Bootstrapping Cognitive Radio Networks

Brent Horine
University of Central Florida

BOOTSTRAPPING COGNITIVE RADIO NETWORKS

by

BRENT H HORINE
B.S. Electrical Engineering, Marquette University, 1983
M.S. Electrical Engineering, University of Central Florida, 1990

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida

Fall Term
2012

Major Professor: Damla Turgut

ABSTRACT

Cognitive radio networks promise more efficient spectrum utilization by leveraging degrees of freedom and distributing data collection. The actual realization of these promises is challenged by distributed control, and incomplete, uncertain and possibly conflicting knowledge bases. We consider two problems in bootstrapping, evolving, and managing cognitive radio networks. The first is Link Rendezvous, or how separate radio nodes initially find each other in a spectrum band with many degrees of freedom, and little shared knowledge. The second is how radio nodes can negotiate for spectrum access with incomplete information.

To address the first problem, we present our Frequency Parallel Blind Link Rendezvous algorithm. This approach, designed for recent generations of digital front-ends, implicitly shares vague information about spectrum occupancy early in the process, speeding the progress towards a solution. Furthermore, it operates in the frequency domain, facilitating a parallel channel rendezvous. Finally, it operates without a control channel and can rendezvous anywhere in the operating band. We present simulations and analysis on the false alarm rate for both a feature detector and a cross-correlation detector. We compare our results to the conventional frequency hopping sequence rendezvous techniques.

To address the second problem, we model the network as a multi-agent system and negotiate by exchanging proposals, augmented with arguments. These arguments include information about priority status and the existence of other nodes. We show in a variety of network topologies that this process leads to solutions not otherwise apparent to individual nodes, and achieves superior network throughput, request satisfaction, and total number of connections, compared to our baselines. The agents independently formulate proposals based upon communication desires, evaluate these proposals based upon capacity constraints, create arguments in response to proposal rejections, and re-evaluate proposals based upon received arguments. We present our negotiation rules, messages, and protocol and demonstrate how they interoperate in a simulation environment.

To my father, Robert A. Horine, who taught me the value of steady, hard work.

ACKNOWLEDGMENTS

I wish to give special thanks to my advisor, Dr. Damla Turgut, who believed in me when I did not. Her persistence convinced me to return to finish.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

This dissertation presents contributions to the fields of link rendezvous and link negotiation for dynamic spectrum access (DSA) in cognitive radio networks (CRN). Since the Communications Act of 1934, spectrum has been centrally planned and highly regulated, leading to static allocations. The dramatic recent growth of wireless services and devices has created an apparent shortage of available spectrum, compelling cellular carriers, for example, to spend nearly a billion dollars to license spectrum suitable for nationwide coverage [1]. On the other hand, recent measurements have revealed that much of allocated spectrum is not actually used a large fraction of time and place [2]. This realization prompted a new thought process on how to best access spectrum. Dynamic spectrum access (DSA) is a relatively new paradigm that allows radios to access fallow spectrum that is licensed to primary users (PU). This access is on a secondary basis, requiring they do not interfere with any PU [3].

While the regulatory environment is rapidly evolving in this domain, researchers are pursuing technical solutions. One of the most common and promising approaches calls for secondary users (SU) to sense the spectrum before accessing it. If they can determine with confidence that the PU is not utilizing the spectrum, they may use it for a short period of time before they need to sense again. Complicated propagation environments lead to sensing errors; concluding that a channel is available when it is not. Collaboration can correct this problem, but presupposes an existing communication channel. This presents a dilemma in the need to have a communication channel over which to collaborate so that the nodes can establish a communication channel.

Furthermore, once we make the regulatory and technical leap to dynamic spectrum access, it seems reasonable to assign other physical layer parameters on a dynamic basis also. These opportunistic radios operate with many degrees of freedom, including frequency, bandwidth, power level, modulation, and potentially many others. Software defined radios (SDR) and modern reconfigurable hardware architectures enable this flexibility; however, adequate control and scheduling mechanisms are lagging. Dedicated control channels and centralized controllers can address the control channel to a degree, but represent single points of failure due to denial of service attacks and congestion. Their use also opposes the secondary usage paradigm. If they are in the secondary usage domain, the dedicated control channel could be occupied by the PU, denying access to the entire SU network. If they are in the primary usage domain, then this places an undue burden on the PU network, decreasing the likelihood that the spectrum license holder will agree to SU access in principle.

Cognitive Radios (CR) and Networks (CRN) are being developed in order to push the decision making ability and responsibility down towards the individual node and to decentralize the process. This allows the nodes to engineer their own communication design specific to the conditions and needs at hand. Rather than a static communication system that is the product of possibly years of engineering design, analysis, and test, the goal is a very dynamic system that can adapt to efficiently utilize the scarce resources to satisfy the user demand. We are interested in studying and providing solutions to two important problems in these systems. The first problem is how disconnected radios can initially find each other in communication space when there are many degrees of freedom in terms of operating parameters, no centralized control, and incomplete information in each knowledge base. To address this question, we have developed our frequency parallel blind link rendezvous (FP-BLR) algorithm. The second problem is how to efficiently allocate limited resources to many users with distributed control and again, incomplete information. To address this question, we present our multi-agent centric technique of argumentation based negotiation in cognitive radio networks (ABNinCRN) work.

Motivation

Clearly a distributed and collaborative approach is needed to control these networks. Collaboration between nodes can improve the accuracy of sensing, but how do you collaborate before you have bootstrapped a network? How do radios find each other when there are many degrees of freedom and no central control? Once these nodes find each other, a process called *rendezvous*, they must decide upon the parameters with which to communicate. While this decision making process can occur over least-common-denominator parameters, it should be highly efficient and robustly resolve arrive at an adequate, if not optimal, solution. While many researchers have attacked this problem, the solution is not at all obvious, complicated significantly by the likely conflicting information and certainly inadequate at each node in the collaboration. While communication links are governed in general by Shannon's Capacity Law, $C = B \cdot log_2(1 + \frac{S}{N})$ (where C is the capacity, B is the bandwidth, and $\frac{S}{N}$ is the signal to noise ratio of the channel), there are an infinite number of solutions to reach a given capacity by trading off bandwidth, signal to noise ratio, and other implicit parameters such as modulation type and depth and coding technique. Somehow, the individual nodes in a CRN have to collaboratively arrive at a suitable decision in a timely manner. We presented our original work in this area in 2007 [4]. From thence, a number of researchers have proposed many algorithms that are predominantly based on frequency hopping sequences. As we will detail later, many of these produce excellent results; however, the implementation of these algorithms in typical radio hardware is not always straightforward. While a few systems, such as Bluetooth and some anti-jam military systems, use frequency hopping for their core communication function, most only have frequency diversity in a quasi-static manner. Dynamic spectrum access brings a more dynamic requirement to the frequency diversity; however, it is possible in implemented systems that the hopping for rendezvous may need to be faster than the operating channel change time. Increasing the hop rate of frequency synthesizers can easily become prohibitively expensive when other parameters must be held constant.

For these reasons, we sought a technique that could easily overlay some of the modern implementations of digital front ends (DFE). A significant trend in wideband systems is to use field programmable gate arrays (FPGA) between the data converters and the baseband processing typically performed in digital signal processors (DSP). Since they are field programmable, they are good architectural choices in adaptable and software defined radios. It is common and relatively straightforward to implement parallel algorithms on these gate arrays. Even the latest generations of DSPs are equipped with multiple, parallel cores. The parallel nature of our algorithm is designed to overlay onto this type of architecture. Furthermore, the push for more data leads to wider radio frequency (RF) bandwidths than previously in common use. Many systems use frequency domain techniques such as orthogonal frequency domain multiplexing (multiple access) [OFDM(A)] [5] and single carrier frequency division multiple access (SCFDMA)[6]. Practical implementations of these waveforms use Fourier transforms. These waveforms are used in WiMAX [7] [8], LTE [9], IEEE 802.11a/g/n WLAN [10], digital television [11], and many more applications. Since the Fourier transformer (usually in the form of a fast Fourier transform (FFT) algorithm) is already present in the radio, we designed our system to operate in the frequency domain.

As we intimated earlier, an *a priori* defined common control channel represents single point of failure that can be exploited by an intentional or even unintentional jammer, denying service to the entire secondary network. Furthermore, it is in conflict with the virtual separation of the primary and secondary users network. Instead, we wish to accomplish rendezvous with minimal knowledge of the behavior and state of other nodes, hence it is a blind rendezvous. Thus, we arrive at our goal of developing a Frequency Parallel Blink Link Rendezvous algorithm.

Once nodes are aware of each other's presence and can begin communicating at some frequency, we may consider how they should allocate the scare resources to provided requests for service. While game theory and other techniques have been used to analyze the optimal allocation, we choose to delve into the process rather than the result. The decision making is easier if all nodes have perfect and complete knowledge. To achieve this, we must exchange information; however,this results in overhead. The lifetime of data before it should be considered invalid is also an issue. In other words, we consume valuable network resources in exchanging information that may be out of date by the time everything is promulgated everywhere. Furthermore, some information at one node may conflict with information at another node. This is common in distributed spectrum sensing where nodes at different spatial locations perceive a unique RF environment. Fusion of this information is a difficult and active research field [12] [13].

With finite radio resources and a high enough demand for those resources, the system will be unable to satisfy all of the requests. If the nodes knew *a priori* all of the requests that were going to be issued during a given time period and the environment in which each node has to operate throughout that time, some optimization process or exhaustive search could find the best solutions for a given set of criteria. This is not practical in most cases, and especially in the particular scenario of secondary usage. Instead, the network nodes must allocate resources as the requests arrive. When requests cannot be satisfied immediately, current systems either queue them or reject them. Similarly, when conditions change, a particular connection may be downgraded to a lower rate or handed to a new cell, in the example of mobile cell phone communications. We envision a system in which the rules for admission or dropping request may be considerably more complex. In one example, requests may have a higher priority than other on-going connections. A naive system may simply drop the lower priority connection in favor of satisfying the new higher priority one. We assert that there is a compromise where one or both connections might accept a lower grade of service, resulting in the new request being at least partially satisfied while the on-going connection is maintained, but at a lower rate. These types of more nuanced solutions might be achievable with a sufficiently complex decision making processes in each node. Without careful design, it will be intractable to maintain and extend when administrators desire to modify the rules. We propose and demonstrate that negotiation can be an effective tool to manage this complexity. Ultimately, these radio nodes are deciding amongst each other on courses of action and then implementing those actions. We model this interaction as a multi-agent system.

Negotiation is often used when two or more parties with potentially conflicting goals and beliefs attempt to reach an agreement. Negotiation between two parties is familiar to most of us whether in attempting to agree on a pay rate for a new job or perhaps finding an agreeable price between buyer and seller for a good at the market. Multi-party negotiations can be considerably more complex, especially when a consensus agreement must be reached rather than a simple majority. In these cases, it is useful to adopt a protocol describing who may issue what kind of proposal to whom at any given time. Without this structure, or something similar, the negotiations will possibly be chaotic and ineffective. Proposals are just one way that a node can communicate with another in the control plane. We have developed a number of message types, or locutions that unambiguously communicate between nodes. Some examples include rejection and acceptance messages in response to a received request. A set of rules determine whether a proposal should be accepted or rejected, or some other action taken. An important aspect in developing a set of rules is ensuring that the negotiations stay within reasonable bounds and do not needlessly consume network resources in a futile attempt at a deal. The resulting framework should be adaptable to different applications without forcing a major rewrite of the control code base.

There remains the problem of incomplete and conflicting information between the agents. An agent issuing a request may learn something of the beliefs and desires of the responding agent based upon receiving a rejection or acceptance message; however, this learning process is not very efficient. For instance, if an agent requests a connection lease on channel C from time T_1 to time T_2 at a rate R, but receives a rejection, it learns a very particular, but coarse fact. It learns nothing about other channels or if a slower rate might be accommodated or if starting at a time T_n between T_1 and T_2 might be acceptable. To speed up the flow of information, messages may be augmented with relevant information. Of course, it may be difficult for the responding agent to determine exactly what is relevant to the originating agent, because it likewise has a very narrow view of the beliefs and desires behind the request. We model the provision and analysis of this additional information as argumentation. Thus we have as a solution to the allocation problem our argumentation based negotiation in cognitive radio networks (ABNinCRN) framework and system.

Overview of Our Solutions

The FP-BLR algorithm is triggered when a disconnected node desires to form a network to conduct some communications, but is unable to find one based upon its current knowledge. It begins by sensing the spectrum for available channels, in other words, channels in which neither a PU nor a SU is operating. It then broadcasts a unique probe signal in one or more of these channels. To fully exploit the parallel nature of the approach, it will use as many as possible according to the capabilities of the hardware. Other radio nodes, if not actively engaged in communications, will reconfigure their hardware to scan the operating band for these probe signals while also conducting spectrum sensing operations in order to prepare itself for transmitting. This means opening its receiver bandwidth to that of the operating band, or some large subset of it depending upon the sample rates in the system. It samples the receiver input and performs a Fourier transform, using an FFT. It then examines the resulting spectrum for the presence of the probe signal in one or more channels. If it finds some, it forms a reply on the intersection of the set of probed channels and its set of available channels. The responding nodes thus communicate some information about their sense of busy versus available channels. Finally, the originating node switches to scanning mode as soon as it finishes transmitting, performing a similar Fourier transform process and scanning for the reply probe signal. This reply probe is similar in nature to the initiating probe, but distinct. If it finds the probe on one or more channels, it reconfigures its hardware for data communications, and finally it responds on the best channel (best in terms of RF channel characteristics, measure by received signal strength for example) with a conventional connection request packet. A suitable media access control (MAC) mechanism, for example, carrier sense multiple access (CSMA) is used by the nodes to avoid collisions.

If the initiating node does not detect a reply probe within a finite scan period, it may repeat the initial probe at a higher power or longer duration, up to an application driven limit. If the total output power is hardware limited, as opposed to regulation driven, it may choose to transmit on a limited subset of the available channels in order to maximize the peak power output at each transmitted frequency. Conversely, can choose to begin the process with a minimum of output power and probe duration in order to conservatively protect ongoing PU communications from unintended interference resulting from imperfect sensing. Most of these parameters are application dependent.

The ABNinCRN process is triggered by a node's agent perceiving a desire to communicate. This may immediately follow a link rendezvous, or it may occur after it has joined a network. We assume an underlying routing protocol that provides the agent with some information about the first node to connect. It sends a proposal message in the form of a request for channel access at a particular rate along with other information relevant to the application, such as anticipated connection duration. The receiving node determines if it has the capacity to support the requested rate. If it can, it will respond with an acceptance message, or forward the request if it is not the final destination of the request. If it cannot, then more advanced negotiations will ensue. In the simplest model, it will respond with a rejection and it will be up to the initiating node to attempt some other method of satisfying its desire. In more complex models, the nodes may engage with other nodes and request modifications of existing connections in order to clear capacity for the new request. All negotiations are accomplished through a small set of messages, augmented by arguments when appropriate to speed the negotiations. Arguments may take the form of explicit information such as "This connection I am requesting has been assigned a high priority by an authority we should both agree upon." They may also ask an implied question through an assertion such as "I am asserting a fairness argument according to our agreed rules." This is implying the question "Do you have an existing connection, without any priority higher than my request, but at a higher rate, that could be downgraded so that we may both communicate at a compromised rate?" The arguments are represented by facts allowing a knowledge base to be built at each node as arguments are exchanged. Each time an agent receives an argument, it will reevaluate the pending proposal and either accept it or reject it. This approach keeps the message exchange straightforward and well bounded, in contrast to needing a large set of question and answer messages. Finally, commitments are made as negotiations proceed and are either satisfied or released at the completion of the negotiations. This helps keep the negotiation stable.

Contributions

The focus of our work is the bootstrapping and evolution of cognitive radio networks in a collaborative fashion. More specifically, the major contributions of this dissertation are as follows:

- *Frequency Parallel Blind Link Rendezvous.* We present our technique for unaided connection with minimal *a priori* information. This protocol features speedy rendezvous, low overhead and early spectral occupancy communications, low overhead and early channel state information communication, protection of the primary user, and elimination of a requirement for a fixed frequency raster or channelization scheme.
- *Link Rendezvous in Software Radio.* We encoded our link rendezvous algorithm in the GNURadio platform allowing automated, distributed rendezvous. This functionality fits seamlessly into the open source system. We have also developed high fidelity Matlab simulators to facilitate deeper inspection of the algorithm.
- *Link Rendezvous Detectors.* We developed and analyzed two different detectors for the multiple tone probe signal.
- *Argumentation Based Negotiation in Cognitive Radio Networks.* Our negotiation protocol using argumentation resolves inconsistent and incomplete information through the exchange and analysis of arguments. The protocol robustly and predictably finds effective solutions, provided one is feasible, with any of a set of initial conditions. If a solution is not feasible, it properly indicates that conclusion. In other words, the negotiation completes, rather than endlessly negotiating.
- *Argumentation Based Negotiation for Radio Resource Allocation.* Our protocol orchestrates compromises which allow more users to achieve some level of resource access

under a rules driven methodology. This also results in more total data transferred in from source to destination, and more active connections supported when the network is severely stressed. This is accomplished by a minor penalty in the number of negotiation messages.

- *Adaptable Framework for ABNinCRN.* We have implemented our framework in software system that could be deployed on actual radios with few changes. Furthermore, we have used this system to simulate multiple network applications, with different rules, and with multiple negotiation strategies. This framework facilitates complex distributed control.

Outline

This dissertation is organized as follows:

Chapter 1 presents the problem definitions. Chapter 2 conducts a literature review covering the background knowledge needed to understand the remaining chapters. This is divided into sections describing rendezvous techniques for cognitive radio networks, negotiation techniques for these networks, and argumentation techniques in negotiations in general.

Chapter 3 presents our Frequency Parallel Blind Link Rendezvous protocol including the philosophy and principles guiding the design, a detailed description of the protocol itself, an analytical analysis of the probability of errors in the rendezvous process, and some results from a software radio implementation. We extend this work by examining a detector based upon cross-correlation in Chapter 4. Here we will describe a simulation system that analyzes several metrics under different conditions including the effects of multipath interference and various decision threshold levels. We will compare and contrast our FP-BLR with some important conventional rendezvous techniques in Chapter 5.

Chapter 6 discusses our argumentation based negotiation framework. Here we present our framework. We illustrate its usefulness and effectiveness through a small network composed of nodes with heterogeneous roles: sinks and sources in one role and relays in another. This scenario involves a complex negotiation due to multiple rules and arguments that may be asserted. In Chapter 7 we discuss and extensively simulate a different application involving homogeneous agents in common roles with a larger network subject it to request rates designed to stress the network. We significantly constrain the negotiation options through our rules in order to demonstrate that overhead due to negotiations can be minimized and limited with an upper bound, while still achieving superior results. In Chapter 8 we present an extensive simulation of this latter scenario. Finally, Chapter 9 presents our conclusions and possible extensions to the work presented.

CHAPTER 2: RELATED WORK

Blind Link Rendezvous

Polson [14] identifies two main approaches toward link rendezvous. The first assumes some infrastructure which transmits a beacon encoded with time and frequency rendezvous information. Cognitive radios (CR) request a time-frequency slot for a specified network and provide location and power information. The infrastructure server recommends a frequency and schedules a time for the CR to check back in. The server has an omniscient view of all CRs in the area and can globally optimize its decisions.

Infrastructure is not always appropriate, especially in a secondary use scenario. Polson also describes two versions of an unaided approach. Both rely on the calling node emitting a probe signal on a selection of available frequencies. Receiving nodes listen on the set of frequencies that they determine are available. When the original probe waveform is detected, the receiving node transmits its own probe-acknowledgement waveform, signalling its readiness to establish a connection. Once connected, the nodes exchange information which expands their knowledge of the spectral environment. In the first described case, a node transmits in random vacant channels while another node monitors its set of vacant channels. The monitoring node can speed the rendezvous by simultaneously monitoring multiple channels provided the hardware supports it. In the second case, the probing and scanning is prioritized according to the largest unoccupied blocks.

Balachandran and Kang propose a set of protocols assuming slotted frequency hopping sequences [15]. They present the probability of achieving a link within a given time for the various protocols. These protocols rely upon timing synchronization through a standard such as GPS or acquisition from a neighboring node.

Han, Wang and Li describe a link establishment process centered around a base station and multiple mobile stations [16]. Their system relies on an interleaved OFDM-based transform domain communication system for establishing the first connection. This spread spectrum technique minimizes the potential for interference with ongoing communications.

Holland, et. al. propose a universal dedicated channel to communicate spectrum resource availability and usage [17]. Each radio periodically broadcasts information about the resources used by the communications it receives. Sutton, et. al. recently described a technique which relies on cyclic signatures embedded in OFDM signals to trigger rendezvous in low signal to noise applications [18]. They also describe a media access strategy, nevertheless, questions remain about the ability of a single channel approach to scale to meet the capacity demands of a crisis.

The collaborative exchange has been studied by a number of authors. Zhao et. al. propose a system in which nodes self-configure into local groups and establish a common control channel to serve that group [19]. Fringe nodes bridge between groups and provide and overall network connectivity. They also propose a modified MAC which leverages this approach. Ghasemi and Sousa investigated collaborative sensing in fading and shadowing channels [13]. They describe a system which exchanges bit information where each node compares the sensed energy to a threshold and sets the bit value accordingly. When a node receives information from other nodes, it performs an OR operation on all of the reporting nodes at a given frequency. Any node reporting a signal therefore dominates the decision. Ganesan and Li describe a relay based approach in a TDMA system in which nodes listen on alternate time slices and pass on information about primary transmitters on their time slice [12].

Sensing technology plays a significant role in facilitating network formation while protecting incumbents from interference. Energy detectors have well known detection limit problems [20] and cannot distinguish a primary from an uncooperative secondary user. To overcome these issues, researchers have studied feature detectors [21], matched filter detectors, waveform sensing [22], and cyclo-stationary detectors [20].

Link rendezvous using non-orthogonal frequency hopping sequences has been studied by DaSilva[23]. In the generated orthogonal sequences (GOS) algorithm, each radio hops according to an a priori determined sequence. A permutation of the sequence is repeated, but interlaced with a single, incrementing value from permutation. This interlacing guarantees that two radios will eventually hop onto the same channel, thus accomplishing rendezvous, no matter what lag they start at in the sequence. The lag accounts for timing differences between the nodes, both at a time slot basis, which affects the sequence position, and within a time slot, which affects the probe and monitoring timing. For their generated sequence, they calculate the expected time to rendezvous for the best channel as

$$E[TTR|best] = \frac{N^3 - 2N^2 + 9N - 4}{2(3N-1)} \tag{2.1}$$

where N is the number of available channels.

and for the worst channel

$$E[TTR|worst] = N^2 \tag{2.2}$$

Theis et al. [24] extend this concept and draw a compelling case for blind link rendezvous in order to fully realize the potential and capabilities of dynamic spectrum access technology. They first develop a rendezvous taxonomy with two branches, similar to Polson's. The first is the aided branch encompassing systems with a dedicated control channel, of which DimSumNet [25] is an often cited example. They assert, as we did earlier [4], that a control channel based system is not scalable, flexible or robust due to the overhead required and the single point of failure vulnerability. Unaided rendezvous represents the other branch of their taxonomy. In this case, radios make their own spectrum decisions in a distributed fashion. Since the radios are distributed, there is still a need for communications in order to collaborate on the decision and to promulgate the results. Their taxonomy further divides the various systems by either a single control channel, multiple control channels, or no control channels. While multiple control channels begins to deal with the bottleneck and single point of failure problems of a single control channel, it still is unreliable with uncertain spectrum sensing and a very dynamic environment. The no control channel approach is manifested by blind link rendezvous where any and all available (i.e. vacant) channels are available for exchanging control information in addition to payload data.

