separate socket connections that would remain open at all times. The method for initial full tracking of the users and subsequent tracking (dithering) updates would remain the same, but now initiated from the laptop and specifics exchanged over the socket connections.

For our prototype implementation a full track will be done initially followed by a dithering track for each of the clients to refine their respective beam. Then we proceed with d receding horizon schedules where only the first interval for each schedule being executed. After d schedules then a dithering track will take place for each of the clients. Then on every w dithering tracks a full track is done. This then repeats with another set of schedules.

Chapter 7 will discuss full implementation configurations and results regarding integration with other aspects of Figure 2.1.

6.2 Further Considerations

6.2.1 Future Estimates

Throughout this work and as shown here we have evaluated and tested models for the purpose of developing an efficient system that will provide adequate estimates of $c_j^i(s)$ for scheduling. This model could be very simple with very little overhead, as long as we can develop a schedule. Estimating the $c_j^i(s)$ values may be better thought of by developing a range of likely values. The lower range of values could then be used for "guaranteed" scheduling and the upper range would be used for "is it even possible" scheduling, see Figure 6.1. Depending on the scenario (mobility and radio environment characteristics specifically), if the $c_j^i(s)$ can be characterized via a distribution, then Monte Carlo techniques could be used to determine the likelihood a schedule is met. This would involve taking numerous sets of samples of the $c_j^i(s)$ values using the distribution (or up to second moment at a minimum), running a schedule for each of these sets of values, and then determine the likelihood of a successful schedule. Techniques to estimate these $c_j^i(s)$ values could include being provided GPS coordinates, trajectory information, etc. for the clients. Actual measurements could be taken, similar to common training and tracking methods, by sending or receiving active probes of known signals in order to estimate path loss exponents, fading effects, etc. for future

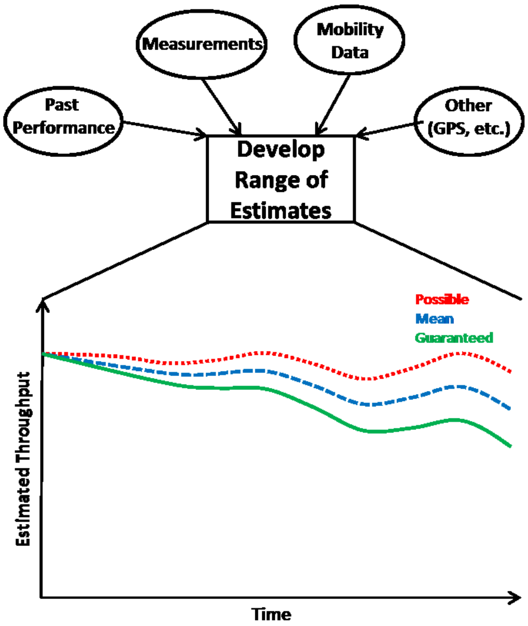


Figure 6.1: Estimating future throughput ranges

transmissions. Future $c_j^i(s)$ values could then be estimated using Kalman filter or other methods. We have already simulated or implemented versions of these techniques as discussed earlier and in Chapter 7.

Another consideration in developing these estimates relates to the time period, T, used. For short T, these estimates and subsequent schedules could be developed with more confidence, however this would create more overhead. This is due to time involved for the method or process being done to update these estimates along with processing and possible additional protocol overhead. This would have to be done more often, resulting in less time for data exchanges. A longer T would require less overhead but the quality of the estimates would degrade over the length of the period. A tradeoff exists. However, if we consider the scenario mentioned of the UAV circling an objective area, then it could learn and continue to refine its knowledge of the radio environment (e.g. an RF map) within the area as it circles[79]. This would help it to be able to establish better estimates over a longer period of time. Such is the method we used in our final implementation as discussed in the next chapter. In order to develop estimates of each of the user's possible throughput rates we had them repeatedly follow the path that they would take through the objective area while capturing what their rates are as they made their way along the path. We then determined an average along the path.

If a schedule can't be developed based on the method being used then the model should be able to determine if it is a result of not enough information regarding the estimates or simply that a schedule is not even possible given the best case estimate scenario. Given the former, it attempts to gain more information, using perhaps one or more of the above techniques, in order to obtain a better set of estimates. Given the latter, then it has determined that a successful schedule is not possible. At this point it becomes another scheduling problem to determine what the best effort options are. Nevertheless, refining and further optimizing the tracking process and developing more accurate future estimates remains an area of future work here and throughout the research community. However, given our earlier results in Chapter 5, Scheduling with Uncertainty, the cost benefit of using complex estimation schemes versus rougher estimates should be weighed depending on the scenario. The added complexity might not be worth it.

6.2.2 Opportunistic Feedback

Another aspect of tracking is with regards to opportunistic feedback. Depending on the model, protocol, etc. being used during data exchanges, some side information should be able to be obtained. As a simple example, we could assume that a link should be able to exchange traffic at 10 Mbps but find that during the data exchange it can only do 5 Mbps or even worse, can't exchange at all due to an equipment malfunction. Information is gained from this and can be used to analyze why execution failed, if it fails, or even if successful then updating estimates. This can then be used to establish better estimates when it returns.

In our implementation, in between tracking iterations and during data exchanges with a client, we obtain the RSSI for the client at that instance from the array and map that value to a rate, see Table 3.4. We then use that as the current interval rate for that client in the receding horizon scheduling as discussed more fully in the next chapter.

6.2.3 Slack Time Options

"Slack time options" as shown in Figure 2.1 include tracking down new users, choosing a set of alternate patterns to explore for clients, addressing maintenance issues, etc. These factors and considerations will largely be implementation dependent but a model developed could serve as a basis for other scenarios.

In our implementation we choose to use slack time to track and refine our estimates for our current clients. We also find it necessary to do a full track periodically where we iterate through all of the beam directions to make sure that the beam being used for a client is appropriate, and to find additional clients. In a situation where users are close or where you have a lot of multipath, there could be instances where a beam is chosen that is appropriate at that particular time but could be where the client is in the sidelobe of a beam pointing in a completely different direction or is a reflected signal. With the dithering method used then the tracking could be "wandering aimlessly" and never recover the general “best beam” based on the client’s general direction with respect to the array. This full track will allow the tracking (dithering) process to regain its “bearings” if necessary.

6.3 Conclusion

For tracking, we find that it is more scenario dependent and build off concepts presented in related work and in preliminary field experiments. Our main method is similar to the dithering method that is used to track satellites. The criteria for tracking, as discussed, should allow for a schedule to be developed to meet the clients’ needs. This implementation does that and if not then it conducts steps to further refine the tracking that was done. This includes a dithering track or a full track to regain a client’s “bearings”, or find additional clients. Our tracking mechanism also uses the data exchanges to obtain additional information opportunistically in order to gain better estimates.

Tracking completes the final major component of our problem as illustrated in Figure 2.1. We proceed to show how we fully implemented a prototype version of it in Chapter 7 - Prototype Implementation.

Chapter 7

Prototype Implementation

7.1 Introduction

The culmination of this work serves to provide a working implementation of Figure 2.1. The scenario involves two mobile clients exchanging data with two static clients at the hub location. See Figure 7.1 The two pairs represent the two conversations that will be ongoing in the scenario.

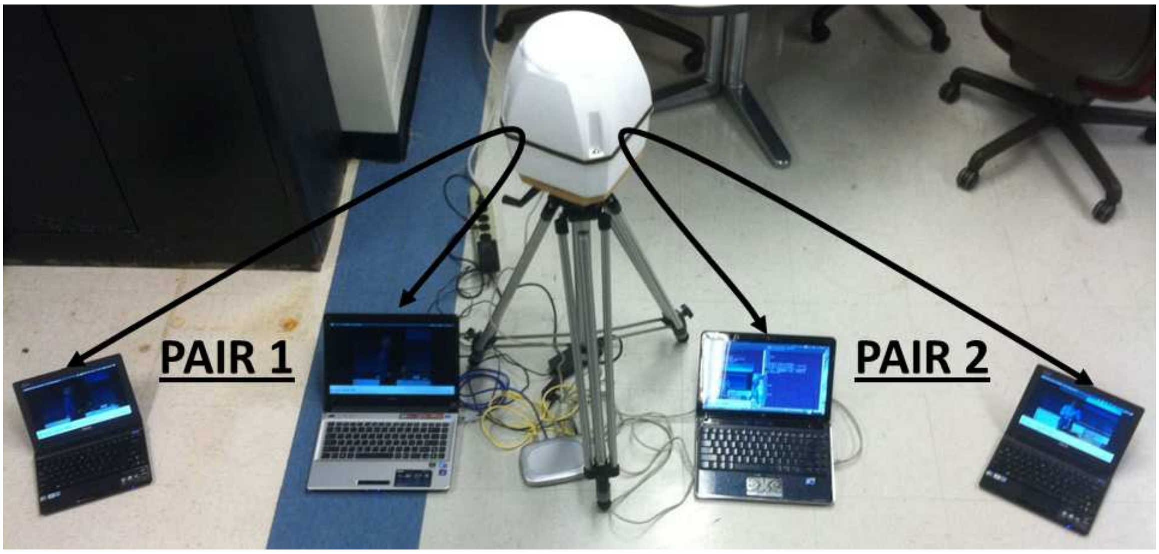


Figure 7.1: Final Implementation set-up

In the depiction the two clients closest to the phased array are directly connected via ethernet to a switch directly connected to the array while the two clients at the ends will be connected wirelessly via the array. The conversations consist of each of the hub location clients streaming a separate video lecture to the respective wireless client. *Pair 2's* hub location client also is running the *C* script that is controlling the antenna. The wireless clients will then move in separate directions (clockwise/counter-clockwise) at walking speeds along the paths as shown in Figure 7.2. *Pair 1* has the shorter path. His farthest distance away is $\approx 140m$ away at the top right corner of his path as shown in Figure 7.2, while *Pair 2's* farthest distance away is $181.1m$ as shown. In the final configuration, of which details will be provided going forward, the intent was to not use omnidirectional at all and if so to show that it would fail. In the array's 16 directional beam default configuration, as given earlier, it meets the demands and the video streams remain reliably up during the duration of following the paths. We turned down the array's transmit power as low as possible to where the transmit/receive modules would still work correctly. This was at roughly $-9dBm$[1] and the video streams would reliably fail with the omnidirectional antenna pattern at distances of no more than $100m$ away from the array. We compare these settings against how it should perform theoretically. Using a basic path loss model, as given in [68], with a transmit power of $-9dBm$, cable losses a liberal $-6dB$, antenna gain of $2.1dB$, a wireless client antenna gain of $5dB$ and a receiver sensitivity of $-91dBm$ at 2 $Mbps$ and a free space loss of $-83.1dB$, then the limit should be $140m$. Therefore, given this liberal calculation and the results then omni-directional is not feasible in this scenario. In the same configuration, but using the directional beams and their $15dBi$ gain, the calculation returns a $10.64dB$ remaining margin out to the full $181.1m$ distance. The transmit power will remain at $-9dBm$ for our implementation.

The period for scheduling is $T = 414s$. This was based off of the average time it would take a client to follow *Pair 2's* path over a repeated number of runs. During these runs we ran a tracking sequence every second, as presented in the last section, for the length of each of the paths. We also used a TCP traffic generator to constantly exchange data with the clients during their separate runs. For *Pair 1's* path the average rate (during exchanges) was $\bar{c}_1 = 9.45Mbps$ and *Pair 2's* path was $\bar{c}_2 = 6.3Mbps$. The streaming video using the *VLC Media Player* is essentially a constant load assumed equal for both clients. We assumed a high load estimate of $1.75Mbps$. Furthermore, to create different load dynamics we set *Pair 1's* configuration to be an *RTP - Real Time Protocol*