They also described the various models and assumptions that can be used to categorize the various systems and rendezvous algorithms. In our work, the first relevant model is the *shared* model, in which there is no control infrastructure or specific roles between radios (e.g. master/slave), but there is a shared agreement of the available channels upon which to rendezvous. Another important model is the *individual* model. It is similar to the shared model, but does not require that each radio observe the same set of available channels nor that they are labeled in the same manner.

They expand upon the earlier work of DaSilva et al. on random sequences and GOS and present the Modular Clock (MC) algorithm. They calculate expected and maximum times to rendezvous (while showing that the random algorithm does not have a maximum). The random algorithm calls for each radio to randomly hop to available channels and attempt a rendezvous. They show that the expected time to rendezvous is given by $E[TTR] = \frac{m_1 m_2}{\overline{m}}$, where each m is the number of available channels, assuming the individual model. The MC model under the shared assumption has an upper limit given in equation 2.3

$$E[TTR] \leq 2p_i + \frac{2p_i}{p_i - 1} \tag{2.3}$$

where p_i is the lowest prime greater than or equal to the number of observed channels m_i by radio i.

and for the individual model, the result is provided in equation 2.4.

$$E[TTR] = \frac{p_1 p_2}{\overline{m}} \tag{2.4}$$

The MC algorithm has certain conditions under which it is impossible to guarantee rendezvous. These involve a difference of open channels assessments, but the chosen prime numbers by each node are the same. The MMC algorithm randomizes the selection of primes in order to avoid these conditions, but there is still not an absolute maximum expected value for the time to rendezvous.

The Ring Walk algorithm of Lin, et al. [26] later evolved into the Jump Stay (JS) algorithm, which they claim has the quickest TTR for sequence based hopping algorithms [27] [28]. The radios hop in a modulo manner through a sequence (jump), but stay on a particular channel for an extended period of time; (stay). They use the term *symmetric* for the previously described *shared* model and *asymmetric* for the *individual* model. The JS routine supports rendezvous for more than 2 radios. It also guarantees rendezvous.

Pu, Wyglinski and McLeron [29] describe a technique where receivers transmit pilot tones on a frequency of their choice. A single transmitter scans for these pilot tones and then visits each in one of three different scanning patterns. During the visit, it inquires of the receiver to determine if it is of the set it wishes to communicate. If it is, it attempts a connection, which if successful, ends the rendezvous process. If not, it proceeds with the scanning, repeating the whole process if necessary.

The first scanning pattern examines the frequencies in ascending channel order. Using the Law of Total Expectations, they determine that the average scanning time is $(M + 1)/2$. The second scanning pattern visits the channels in order of descending power. This approach also yields an average scanning time of $(M + 1)/2$. Finally, the third scanning pattern groups the pilot tones into clusters and visits each in descending order based upon the number of pilot tones in each cluster. They develop an upper bound that indicates this approach has a shorter average scanning time than the previous two scanning rules. They do not describe precisely how the clusters are determined, but it appears that each is bounded by blank channels. Three scenarios are used to validate the theory. The first scenario uses a uniform distribution to allocate frequencies, while the second uses a normal distribution. The third approach chooses open frequencies from measured data in a paging band in the Worcester, MA area. Scenarios 1 and 2 produce results consistent with the $(M + 1)/2$ theory. Scenario 3 produces the best results, as predicted.

The authors do not say whether or how receivers choose unique frequencies, although all of the presented data indicates that they do. They state that the transmitter initially sweeps through the band to catalog the pilot tones. They do not describe unique features of the pilot tones that would allow the sweeper to distinguish between receivers or primary user transmitters. Finally, they make a key assumption that the time to find a target receiver within a cluster is negligible compared to the time to scan the clusters. This is not validated according to various quantities that might be present in each cluster.

We first presented our proposed link rendezvous protocol in our previous work [4] where we explored the detection limits by prototyping key parts of it in the open source project GNU Radio [30]. We followed up with a performance analysis in [31] were we developed an analytical relationship to the false alarm rates. This and subsequent work will be discussed in the following chapter.

Negotiation and Argumentation

Negotiation in Dynamic Spectrum Access Systems

Negotiation in a broad sense occurs in many situations of a cognitive radio network practicing dynamic spectrum access. A number of researchers have applied various techniques to allow a cognitive engine in a radio agent to negotiate either with itself (essentially a self-reasoning process) or with other radio agents. Mitola has emphasized cognition at various levels of a radio's protocol stack, particularly at the application level where it interacts heavily with the user and his environment [32]. He also emphasizes case-based reasoning (CBR) as a means of synthesizing new solutions based upon past successes [33]. Rondeau and Rieser investigated using genetic algorithms in conjunction with CBR to find novel physical link layer parameter sets [34, 35]. Neel, et. al. have discussed using game theory extensively to analyze cognitive radio networks and discover algorithms for achieving cooperative behavior [36, 37].

Denker et. al. propose extending their XG-style policy reasoner to return failure reasons and constraints instead of a simple "no" answer [38]. This introduces negotiation into the system strategy and policy reasoner interaction, and, as we shall see later, is actually a form of argumentation.

Kulik, Heinzelmand and Balakrishan use negotiation protocols to eliminate the transmission of redundant data in sensors networks [39], with the goal of reducing energy consumption. They tag sensor data with meta-data and connect to the application layer to determine what data is needed where. They define three messages, ADV, new data advertisement; REQ, request for data; and DATA, data message.

Argumentation Based Negotiation

Argumentation has roots in philosophy and can trace back to Aristotle [40]. In recent years, a number of researchers have begun to apply these concepts to artificial intelligence problems, first in self-reasoning applications and later in multi-agent systems. Researchers in multi-agent systems often assume the agents are autonomous and rational [41]. Autonomous in this context means that the agent will make its own decision on whether to honor a request from another agent or whether it will initiate some action as a result of sensing something in its environment. Acting autonomously does not mean that the agent is not constrained by some set of rules, but that within these rules, it is free to decide on its course of action. Being rational means that it will act in its own interest. Note that a rational agent can be encouraged to act with the appearance of altruism if it gains some value, however contrived, from that altruism. Likewise, its own interests may be compromised by a decision at the present, but enhanced in the future as a result of a promised reward for acting in the present.

Because one agent cannot force another agent to obey it, they are compelled to negotiate with each other in order to achieve theirs goals. In fact, negotiation is even more important when one cannot guarantee that the agents will act rationally. This might occur if the agent's knowledgebase is inconsistent or incorrect. Then the agent might make a decision which it thinks is in its interest; however, from an omniscient point of view, it is detrimental, and therefore, irrational.

As Rahwan et al. point out in their survey paper [42], "negotiation is needed when agents have competing claims on scare resources, not all of which can be simultaneously satisfied." In our context, the primary resource is spectrum, dimensioned by frequency, space, and time. They further assert that argumentation based negotiation can reach superior agreements faster by exchanging additional information about their beliefs and other mental attitudes in order to justify its own stance or influence another's stance. Specific types of information include a critique of another's statements (e.g. proposal), a justification of its own proposal, and information which changes the course of the negotiation process (e.g. introduce a relay node into a peer-peer link negotiation). Furthermore, threats or promises of a reward can be used to influence another's value system. For a detailed discussion on the limitations of game theoretic and heuristic approaches and how argumentation can overcome them, refer to Rahwan et al. [42].

Dung [43] presents a logical view of argumentation via the concept of acceptability. If an agent cannot produce a counter-argument which defeats an argument under consideration, then that argument under consideration must be accepted. Dung asserts "whether or not a rational agent believes in a statement depends on whether or not the argument supporting this statement can be successfully defended against the counterarguments."

Sierra et al. focus on negotiation as persuading another agent to do something for it [44]. They emphasize the social roles that agents have with respect to one another. Their framework is based upon dialogs and the tuple $<Agents, Roles, R, L, ML, CL, Time>$, where *Roles* is a set of social roles such as supervisor, subordinate, etc. that an agent could assume. R is the mapping of agents to those roles. L, ML, and CL represent the logical language, meta-language, and communication language respectively. The negotiations take place within a negotiation thread, with access to the history of negotiations. The thread begins with an offer or request dialog, followed by a series of proposals (including counter proposals) which may be rejected or accepted or one agent may withdraw from the negotiations. Each proposal message may include an offer, a threat, a reward, or an appeal. These are interpreted in the context of an authority graph generated from the roles of the respective agents.

Parsons et al. [45] describe an agent architecture based upon multi-context systems [46]. Their approach emphasizes that in multi-agent systems, agents may have differing logic reasoning systems. This results in the possibility that an argument may be well grounded in one agent, but not in another. The implication is that the rules of inference and the bridge rules in the multi-context system must be associated with the argument so that another agent can properly interpret it.

They also define the notions of consistency, rebuttal, and undercut and use these to develop a formal model of argumentation based reasoning. They develop a negotiation system where agents can construct arguments to justify their proposals, critique other proposals, and exchange arguments to influence each other with the goal of achieving an agreement.

Amgoud et al. explored how to generate and interpret arguments in the context of a negotiation [47]. They cast the negotiation process in a series of dialog moves. Each dialog move (e.g. assert, question, challenge, request, promise, accept, and refuse) has a rationality, dialog, and and update feature. The rationality reflects the preconditions for a particular move. The dialog indicates how it interacts with the other agent. Update determines what actually happens when the dialog is invoked, primarily triggering a communication act and possibly a knowledgebase update.

Finally, Parsons et al. [48] present a detailed analysis of how to specify dialogs used in argumentation. This work emphasizes a continual analysis of commitments made by each agent in the context of Walton and Krabbe [49]. These commitments are collected in a public commitment store and can be used to enforce consistency and follow through on agents.

Applications to Networks

We have investigated argumentation based negotiation as a tool to solve a variety of problems in cognitive radio networks. In particular, we studied reservation of network resources. Our current work draws heavily on protocols in Internet type networks. We now briefly survey some of the techniques. The Resource Reservation Protocol described in RFC 2205 [50] uses path and reservation messages to reserve capacity for streaming data. RFC 4495 [51] extends this by allowing for the bandwidth of an existing connection to be downgraded. A great deal of research has focused on the challenge of guaranteeing a specified quality of service (QoS) in mobile ad hoc networks (MANET). Some of these proposed solutions work with scheduled media access control (MAC) layers, while others work with contention based MACs. One of the challenges of these systems is having to deal with local information, keeping exchanged information fresh, and estimating channel capacity conditions [52]. Pitt, et al [53]use a norm-governed multiagent system to address quality of service provisioning in MANETs. Thier system runs over a Multimedia Network Support Platform (MNSP). The MNSP provides the communication services while the multiagent system provides the decision making services through a deliberative process according to norm-governed policies and protocols. Continuing this work, Manvi, et al [54] use multiagent techniques to disseminate information in vehicular ad hoc networks, using both static and mobile agents. Förster surveys machine learining techniques in wireless ad hoc networks in [55], covering reinforcement learning, swarm intelligence, heuristics, and mobile agent technologies. Liu and Issarny study trust relationships through reputation for MANETs in [56]. Finally, Gan, Liu, and Jen address energy efficiency using agent techniques in [57].

CHAPTER 3: BLIND LINK RENDEZVOUS IN THE FREQUENCY DOMAIN

In this chapter, we describe our blind link rendezvous protocol, and present a feature detector algorithm. Subsequent chapters will introduce a cross-correlation detector and compare our results to conventional frequency hopping sequence based rendezvous protocols. The goal is to create a system that results in a rendezvous between two radios operating in a pure secondary use scenario. This implies the absence of a control channel for reasons discussed in the Introduction. Furthermore, we seek to speed the rendezvous process by taking advantage of modern digital radio architectures, and specifically multichannel systems such as OFDM. The typical cognitive radio hardware includes reconfigurable hardware in its digital front end (DFE), such as FPGAs, DSPs, and DPUs. By reconfiguring the DFE for wide band operation, the rendezvous initiator can probe several channels simultaneously. Likewise, potential responders can simultaneously scan multiple channels.

We begin with a description of the algorithm. Then we discuss some of the more important metrics. Following this is an analysis of the performance of the system and simulation results.

Frequency Parallel Blind Link Rendezvous Protocol Description

The rendezvous process begins with sensing the spectrum to determine which channels are available for secondary use. The node wishing to originate the connection then transmits a unique probe signal on these channels. This probe signal possesses unique frequency domain characteristics. One example is the signal described by equation (3.1). This sum of sinusoids is easy to generate using software defined radio technology. The transmitted signal is the aggregate of these probes on each channel.

$$a(t) = \sum_{j=1}^{J} A_j \cos(\omega_j t) \tag{3.1}$$

where there are J sinusoids in the probe signal, each with relative frequency ω_j and amplitude A_j. This probe can then be modulated to multiple channel locations, ω_{c_m} with an ouput amplitude of A_0 as described in equation 3.2

$$s(t) = A_0 \sum_{m=1}^{M} \cos(\omega_{c_m} t) \sum_{j=1}^{J} A_j \cos(\omega_j t) \tag{3.2}$$

where M is the number of channels to probe and typically $\omega_c >> \omega_j$.

Nodes on standby continuously monitor the spectrum for this pattern. SDR technology allows the radio to scan many channels at once. An entire band can be sampled, based upon the performance of the A/D converter and system processing speed. A Fast Fourier Transform (FFT) is performed on the resulting sample stream. A feature detection algorithm searches for the well defined pattern given in equation (3.2). The receiving node collects all of the probe signal occurrences it finds within its scanning range and chooses which channel to use in response. It then transmits a similar but distinct pattern of its own on the chosen frequency.

After the initial transmission, the calling node switches to a listen mode. It scans all of the frequencies on which it originally transmitted, looking for the reply pattern of sidebands. In dense RF environments, it might receive a reply from more than one responding node. A collision detection based media access control (MAC) layer can be used to sort out responses. The calling node chooses the final frequency on which to connect from the entire set of responses. This decision may be based upon signal strength or other preference ranking. The node is finally ready to establish a connection. It broadcasts a connection request code using some reasonably lowest common denominator radio waveform parameters on the chosen frequency.

After transmitting the probe reply signal, idle receivers enter a listen mode for a connection request. Upon detecting a connection request, the receiver transmits a connection response message directly to the originating node. This message is the first unicast message and can include information about the node such as the services it can provide and connection parameter preferences. The originating node chooses to which destination node to connect, which finishes the rendezvous process.

The hypothesis that leads to the use of a probe signal detected in the frequency domain is that the transmitter can secure the attention of a receiver with a minimal energy, bandwidth and duration. This is achieved primarily by an SDR's ability to monitor multiple channels simultaneously. Because this protocol probes many frequencies at once, and operates in the frequency domain, we call it the Frequency Parallel Blind Link Rendezvous or FP-BLR.

Performance Metrics

The performance of the Frequency Parallel Blind Link Rendezvous protocol is driven by both the specific design of the probe signal and the receiver design. In a general sense, the frequency parallel nature of the protocol sets the context of the performance, in that it is a wideband technique. In this section, the metrics of concern in link rendezvous are reviewed. Then the design parameters driving these metrics are described. There are several different ways to evaluate a rendezvous protocol, depending upon the application domain.

We enumerate several here and discuss their implications.

Time to Rendezvous

Many researchers have focussed on the time to rendezvous metric measured as both the expected and maximum (if it exists) values. This is always important in DSA applications because an exceedingly long rendezvous time may run into problems with stale spectrum sensing data. Once rendezvous is accomplished, then actions due to spectrum disruptions can be coordinated before clearing the channel. For long communication exchanges, the rendezvous time is less significant. For short conversations, it could become a significant percentage of the communication time. Ultimately, this is a function of how many channels are available; however, we can examine some existing rendezvous algorithm to get an understanding of the order of magnitude. Bluetooth pairing or bonding can take up to 30 seconds. GPS acquisition in a cold boot situation can take up to 60 seconds if not aided.

Energy to Rendezvous

The total energy used to accomplish rendezvous is a combination of the power used for probe signals, the duration of the probe signals, and the total number of rounds needed for rendezvous. The energy required for rendezvous is a function of the rendezvous algorithm, the number of SUs, the number of PUs, the distribution of SUs, the number of frequency channels, and the complexity of the propagation environment.

Likelihood of Interference

It is well known that local spectrum sensing is susceptible to errors such as the hidden node problem. Multipath interference and shadowing cause highly localized drops in the received signal power, leading to spectrum sensors erroneously concluding that no primary user is present. Naive rendezvous techniques can unintentionally cause interference to the primary user. This metric can be quantified by the likelihood of interference per rendezvous. Because there are a great many factors affecting rendezvous, it is illustrative to consider a modified form of receiver operating characteristic (ROC) chart plotting the likelihood of interference versus the likelihood of rendezvous. We expect various rendezvous algorithms of trace out different regions of this chart. In addition, the modified ROC is parameterized by the propagation environment.

Probability of False Negatives and Impact of False Positives

All communication systems are susceptible to errors. False negatives are also known as misses. An initiating node attempts a rendezvous, and there is at least one other node that could respond; however, it fails to recognize the rendezvous. This at least delays the connection. The initiating node may continue to attempt rendezvous, wasting energy and polluting the spectrum with futile energy.

False positives occur when a scanning node makes the decision that a rendezvous has been attempted, when it has not. In this case, the scanning node may expend energy attempting to conclude the rendezvous process and again pollute the spectrum by emitting energy that is not serving a useful purpose. For certain applications, for example where mains power is available, false positives may be less severe than false negatives, which actually deny the communication mission of the radios.

Immunity (or Robustness) to Denial of Service Attacks

It has long been recognized that dedicated control or beacon channels induce a susceptibility to denial of service attacks. A jammer at this frequency can prevent the entire secondary network from rendezvous. This problem influences the general algorithm design and has led to both wideband rendezvous techniques and narrowband, hopping sequence based techniques. Few researchers have successfully quantified this metric for particular rendezvous algorithms.

Security to Intercept or Penetration

Some applications, such as military operations, must pay special attention to interception or penetration of networks by adversaries. Typically spread spectrum techniques are used to provide low probabilities of intercept and encryption is applied to prevent data loss (or conversely data spoofing) in the event the physical (PHY) layer is compromised. They also use the minimum power necessary to achieve low error rates and when appropriate, focus the radiated energy using directional antennas. Spread spectrum poses unique problems to DSA applications and blurs the concept. This metric is difficult to quantify without specific context; although we can consider the issues qualitatively. The rendezvous process presents another vulnerability to attack and should be secured. Algorithms that use a low probability of intercept (LPI) waveform as probe and response signals are potentially superior to those using waveforms with distinct features.

Those techniques that use a common signaling channel can use encryption and spread spectrum techniques to improve upon the case of clear, narrowband beacons. Sequence hopping techniques present some problems since a rouge transmitter or receiver only has to happen upon the same channel as an authorized node to signal a rendezvous action. While hopping sequence based rendezvous research has focussed on finding sequences that minimize the expected and maximum time to rendezvous, random hopping can still achieve rendezvous (on average, although there is no maximum). Wideband techniques that emit probe or attention signals in multiple channels simultaneously actually communicate information about what channels it finds clear. This information can potentially be exploited by an adversary. We design our rendezvous algorithm to start with minimal power and duration and gradually increase those parameters until rendezvous is achieved. Furthermore, we present a probe and response signal based upon chaos modulation, which exhibits low probability of intercept.

Complexity

Complexity is always a practical consideration in wireless communications. Here we are concerned mainly with the impact on the hardware expense of the system. Complexity can impact power consumption and time, but these are captured by other specific metrics. One should exercise caution when evaluating this metric. The rendezvous algorithm is not a core mission function, but rather a support function. The complexity of the system may be driven by the core communication function. In this case, provided that the rendezvous approach uses similar algorithms and hardware as the core communications, there may not be a significant system level impact, even when the rendezvous itself is complex. On the other hand, we must be careful that the hardware performance demands of the rendezvous does not exceed that of the communications. This is relevant in the FP-BLR. Both FP-BLR and OFDM use FFTs in their processing. Typical frame sizes in OFDM are 512-2048. This is short for the FP-BLR. Additionally, the wide bandwidth nature of the frequency domain scanning means that sample rates must be sufficient to satisfy the Nyquist condition. At first glance, this might seem to be a problem since increasing sample rate while maintaining dynamic range results in very costly data converters, particularly the analog to digital converter. In practice, however, considerations such as crest factor reduction and digital predistortion, both commonly used functions in OFDM systems, require higher sample rates with good dynamic range. It is very possible that these other considerations already justify the added expense of faster sample rates.

On the other hand, frequency hopping systems have their own set of challenges. These include expensive synthesizer sections and a very challenging power amplifier linearization problem, making compensation of nonlinear memory effects especially challenging. It should be noted that after the signal is hopped, the remaining RF chain, whether analog or digital must be wideband. Cognitive radios for SU paradigms need to be to change channels quickly, so a fast synthesizer is probably available. On the other hand, in order to achieve rendezvous quickly, a very fast hop rate is required. This may place additional burdens on the synthesizer, well beyond the core communications, possibly putting the device in a steep cost versus performance curve region. There are few absolutes in this metric; however, careful attention to it will likely pay dividends in a cheaper system.

Energy Required in Standby

Blind link rendezvous algorithms rely upon nodes that may not be actively seeking to communicate. Instead of passively listening on a single assigned channel, these nodes must scan the operating band for signals coming from those nodes that do want to communicate. If the radios are very active, this passive scanning is a small fraction of the time, and its impact is small. If the radios are rarely used, then the power in scanning is a much higher fraction of the total power budget. The transmitter power amplifier normally dominates a radio's power budget for all except very short range radios. Nevertheless, if the scanning operation involves complicated processing, the unit is battery operated, and it spends a lot of time in scanning mode, the power drain can be significant. For example, most Bluetooth devices must be manually placed in page scan mode in order to prepare for rendezvous rather than continuously scanning. Algorithm complexity can be calculated and used to estimate the power consumption, but accurate and precise analysis must involve physical implementation.

The ideal link rendezvous algorithm design optimizes some weighted sum of these metrics. It is easy to imagine some application domains where time to rendezvous is the most critical metrics and other domains where minimizing the likelihood of interference to primary users is the most important metric.

Feature Detector

Our early work on detecting the probe signal examined the spectrum at discrete locations for the expected tones and compared their relative amplitudes. If they fell within a tolerance window of the expected amplitudes, the algorithm concluded the presence of a probe signal. The decision statistics are ratiometric in nature and can be summarized as in equation (3.3).

$$H1 : \bigcap_{n=1}^{N} [A_n A_o - tol < S_n < A_n A_o + tol] \tag{3.3}$$

where $H1$ represents the hypothesis that an attention signal is present, A_n is the design relative amplitude for the n^{th} sidetone, tol is a tolerance to account for nonlinearities and some noise margin, S_n is the sidetone amplitude, N is the number of sidetones and A_o is the carrier amplitude. The alternate hypothesis, $H0$ is assumed when any one these conditions are not met and indicates that there is not an attention signal present.