[1] based on modification of the txpower settings

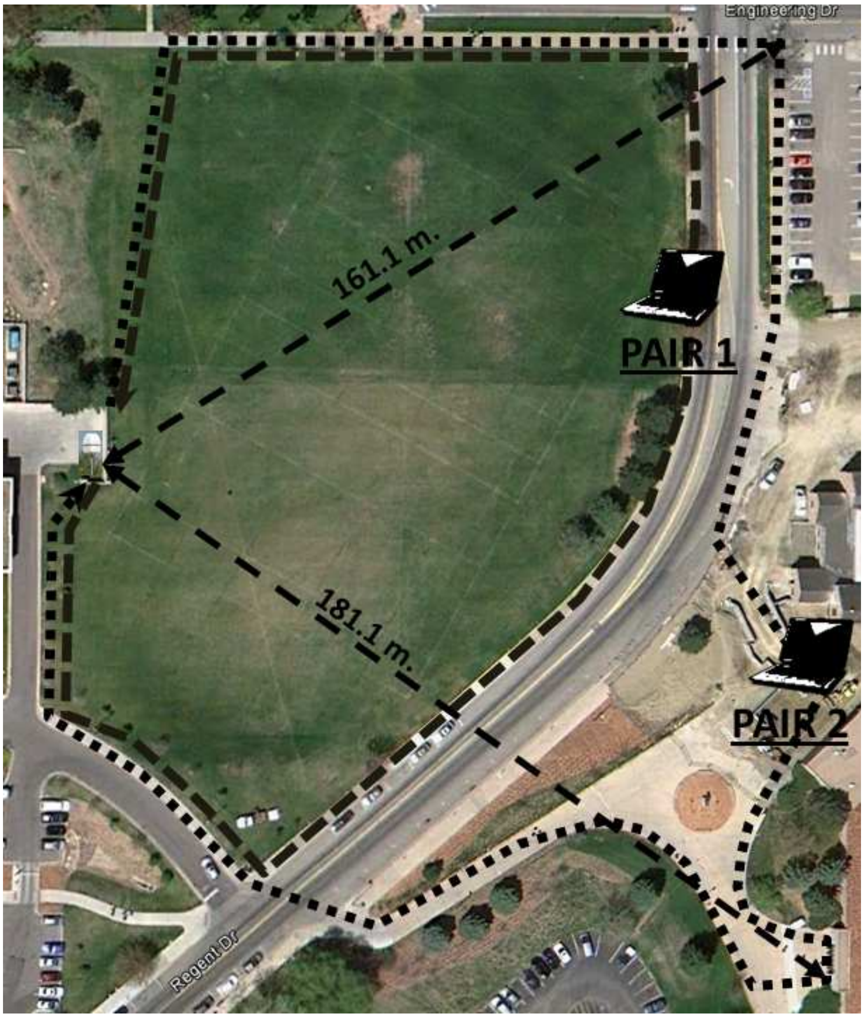


Figure 7.2: Final Implementation client paths

feed that uses the *UDP - User Datagram Protocol* with some additional enhancements for jitter and out of sequence issues. For *Pair 2's* feed we set it up as an *http* feed with a $20s$ buffer. These differences in load dynamics will factor into the scheduling later on.

For the remainder of this chapter we step through Figure 2.1 as it relates to our implementation.

7.2 Input

The input is as we have presented in the Introduction. It is $n = 2$ wireless clients having conversations with two directly connected clients. The array uses any of its $m = 16$ default directional beams.

7.3 Tracking

Tracking follows the implementation presented in Section 6.1.2 where $d = 4$ and $w = 20$. Given this configuration, in testing we find that a full track averages $2.5 \pm .1s$. If we *fping* a range of five IP addresses, with only these two clients available, then this jumps to $4.4s$ (due to timeout times) and subsequently to $6.6s$ with ten IP addresses. A dithering track averages just over $300ms$ per client. Therefore, implementation using the *fping* polling mechanism is expensive and can further be optimized as a topic of future work.

In this prototype implementation, four receding horizon schedules are done and the first interval executed (up to one second each) for every dithering track and twenty dithering tracks (four schedules each) were done for every full track. Therefore a dithering track was conducted at least every $4.6s$ and a full track at least every $93.9s$. These times assume every interval is used up and the clients share no beams involved in the dithering tracks as this would reduce the time needed.

7.4 Attempt to Schedule/ Best Effort Options

7.4.1 Attempt to Schedule

The tracking and opportunistic feedback would provide the rates that the scheduler would use as discussed in Chapter 6. The scheduling follows a receding horizon approach that is done every second. For these two clients we set up a linear program. The objective is to maximize the throughput for the two clients for the next interval and a subsequent interval consisting of aggregates for the remainder of the period. Using a receding horizon, this entire schedule is calculated every second but only executed for each second (the next or immediate interval), or $p(s)$ as depicted here. Therefore $p(s+1)$ is always calculated but never executed. The constants are $\lambda_j = 1.75Mbps\ \forall j$ and $T = 414s$. The number of clients here is $n = 2$. The size of a client j's buffer at interval s is $b_j(s)$. The linear program then is as follow:

$$\begin{aligned}
GIVEN:\ & T, n, \{\lambda_j | j = \{1, ..., n\}\}, \{b_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\}, \\
& \{c_j(s) | j = \{1, ..., n\}, s = \{1, ..., T\}\}, \{\bar{c}_j | j = \{1, ..., n\}\}; \\
OBJECTIVE: \max\ & \sum_{j=1}^{n} [p_j(s)c_j(s) + (T-1)p_j(s+1)\bar{c}_j - p_j(s)] \\
& s.t. \\
& p_j(s)c_j(s) \leq \lambda_j + b_j(s)\ \forall j, s \\
& p_j(s)c_j(s) + (T-1)p_j(s+1)\bar{c}_j \leq T\lambda_j + b_j(s)\ \forall j, s \\
& \sum_{j=1}^{n} p_j(s) \leq 1\ \forall s \\
& p_j(s) \geq 0\ \forall j, s
\end{aligned}$$

The schedule is based on two intervals, s and $s+1$. The first constraint shows that for the next (immediate) interval, s, the client's throughput cannot exceed the new offered load as well as what exists in the buffer (up to the buffer's maximum size). Since the first client doesn't have a buffer then $b_1(s) = 0\ \forall s$ and any load that isn't carried in the last interval is then lost. The second client has a $20s$ buffer and so he can accumulate up to $(20\lambda)Mb$ worth of load. The second constraint then represents that a client's throughput over the entire period cannot exceed the offered load plus the size of the buffer. Therefore the second interval $s+1$ represents an aggregate of the remaining $T-1$ intervals. Without loss of generality we drop references to the beams used since we assume that a client only uses one beam per interval anyways and they are especially not relevant after the immediate interval (since we only execute the immediate interval). The third constraint then is that only one client (using only one beam) can be scheduled at any given point in time. It should also be noted that an interval can be broken up but the sum of the allocations can be no more than $1s$ (the length of the interval in this implementation). The fourth constraint is straightforward.