We analyzed the noise by superimposing AWGN at various signal to noise ratios, using the tolerance window as integration limits. Both the carrier and each side tone are independently affected by noise, assumed to be additive white gaussian noise (AWGN) in this analysis. The decisions are made after a magnitude operation; therefore, the gaussian distribution is transformed to a central chi-square distribution with 1 degree of freedom (where the noise is assumed to be real). The probability density function is given in equation (3.4).

$$f(y) = \frac{1}{\sqrt{2\pi y}\sigma_n} e^{-y/2\sigma_n^2} \tag{3.4}$$

where σ_n is the noise variance.

The decision is found by measuring the amplitude of the carrier, then measuring the amplitude at the side tone frequencies. If the relative amplitude between the side tone and the carrier is within the tolerance window, the decision is true, indicating that an attention signal is present. A false positive occurs when the noise at a carrier frequency is higher than the noise at all of the side tone frequencies by an amount equivalent to the specified relative amplitudes. A false negative occurs when the noise affects the carrier and the side tones in such a way that one or more of the signals is pushed out of the tolerance window.

The probability that the side tone amplitude is within the tolerance is conditioned upon the measured carrier amplitude. The conditional probability that the side tone is within the tolerance is found by integrating the noise probability distribution function through the tolerance window. This is illustrated for one side tone in equation (3.5).

$$P(fp|Pc) = \int_{-tol}^{+tol} f(y)\, dy \tag{3.5}$$

where $P(fp|Pc)$ is the probability of a false positive (fp) given a measured carrier amplitude (Pc), and tol is the tolerance used for the decision.

The total probability is found by multiplying the conditional probability by the probability of measuring that particular carrier amplitude and integrating the product from zero to infinity. This is described by equation (3.6). For more than one side tone, the probability decreases as in equation (3.3).

$$P(fp) = \int_{0}^{\infty} P(fp|Pc) p_c(x)\, dx \tag{3.6}$$

where $p_c(x)$ is the probability density function for the noise at the carrier location.

Equation (3.6) can be used to calculate the probability of false positives versus the relative amplitude of the tones as in Figure 3.1. We also calculate the probability of false positives versus SNR in Figure 3.2. Finally, the probability of negatives versus the tolerance window is shown in Figure 3.4.

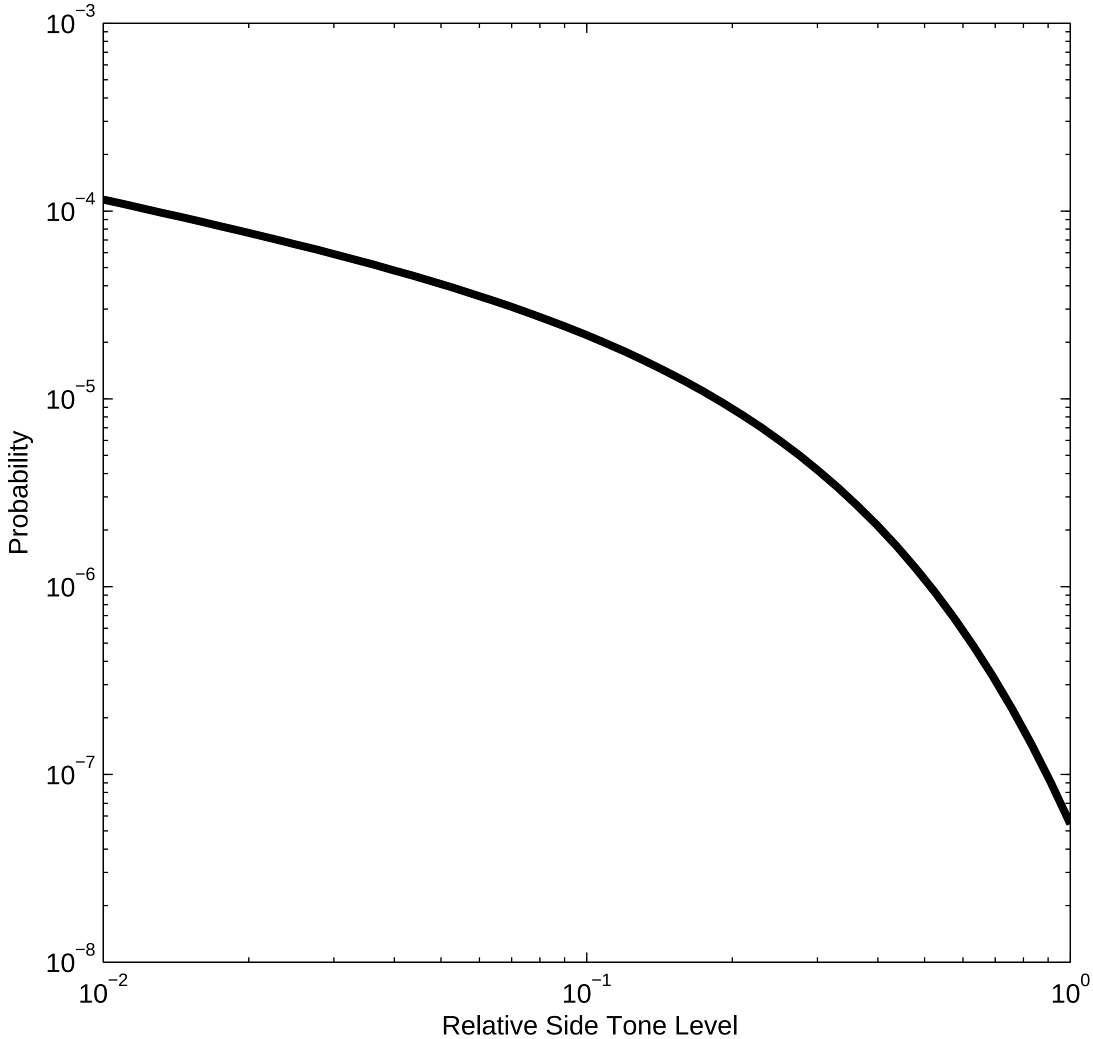

Figure 3.1: Probability of false positive versus relative amplitude of tones

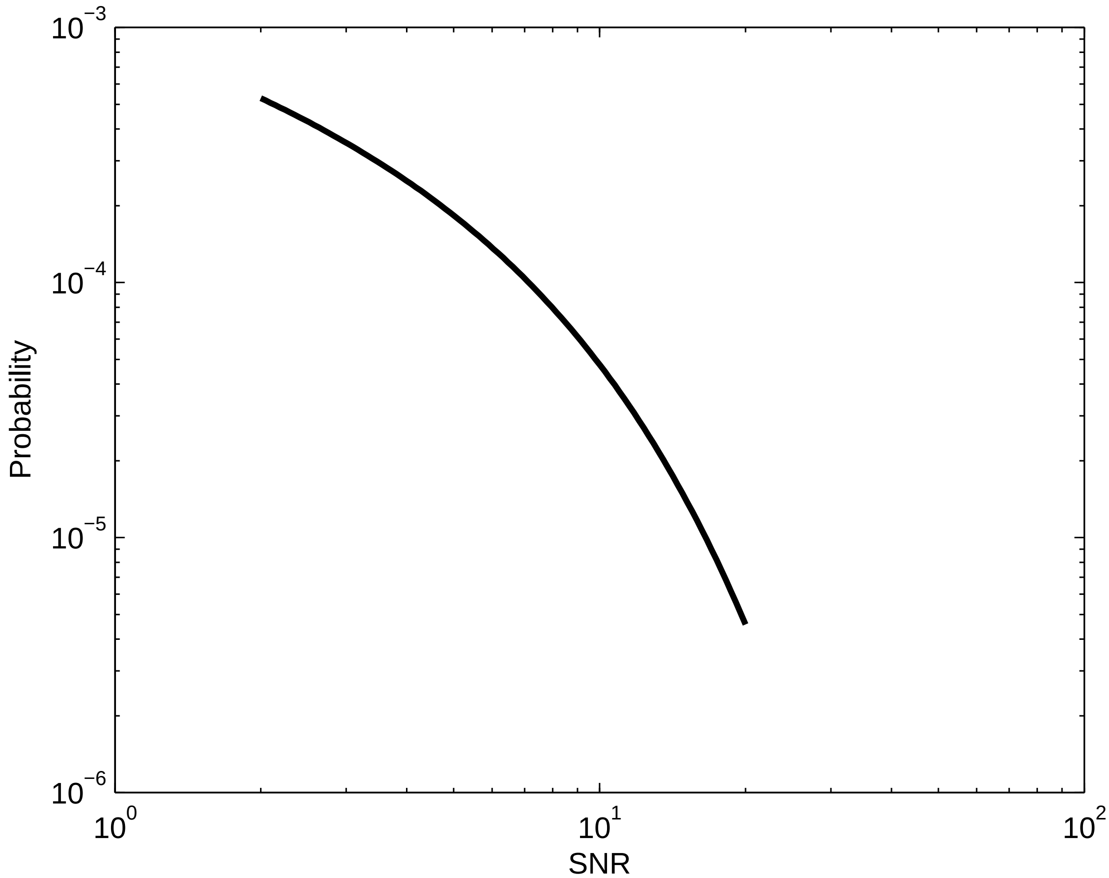

Figure 3.2: Probability of false positive versus signal to noise ratio

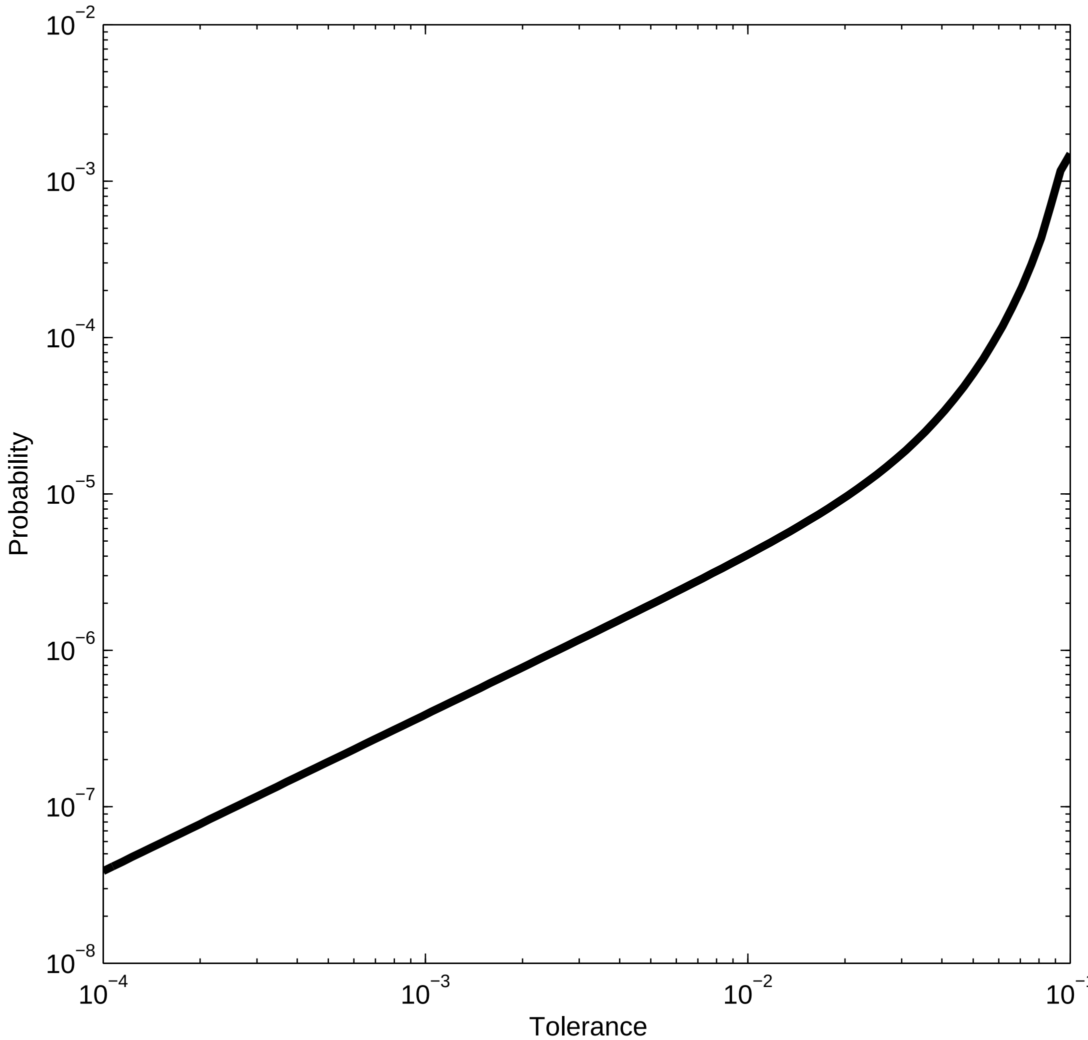

Figure 3.3: Probability of false positive versus tolerance window

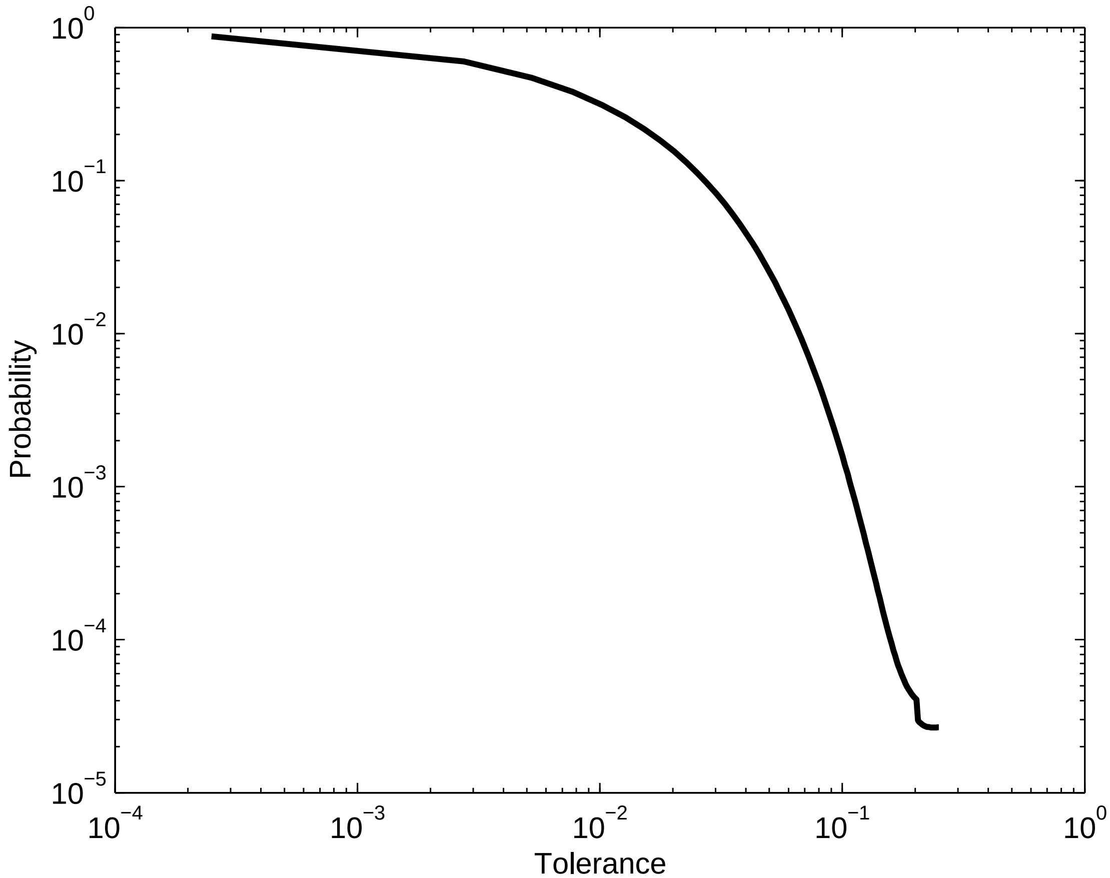

Figure 3.4: Probability of false negative versus tolerance window

The probability of suffering a false positive decreases as the side tone level increases towards the reference tone. Although having a distinctly different amplitude on the side tone may produce a discrimination ability relative to a primary signal, using larger differences buries the side tone in the noise. As expected, the probability of false positives decreases with increasing signal to noise ratio. The probability of a false positive increases with wider tolerance windows. Conversely, the probability of false negatives decreases with wider tolerance windows. All of these are for a pair of tones; one reference and one side tone. Using additional tones improves the results as will be seen in later simulations.

FP-BLR Simulation Study

In order to validate our ideas, we developed two different simulation studies. In order to demonstrate how this can be used in practice, we extended the open source software defined radio, GNURadio [30]. We also developed a number of tools in Matlab, enabling us to study the intermediate steps and performance in great detail.

GNURadio

The GNURadio platform is composed of a library of C language routines that are connected together according to a flowgraph described in Python code. It has device drivers to send and receive data to hardware digital upconverters, downconverters, analog to digital converters, and digital to analog converters to actually transmit and receive waveforms over the air. It can also be configured in a loopback topology with simulated channels. The feature detector was built according to this latter configuration. This required developing new functionality in C to search the spectrum for the probe signal [4]. Much of the rest of the functionality is achieved through Python code. These tests focused on detecting the attention signal without *a priori* knowledge of the center frequency beyond a band specification. The enhanced platform also provided an empirical understanding of how the choice of window function, sidetone amplitude tolerance and number of sidetones affect the detection limits.

The parameters for the test are listed in Table 3.1. The tolerance and gaussian noise amplitude were varied during the individual tests as was the window function applied during the FFT. The frequencies and amplitudes have relative dimensions. The same results are achieved if the units are kHz, MHz, or GHz. The sidetones are chosen so that they are separated by an integer number of bins from the carrier, based upon the sampling frequency and number of FFT points.

Table 3.1: Nominal Experimental Parameters

Parameters	**Value**
Carrier Frequency	1000
Sampling Frequency	2500
Carrier Amplitude	1.0
Offset Frequency 1	12.207
Offset Frequency 2	24.414
Offset Frequency 3	48.828
Relative Sidetone Amplitude	0.20
FFT Length	8192

Averaging was not utilized. Effectively, the transmission time was equivalent to N/F_s. For convenience, the system built up attention signal from separate RF signals, rather than modulating an RF carrier with the baseband tones. Gaussian noise was added at the source. No filtering was applied, so there is some noise being aliased into the measurement.

Matlab Simulation Study

We developed a simulation in Matlab to evaluate the performance of the Multi-Tone probe signal in both AWGN and multipath channels. In each trial, a set of clear channels is chosen from a uniform distribution. Probe signals are transmitted in each available channel. The composite signal is then corrupted by the channel. The receiver Fourier transforms the received signal. The resulting spectrum is correlated with a copy of the uncorrupted probe signal. The minimum magnitude of the correlation is subtracted from the maximum magnitude of the correlation and a threshold is set by a variable fraction of the difference. The default fraction is 0.65. Each correlation peak above the threshold is then labeled a hit. Recorded hits are binned into the original channels and compared to the transmitted channels. True positives are indicated when the binned hits match the transmitted channels. False positives are indicated with a binned hit, but no transmission. Finally, false negatives are indicated when the signal was transmitted, but a hit was not recorded at that channel. The True Positive and False Negative probabilities are found by dividing by the total number of transmission locations while the False Positive probability is found by dividing by the total number of channels.

$$Pr(TruePositive) = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{M_n} H_m^i \wedge T_m^i}{\sum_{n-0}^{N-1} M_n} \tag{3.7}$$

$$Pr(FalseNegative) = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{M_n} H_m^i \wedge T_m^i}{\sum_{n-0}^{N-1} M_n} \tag{3.8}$$

$$Pr(FalsePositive) = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{M_n} H_m^i \wedge T_m^i}{N * M} \tag{3.9}$$

Where H_m^i is the estimated decision and T_m^i is the actual.

Illustrative Example

GNURadio Example

As described previously, the time-truncation of the attention signal causes a degradation of the frequency resolution and the appearance of leakage. The degradation of the frequency resolution is due to the main lobe widening. If the signal was infinitely long and the analysis was likewise infinite, the carrier and sidetones would appear as infinitely narrow pulses in the frequency domain. Instead, the main lobe and the sidetone lobes are broadened significantly. The broadening is smooth and consistent between the carrier and the sidetones. This results in false positives in frequency bins adjacent to the carrier bin, since the test is based upon a relative amplitude. Fortunately, these false positives appear symmetrically around the true carrier. A post processor could be used to perform a frequency average. The performance as the number of sidetones varies is summarized in Table 3.2. It lists the number of false positives found in a single scan of the spectrum. A rectangular window was used with 8192 FFT points. As the noise increases, the single sidetone has more false positives. On the other hand, the double and triple sidetone versions manage to properly identify the signal with no false positives. The tolerance was held constant at 0.1. This causes an increase in false positives at the higher signal to noise ratios. An adaptive approach to setting the tolerance would drive the false positive rate to near zero for low levels of noise. Clearly, the double sidetone version performs better than the single sidetone; however, there is little to be gained with the triple sidetone. It should be noted that the single tone performance can be significantly improved at moderate to low noise levels by adjustment of the tolerance value. Additionally, the false positives at the lower noise levels are generally in adjacent bins.

Table 3.2: False Positives Versus Noise by Number of Sidetones

Noise Amplitude	**Sidetone 1**	**Sidetone 2**	**Sidetone 3**
0.01	21	7	7
0.1	48	8	6
0.5	35	2	1
1.0	30	0	0
2.0	33	0	0
3.0	33	0	0

Matlab Example

This section presents an illustrative example to highlight to advantages of the FP-BLR protocol. We use a total bandwidth of 100 MHz, a channel spacing of 1 MHz (resulting in 100 total channels), and a sampling frequency of 204.8 MHz (convenient to a FFT array size of 2048 points). In each of 10,000 trials, we select a number of available channels from a uniform distribution. The received signal is Fourier transformed using a FFT size of 2048. The minimum magnitude of the correlation is subtracted from the maximum magnitude of the correlation and a threshold is set by a variable fraction of the difference. The default fraction is 0.65.

The probe signal consists of 5 tones, centered about the channel center frequency and spaced at 100 kHz intervals. The outer pair of tones are at a normalized amplitude of 1.0. The center tone is at a normalized amplitude of 0.001, while the middle pair are set at 0.02.

We first examine an AWGN channel without multipath or primary user interference. Fig. 3.5 illustrates the probability of true positive probe signal detection versus signal to noise ratio with a threshold fraction set at 0.65. As expected, the probability approaches 1.0 with a stronger signal. There appears to be a floor at low SNRs, which is a consequence of how the threshold is determined. Since the threshold is determined by the received correlated spectrum, eventually, the algorithm will select hits on many channels due to noise alone. Many spectrum sensing algorithms, especially those based upon energy detection, require a noise estimate. This could then be used to provide a better ground truth for the threshold. The Receiver Operating Characteristic (ROC) is often used to illustrate the tradeoff between True Positives and False Positives as a function of the threshold. The results for the AWGN channel are presented in Fig. 3.6. The curve exhibits a sharp transition, which allows the choice of a threshold that produces a high rate of correct decisions, with a false positive rate on the order of 0.08.

We also examine the false negatives versus SNR in 3.7.

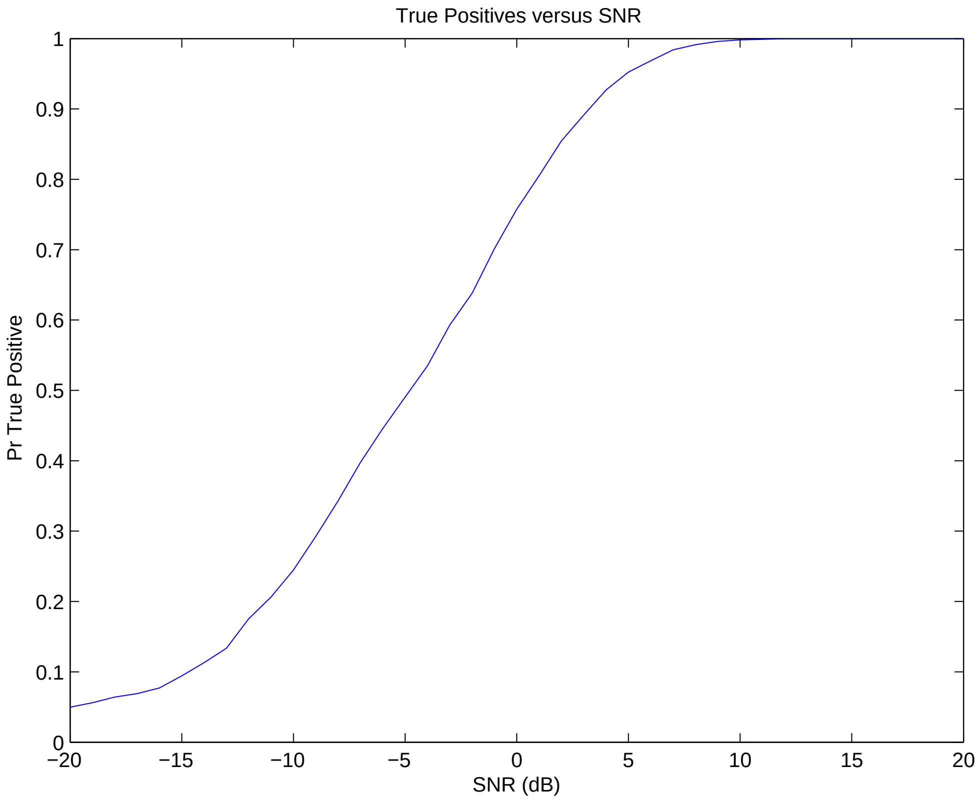

Figure 3.5: Probability of Probe Signal Detection Versus SNR

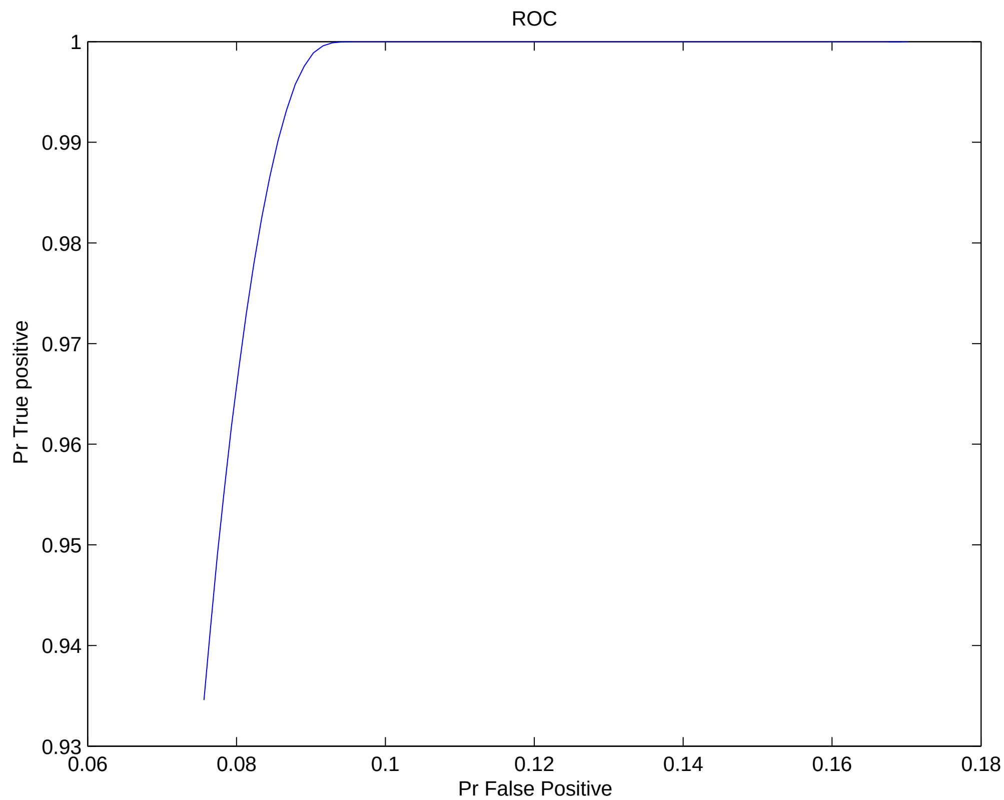

Figure 3.6: Receiver Operating Characteristic for Multi-Tone Probe Signal with Varying Threshold

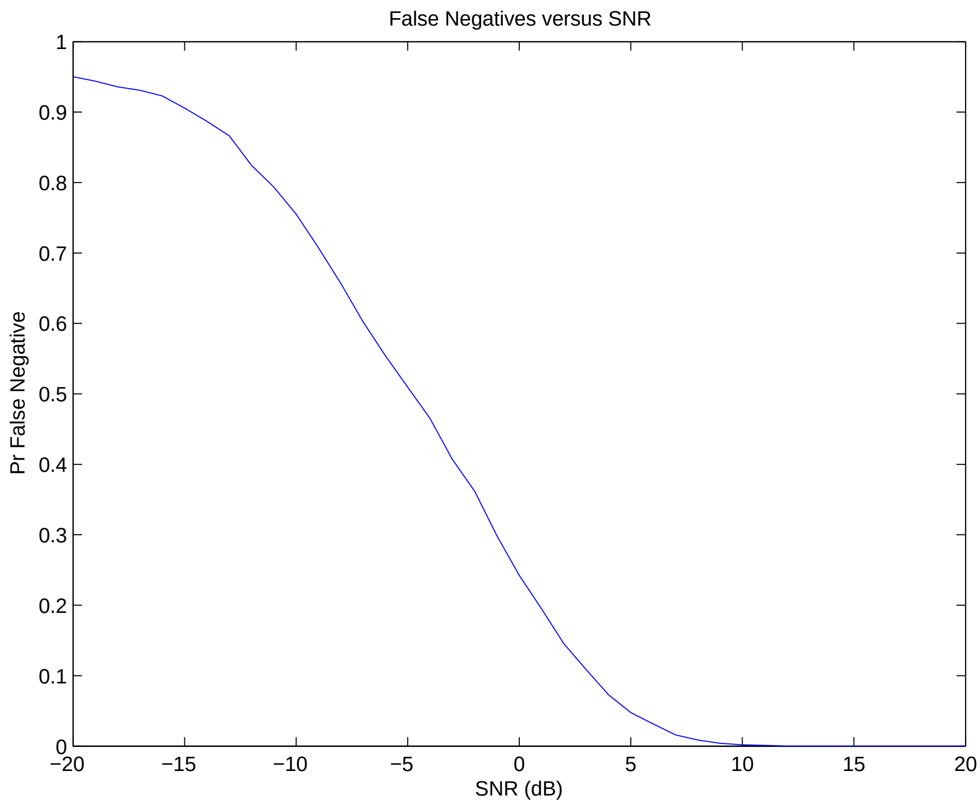

Figure 3.7: Probability of False Negatives Versus SNR

We created a multipath channel similar to COST 207 with a vehicle speed of 166 km/hr, a carrier frequency of 2.45 GHz, and paths according to Table 3.3. We repeat the analysis and present the results in Fig. 3.8. The sharpness of the ROC transition is degraded compared to the AWGN case and the curve moves to the right; however, it is still clear that a proper threshold can be set with a good tradeoff. This is clarified in 3.9.

Table 3.3: Multipath Tap Delays and Weights

Tap	1	2	3	4	5	6
Delay (usec)	0.0	0.4	1.0	1.6	5.0	6.6
Weight (dB)	-3	0	-3	-5	-2	-4

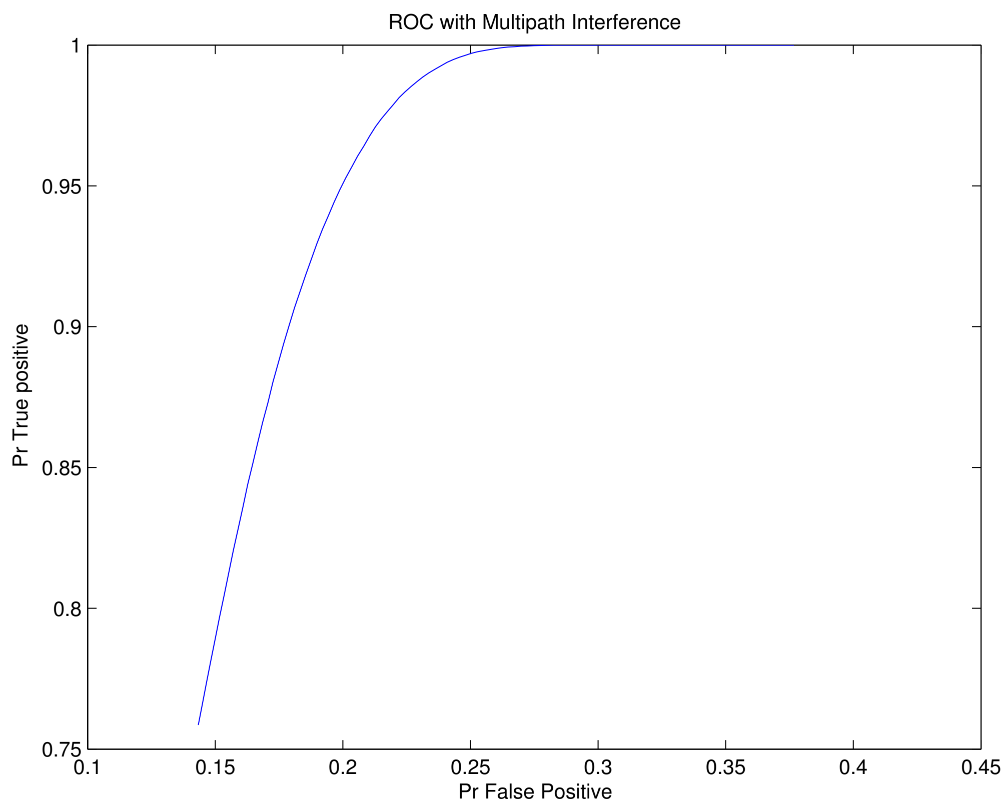

Figure 3.8: Receiver Operating Characteristic for Multi-Tone probe signal with varying threshold

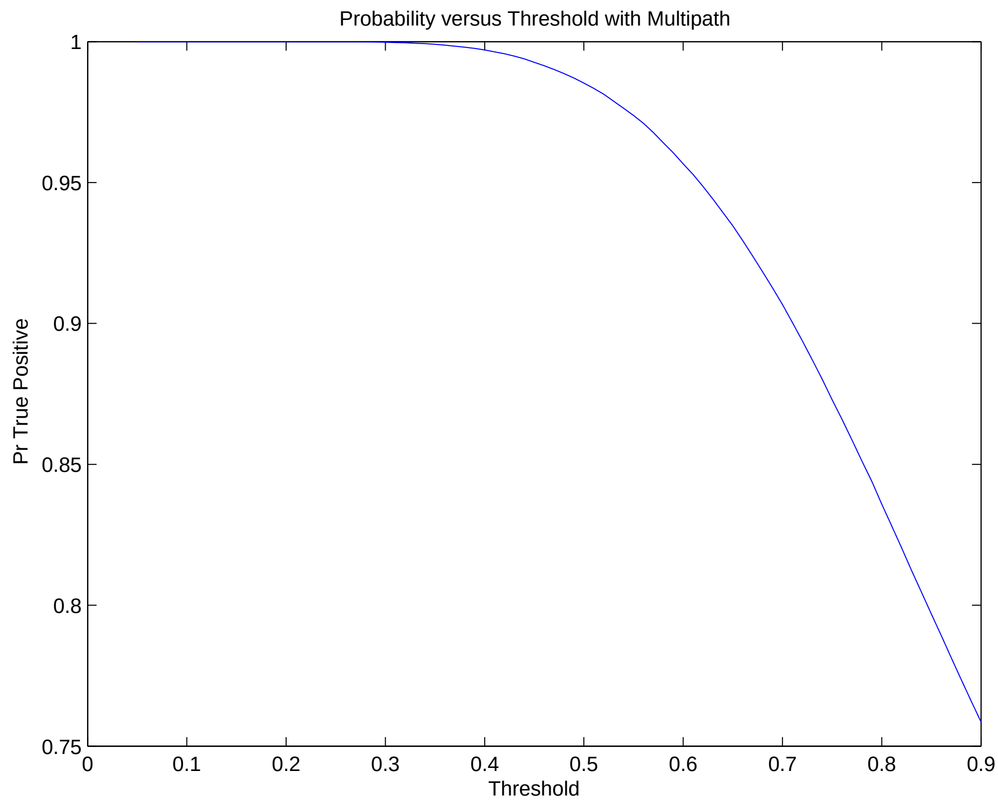

Figure 3.9: Probability of Probe Signal Detection Versus Threshold with Multipath

CHAPTER 4: CROSS CORRELATION DETECTOR FOR BLIND LINK RENDEZVOUS

The feature detector only looks where it expects the sinusoids. This means the technique is ignoring the fact that a lack of energy at a particular frequency is also information. A cross-correlation based detector can potentially extract more information. Ideally, we could create probe signal based upon a code such as a Barker code, which exhibits excellent distance between the correlation peak and the signal at other lags. The Barker code can produce processing gain depending upon its length. This requires decoding phase information and leads to a more complex receiver. Depending upon the bandwidth and the complexity of the channel, an equalizer may be required in order to achieve the processing gain. Alternatively, a receiver could simply use the magnitude of the signal and ignore the phase. Since simplicity is especially important during the rendezvous phase, this latter approach is chosen.

As before, the received signal is Fourier transformed. Instead of then examining each candidate tone location for the expected amplitude, the result is correlated with the Fourier transform of the probe signal (4.1) as in (4.2). This produces a correlation peak everywhere the probe signal was transmitted. In this and following equations we assume only one channel is probed for clarity.

$$P(k) = \sum_{n=0}^{N-1} p(n)w(n)e^{-j2\pi kn/N} \tag{4.1}$$

where N is the transform length and p is the probe signal. Here, k is the frequency parameter and $k \in [0, K-1]$, where $K = N$ and is the number of FFT points used in the transform. It is important to keep the same frequency spacing for both the received signal and the probe prototype signal.

$$\hat{R}_{X,P}(l) = \begin{cases} \sum_{k=0}^{K-l-1} X(k+l)P^*(k) & \text{if } l \geq 0 \\ \hat{R}^*_{X,P}(-l) & \text{if } l < 0 \end{cases} \tag{4.2}$$

At this point, one observes that the performance of the cross correlation detector is a function of at least three issues. The first is the noise performance of the detector. The second is the auto-correlation properties of the probe signal. The third is the spectral analysis technique used. So that we may focus on the noise performance in this analysis, we will simplify the transformed signal to consist of spectral lines only. This results in equations (4.3), (4.4), and (4.5).

$$X(k) = C(k) \sum_{j=1}^{J} A_j \delta(\Omega_j - k) + \eta(k) \tag{4.3}$$

$$P(k) = \sum_{j=1}^{J} A_j \delta(\Omega_j - k) \tag{4.4}$$

$$\begin{aligned} \hat{R}_{X,P}(l) &= \sum_{k=0}^{K-l-1} X(k)P(k+l) && (4.5) \\ &= \sum_{k=0}^{K-l-1} \left\{ \left[C(k) \sum_{j=1}^{J} A_j \delta(\Omega_j - k) + \eta(k) \right] \left[\sum_{q=1}^{Q} A_q \delta(\Omega_q - k + l) \right] \right\} && (4.6) \\ &= \sum_{k=0}^{K-l-1} \left\{ C(k) \sum_{j=1}^{J} A_j \delta(\Omega_j - k) \sum_{q=1}^{Q} A_q \delta(\Omega_q - k + l) + \eta(k) \sum_{q=1}^{Q} A_q \delta(\Omega_q - k + l) \right\} && (4.7) \end{aligned}$$

We have used different summation indices for the received signal (j) and the probe (p). Also, $C(k)$ represent the channel gain at each frequency, and is assumed to be flat over the bandwidth represented by a single spectral sample. The tone frequencies in the sampled domain are represented by Ω_j and Ω_q. Finally, the A_j and A_q are the magnitudes of the signal and probe respectively, implying that we are working with real values at this point. This is so we do not have to perform any equalization or channel estimation and can ignore the phase. The spectral lines are represented by Kronecker delta functions, δ which for our purposes have a magnitude of 1 when its argument is zero.

The first term is only nonzero when $\Omega_j - k = \Omega_q - k + l = 0$. Obviously, this happens when $l = 0$ and $\Omega_q = \Omega_j = k$, but it also occurs when $l = 1$ and $\Omega_q = \Omega_j - 1$ at certain k values. Likewise, the condition is also satisfied when $l = -1$ and $\Omega_q = \Omega_j + 1$. These alternative conditions (besides $l = 0$) lead us to observe that the auto-correlation properties of the probe signal are important. We can also observe that when $l = 0$,

$$\hat{R}_{X,P}(0) = \sum_{j=1}^{J} A_j^2 + \sum_{j=1}^{J} A_j \eta(k) \tag{4.8}$$

where we have assumed $C(k) = 1$ for clarity. Here we observe that the cross-correlation of multiple tones is effectively signal averaging. In other words, we are coherently adding the signal tones whereas the noise samples tend towards a zero mean. It is well known that signal averaging of a stationary signal in the presence of zero mean noise improves the signal to noise ration at a rate of $\sqrt{J}$. When viewed alone, this implies that we can improve our SNR, and therefore our detection range, by increasing the number of tones per channel; however, this is not without cost. The transmitter may be constrained by either total power transmitted, which will result in a decreased per tone power with additional tones, or it may be constrained by an interference condition. Modern wideband waveforms of primary users are designed to tolerate narrowband interference (or conversely a narrowband multipath induced channel response null), with minimal impact to bit error rate. This is especially the case with OFDM where the redundancies associated with error control coding are normally spread to different subchannel frequencies. This protection will decrease with the number of tones in the probe signal. Finally, for a fixed channel bandwidth, increasing the number of tones reduces the frequency spacing, placing additional burden on the spectral analysis algorithm to resolve each tone.

Finally, the energy in each frequency channel is compared to a threshold to determine hits. This threshold can be determined using Receiver Operating Characteristic curves as will be shown later. In order to illustrate the inner workings of the algorithm, we created a scenario with 20 total channels. We randomly chose 5 available channels and transmitted a probe signal in each of those as shown in Figure 4.1. In Figure 4.2, we give the spectrum of the received signal with a signal to noise ratio of 0 dB.

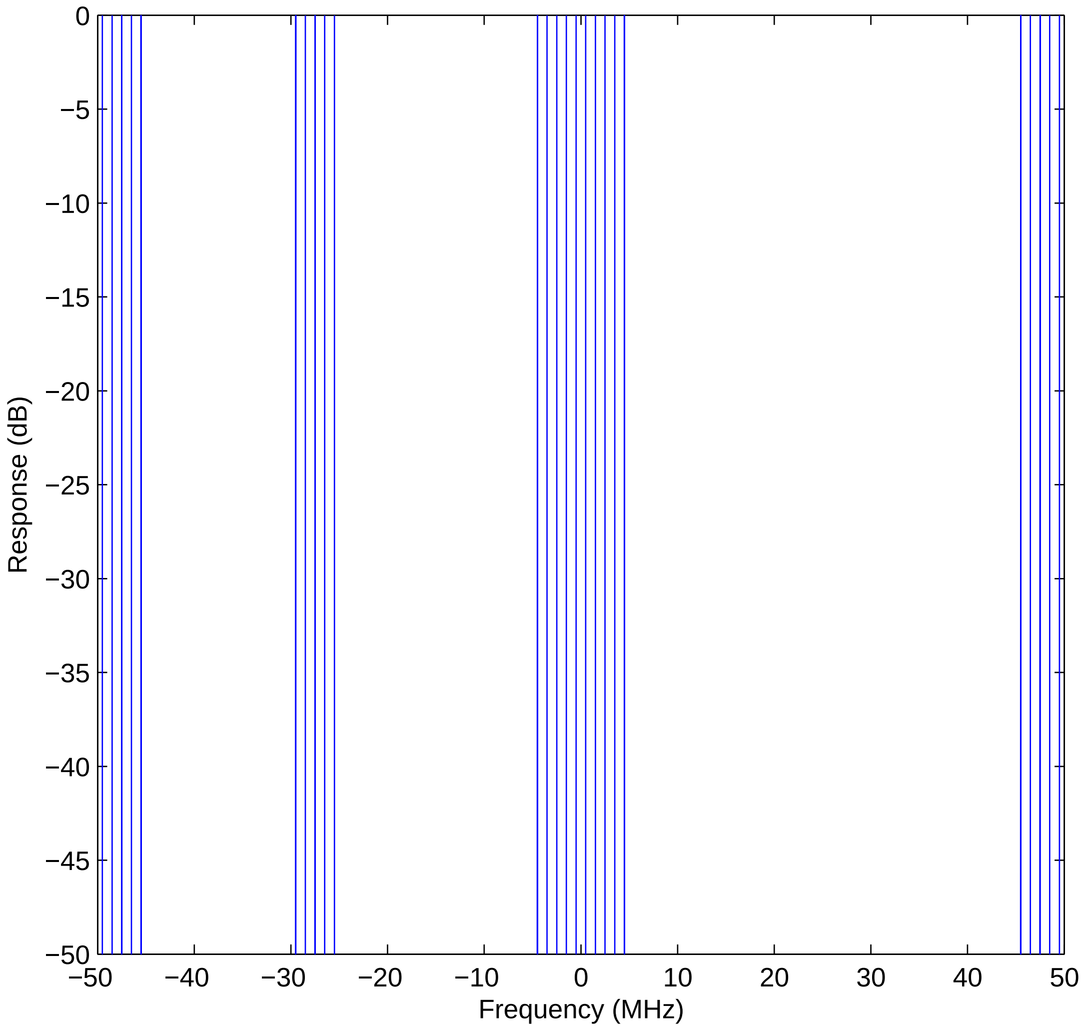

Figure 4.1: Spectrum for 5 Sent Channels as Transmitted

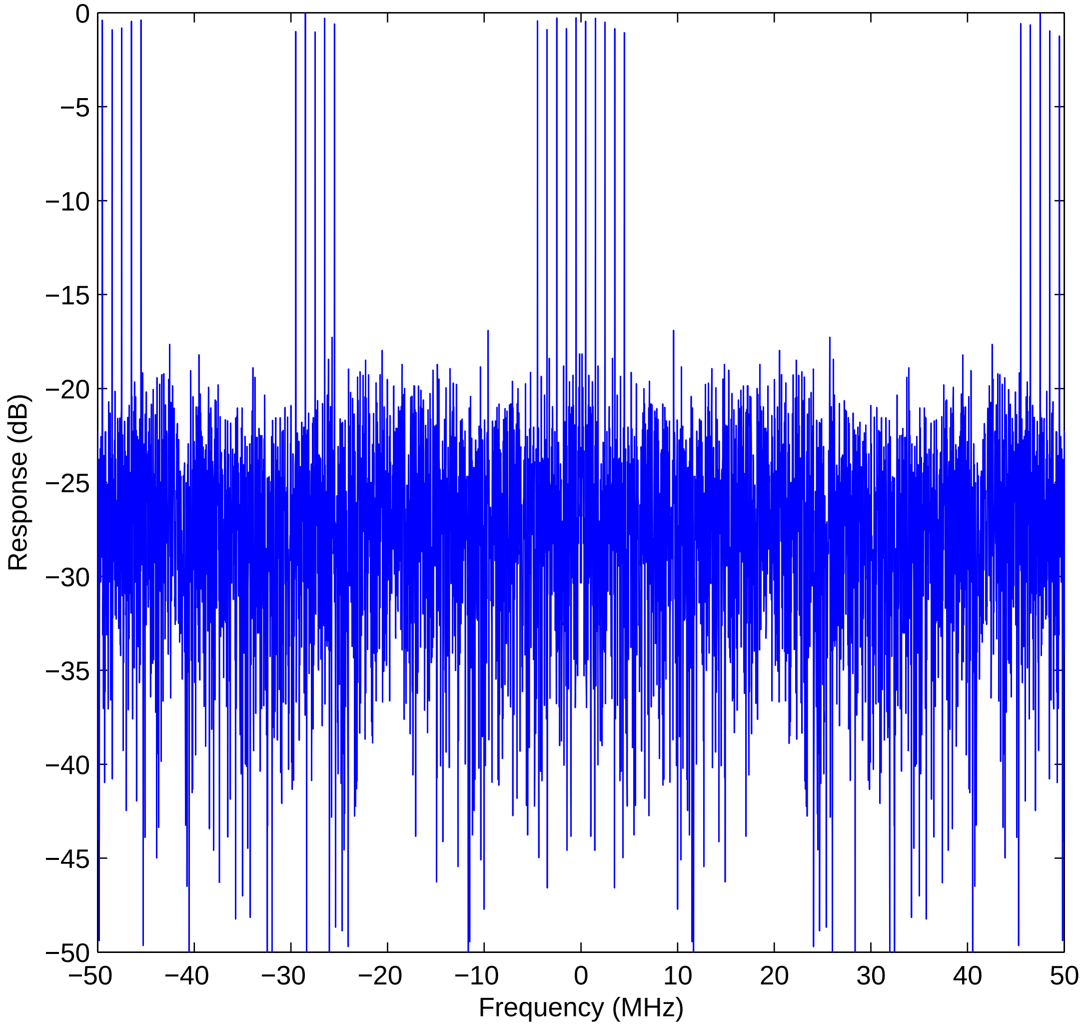

Figure 4.2: Spectrum for 5 Sent Channels Received at 0 dB SNR

The output of the cross correlator is presented in Figure 4.3. The threshold in this case is derived by multiplying the difference between the maximum and minimum value by 0.65. This *ad hoc* approach to the threshold yields reasonable results. The hits are shown in

Figure 4.4.