The rates for the first interval for each client were based on the last updated value for a client based on a track or opportunistic feedback. Opportunistic feedback would be used based on data exchanges, in between tracking iterations. After each data exchange we obtain (via the socket connection) what the array sees as the RSSI value for that client given the recent exchange. That value would then be used after being mapped to a rate. If a track was just conducted then the rate based on that result would then be used. We follow the UAV scenario presented earlier that we assume a constant circling of an objective and therefore for the second interval we simply use the sum of the average rate over the remaining intervals in the period. As given previously, the average rate over T for client j is depicted as $\bar{c}_j$.

7.4.2 Can it be Scheduled / Need more Tracking

With regards to "Can it be scheduled," in our implementation we assume that the scheduler will always have enough information to be able to schedule it. We provide the scheduler a very conservative load value for each client (to represent the load of the streaming video) and default values for a client's rate are zero if not obtained otherwise. If it still cannot be scheduled then we assume that the system just cannot meet the offered load, whether having more tracking or not. This is in regards to "Need more tracking." As mentioned earlier, tracking is conducted periodically in our implementation anyways.

7.4.3 Best Effort Options

Therefore if the system cannot schedule to meet all of the offered load then it will execute any schedule maximizing the throughput and subject to the constraints. No user is given any priority.

7.5 Execute Schedule / Slack Time Options

7.5.1 Execution

The schedule is based on a 1s interval mentioned earlier. Since we are doing a receding horizon approach then all we need is the $p_j(s) \; \forall \; j$ for the next interval s. The actual schedule implementation here then is based on a physical diversity approach. To execute at each s the system will point the antenna towards each client (based on their most recent updated best beam) for their respective proportion of the second allocated to them.

Streaming video provides a tangible method of determining whether this implemenation works or not. Execution of this implementation was successful. Execution of the script and of the schedules themselves had the net result of both streams remaining up, without any drops, during the entire time that the script was running. In fact, during the instances where the script ended before shutting it off, then one of the clients would immediately lose the stream if they didn't have a buffer and if they did then they would lose it shortly thereafter. This was a result of the script stopping and then the array remaining on the last beam that it was pointing towards - one or the other client would be isolated. For further verification, during the execution of each schedule the script would write to a file how much time was allocated to which client using which beam. During each run the system generally followed the path that each client was taking and allocated times consistent with the client's distance from the array during the course of them running the path. Variations occurred due to fading, multipath, etc. but the results are consistent for what was expected. The following Figure 7.3 shows the sequence of beams used by each client over the intervals for the duration of the test. It can be seen that the clients follow opposite (clockwise) directions as one has increasing beam numbers and the other decreasing. The top left of Figure 7.4

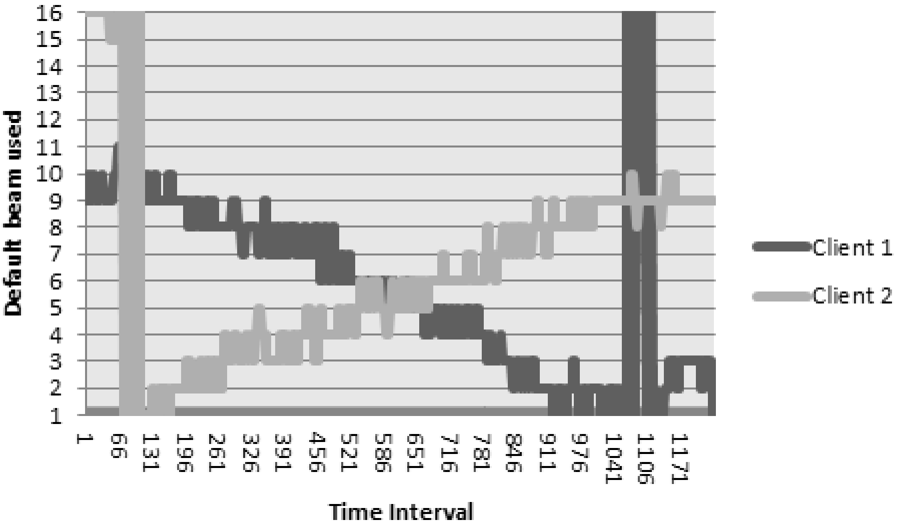


Figure 7.3: Beam used per client per interval

is then a stacked graph that shows the amount of time allocated to each client for each interval over the duration of the test, as well as any slack that is remaining. You'll notice a fixed number of allocation values used. This is a direct reflection of the fixed discrete values in 802.11 that the RSSI values can be mapped to as well as the assumed fixed constant fluid flow type of load expected every s, $1.75 Mbps$ here. It was also verified that at each instance the sum of the allocations were ≤ 1.

7.5.2 Performance vs. Other Schemes

Subsequently we tested our system against other implemented schemes. They consisted of:

- Greedy - The greedy algorithm would choose the client that had the highest rate for the interval. If the clients had the same rate then it would randomly pick one or the other.
- Greedy with Load Constraint (Greedy w/L) - Similar to greedy but this method would prefer the higher rate user up to fulfilling the conservative load of $1.75 Mbps$ as given earlier. At that point it would switch over to the other user up to the same point or until the interval is used up.

- Proportional Greedy (GreedyPr)- This method would allocate to each user the same proportion of the 1s as their proportion of combined rate. For example, if one user has a rate of 6 Mbps and the other user has a rate of 1 Mbps then the former would get $\frac{6}{7}s$ and the later would get $\frac{1}{7}s$ allocated for that interval.

Figure 7.4 highlights our results. Our Linear Programming scheme outperforms them all.