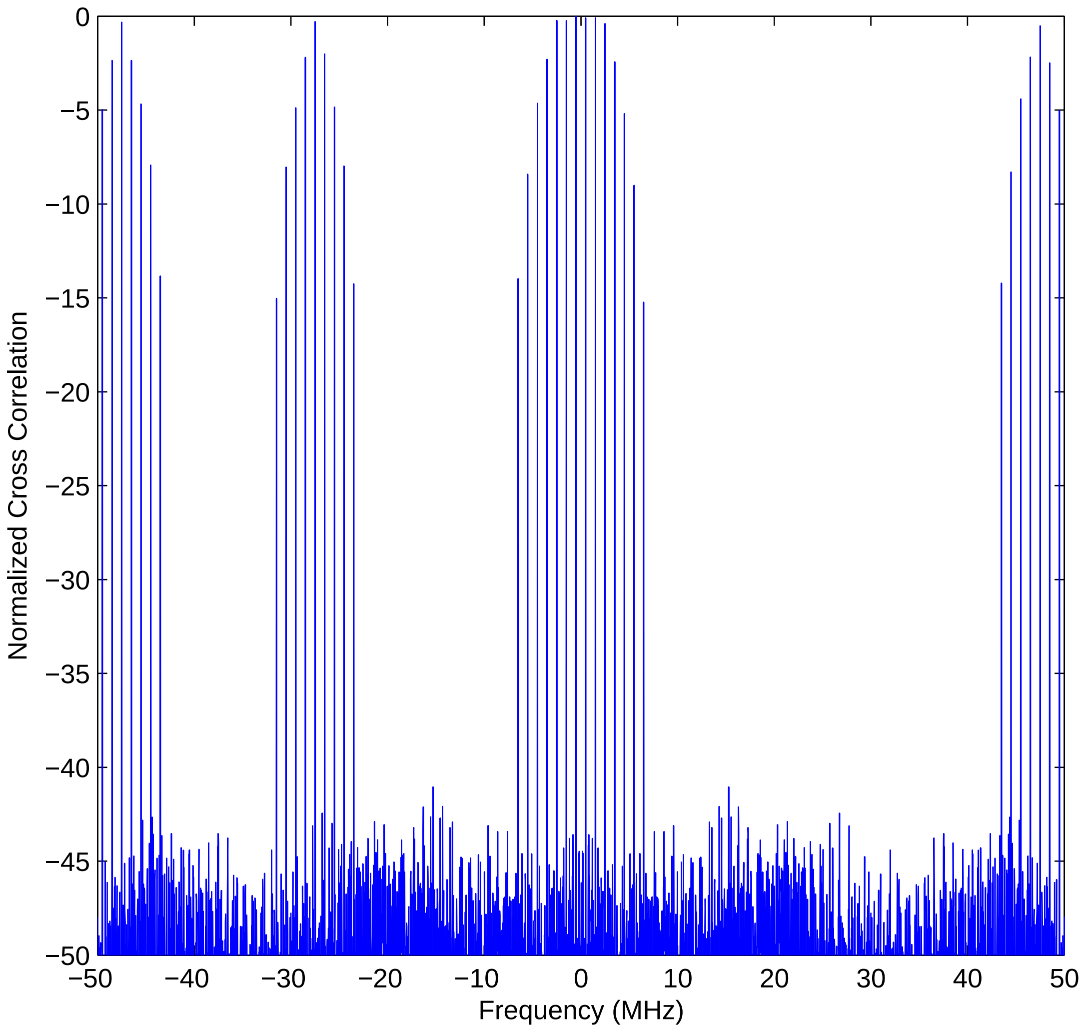

Figure 4.3: Cross-Correlation for 5 Sent Channels at 0 dB SNR

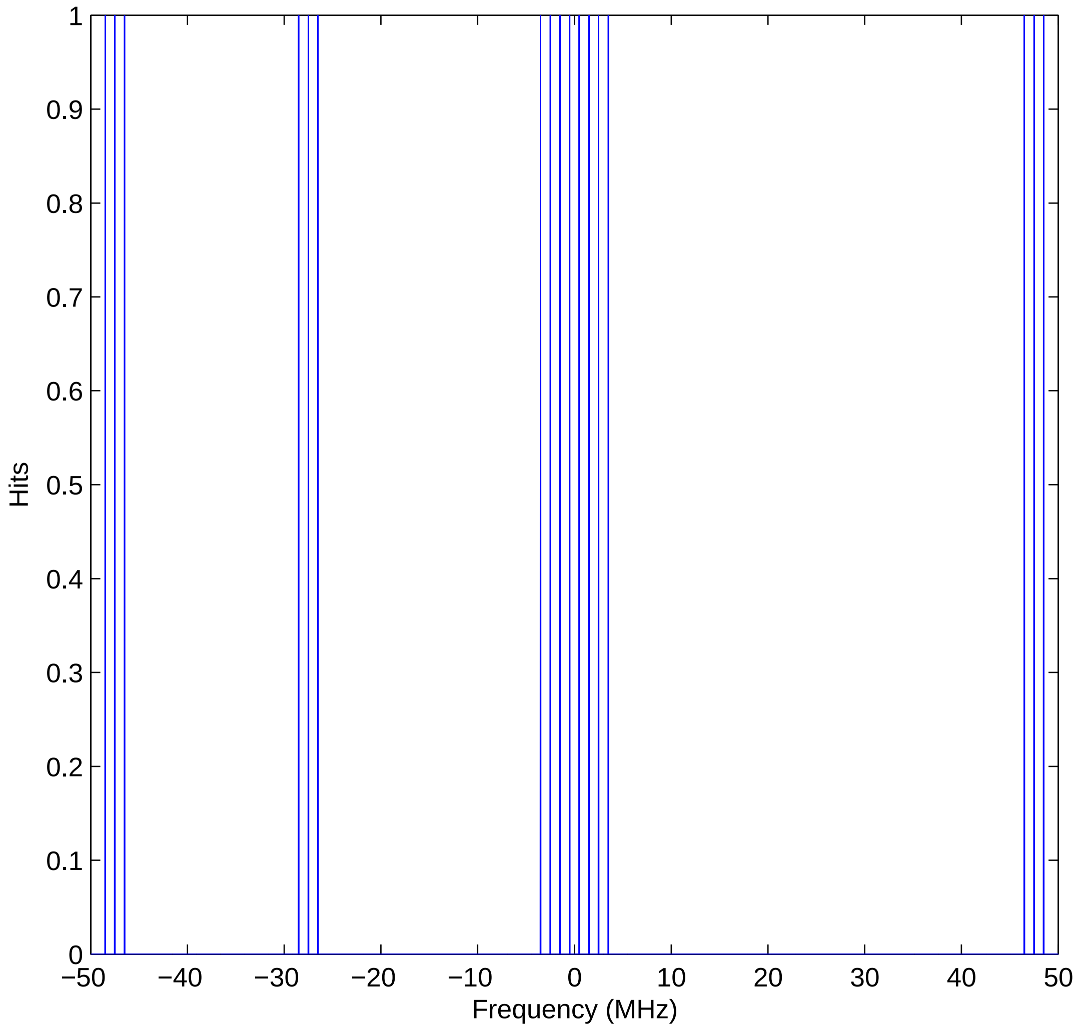

Figure 4.4: Hits for 5 Sent Channels at 0 dB SNR

Common Detector Issues

We have described two possible detectors for our probe design. There are some considerations in common between them. We will briefly discuss the options for performing the transform into the frequency domain, followed by a discussion of the pros and cons of each detector.

Design Parameters

The multiple tone rendezvous scheme we presented in [4] and [31] is elegantly simple. The tones are easy to generate in the transmitter and easy to detect in the receiver. A software defined receiver can adapt to a bandwidth covering many channels, sample the received signal, and perform a Fast Fourier Transform (FFT) on the signal. The receiver then correlates the frequency domain with a FFT of the probe signal. Peaks in the cross correlation indicate the presence and frequency of the probe signal. Notice that no synchronization is required. On the other hand, it is susceptible to false negatives due to multi-path interference, and false positives due to noise and signals that randomly match the pattern. The problem of false positives due to matching the random fluctuations of a noise signal can be mitigated by proving the match over a longer duration, in other words, more FFT frames. Presumably, the noise will change during this time and cause later tests to fail, resulting in a true negative.

The false positives due to noise can be mitigated if an absolute amplitude check is done in conjunction with the cross-correlation. The positive hypothesis would only be chosen if the signal strength was above a threshold. The setting of a threshold implies a knowledge of the noise level, which is a common estimation step in spectrum sensing.

Multipath Susceptibility

A potential shortcoming of the multi-tone approach is its susceptibility to multipath distortion. Complicated propagation environments can lead to reflected signals arriving at the receiver with different time delays. These will interfere with each other, constructively at some frequencies and destructively at others. In some cases, the fading due to destructive interference can be severe causing amplitude nulls, although the bandwidth of these fades tends to be small. If this fade occurs at the same frequency as a tone of the probe signal, then the test could fail if the required match is too tight.

Although it is plain to see that this is a potential problem, it is not obvious regarding how severe the problem is. Multipath can affect all wireless systems to one degree or another. In mobile communications, the environmental effects are constantly changing. If a fade causes a problem during one instance, it may not cause a problem during the next instance if either the transmitter or receiver moves. The coherence bandwidth measures the range of frequencies over which the amplitude fading is constant,

$$W_c = \frac{2\pi}{D} \tag{4.9}$$

where D is the delay spread of the various signals arriving at the receiver. Wide delay spreads cause narrow, null-like fades. Our earlier implementation examined each peak individually. A tolerance about each projected amplitude level provided a 0-1 test. We have now implemented a correlation-based algorithm that provides a soft decision, which can be hardened via a threshold. The correlation-based algorithm allows for some distortion because it includes more information, such as the nulls between the peaks.

Receivers often use equalization to solve the multipath problem, but this is a complex operation and greatly diminishes the elegance of the multi-tone approach. Ultimately, if the multipath is so severe that the rendezvous algorithm fails, it may not be a suitable channel for communications, rendering the issue moot. Because our technique calls for the initiator to emit probe signals on several channels (if enough are available), this process automatically selects channels with good propagation conditions.

Spectral Analysis Techniques

Both detectors operate in the frequency domain, necessitating a transform. The most straightforward technique is to use a fast Fourier transform (FFT). It is well known that a Fourier transform on a finite length sequence results in leakage and increased variance [58]. In other words, sinusoids will occupy more than one frequency bin and sidelobes will be present, which may be interpreted as signals by a detector. A finite length sequence implies that the time series has been windowed by a rectangular function. Many other window functions exist with different tradeoffs between the main lobe width (leakage) and the sidelobe level (variance). The rectangular window function has the smallest leakage. Some window functions feature constant sidelobe levels, while others have tapering sidelobe levels. These differences in the window functions arise from the mathematical equation used to define the function. Any of the window functions can be lengthened to narrow the main lobe width. This is equivalent to using a longer time series as input. In the FP-BLR, this means that the transmitter is transmitting longer, consuming more energy and increasing the likelihood of harmful interference. The duration of the tone affects how closely spaced and large of relative amplitude difference is achievable for the pattern. Denote the overall rendezvous bandwidth as BW_r and the individual channel bandwidth as BW_c. The number of channels is then $N_c = BW_r/BW_c$. If we use a FFT size of $NFFT = 2048$ and a sampling frequency of

$F_s = 204.8MHz$, then the bin spacing is $\Delta f = F_s/NFFT = 100kHz$. This is the best that can be achieved; however, spectral leakage means that some of the main lobe energy will be found in adjacent bins. In general, the main lobe width determines how close of frequency spacing can be used in the probe signal. The sidelobe level affects how different the levels of the tones can be before the smaller signal is lost in the sidelobes of the larger signal.

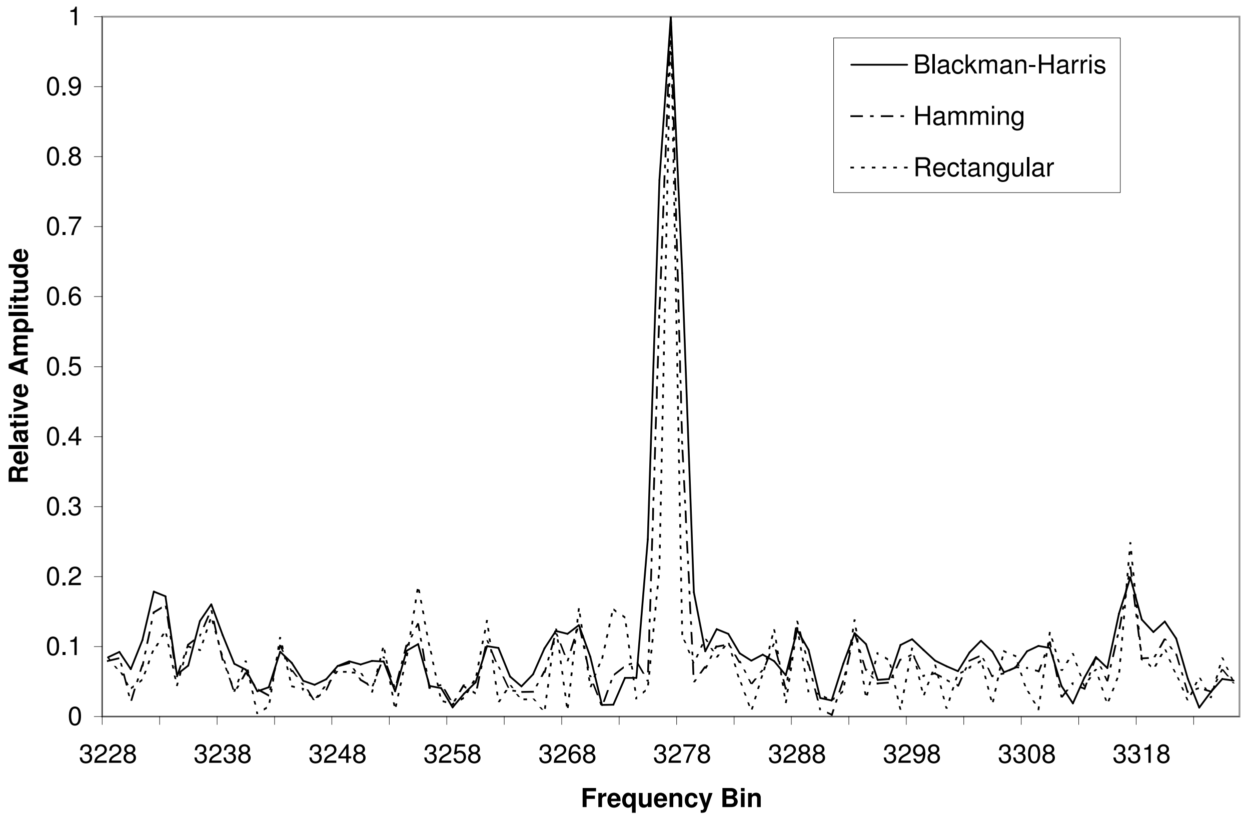

Figure 4.5: Three Window Functions

Since the tones are deterministic, additional or longer FFTs can be employed to extract the signals from the noise using a Welch method. The FFT length N leads to a processing gain.

We conclude that the noise performance can be improved by longer processing time, up to the transmitted time length. The longer the signal is transmitted, the greater likelihood it will cause harmful interference to a primary user, if the spectrum sensing has not been perfect.

$$SNR(l) = N\frac{A^2}{\sigma_v^2} \tag{4.10}$$

where N is the length of the FFT, A is the amplitude of the signal, and σ_v is the noise variance.

We studied a variety of window functions, modifications to the standard periodogram, and parametric methods in order to tradeoff the frequency resolution with the dynamic range. Additional considerations include the ability to average results to improve SNR, the overall complexity, and compatibility with on-board hardware and communication algorithms. For parametric approaches, we found that the Eigen approach provided excellent results. The main challenge in using any of the model based approaches is the need to estimate the size of the model. In out case, this is tightly related to the number of received sinusoids. Unfortunately, this is unknown, since we do not know how many channels will be probed. Furthermore, we will likely have waveforms from primary users in the band also. The result is not only a complicated spectrum, but one that we cannot easily predict the order. We conclude that non-parametric approaches are a poor fit in most realistic applications.

For the scenarios we have studied, frequency resolution has been more critical than dynamic range. This holds even stronger with narrower channel band plans. This is likely to be the case in many secondary use applications, such as we see in the current TV White Space application. In high SNR applications we could take advantage of a high dynamic range probe to offer more discrimination power between primary user waveforms and the probe waveform. This advantage is lost when the SNR is low, because the lower amplitude tone will drop below the noise floor before the overall detection limit is achieved. For these reasons, we favor the non-parametric periodogram method with a rectangular window for its good frequency resolution and compatibility with hardware and software that supports OFDM communications. The Welch method of overlapping and averaging multiple FFT frames offers superior dynamic range, but degraded frequency resolution. It may be useful in those applications that have wideband channels and favor waveform discrimination.

Finally, there are some application specific opportunities for certain transforms. Our primary motivation for exploring short time transforms is to minimize the required length of transmission, not the computation complexity. On the other hand, the sliding Goertzel algorithm [59],[60] can be used to examine a relatively small number of frequency locations in a large spectrum with very good computational efficiency compared to a full FFT. This could be used to reduce the complexity and power consumption of mobile radios.

Comparison of Detectors

We have presented a feature detector and a cross-correlation detector. In the context of our considered applications, it is not possible to declare on completely superior to the other. They each have their advantages and disadvantages. Both require the setting of a threshold or tolerance in order to minimize false positives while maximizing true positives. Setting the threshold very low or the tolerance very wide, ensures that all of the probes will be detected; however, this will also result in many false positives triggered by noise or the primary user's waveform. Both can be made insensitive to the phase, avoiding the need to estimate the channel and also providing the opportunity for the transmitter to adjust the phase to minimize the PAPR of the composite signal. This is particularly important since digital pre-distortion (DPD) may not be available for the rendezvous waveform. Many radios use digital pre-distortion to effectively linearize the power amplifier chain, resulting in more power efficient amplification. A key step in the DPD process is to oversample the waveform by a factor of 5 to account for intermodulation products out to fifth order. The signal is often upsampled by another factor of at least 2 before sending it to the digital to analog converter. The resulting high sample rate allows us to send our wideband waveform through the same system without having to change the clocking. Changing the clocking would make the partial reconfiguration of the FPGA impractical. On the other hand, since the wideband composite probe signal will be already challenging the Nyquist limits set by the sample rate, it is not possible to perform DPD processing on the probe. This means it should exhibit inherently low PAPR. With flexibility in the transmitted phase, the transmitter can vary the phase in order to find solutions where the sinusoids add in such a way to minimize the PAPR. Since there is a finite number of channels and tones for each probe, and they are all well defined, these solutions can be precalculated and stored in a lookup table for fast access at run time.

Continuing with their advantages in common, the rigor of the $\mathcal{H}_1$ match can be adjusted. For the feature detector, the strict intersection of all sidetone matching can be relaxed by summing a unity value for each matched sidetone. The sum can then be compared to the maximum possible sum, such as 4 out of 5 match. A threshold then determines the resulting hypothesis. This allows for some channel distortion. The relaxation of the cross-correlation detector is even more straightforward. It is accomplished by direct change of the threshold value.

Our simulations indicate that the feature detector is more sensitive to at least some types of multipath interference. This can be mitigated by the relaxation of the matching criteria discussed above. Whether or not this is a negative feature depends upon perspective. If severe enough, multipath interference will result in misses or false negatives. This is an indication of a poor channel; therefore, the miss is a positive event. It will prevent the system from attempting to use the poor channel and favor a better channel. This is an advantage if a better channel exists. This is a function of the band occupancy, and the nature of the interference. It only becomes a negative issue, if there are no other channels available and the interfered channel is not so poor to completely inhibit communications.

The feature detector may offer some advantage when attempting to discriminate between probe signals and PU waveforms, or even against other SU waveforms. If enough SNR is available, the probe can be designed with a large dynamic range. In particular, its power spectral density (PSD) can be made distinctly different from the PU's waveform. For example, if the PU uses OFDM, the PSD will be flat-topped with slightly rounded corners and very steep sides. If the relative amplitudes of the tones are adjusted to present a concave shape over the same bandwidth, the discrimination between the two waveforms will be easier. The feature detector preserves this shape during its operation. On the other hand, the cross-correlation process averages the effect. A higher value in one location can cancel out the effects of a lower value at another location. This leads to weaker discrimination power.

Assuming a fixed band plan, the feature detector will have less complexity than the cross-correlator. The feature detector only has to examine a few specific locations, while the cross-correlator must examine all of them in an iterated manner. For processing constrained hardware, the feature detector can be easier to implement. Also, by minimizing the processing time, frames may be processed at a faster rate. This results in a higher probability of intercept of the probe signal.

Finally, the cross-correlation process provides a floor response. This can be used to estimate a threshold, as is done in the current simulator. The peak is relative to the floors. If the probe waveform is made narrow compared to the channel spacing, one can be assured of a spectrum location that should not have a correlation peak. These locations can be used to establish a lower bound for the threshold. The feature detector does not have this ability. The feature detector can greatly benefit from a noise estimate in reducing the number of false positives. From the noise estimate, it could establish a threshold, below which the feature detection would not occur. The challenge is deriving this noise estimate. Without synchronization with other SU nodes and with the PU nodes, no quiet periods or locations can be reliably found.

In conclusion, both detectors have their merits and shortcomings. Application specific criteria will guide the ultimate choice.

CHAPTER 5: COMPARISON TO LEGACY RENDEZVOUS

In this chapter, we provide a brief overview of the legacy approach to blind link rendeszvous, frequency hopping sequences. We then discuss our approach in comparison to the legacy. Finally, we summarize our research in FP-BLR.

Frequency Hopping Blind Link Rendezvous

Many of the existing rendezvous techniques involve frequency hopping to channels according to specific sequences [24][26][28], or even random sequences [23]. Researcher in this area have typically assumed that the dwell time at each frequency hop is a constant across all protocols, so they ignore it in their analysis. While this is a legitimate assumption when comparing against frequency hopping protocols, it creates a problem in comparing to the FP-BLR protocol. Equating the two approaches in a tradeoff between range and energy required for rendezvous is an effective way of establishing the dwell time. Once the dwell time is fixed, the expected time and maximum time to rendezvous can be calculated. This analysis assumes that the transition time from one frequency to another is negligible.

The upper bounds of expected time to rendezvous (E(TTR)) and the maximum time to rendezvous (MTTR) can be calculated for many sequences and protocols. These values are in terms of the number of hops. The Jump-Stay of Liu, et al [28] is one of the more recent and highly performing algorithms. Its upper bound of E(TTR) is given in (5.1) and upper bound of MTTR in (5.2) for the 2-user asymmetric model. The asymmetric model means that each node has a potentially different assessment of available channels. This is more demanding than the symmetric model.

$$E(TTR) = 2MP(P-G) + \frac{M + 5 - P - \frac{2G-1}{M}}{P} \tag{5.1}$$

$$MTTR = 2MP(P-G) + 3P \tag{5.2}$$

where M is the number of channels in the band plan, P is the next greater prime number than M, and G is the number of commonly available channels between the radios. Obviously, G can range from the trivial case of 0 to a maximum of M. For band plans with 20 channels, values of E(TTR) and MTTR range from the 2000's to over 20,000. Roughly speaking, this amounts to a minimum of a 100 fold increase in hops relative to the number of channels.

Comparison of FP-BLR with FH-BLR

The FP-BLR is by design a wideband protocol. The corresponding noise bandwidth is wider than that optimally used in a frequency hopping approach by a factor corresponding to the total number of channels in the operating band. Additionally, in FP-BLR the transmitter operates on multiple channels at one. This can potentially limit the power in each channel, since a transmitter can only emit a maximum amount of power. This power is spread throughout the number of probed channels. Whether this is the limiting factor on emitted power spectral density, or it is regulation limited depends upon the hardware design.

The research community working on frequency hopping sequences has not yet focused on what type of signal to emit during each hop and what type of detector to use to sense the rendezvous. In most published works, the implication is that a single tone is transmitted in the channel and an energy detector is used. This may be suitable for narrow band channels. Unfortunately, energy detectors suffer from a threshold setting problem. If the receiver does not have the ability to sound the channel, it is difficult for it to estimate the noise floor. This complicates the setting of the threshold. Receiver operating characteristic curves can be used to trade off the failure rate with the threshold, but this makes direct comparison with our technique arbitrarily dependent upon threshold setting. One may anticipate that systems with wider individual channels might use more sophisticated signal processing techniques such as a broadband signal and a cyclo-stationary detector in order to avoid the noise uncertainty issue. In these cases, the detection algorithm complexity may limit the hop rate.

We can provide some tangible comparison by examining the noise bandwidth impact from moving from the full spectrum to a single channel. If we hold the signal power constant, the noise power scales by the number of channels in the band plan, conventionally expressed in decibels as $10\log_{10}(M)$, where M is the number of channels in the band. For 5 MHz channels in a 100 MHz band, this is a 13 dB penalty in signal to noise compared to examining a single channel. For 1 Mhz channels in the same band, the penalty is 20 dB. We can interpret this from several different viewpoints. The most punishing way is to consider increasing the output power by 13-20 dB to make up the difference. This is a huge penalty, and not very realistic. One could also make up the difference by sacrificing the physical range of rendezvous, but again it is not practical to make up this difference.

Instead, we should consider the impact of integration time in the detector. The dwell time in the frequency hopping system will be long enough to integrate the deterministic signal above the random noise by an amount consistent with the threshold setting. This has to be multiplied by the expected (E(TTR)) and maximum time to rendezvous (MTTR), according to the sequences used, in order to calculate the total time required. Again using the Jump Stay algorithm, assume that $M = 100$, making $P = 101$, and assume $G = 20$. The $MTTR = 2,454,603$ and $E(TTR) = 1,636,565$ time slots. By comparison, assume $M = 20$ (equivalent to 5 MHz channels in a 100 MHz band), making $P = 23$, and assume $G = 17$, or almost all of the channels are free. Now $MTTR = 8349$ and $E(TTR) = 5528$.

In our algorithm, the fundamental time constraint is the spacing of the tones. In the simplest case, use use a single tone per channel and the detector is simply an energy detector. Alternatively, as described previously, we can use multiple tones and use a pattern detector, possibly gaining some processing gain. In either case, the reciprocal of the time length of the FFT frame determines the frequency resolution (along with second order effects such as the window function used). Repeated processing and summing of the spectrum arrays can integrate the deterministic signal out of the noise, improving at the rate $\sqrt{N}$, where N is the number of averages, at the expense of some minor main lobe broadening (decreasing the frequency resolution).

Furthermore, since the receiver is not synchronized to the start of the attention burst, the frame handed off to the FFT operation might capture as little as one half of the burst. In order to maintain the resolving power, the actual pulse should be twice the minimum length. This is less of an issue if averaging is used and the hardware is fast enough to process sequential frames. (This would normally require multiple cores operating in a ping-pong fashion.) A similar issue confronts the frequency hopping protocols, where the lack of coordinated timing means that each frequency hop may have an overlap of as little as 50%. There is a further ambiguity that affects both systems. This involves which node transmits and which one listens during a rendezvous period. In many cases, a particular node initiates the communication by transmitting, while others are monitoring. Nevertheless, there is some handshaking required once both radios land on the same channel. Timing ambiguity requires guard bands until the nodes have synchronized.

Let us assume a single tone per channel with an energy detector. To resolve a 1 MHz channel, the time involved in the FFT has to be at least $1\mu S$; however, the window function will broaden the main lobes, so we will assume $3\mu S$. To account for a factor of 100 difference in noise bandwidth between the FH and the FP approaches, we can average such that $100 = \sqrt{N}$, or $N = 10,000$. Similarly, with 5 MHz channels, $20 = \sqrt{N}$, or $N = 400$. While the number of frame averages is several orders of magnitude lower than the number of hops, we have not accounted for the relation between hop dwell time for the frequency hops and the frame time length. There is a minimum dwell time for a given channel bandwidth that is a function of the associated filter group delay. To this point, we have made many assumptions, including the band channel plan (M), the number of common available channels (G), main lobe broadening due the the FFT window function, the tone spacing in the probe signal, and now the filter group delay. While we have begun to understand the tradeoffs between the two general approaches, we are finding diminishing returns in being able to draw firm conclusions about the general superiority of one technique over another.

Because researchers in FH-BLR have generally abstracted the actual signal detection, it is difficult to address the discrimination between PU signal, SU signal, and rendezvous signal. Other researchers have independently addressed this when dealing with spectrum sensing and a great number of algorithms have been proposed. Some involve feature detection of particular parts of the waveform, such as a preamble or midamble. Others achieve slightly more generality by using cyclo-stationary techniques to identify known periodicities. These all require complicated processing that may further slow down the hop rate. Although they are complicated, they may already be available in the spectrum sensing functionality. At the current time, most systems still use the energy detector because the more exotic techniques require too much processing power and time or too many assumptions about the transmitted signal. We conclude that for fast hopping rendezvous, a specific probe and pattern matching technique, much like ours will be needed. Ultimately, this means the fundamental difference between the FP-BLR and FH-BLR algorithms is that the FP-BLR probes multiple channels simultaneously. The other differences exist simply because the FP-BLR is farther ahead in developmental maturity than the FH-BLR for PU-SU DSA applicatons.

We believe there is sufficient cause to believe that these techniques have somewhat overlapping trade-spaces so that in come cases one would be favored, in others the opposite would prove a better choice, and in some cases, both may substantially meet the requirements. We assert that in practice, the choice of one over another approach will likely be driven by the radio hardware and its most common waveform. As we mentioned in Chapter 1, our design was inspired by the prevalence of highly parallelized FPGAs and multi-core processors in many cognitive radios, and even simpler, wide bandwidth adaptable radios, and the increasing dominance of frequency domain waveforms such as OFDM. The FFT is an essential part of processing an OFDM waveform. Typical frame sizes are 512-2048 for WiMAX and LTE waveforms. Of course, this is done over a 5-20 MHz bandwidth. Achieving good frequency resolution over a whole band will require larger sizes. This does not change the fundamental algorithm, only the size of the memory, especially if it is implemented in a DSP or GPU device. The FPGA (or equivalent functional part) must be reconfigured to handle the wider bandwidths. Filters need to be widened, but this is easy since wideband filters are shorter than narrowband filters for a similar shape factor. Sample rates may be a problem because the Nyquist condition must be satisfied with the wider bandwidths in order to avoid aliasing. Increasing the sample rates could force an increase in clock rates; something that is not normally feasible by simple partial reconfiguration of a FPGA. Fortunately, many systems are oversampled near the data converter side already. This is because of the DPD issues discussed in the previous chapter and the better performance from the data converters (relative the the main signal bandwidth). A combined filter and sample rate converter often dominates the FPGA design. By simply bypassing a decimation stage, a wider bandwidth can be achieved with a higher sample rate. In conclusion, the FP-BLR algorithm requires some demanding signal processing; however, it is likely already available, especially when a core OFDM waveform is one of the supported waveforms.

On the other hand, a dynamic spectrum access radio is designed to change frequencies quickly in order to take advantage of short windows of available spectrum. In every design there is a limit to how fast the radio can hop. If the frequency hopping rate of the rendezvous process must be faster than the sustained channel change rate of the core waveform, then the design may be challenged in a significant way. Hopping rate of a frequency synthesizer is typically in conflict with the goal of low noise. If the core waveform is a frequency hopping waveform such as Bluetooth, then this is not an issue. Some military frequency hopping waveforms hop very quickly in order to minimize detection and jamming susceptibility. Of course, these are not normal candidates for dynamic spectrum access since each channel has to be available for secondary use when it hops to it. All of these considerations affect the practicality of a frequency hopping rendezvous technique.

Conclusion and Discussion of FP-BLR

In conclusion, the Frequency Parallel Blind Link Rendezvous protocol can quickly facilitate the initial connection of cognitive radios in an infrastructure-less dynamic spectrum access environment. The algorithm assumes that spectrum sensing is imperfect, and therefore seeks to exchange spectrum occupancy data at the earliest opportunity. We have provided both analytical and simulation based performance analysis in terms of probabilities of errors and provided guidance through receiver operating characteristics curves on threshold choice. We discussed at length how our Frequency Parallel technique compares with Frequency Hopping techniques and concluded that there are applications that may favor each one. This will likely be driven by the nature of the underlying hardware, which is in turn driven by the typical waveforms it processes.

CHAPTER 6: ARGUMENTATION BASED NEGOTIATION IN COGNITIVE RADIO NETWORKS

We briefly presented our negotiation concept in Chapter 1. We now develop the framework in detail. This framework will first be applied to a scenario with a network of source and sink radio nodes that communicate through relay nodes. In this case, the basic radio nodes always communicate through the relay nodes and will initiate connections. The relay nodes accept or reject the requests based upon their available capacity to handle the additional load. We consider the situation where radio nodes seek to establish long lived connections and therefore need a lease. Rather than being immutable, we explore the idea of flexible terms where the granted service might be subject to modification later on in the connection. A set of rules control this process. This flexibility offers a rich negotiating domain to test our framework. One of the important goals of the system is that negotiations should terminate in a finite time length with a firm upper bound. The example developed in this chapter is a one shot negotiation. In other words, only a single negotiation round for a single lease is considered. In Chapter 7 we will develop a system that involves many initial requests from any node in the network throughout the simulated day.

Proposed Negotiation Protocol

A negotiation protocol describes what can be done at each step of the process. This may be to issue a proposal, accept a proposal, reject a proposal, amend a proposal, critique a proposal, offer a counter proposal, ask a question, or offer supplementary information. The philosophical study of argumentation in human dialog can be used to guide the development of a protocol [49]. This ultimately sets rules governing what statements can be uttered in a particular context and what the implications are of that utterance [47]. This analysis begins with the context describing the current state of the negotiation. For example, a context could be that another agent has submitted a proposal for consideration. One could imagine a number of legal responses in this context; however, in artificial intelligence applications, it is appropriate to consider constraining the set of legal responses. This not only simplifies the development effort, but also makes the system more predictable, potentially more stable, and likely faster. These issues are all critically important in our application.

Our intention is to use argumentation to speed the negotiation process and to find solutions not otherwise obvious under plain negotiation. This introduces another set of question addressing what kind of argument can and should be provided in each state. As discussed by Parsons and McBurney [61], argument selection can be confident, careful or thoughtful. Confident agents send any proposal for which they can generate an argument out of their knowledgebase, even if it is rebutted by stronger arguments. Careful agents only assert those arguments which are not rebutted, while thoughtful agents only choose arguments which are not undercut. Rebuttal in this case means that one argument has the complemented conclusion as another argument. For examle, one argument could include data from an agent's own spectrum sensor that indicates that a frequency is unoccupied. Another argument includes data received from another agent which indicates that the same frequency is occupied. Undercut is similar, but rather than operating on conclusions of each argument, it operates between the conclusion of one argument and the premises or support of another. When an argument is received, the agent must evaluate it. Similar to selection, evaluation can be credulous, cautious or skeptical. Credulous represents the lowest burden of acceptance, while skeptical is the highest.

Our framework assumes that any continuous parameters, such as rate, are discretized into a finite set of values. This means that the agent selects values rather than calculating a precise value. These selected values are formed into offers, the set of which is represented by $\mathcal{O}$. From the perspective of a radio node, there is a single negotiation with offers for a single communication link. On the other hand, from the perspective of a relay node, there are multiple simultaneous negotiations, each involving a communication link. The relay node is concerned with a conjunction of offers and commitments. For example, in scenario 1, it initially considers the conjunction of the commitment of the ongoing communication between $N1$ and $N2$ and the proposal for communication between $NX1$ and $NX2$. Ultimately, it opens a negotiation with $N1$ to downgrade its link with $N2$ and considers this negotiation in concert with the proposal from $NX1$.

Table 6.1: Possible offers from NX1 with two relay nodes

Offer$< NX1, NX2, R1, HDV >$
Offer$< NX1, NX2, R1, VIDEO >$
Offer$< NX1, NX2, R1, VOICE >$
Offer$< NX1, NX2, R2, HDV >$
Offer$< NX1, NX2, R2, VIDEO >$
Offer$< NX1, NX2, R2, VOICE >$

According to [62], arguments can be classified as *epistemic* or *practical*. Epistemic arguments arise from and justify beliefs. Practical arguments justify offers and are built from both beliefs and goals. In scenario 2, the HIGH priority of the link between $NX1$ and $NX2$ is a fact that can form an epistemic argument. An example of a practical argument is the desire or goal of $NX1$ to communicate with $NX2$. Epistemic arguments are represented by $Arg_e(\mathcal{L})$. and practical arguments by $Arg_p(\mathcal{L})$. Some examples are listed in Table 6.2.

Table 6.3 lists several facts that can be used to generate an epistemic argument. The first line could be used to argue for the video link between $NX1$ and $NX2$ to be granted due to its high priority status. The second and third lines could form an argument to convince

Table 6.2: Possible Practical Arguments of NX1

Desire$< NX1, NX2, HDV >$
Desire$< NX1, NX2, VIDEO >$
Desire$< NX1, NX2, VOICE >$

$Relay2$ to accept a hand off. The last line could be used by $NX1$ to argue that it should be granted a VID link to $NX2$ by asserting Rule 1.2.

Table 6.3: Facts leading to Possible Epistemic Arguments

Fact$< NX1, NX2, VID, HIGH_PRIORITY >$
Fact$< Relay1, CONNECTIONS, 2 >$
Fact$< Relay2, CONNECTIONS, 0 >$
Commitment$< N1, N2, Relay1, HDV >$

Each exchange in a negotiation is a message that may be a proposal, an argument, acceptance or rejection of a proposal, and agreement or disagreement with an argument. Proposals and responses can be augmented with an argument, although arguments can be sent on their own. This system implies that participating nodes keep state about the negotiation process.

Argument Generation and Evaluation

For each argument, we consider when it applies, how to evaluate it for acceptance or rejection, and how to respond for each result. We also consider how the issuing agent should respond if the receiving agent accepts or rejects the argument. Our arguments are narrowly defined in the context of our carefully designed scenarios. This allows us to reasonably bound the conditions for generation and evaluation. We analyze each argument individually, starting with the fairness argument.

Fairness

Applies Assert when a proposal has been rejected after asking for less than the highest rate handled by the relay. The requesting node will typically not know the highest rate currently handled by the relay, but will know the highest rate defined in the negotiating system. In this case, that is HDV. This argument cannot be asserted when requesting HDV, but it can when requesting VIDEO or VOICE.

Evaluation The receiving agent should agree with this argument if it can find at least one connection with normal priority at a rate greater than the one requested.

Acceptance The receiving agent should reevaluate the associated proposal to see if there is a connection that can be downgraded and issue a proposal accordingly.

Rejection The receiving agent should send a DISAGREE message to the originating agent, which will in turn attempt to find another argument to support the proposal, or issue a new proposal with a concession.

Priority

Applies Assert when rejected, if the requesting node has priority.

Evaluation Accept if there is a higher or equal rate connection without priority.

Acceptance Create DOWNGRADE proposal with a priority argument and send to a node of the existing connection without priority

Rejection Send DISAGREE message to originator. The originating node may concede and issue a proposal at a lower rate, and subsequently argue for fairness.

Favor Increase in Connections

Applies Assert by a relay node when another relay node has rejected a HANDOFF proposal

Evaluation Accept if it can support the HANDOFF without having to drop an existing connection

Acceptance Reevaluate the original proposal that triggered the HANDOFF proposal and accept it if possible.

Rejection Send a REJECT notice to the node that originated the proposal that triggered the HANDOFF proposal (if no other relays are available). The originating node may attempt a new argument, such as priority, or concede and submit a proposal at a lower rate.

Performance of ABN Framework

The key metrics in negotiation are the speed at which a deal can be met, the quality of the deal, the robustness in proceeding to a deal, and the ability to conclude negotiations even when a deal cannot be met. From our perspective, optimality of the deal is preferred; however, not necessarily at the cost of stability and speed. This is a function of the application domain. A negotiating system governing spectrum allocation of femto cells in a 4G network might favor spectrum efficiency over speed of negotiation, since they operate in a relatively static environment. A DSA system operating in the public safety band encounters a much more dynamic environment, thereby favoring speed of negotiation over optimum spectral efficiency. Although Amgoud has made progress towards demonstrating that ABN can reach superior deals [62], optimality is not yet within reach.

Simulation Study of Argumentation in Cognitive Radio Networks

In order to evaluate the robustness and effectiveness of argumentation based negotiation in finding quality solutions, we eveloped a software platform that performs the negotiations autonomously. The argumentation based negotiation system should process any set of initial conditions and properly reach the best outcome under the set of rules, even if that is a "no deal". This can be validated by a computer program that accepts any initial conditions and calculates and displays the correct deal. The system prints an optional trace of messages that reveals the internal decision making process.

Simulation environment and metrics

We created a Java based simulation program which accepts a set of initial conditions and negotiates a solution according to the set of rules adopted. Proposals, arguments, and responses are exchanged by messages directed from an originator to a target. A proposal for a connection includes a source, destination, communication grade, and relay. Note that the originator and target do not necessarily correspond to a source or destination. In fact, a radio node acting as a source will typically originate a proposal targeted to the relay it wants to handle its traffic. The basic problem topology is give in Figure 6.1

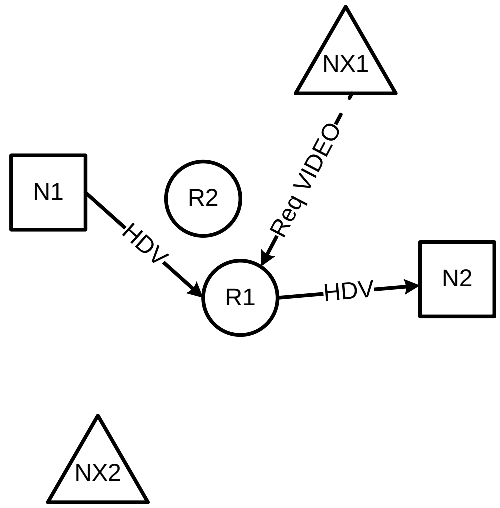

Figure 6.1: Typical Network Topology for Argumentation Study

Both relays and basic radio nodes inherit from a common object, ABNAgent, that supports negotiations. Each negotiating agent maintains a knowledge base of facts that originate from either the initial conditions or arguments. These facts include the existence of different entities, such as Relay2, and connection priority. A list of existing connections is maintained so that agents can calculate their current load and find existing connections to modify (e.g. handoff or downgrade) in order to support a new proposal. Finally, a list of pending proposals is managed as agents issue new proposals in response to received or rejected proposals. This mechanism allows a relay node, upon receiving a request for a new connection that exceeds its capacity, to negotiate with a node in an ongoing communication to modify its connections, and ultimately respond to the original requesting node, based upon the outcome of the secondary negotiation.

A negotiating agent moves through several different states during negotiations, as illustrated in Figure 6.2. Negotiation starts when an agent is seeded with a desire, such as communicate

HDV with NX2. This desire becomes an intention as it creates a proposal to a relay entity, found by examining its knowledge base. The node then enters a waiting state until it receives a response.