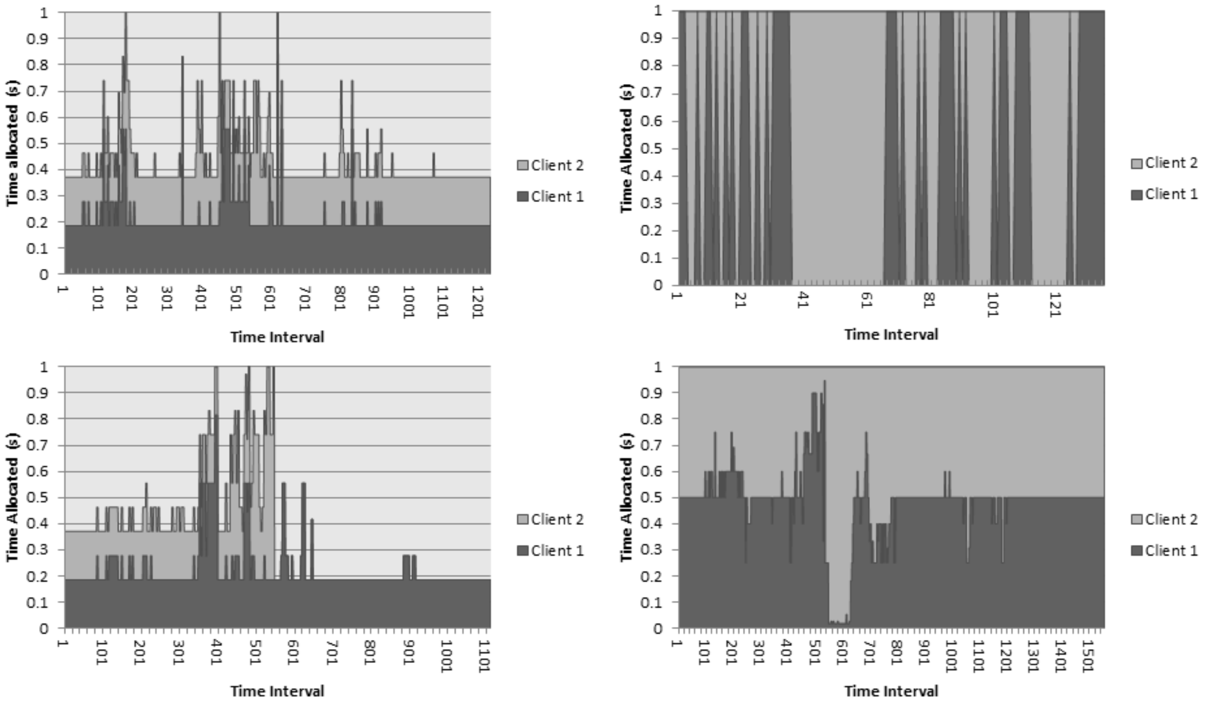


Figure 7.4: Allocations per Client per Interval: Clockwise from top left: LP Scheme, Greedy, GreedyPr, Greedy w/L

Both feeds remain up during the duration of the entire test, as earlier. Furthermore, as the figure shows, it maximizes the amount of slack that exists in the schedule in order to allocate for tracking, maintenance, etc. The *Greedy* scheme was run multiple times but continued to fail early on into the tests as the results show. Furthermore, it uses up the entire slot resulting in no slack. The *Greedy w/L* scheme performed fairly well and had more slack in the system but would fail once the *Pair 2* client got to the farther location of the path. The misallocation and lack of balance (current vs. future) in fulfilling the loads repeatedly resulted in the *Pair 2 (http configuration)* dropping the feed and not re-gaining it once it had used up its 20s buffer. The point at which this occurs is at roughly interval 550 in Figure 7.4. Finally, practically the *GreedyPr* performed the best of these other schemes. It would lose the feed briefly at the farther locations but then was usually able to regain it. However, this method has no slack. The using up of slack time whether needed or not helped the struggling client recover the feeds as it was easier for it to make up any back log. Nevertheless, based on performance and available slack time, our linear programming scheme easily outperforms these other implementations.

7.5.3 Execution Successful

The overall execution was successful then. Following each schedule execution, if it was successful then there was inherently slack in the system and our implementation would automatically proceed to slack time options, discussed next. If it wasn't successful then the system would still proceed, however it gains opportunistic feedback (to refine or update estimates for the current beam). This would then be used for the next schedule or the system would proceed to track anyways and perhaps also find a better beam. In the case of the client with a buffer then the buffer would help account for any deficiency in meeting the requisite offered load. This highlights additional tools that can be used to further refine this implementation and help account for instances for which the execution is not successful, or not completely successful.

7.5.4 Slack Time Options

Based on our implementation as discussed earlier, our slack time options consisted of being able to move to the next step quicker. This means moving to developing and executing the next schedule, for which our horizon window was shorter in that instance, or moving immediately to a full track or dithering track based on the timing sequences discussed before.

7.6 Conclusions

In conclusion this implementation has shown that the problem overview as provided by Figure 2.1 provides a robust roadmap for key components needed to develop a reliable electronic beamforming antenna system that can service dispersed users in a dynamic environment. This implementation also illustrates the ability to create it given the tools we have developed, refined, and integrated with respect to tracking, scheduling, and practical modifications. Along with this success we also identified that the tracking process can be somewhat expensive however this cost could be mitigated if implemented fully on the array.

Chapter 8

Conclusions and Future Work

8.0.1 Conclusion

This work addressed a general slow mobile wireless hub and spoke scenario where the hub's antenna has electronic switched beam antenna capability and the clients have only a fixed antenna capability. The eventual goal was to implement this on a hardware radio system described in Chapter 3. From Figure 1.10, the general problem was to find an efficient way for a hub (or possibly multiple hubs) to service its clients successfully. More specifically, we wanted to be able to transfer the data for each client up to their offered load given that only one client/beam pair can be serviced at any given point in time.

We formulated the general problem around Figure 2.1. It does not present a solution but simply breaks the question down into smaller manageable components while presenting the criteria necessary in order to be able to proceed from one component to the next. These components were then collectively separated into scheduling, tracking, and execution.

We took a linear programming approach and used it to maximize client throughput while considering delay constraints as applicable. We showed how to incorporate l_1 norm regularization to account for non-linear factors such as beam-switching while still maintaining optimality and maximizing slack time. We then optimized the beam-switching sequence using a shortest path first approach to further mitigate the inter-switch delay and associated protocol specific overhead and help create additional slack time. We illustrate that this system can easily be extended to a networked system of multiple hubs. Finally, we present what might happen in a situation where a schedule cannot be developed, either due to not enough information being available or not enough resources being available to meet the demand. These factors are largely scenario dependent. Throughout the scheduling process, a key problem is obtaining the current possible rates and offered loads for clients as well as future estimates of what these rates and loads might be in the future. This transformed our scheduling into a scheduling with uncertainty problem.

We started with a simple receding horizon type approach to address the uncertainty associated with scheduling client data exchanges due to their mobility. We showed that by using a receding horizon approach with a linear program that aggregates future time periods into a small number of intervals not only reduces the complexity of the scheduling but also allows us to use less precise future information. We found that good performance is possible in the simplest case that only considers the current time interval and an aggregate of all future bins. This greatly simplifies the linear program (and scheduling more generally). Though simple, this approach does significantly better than a simple greedy approach which looks only at current information. These results suggest that we can efficiently schedule communication traffic well using only imprecise information. We extend these results to our prototype implementation as presented in Chapter 7.

We find that tracking is more scenario dependent and build off concepts presented in related work and in preliminary field experiments. Our main method is similar to the dithering method that is used to track satellites. The criteria for tracking, as discussed, should allow for a schedule to be developed to meet the client's needs. Our implementation does that and if not then it conducts steps to further refine the tracking that was done. This includes a dithering track or a full track to regain a client's "bearings," or find additional clients. Our tracking mechanism also uses the data exchanges to obtain additional information opportunistically in order to gain better estimates to use for subsequent schedules where a track might not take place.

Finally, our prototype implementation shows how all of these areas are integrated in a full implementation of a solution to the problem. Our implementation shows that the problem overview as provided by Figure 2.1 provides a solid roadmap for this problem. This implementation also shows that one can be developed given the tools we have developed, refined, and integrated.

8.1 Future Work

Tracking is a key area that can be further optimized. Our scheduling with uncertainty results illustrated that rough estimates could perform fairly well and one must consider the tradeoff of complexity associated with more complex estimation schemes versus the additional performance gained. However, our use of *fping* as a probing mechanism in conjunction with the tracking mechanism we present is expensive and could be further optimized. Not only the process, timings, sequences, etc. for which it is conducted but also the specifics of an *icmp* polling mechanism as well as a socket connection that both use multiple layers of the OSI stack. Tracking when implemented on the array would reduce some of this overhead. Nevertheless this process can be further coupled and optimized.

With regards to scheduling we would introduce a control loop within the scheduling process to help account for variability of the load. In our prototype implementation we assumed a constant flow of up to an upper bound for the load but introducing a control loop would provide increased fidelity in scheduling in dealing with the variability that it actually exhibits.

Bibliography

[1] Eric William Anderson. Integrated Scheduling and Beam Steering for Spatial Reuse. PhD thesis, University of Colorado at Boulder, 2010.

[2] Matthew Andrews and Lisa Zhang. Scheduling over a time-varying user-dependent channel with applications to high-speed wireless data. Journal of the ACM, 52(5):809–834, 2005.

[3] S. Bandyopadhyay, K. Hasuike, S. Horisawa, and S. Tawara. An adaptive MAC protocol for wireless ad hoc community network (WACNet) using electronically steerable passive array radiator antenna. IEEE Global Telecommunications Conference, 2001. GLOBECOM '01, pages 2896–2900, 2001.

[4] Lichun Bao and J. J. Garcia-Luna-Aceves. Transmission scheduling in ad hoc networks with directional antennas. In MobiCom '02 Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, 2002.

[5] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. Mathematical Programming, 99(2):351–376, 2004.

[6] Paul Bender, Peter Black, Matthew Grob, Robert Padovani, Nagabhushana Sindhushayana, and Andrew Viterbi. CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users. IEEE Communications Magazine, pages 70–77, 2000.

[7] Daniel Bennett and Timothy Brown. Wireless data scheduling of clients serviced using uav mounted beamforming antennas. In Proceedings AIAA Infotech @ Aerospace Conference, 2012.

[8] D. Bertsimas and J. Ni no Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. Operations Research, 48(1):80–90, 2000.

[9] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, New York, 2004.

[10] Michael Buettner, Eric Anderson, Gary Yee, Dola Saha, Anmol Sheth, Douglas Sicker, and Dirk Grunwald. A phased array antenna testbed for evaluating directionality in wireless networks. In MobiEval '07 Proceedings of the 1st International Workshop on System Evaluation for Mobile Platforms, pages 7–12, 2007.

[11] Antonio Capone, Ilario Filippini, and Fabio Martignon. Joint routing and scheduling optimization in wireless mesh networks with directional antennas. In The IEEE Communications Society's ICC Proceedings, pages 2951–2957, 2008.

[12] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized energy efficient broadcast for wireless networks with directional antennas. In Proc. Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET 2002), 2002.

[13] Lijun Chen, Steven H. Low, and John C. Doyle. Random access game and medium access control design. IEEE/ACM Transactions on Networking, pages 1–14, 2010.

[14] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Computer Networks and ISDN Systems, 17(4):1–14, 2005.

[15] Romit Roy Choudhury and Nitin H. Vaidya. Deafness: A MAC problem in ad hoc networks when using directional antennas. Technical report, Dept. of Computer Science, Dept. of Electrical and Computer Engineering, and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 2003.

[16] Romit Roy Choudhury, Xue Yang, Ram Ramanathan, and Nitin H. Vaidya. Using directional antennas for medium access control in ad hoc networks. In MobiCom '02 Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, pages 59–70, 2002.

[17] Vašek Chvátal. Linear Programming. W. H. Freeman and Company, New York, 1983.

[18] Fidelity Comtech. Phocus Array System Manual, 2009.

[19] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory Second Edition. John Wiley and Sons, Inc., Hoboken, NJ, 2006.

[20] J.-C. Cousin C.Q. Hien, J.-M. Conrat. Propagation path loss models for lte-advanced urban relaying systems. In IEEE International Symposium on Antennas and Propagation (APSURSI), pages 2797–2800, 2011.