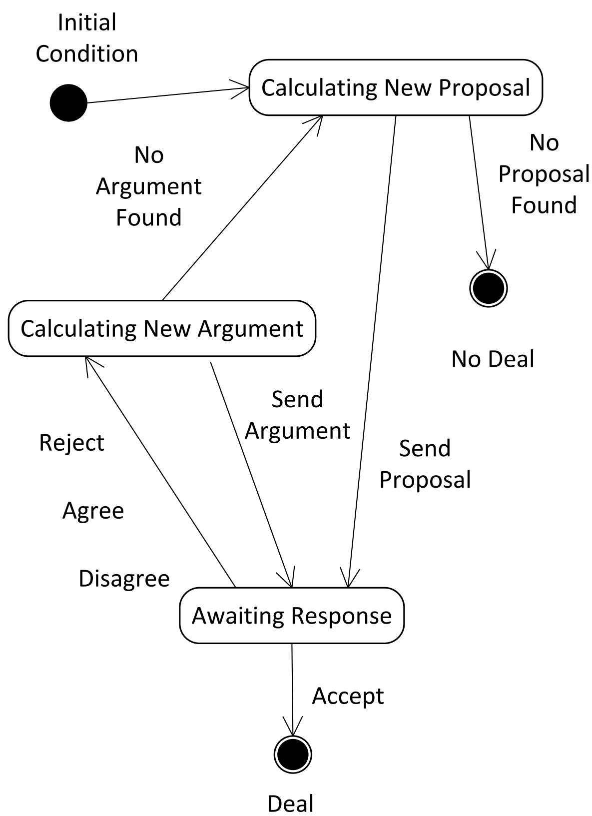

Figure 6.2: Radio Node Negotiation State Diagram.

Valid responses from the relay node can be acceptance or rejection of the proposal. The originating node then attempts to create an argument supporting the original proposal. If it can find one, it sends it; otherwise, it attempts to create a new proposal. For example, if it cannot find an argument to support an HDV connection, it may create a new proposal to create a VIDEO connection. Each argument can only be sent once. When the arguments supporting a proposal are exhausted, then a concession is required in order to submit a new proposal to the same agent. When the originating node is unable to generate a new argument or proposal, it reaches a "no deal" conclusion.

Agents evaluate proposals by calculating the new load and comparing that to the agent's maximum capacity. If they can support the new connection in addition to their current connections, the agent will accept the proposal. The exception to this is a HANDOFF request to a relay node. In this case, the default is to refuse unless there is a supporting argument. This seems reasonable since a handoff to a different relay carries with it some risk of a dropped connection if the end nodes cannot reach it.

In general, nodes issue their strongest arguments first; however, it is not always possible to ascertain the most effective argument. Knowledge of another relay node can be more effective than asserting priority, unless that other relay node is already carrying priority traffic at its maximum capacity.

We are able to seed the simulation with various initial conditions, observe the negotiation process and examine the resulting deal. The simulation starts with an existing connection and negotiates a new connection, as in the scenarios described earlier. The various initial conditions are the capacity of the relay nodes, the grade of the existing connection, the priority condition of both the existing connection and the proposed connection, the existence of a second relay node, and the desired grade of the new connection. A large subset of the possible trials are presented in Table 6.9 in the next section.

Illustrative Example of Argumentation in Cognitive Radio Networks

In order to illustrate the power of argumentation based negotiation in cognitive radio networks, we have created a set of scenarios where radio nodes negotiate for access with a certain grade of service in compliance with a set of rules. Each scenario consists of a set of radio nodes that communicate through relay nodes. They can communicate voice, standard video, or high definition video. Table 6.4 summarizes the rates assumed for each communication grade. There is an order to these in terms of the rate. Furthermore, each communication link can have either normal or high priority. Finally, there may be either one or two relay nodes.

Table 6.4: Communication Grades and Rates

Grade	Rate	Unit
Voice	256	kbits/s
Video	3.5	Mbits/s
HDV	8	Mbits/s

The negotiation starts with a set of initial conditions composed of an existing communication link between nodes $N1$ and $N2$ as shown in Figure 6.3, and a desire for $NX1$ to communicate with $NX2$, with a certain priority. Each radio agent's knowledge base includes specific and potentially different information about the topology, for example, the existence of a second relay and the total capacity carrying capability of each relay node. The first step is for $NX1$ to send a proposal to communicate to $Relay1$. Negotiations are resolved according to a specific set of rules, listed in Table 6.5. Note that these are not the only set of rules one could compose for this problem; however, they are chosen to lead to reasonable results.

Table 6.5: Rules for Negotiation

R1	All connections have a right to a class of communications before any have a right to a higher class(fairness).
R1.1	All connections have a right to AUDIO before any have a right to VIDEO or HDV.
R1.2	All connections have a right to VIDEO before any have a right to HDV.
R2	HIGH priority connections can force a NORMAL priority connection to downgrade one level.
R3]	HIGH priority connections cannot force a disconnection of any existing connection.
R4	Relays must accept handovers if it increases the total number of connections in the network.

The first scenario starts with an ongoing HDV communication between $N1$ and $N2$ through relay R. The node $NX1$ proposes an HDV connection to $NX2$ through R. The agents follow a negotiation listed in Table 6.6 and Figure 6.4, and reach the end-state (see Figure 6.5). This scenario shows concession by $NX1$ to drop to VIDEO in order to argue fairness (Rule R1.2), and a concession by $N1$ when presented with that argument. It is implied that the relay node R can support one HDV or two VIDEO connections, but not an HDV and VIDEO connection simultaneously.

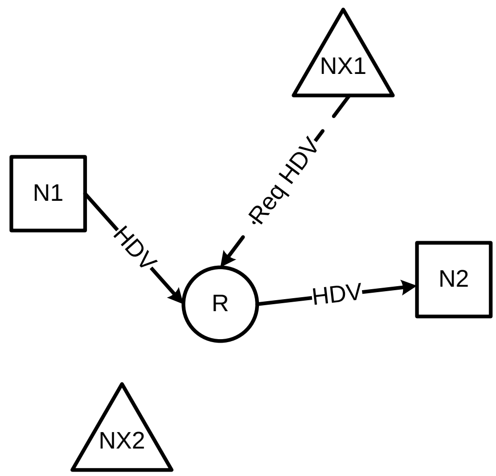

Figure 6.3: Scenario 1 and 3 Initial Conditions.

Table 6.6: Scenario 1

Initial Conditions	Existing HDV between N1 and N2
NX1 → R	Request HDV connection to NX2
R → NX1	Denied. No capacity
NX1 → R	Request VIDEO connection to NX2
R → NX1	Denied. No capacity
NX1 → R	Argue Rule R1 (fairness)
R → NX1	Agree Rule R1 applies
R → N1	Downgrade to VIDEO. Rule R1 applies
R → NX1	Request Granted. VIDEO to NX2

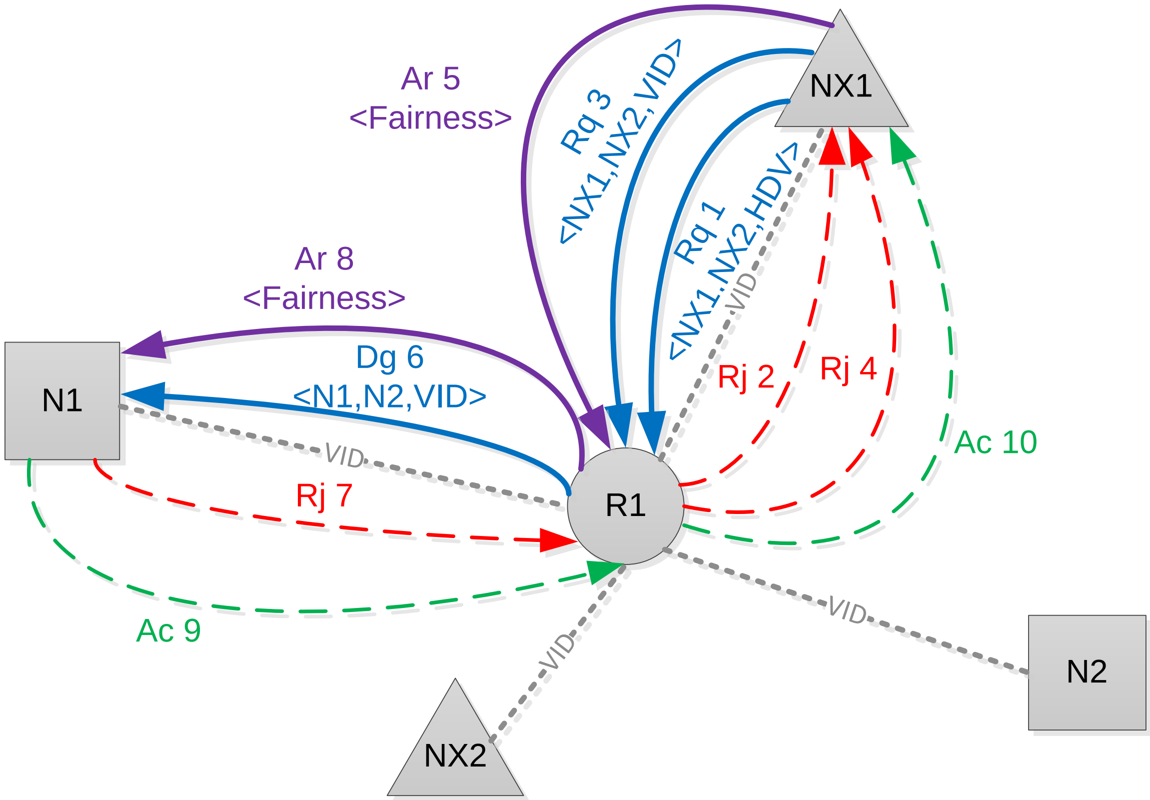

Figure 6.4: Scenario 1 Negotiations

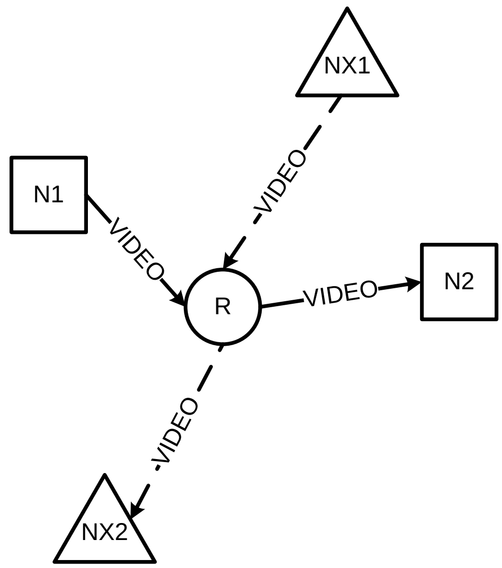

Figure 6.5: Scenario 1 Final Conditions.

The second scenario introduces a second relay, $R2$, known to the first relay, $R1$. The node $NX1$ is aware of $R1$, but not $R2$. The relays negotiate a handoff as described in Table 6.7 and Figure 6.7. The initial condition is given in Fig. 6.6, while the final condition is shown in Figure 6.8.

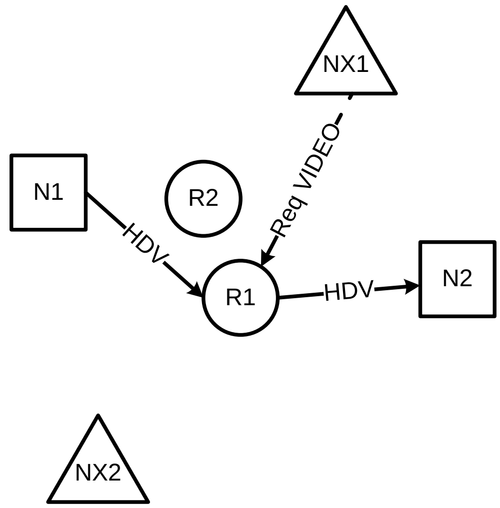

Figure 6.6: Scenario 2 Initial Conditions.

Table 6.7: Scenario 2

Initial Conditions	Existing HDV between N1 and N2 Two Relay nodes
NX1 → R1	Request VIDEO to NX2
R1 → NX1	Denied
NX1 → R1	Argue Rule R1 (fairness)
R1 → NX1	Agree Rule R1
R1 → R2	HANDOFF request Connection N1-HDV-N2
R2 → R1	HANDOFF denied
R1 → R2	Argue Rule 4
R2 → R1	HANDOFF accepted
R1 → NX1	Granted

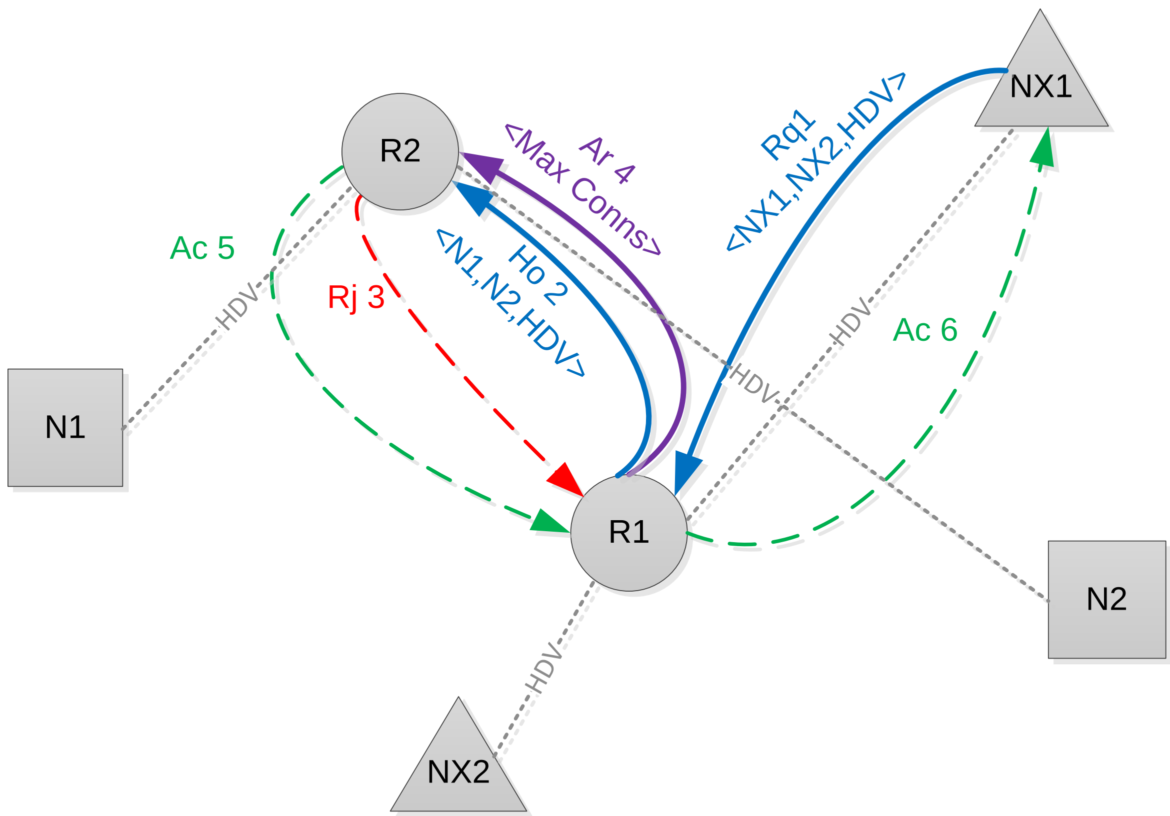

Figure 6.7: Scenario 2 Negotiations

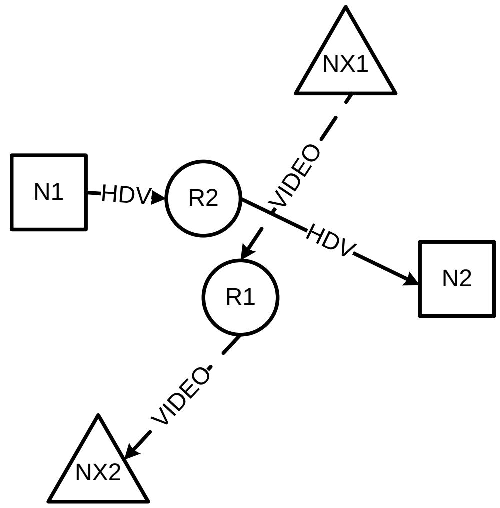

Figure 6.8: Scenario 2 Final Conditions.

The initial conditions for scenario three are the same as scenario one presented in Figure 6.3. Here, the connection that $NX1$ seeks to establish has a high priority. This negotiation follows Table 6.8 and Figure 6.9, with the end state given in Figure 6.10. This negotiation forces the initial connection to concede when confronted with the priority argument. In this case, the relay node has a greater capacity than previous scenarios, being able to handle one HDV and one VIDEO, but not two HDV connections simultaneously. These scenarios provide a rich domain in which to study argumentation in negotiation for cognitive radio networks.

Table 6.8: Scenario 3

Initial Conditions	Existing HDV between N1 and N2 One Relay node. NX1 Priority
NX1 → R	Request HDV to NX2
R → NX1	Denied
NX1 → R	Argue Rule R3 (Priority)
R → NX1	Agree Rule R3
R → N1	Downgrade to VIDEO
N1 → R	Denied
R → N1	Argue Rule R3. Node Request HDV with Priority
N1 → R	Accept downgrade to VIDEO. Agree Rule R3
R → NX1	Accept HDV to NX2

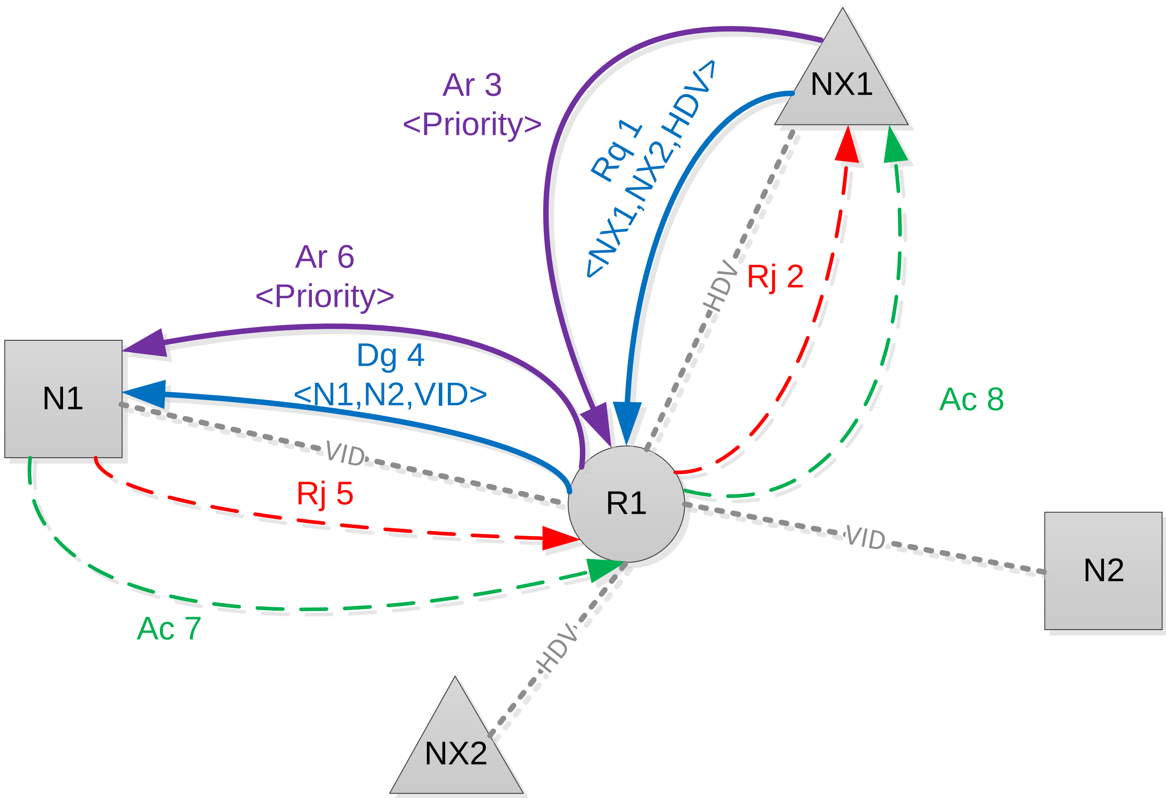

Figure 6.9: Scenario 3 Negotiations

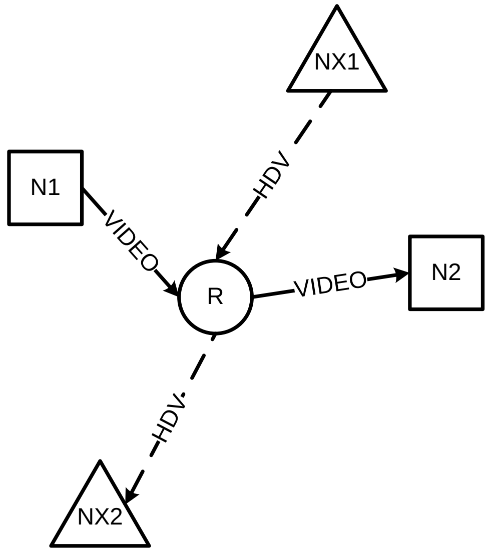

Figure 6.10: Scenario 3 Final Conditions.

We also ran the simulation for a wider set of initial conditions and desires, with the results presented in Table 6.9. In all cases, the capacity of the relay nodes is 12M, so that each relay can support either one HDV and one VIDEO connection or two VIDEO connections. The inability to support two HDV connections through a single relay creates a resource constrained environment, necessitating negotiations. We also present some trials where the capacity is not exceeded by the additional connections, to demonstrate the stability of the system to all types of conditions.

Table 6.9: Simulation Results

Trial	Existing			Desired		Solution		
	Service Class	Priority	Relays	Service Class	Priority	Prior Service	New Service	Num Messages
1	HDV	NONE	1	HDV	NONE	HDV	VID	4
2	HDV	NONE	1	HDV	HIGH	VID	HDV	7
3	HDV	NONE	2	HDV	NONE	HDV	HDV	3
4	HDV	HIGH	1	HDV	NONE	HDV	VID	4
5	HDV	HIGH	1	HDV	HIGH	HDV	VID	10
6	HDV	HIGH	2	HDV	HIGH	HDV	HDV	3
7	VID	NONE	1	HDV	NONE	VID	HDV	2
8	VID	NONE	2	HDV	NONE	VID	HDV	2
9	VID	HIGH	1	HDV	NONE	VID	HDV	2

Simulation results

In reviewing the resulting solutions for a wide variety of initial conditions in Table 6.9, we observe that the argumentation based negotiation leads to reasonable results. This is noteworthy since the design of the rules and negotiation system were based upon three scenarios, yet when many different combinations of the initial conditions are considered, the system still behaves well.

An often cited advantage of argumentation systems is the ability to find solutions that basic negotiations cannot. We see this in particular when introducing a second relay node. The originating node, $NX1$, does not know of the existence of the second relay node; however, the first relay node does have this knowledge, and acts accordingly to find a solution where no concessions are required (other than a handoff). Another situation arises when both the existing and proposed connection have a high priority and are requesting more aggregate resources than the relay can provide. This system results in an orderly conclusion, although a concession is required.

The simulation and scenarios are setup to send proposals and arguments separately. This helps to expose more detail about the negotiating process; however, one could also design the system to send the strongest argument available, whenever sending a proposal. While this would reduce the communication overhead in an implemented system, it has no significant impact on the negotiation process, except in those cases where a proposal is accepted without regard to any argument. These are trivial conditions where negotiation is not required. None of the trials listed in Table 6.9 resulted in a "No Deal", because the initiating agent preferred a lower rate to no rate at all. In other words, if rejected when requesting HDV service and no arguments prevailed, then $NX1$ will ask for VIDEO service instead. In case this is not acceptable, a minimum communication grade fact can be added to $NX1$'s knowledge base. The NewProposal algorithm will check that before concession and conclude that no proposal can be generated, resulting in the "No Deal" termination condition.