[21] Hongning Dai, Kam-Wing Ng, and Min-You Wu. An overview of MAC protocols with directional antennas in wireless ad hoc networks. In ICCGI '06 Proceedings of the International Multi-Conference on Computing in the Global Information Technology, page 84, 2006.

[22] J. Deng and Z. Haas. Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks. In ICUPC, 1998.

[23] Arthur C. Densmore and Vahraz Jamnejad. A satellite-tracking K and Ka band mobile vehicle antenna system. IEEE Transaction on Vehicular Technology, 42(4):502–513, 1993.

[24] Alexandra Duel-Hallen, Shengquan Hu, and Hans Hallen. Long range prediction of fading signals: Enabling adaptive transmission for mobile radio channels. Technical report, North Carolina State University, 2000.

[25] J. S. Elston, J. Roadman, M. Stachura, B. Argrow, and E. Frew. The tempest unmanned aircraft system for in situ observations of tornadic supercells: Design and vortex2 flight results. Journal of Field Robotics, 28:461–483, 2011.

[26] J. Zhao F. Ordóñez. Robust capacity expansion of network flows. Networks Journal, 50(2):136–145, 2007.

[27] Qixing Feng, Joe McGeehan, Eustace K. Tameh, and Andrew R. Nix. Path loss models for air-to-ground radio channels in urban environments. In Vehicular Technology Conference. VTC2006, pages 2901–2905, 2006.

[28] Michael P. Fitz. Fundamentals of Communications Systems. The McGraw-Hill Companies, Inc., New York, 2007.

[29] John A. Fleming. The Principles of Electric Wave Telegraphy. Longmans Green, London, 1906.

[30] M.P. Friedlander. Exact regularization of linear programs. Technical report, Department of Computer Science Tech. Rep., University of British Columbia, 2005 (revised 2006).

[31] Matthew S. Gast. 802.11 Wireless Networks: The Definitive Guide. O'Reilly and Associates, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2002.

[32] G. Grimaud, A. Honore, H. Meunier, and D. Simplot-Ryl. Efficient packet scheduler for wireless ad hoc networks with switched beam antennas. In 20th International Conference on Advanced Information Networking and Applications, 2006. AINA 2006., pages 523–527, 2006.

[33] Bruce Hajek. Notes for ECE 534 an exploration of random processes for engineers. University of Illinois, Urbana-Champaign, 2008.

[34] Daniel Henkel. The Impact of Controlled-Mobility Helper Nodes on the Performance of Challenged Networks. PhD thesis, Technical University of Ilmenau and University of Colorado at Boulder, 2009.

[35] Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research, 165:289–306, 2005.

[36] Ju-Lan Hsu and Izhak Rubin. Performance analysis of directional CSMA/CA MAC protocol in mobile ad hoc networks. IEEE Communications Society, pages 3657–3662, 2006.

[37] Xiaoxia Huang, Jianfeng Wang, and Yuguang Fang. Maximum flow problem in wireless ad hoc networks with directional antennas. Optimization Letters (2007) Springer-Verlag 2006, 1:71–84, 2007.

[38] Zhuochuan Huang, Chien-Chung Shen, C. Srisathapornphat, and C. Jaikaeo. A busy-tone based directional MAC protocol for ad hoc networks. In Proceedings MILCOM 2002, volume 2, pages 1233–1238, 2002.

[39] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Technical report, Digital Equipment Corporation Eastern Research Lab, 1984.

[40] Gentian Jakllari, Wenjie Luo, and Srikanth V. Krishnamurthy. An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas. IEEE Transactions on Wireless Communications, 6(3):1–12, 2007.

[41] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of cdma-hdr a high efficiency-high data rate personal communication wireless system. In Vehicular Technology Conference Proceedings VTC2000, 2000.

[42] Feng Jiang and A. Lee Swindlehurst. Optimization of UAV heading for the ground-to-air uplink. Technical report, Department of Electrical Engineering and Computer Science, University of California, Irvine, 2011.

[43] E. A. Jorswieck, P. Svedman, and B. Ottersten. Performance of TDMA and SDMA based opportunistic beamforming. IEEE Transactions on Wireless Communications, 7(11):4058–4063, 2008.

[44] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. Palgrave Macmillan Journals on behalf of the Journal of the Operational Research Society, 49:237–252, 1998.

[45] Murad Khalid, Xuan Hung Le, In ho Ra, and Ravi Sankar. ADMAC with integrated destination discovery for ad hoc networks. IEEE Communications Letters, 14(8):704–706, 2010.

[46] Leonard Kleinrock. Queueing Systems Volume I: Theory. John Wiley and Sons, Inc., New York, 1975.

[47] Leonard Kleinrock. Queueing Systems Volume II: Computer Applications. John Wiley and Sons, Inc., New York, 1976.

[48] Young-Bae Ko, V. Shankarkumar, and N. H. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In INFOCOM 2000. IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings, pages 13–21, 2000.

[49] Thanasis Korakis, Gentian Jakllari, and Leandros Tassiulas. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. In MobiHoc '03 Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pages 98–107, 2003.

[50] V. Krishnamurthy and R.J. Evans. Hidden markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking. IEEE Transactions on Signal Processing, 49(12):2893–2908.

[51] Vikram Krishnamurthy, Bo Wahlberg, and Frank Lingelbach. A value iteration algorithm for partially observed markov decision process multi-armed bandits. Mathematics of Operations Research, pages 133–152, 2005.

[52] Arvind Kumar and Rajeev Tripathi. Impact of antenna beam joining on resource utilization in wireless mesh networks using smart antenna. In Int'l Conf. on Computer and Communication Technology ICCCT'10, pages 377–382, 2010.

[53] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. MANET simulation studies: The incredibles. Mobile Computing and Communications Review, 9(1):50–61, 2005.

[54] Dhananjay Lal, Rishi Toshniwal, Rajesh Radhakrishnan, Dharma P. Agrawal, and Jr. James Caffery. A novel mac layer protocol for space division multiple access in wireless ad hoc networks. In 11th International Conference for Computing Communications and Networks, 2002.

[55] Dongho Lee and Kwangsue Chung. RA-MAC: An energy efficient and low latency MAC protocol using RTS aggregation for wireless sensor networks. In 2008 IEEE International Conference on Advanced Technologies for Communications, pages 150–153, 2008.

[56] Zukui Li and Marianthi Ierapetritou. Process scheduling under uncertainty: Review and challenges. Computers and Chemical Engineering, 32:715–727, 2008.

[57] Xiaojun Lin, Ness B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. IEEE Journal on Selected Areas in Communications, 24(8):1452–1463, 2006.

[58] Jacob Mattingley, Yang Wang, and Stephen Boyd. Receding horizon control - automatic generation of high-speed solvers. IEEE Control Systems Magazine, pages 52–65, 2011.

[59] Kamesh Medepalli and Fouad A. Tobagi. Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. IEEE Communications Society for Proceedings IEEE INFOCOM, 2006.

[60] Katta G. Murty. Network Programming. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1992.

[61] Woongsoo Na, Laihyuk Park, Gunwoo Lee, Sungrae Cho, Seung Eun Hong, and Woo Yong Lee. Performance analysis of circular directional MAC protocol. 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN), pages 183–186, 2010.

[62] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. 2000 IEEE Wireless Communications and Networking Conference, 2000. WCNC 2000, 3:1214–1219, 2000.

[63] Omar A. Nasr, Mihaela van der Schaar, and Babak Daneshrad. A unique beamforming-based equilibrium in multi-user random access SIMO networks. IEEE Communications Letters, 14(2), 2010.

[64] Vishnu Navda, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel, and Samir Das. Mobisteer: Using steerable beam directional antenna for vehicular network access. In Proceedings of the 5th International Conference on Mobile Systems, Applications and Services. MobiSys '07, pages 192–205, 2007.

[65] Daniel P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. IEEE Journal on Selected Areas in Communications, 24(8):1439–1451, 2006.

[66] Jaeok Park and Mihaela van der Schaar. Stackelberg contention games in multiuser networks. EURASIP Journal on Advances in Signal Processing, 2009:1–30, 2009.

[67] Kishore Ramachandran, Ravi Kokku, Karthikeyan Sundaresan, Marco Gruteser, and Sampath Rangarajan. R2d2: Regulating beam shape and rate as directionality meets diversity. In Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services. MobiSys '09, pages 235–248, 2009.

[68] Theodore S. Rappaport. Wireless Communications: Principles and Practice, Second Edition. Prentice-Hall, Inc., Upper Saddle River, NJ 07458, 2002.

[69] Simon R. Saunders and Alejandro Aragon Zavala. Antennas and Propagation for Wireless Communication Systems Second Edition. John Wiley and Sons Ltd, The Atrium Southern Gate, Chichester, West Sussex, PO19 8SQ, England, 2007.

[70] B.F. La Scala and B. Moran. Optimal target tracking with restless bandits. Digital Signal Processing, 16:479–487, 2006.

[71] Harkirat Singh and Suresh Singh. A MAC protocol based on adaptive beamforming for ad hoc networks. In The 14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings, pages 1346–1350, 2003.

[72] Harkirat Singh and Suresh Singh. Smart-802.11b MAC protocol. 2004 IEEE International Conference on Communications, 6:3684–3688, 2004.

[73] Harkirat Singh and Suresh Singh. Tone based MAC protocol for use with adaptive array antennas. Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, 2:1246–1251, 2004.

[74] IEEE Computer Society. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

[75] Henry Stark and John W. Woods. Probability and Random Processes with Applications to Signal Processing Third Edition. Prentice Hall, Inc., Upper Saddle River, NJ 07458, 2002.

[76] Karthikeyan Sundaresan, Kishore Ramachandran, and Sampath Rangarajan. Optimal beam scheduling for multicasting in wireless networks. In Mobicom '09, 2009.

[77] Mineo Takai, Jay Martin, Aifeng Ren, and Rajive Bagrodia. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. MOBIHOC '02, 2002.

[78] Naishuo Tian and Zhe George Zhang. Vacation Queueing Models Theory and Applications. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA, 2006.

[79] N. Wagle and E. Frew. Spatio-temporal characterization of airborne radio frequency environments. In IEEE GLOBECOM Workshops (GC Wkshps), pages 1269–1273, 2011.

[80] P. Whittle. Restless bandits: Activity allocation in a changing world. Journal of Applied Probability, 25:287–298, 1988.

[81] Wayne L. Winston. Operations Research: Applications and Algorithms Third Edition. Wadsworth Inc., Belmont, CA 94002, 1994.

[82] Evşen Yanmaz, Robert Kuschnig, and Christian Bettstetter. Channel measurements over 802.11a-based UAV-to-ground links. In IEEE GLOBECOM-WiUAV, Houston, TX, 2011.

[83] Yonggang Yao and Yoonkyung. Another look at linear programming for feature selection via methods of regularization. Technical report, Department of Statistics, The Ohio State University, 2010.

[84] Gary Vincent Yee. Ordering and Combinatorial Effects of Wireless Optimizations in Beam Forming 802.11 WLANs. PhD thesis, University of Colorado at Boulder, 2010.

[85] Yung Yi, Junshan Zhang, and Mung Chiang. Delay and effective throughput of wireless scheduling in heavy traffic regimes: Vacation model for complexity. In Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing. MobiHoc '09, pages 55–64, 2009.

[86] H. Zhang, Y. Jiang, K. Sundaresan, and S. Rangarajan. Wireless data multicasting with switched beamforming antennas. Technical report, NEC Labs America, 2010.

[87] Honghai Zhang, Yuanxi Jiang, Karthik Sundaresan, Sampath Rangarajan, and Baohua Zhao. Wireless data multicasting with switched beamforming antennas. Mini-Conference an Proceedings IEEE INFOCOM, pages 526–530, 2011.

[88] Z. Zhang. Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas. In IEEE ICC, 2005.

[89] Zhensheng Zhang. DTRA: Directional transmission and reception algorithms in WLANs with directional antennas for QoS support. In IEEE Network, 2005.

[90] Shengli Zhou and Georgios B. Giannakis. How accurate channel prediction needs to be for transmit-beamforming with adaptive modulation over rayleigh MIMO channels. IEEE Transactions on Wireless Communications, 3(4):1285–1294, 2004.