Although a "No Deal" termination seems like a failure, it is an acceptable response. It is crucial that the negotiations terminate when a solution is not available, in order to minimize futile resource consumption. The system can handle a "No Deal" response by queuing the request, or take other action as dictated by the higher level application.

The arguments in the simulation supporting these scenarios are assertions, but in some situations, there is an implied inquiry. Sending a high priority argument is an assertion.

Sending a fairness argument is an assertion, with an implied inquiry. In our scenarios, the proposing node does not know the communication grades of the other connections that the relay is carrying. If the relay accepts the proposal (after internally agreeing to the argument), the proposing node can conclude that there had been a connection at a higher rate and that it did not have a high priority. If the relay agrees to the argument, but is still unable to accept the proposal, then the proposing node can infer that the relay is carrying a large number of minimal grade (AUDIO) connections with a limited maximum capacity, and is unable to find capacity due to rule R3. If the relay disagrees with the argument, then it implies that all other connections have priority asserted also. Rather than issuing an inquiry first, and then a fairness argument, it is more efficient in many cases to directly issue the fairness argument.

Furthermore, there is an implied trust, in that arguments are accepted if they cannot be defeated, consistent with Dung's acceptability criteria [43]. This is illustrated by the relay asserting the fairness argument towards N1. This node accepts the argument because it cannot defeat it. An interesting attack relationship is found when both the initial and the proposed connections have a high priority. Each priority assertion attacks the other, resulting in a stalemate. With no other options, such as a second relay, the proposing node is forced to concede to VIDEO grade as a *de facto* first-come-first-served rule applies.

Rogue radio nodes could simply assert that they have a High Priority status, even when they do not. One way of dealing with this is to require the node to send a certificate traceable to an authority that grants the priority status. Network admission control techniques can also be deployed to ensure that only well behaving nodes are allowed entry into the negotiating process. Finally, the argumentation system can be extended to demand that the radio node present arguments supporting its High Priority status assertion. A typical argument might state that the information is of a tactical nature, where the success of a mission and preservation of lives is at stake. While this may allow the relay node to arbitrate with a finer grained resolution between potentially conflicting priority assertions, some secure authentication mechanism will eventually be needed to ensure that the arguments are based in truth. The level of secure authentication is application dependent.

Overhead is a concern in capacity constrained networks. We introduce overhead by the act of negotiating and further add to it when exchanging arguments. Of course, much research is directed at finding solutions without needing to incur the overhead of negotiation. Nevertheless, one can easily conceive of circumstances where the nodes of a system are heterogeneous (in terms of capability) and the local information at each node is incomplete or even inconsistent with other nodes. Information exchange can clearly lead to superior configurations relative to solutions without any information exchange. Argumentation actually improves the overall overhead by providing information to guide a negotiation, rather than letting the negotiation iterate throughout a search region. In a sense, it is the tool to rapidly and directly move from a game of imperfect information to one of perfect information.

The actual calculation of overhead is dependent upon application and domain issues such as the length of the aforementioned certificate authenticating the High Priority argument. It is also a function of whether or not anticipatory arguments are sent with an original proposal, or in response to a rejection. For the scenarios we have described, the strongest argument, or even all arguments, could always be sent to advantage. As systems scale with complexity of rules and arguments it will become more important to choose the best (i.e. strongest or most relevant) argument. By sending only the strongest argument, the task of evaluating the arguments is eased (at the expense of more difficulty in choosing the best one). The number of message exchanged column in Table 6.9 provides a qualitative examination of the overhead required. These values include the original proposal and the accept message.

Discussion of Complex Argumentation in Cognitive Radio Networks

In this chapter we have applied our negotiation framework to small network with a complex negotiation domain. We have demonstrated how passing information through arguments can influence the negotiation path and possibly uncover solutions that were not otherwise available to all of the nodes. The arguments do not force a decision, but rather prompt the nodes to reconsider rejected proposals in light of the new information. These arguments included priority assertions, fairness appeals, and entity notices. In some cases, these arguments will result in current connections being downgraded or handed off to other relay nodes. The flexibility provided by the rules-based concept that existing connections can be modified, is managed and leveraged by the negotiation process. This chapter has focused on a single round of negotiation. In the next chapter, we scale this up to include multiple rounds of negotiation as nodes request connections and previously granted connections complete, freeing up capacity for new ones.

CHAPTER 7: NEGOTIATION WITH HOMOGENEOUS ROLES

In this chapter we will consider a system that involves multiple rounds of negotiation, with each round representing a new connection request. The connections will be long running compared to the negotiation time. They will eventually expire, freeing up capacity for new connections. We purposely choose to significantly constrain the negotiation options, for example, only allowing a single attempt at rerouting around a at-capacity node instead of exhaustively searching every route. This is done to demonstrate the we can achieve superior results with a very short negotiating time, and therefore little impact on network overhead and connection setup time.

Specifically, we consider the problem of allocating link capacity for long duration streaming media of three discrete grades: high definition video (HDV), standard definition video (VID), and audio (AUD). The links between source and destination are capacity limited at high network loads, causing requests for new leases to be rejected. We are interested in ways to recover from these rejections. This can be accomplished by attempting alternate routes, or arguing that the requested connection has a higher priority than one or more existing connections (granted by some higher, agreed-upon authority). It is possible for an ongoing connection to be downgraded from a higher level to a lower one under certain conditions, although no connection will be dropped in favor of a new one.

Negotiation Rules

The negotiations are governed by a set of rules, some explicit and others implicit. These rules determine acceptance of proposals and arguments. They also determine which actions to take at any give point in the negotiations, for example, in response to a received message. This begins to define the negotiation protocol. We develop it more explicitly in the next sections.

Implicit Rules

The nodes trust other nodes' arguments, such as priority. Nodes are conservative in that they do not ask for a higher grade of service than they need. Relay nodes do not discriminate between their sourced or sinked connections and a relayed connection. Agents know the capacity and actively maintain knowledge of the current load of their one-hop routes. We assume that these nodes use a routing protocol of similar nature to table driven routing [63].

Explicit Rules

A node shall attempt to create a connection along the shortest route. Each node along the path shall accept the proposal if it has enough remaining capacity to support the requested grade of service. Otherwise, it will reject it. Received rejections result in an identical request, but along the next shortest route. A priority connection can force a current, non-priority connection to downgrade from HDV to VIDEO in order to free up capacity to support the new priority connection. Conversely, a node shall accept a downgrade proposal when presented with a priority argument, unless it can assert its own priority for the specified connection.

Because of the plurality of routes in a densely connected network and the additional considerations due to priority arguments, many solutions may be possible at any given negotiation step. In highly dynamic mobile radio networks, possibly using dynamic spectrum access, lengthy negotiations may not be practical. The limits are application dependent. This work uses fairly severe limitations on the number of negotiation options to be explored by any agent. In particular, if a given route includes an over-capacity link, another route will be chosen (if available), but if that one is also over-capacity, no more routes will be explored. If the desired connection has a high priority status, a single attempt will be launched from the originator, before failing. Likewise, when a priority argument is received after rejecting a proposal, the node will search for a connection to downgrade in order to clear capacity for the priority connection. The downgrade proposal will then be sent to the source node for that connection. It is possible that the connection will also have a high priority, in which case the proposal will be rejected. Only a single attempt is made for each argument. This approach avoids the danger of consuming excessive network resources in negotiation at the expense of a higher failure rate. Obviously, some applications will call for more diligent negotiation.

Negotiation Protocol

The rules and messages combine to a set of algorithms that represents the negotiation protocol. Negotiations start when a node has a desire to communicate with another at a specific rate. If appropriate, a high priority status is attached to the desire. This becomes an intention with a start time, duration, and a route. A proposal in the form of a request is then created and dispatched to the first node in the route.

Requests or proposals are accepted if sufficient excess capacity exists. This process is described in Algorithm 1. The agent examines the outbound connection at each node and compares the current load plus the proposed load to the capacity limit. If it is below the limit, the proposal is forwarded. Notice that if the current node is the final destination, the proposal can be immediately accepted, since the capacity has already been verified by preceding nodes. Actual acceptance from intermediate nodes is deferred until the destination nodes have accepted the proposal. When an intermediate agent forwards a proposal, it reserves the capacity on the assumption that the connection will ultimately be accepted by all nodes along the route. Eventually, the node will receive a response message from the node to which it forwarded the proposal. If it is an acceptance, then the reservation becomes a commitment for the duration of the connection. If it is a rejection, the reservation is cancelled. In either case, the message is forwarded along the reverse route back to the originating node. Timeouts can be used to clean up the negotiation state in case of a broken link.

Algorithm 1 Handle Proposal algorithm

```
function handleProposal(Proposal proposal)
    if isFinalDestination() then
        new_msg ← accept(proposal)
        connections.add(new Connection(proposal))
    else
        if current_load + proposed_load ≤ maximum_capacity then
            new_msg ← forwardMessage(proposal)
        else
            new_msg ← reject(me, proposal)
        end if
    end if
    return new_msg
end function
```

When an initial proposal is rejected, the originating agent can either issue a proposal along a different route, or issue an argument, if one is available, as shown in Algorithm 2.

If an agent receives an argument, it retrieves the cached proposal to which it refers (Algorithm

Algorithm 2 Handle Reject algorithm

```
function handleReject(Reject reject)
    if isOriginatingNode() then
        if num_attempts = 1 then
            route_num ← route_num + 1
            new_msg ← createProposal(route_num)
            num_attempts ← num_attempts + 1
        else
            if havePriority(desire) then
                new_msg ← createArgument(priority)
            else
                Fail                                    ▷ Negotiation fails and terminates
                new_msg ← null
            end if
        end if
    else
        new_msg ← forwardRejection(reject)
    end if
    return new_msg
end function
```

3). In the case of a priority argument, it attempts to find an existing connection that it can downgrade in order to clear capacity for the proposed connection. Assuming it finds one, it then issues a downgrade proposal to the originator of that connection. This gives that agent an opportunity to issue its own argument, for example, it also has priority, against the downgrade. Otherwise, it accepts it.

Agents handle a downgrade request by forwarding to the originating agent of the connection to be downgraded, which is the destination of the message route. As shown in Algorithm 4, the agent retrieves the connection from its database and checks to see if it has a high priority status. If it does, it rejects the downgrade request; otherwise, it accepts it and makes the appropriate changes to its connections and link loading databases. In either case, the message is sent back along the reverse route.

Acceptances from both proposals and downgrade requests are handled in Algorithm 5. Note

Algorithm 3 Handle Argument algorithm

```
function handleArgument(Argument arg)
    fact ← arg.fact
    if priority then
        connection ← findDowngradeConnection
        if connection ≠ null then
            new_msg ← createDowngrade(connection)
        else
            new_msg ← reject
        end if
    end if
    return new_msg
end function
```

Algorithm 4 Handle Downgrade algorithm

```
function handleDowngradeProposal(Downgrade dg)
    if isDestinationNode(dg) then
        dg_connection ← findDowngradeConnection(dg)
        if isPriorityConnection(dg_connection) then
            new_msg ← createReject(me, dg)
        else
            new_msg ← createAccept(dg)
            updateLinkLoads(dg_connection)
        end if
    else
        new_msg ← forwardMessage(dg)
    end if
    return new_msg
end function
```

that the important task of updating the connections and link databases is managed here.

If the node issuing the downgrade proposal is an intermediate node in the downgraded connection's route, it needs to notify the other agents in the opposite direction from the originator. This process is described in Algorithms 6 and 7 and is accomplished through a notification message and a corresponding confirmation. No decisions are required at any of the nodes in this message route. As before, actual changes to the connections and link loading databases are deferred until the confirmation message is received.

Algorithm 5 Handle Accept algorithm

```
function handleAccept(Accept accept)
    if fromProposal then
        updateLinkLoading()
        Connections.add(connection)
        if !orginatingNode then
            new_msg ← forwardMessage(accept)
        end if
    else if fromDowngrade then
        if finalDestination original proposal then
            new_msg ← accept(proposal)
        else
            new_msg ← forwardMessage(accept)
        end if
    end if
    return new_msg
end function
```

Algorithm 6 Handle Notify algorithm

```
function handleNotify(Notify notify)
    if isDestinationNode(notify) then
        new_msg = createConfirm(notify)
    else
        new_msg = forwardMessage(notify)
    end if
    return new_msg
end function
```

Negotiation Messages

The various radio agents or nodes communicate with each other via a small number of message types. Because communication overhead should be minimized in capacity constrained systems, these messages are very compact.

Propose This message requests a lease to communicate from a source node to a destination at a specific grade of communication . It specifies a start and stop time. It also includes a route.

Algorithm 7 Handle Confirm algorithm

```
function handleConfirm(Confirm confirm)
    if ( thenisDestinationNode(confirm)
        proposal ← retrieveOriginalProposal()
        if isDestinationNode(proposal) then
            new_msg ← createAccept(proposal)
            updateLinkLoads(proposal)
            addConenction(proposal)
        else
            new_msg ← forwardMessage(proposal)
        end if
    else
        new_msg ← forwardMessage(confirm)
    end if
    return new_msg
end function
```

Accept The Accept message indicates that the proposal is acceptable and flows back through the reversed route to the proposal originator.

Reject The Reject message is issues when a node is unable to support the proposal due to a capacity constraint. It includes an argument specifying the node that failed the capacity constraint. This information can be used by the proposer to synthesize a counter proposal.

Argue The Argue message sends a fact supporting a proposal. A priority assertion takes on unknown, normal, or high priority status. High priority statuses are associated with particular connections and can force a downgrade of another connection in order to support the priority one.

Downgrade A downgrade message is a specialized proposal sent when a relay does not have the capacity to support a proposal, but has received a priority argument. The downgrade is sent to the originator of a downgrade-able connection.

Notify The Notify message is an informative message that spreads the news of an accepted downgrade to other nodes in the connection route that is not on the path of the original downgrade and accept message pair.

Confirm The confirm message is sent by the ultimate destination of the notify message back to the originator of the notify. This is primarily used to synchronize message passing so that node that initiated the downgrade proposal can either accept the original proposal (if it is the final destination) or forward it.

Metrics

The performance measures include the overall network throughput and the satisfaction rate of requests under various loading conditions. We also investigate the satisfaction of individual links in the context of downgrades during the connection lifetime due to priority or fairness assertions.

An additional and very significant measure in negotiation is the number and total size of messages used to complete the negotiations. The number of messages can be minimized by passing all of the data in the knowledge base in each message. On the other hand, this can increase the total load, especially when no deal is possible or a simple deal is possible. In this case, the transmission of the extra data is either futile or unneeded. With a small network load, it is likely that initial proposals will be immediately accepted and there is no need to transmit excessive information. With very heavy loads, it becomes more unlikely that a deal can be reached; the network is at full capacity. Of course, the node may still wish to engage in negotiation and should send its strongest argument. It is reasonable for the originating node to consider its own knowledge of the network load and adapt its negotiating strategy accordingly. In the current work, we choose to transmit less information per message. This approach allows us to see the progress of decisions in the negotiation process more clearly, since the result of each significant decision is a message. While we measure the number of messages required to complete a negotiation, one should be cautious about drawing final decisions based upon this information.

The overall achieved performance is characterized in several ways. The total carried network load, or throughput is found by summing the grade for each unique connection. This can be further analyzed by comparing it to the requested capacity. This will account for the impact of downgrades. Similarly, satisfaction measures how good of a deal is reached for a request, with zero signifying a complete rejection, one signifying acceptance of the original request, and a number in (0,1) reflecting the degree of satisfaction when the achieved grade is less than the requested grade. The total number of unique connections measures the ability of the system to support as many users as possible. Finally, the number of hops it takes to support the connections measures the impact of rerouting to non-optimal routes.

Conclusion of Design of Negotiation with Homogeneous Roles

We have designed a system to conduct negotiations for leases in a homogeneous network over a sustained time period. In contrast to the previous chapter, we have significantly constrained the negotiation domain. This limits the overhead incurred and allows us to study the ability of the system to yield improvements under these conditions. We next discuss our updated simulator followed by extensive simulation results.

CHAPTER 8: ARGUMENTATION BASED NEGOTIATION SIMULATION

We first describe our updated simulator, which handled multiple lease requests, homogeneous roles, and multi-hop networking. Then we present a scenario and simulation results.

Simulation for Multiple Lease Requests

We developed a simulator to compare the performance of the negotiation technique to a variety of loads. Each of the messages described in the previous chapter inherit from a common class. This class encapsulates the routing data common to all messages.

A topology class controls the generation of a specified number of nodes, randomly determines the connectivity between each pair of nodes, and calculates a number of routes between each pair. This simulation assumes homogeneous links with a specified maximum capacity, although nothing in the protocol precludes heterogeneous links. Using identical links with either no capacity or $MAX\_CAPACITY$, the results are more clearly discerned. Options are available to study both a densely and a sparsely connected network, but the settings are adjusted to ensure that all nodes can be reached from any node. In other words, these are all topological spaces. Because of this, failures can be attributed to a load versus capacity constraint rather than a disconnected graph.

A scenario generator issues desires based upon a Poisson distribution, parameterized by its mean, λ. It is called once per simulated second. This results in a mean request rate of λ requests per simulated second. It creates an intention from this desire by randomly chooses an originating node and a distinct destination node according to a uniform distribution. It enforces the distinct condition by repeatedly drawing a node number from the distribution until it is different than the originating node. The requested connection grade is chosen from a uniform distribution, $[0,1]$, such that if the drawn number is in $[0.67, 1.0]$, a HDV connection is requested. A VID connection is used if the number drawn is in $[0.33, 0.67)$. Otherwise, an AUD connection is requested. A high priority status is set for the connection if a uniform distribution generator provides a value in $[0.0, 0.75)$. Finally, the start time of the connection is set to the current time (assuming that the negotiation time is negligible) and the stop time is set such that the duration is drawn from a normal distribution with mean of 3600, i.e. one hour and a standard deviation of 600 seconds, i.e. 10 minutes.

If an intention is generated at a particular time step, it is set in the originating node and that node subsequently attempts to generate a proposal based upon that intention. The proposal generation process can fail if there are no outbound links with sufficient capacity, and in the case of the argumentation agent, no argument can be formed. This is logged as a failure. A message queue dispatches messages to the target agents message handling routine.

At each time step, each node cleans up any expired connections. Then the scenario generator is called. If one or more intention is generated, the negotiations are conducted. Finally, the results are scored. Scoring involves retrieving a set of distinct connections from the agents. The grade of each connection is summed to calculate the total network load, along with counting the number of connections to assess how many users are supported.

Scenario for Multiple Lease Requests

The simulator allows us to investigate a number of scenarios using different agent models. As a baseline, we developed a radio agent (RadioNode) that simply requests the shortest path. If the capacity is available, it is accepted, otherwise it is rejected. A slightly more sophisticated model (RadioNodeReRoute) will attempt the next longest route upon receiving a rejection. Our RadioNodeABN conducts negotiations according to the rules we have discussed in sections 7 and 7. Finally, we also implemented an oracle (RadioNodeOracle) that has full knowledge of the link loadings throughout the network. When proposing, it is able to choose a route that it knows will succeed, if one exists. On the other hand, it allocates on the fly, temporally. It also chooses the shortest available route, rather than another possible option, the least loaded route. Most importantly, it does not reconsider existing routes when analysing current requests. When considered within the context of the degrees of freedom associated with our rules, an optimum approach is ill-defined. Our rules favor supporting as many connections as possible, while respecting a possible priority condition. This is different than maximizing total data throughput in the network or conventional fairness criteria.

We consider three connections grades, high definition video (HDV), regular video (VID), and audio (AUD). We assume data rates for each of these grades as listed in Table 8.1. Table 8.2 lists the remaining simulation parameters. The sparsely connected scenario is diagrammed in Figure 8.1 where the available links are indicated by line segments between nodes. The densely connected scenario is provided in Figure 8.2.

Table 8.1: Connection Rates

Grade	Rate (kbps)
HDV	8000
VID	3500
AUD	256

Table 8.2: Simulation Parameters

Parameter	Min	Max
Number of nodes	7	
Capacity	48 Mbps	
Lambda	0.0001	0.05
Request Rate	0.36/Hr	180/Hr
Connection density	sparse	dense

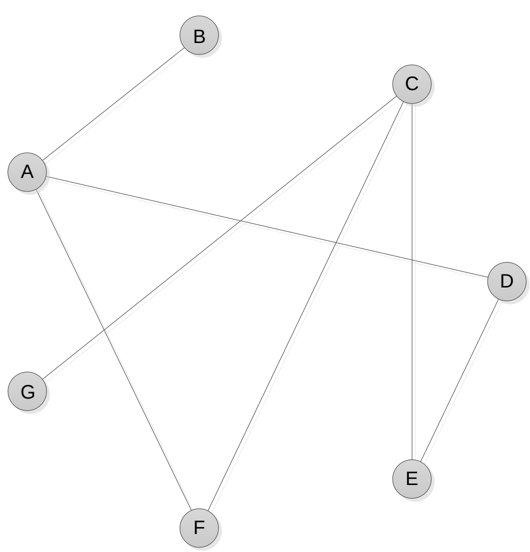

Figure 8.1: Sparsely connected network

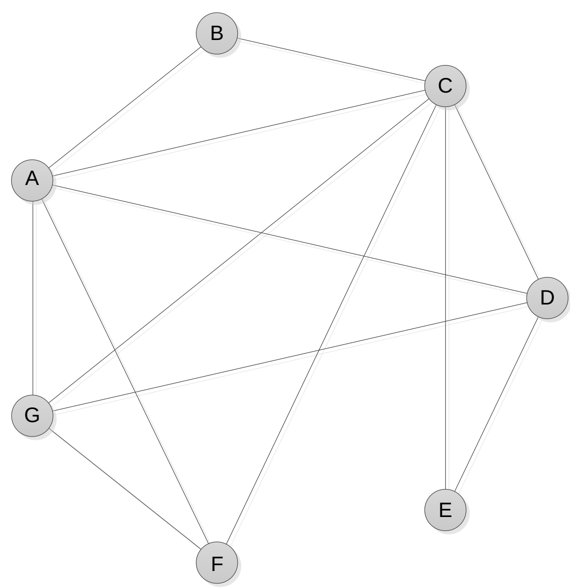

Figure 8.2: Densely connected network

The exchange of one of the more complicated negotiations at time step 23347 (6:29:07) when $\lambda = 0.017$ (61.2 requests/hour) is listed in Table 8.3 and diagrammed in Figure 8.3. The initial proposal is for VIDEO from node F to node E. It is rejected along the route by node C. This information is passed back to node F, which then finds a new route from F to E that does not include A. This route also fails, this time at node A. Node F then asserts a high priority status through an argument targeting node A. Node A finds a connection from D to B that passes through A that can be downgraded from HDV to VID to clear up room for the proposed connection. It then sends a downgrade request to node D, which accepts the request. Since node A is just an intermediate hop on the downgraded connection's route, it sends a notification in the opposite direction to the destination node, i.e. towards B. Once A receives the confirmation associated with the notification, it again retrieves the original proposal to forward to the next node on the route. Finally, accept messages flow back to node F and the negotiation concludes successfully, at the expense to the connection between D and B of a single level downgrade. The confirmation message actually arose in order to keep the simulation synchronized in terms of message passing. It does seem useful in an implementation in order to roll back the transaction if a link is broken somewhere. The protocol as outlined here can be made transactional to a point if reliable message handling is used and a rollback message added. This would be necessary to keep the agents' knowledge bases consistent in terms of active connections and link loadings. Rollback of a downgraded connection would be the greatest challenge, mostly complicated by the record keeping necessary to perform a rollback.

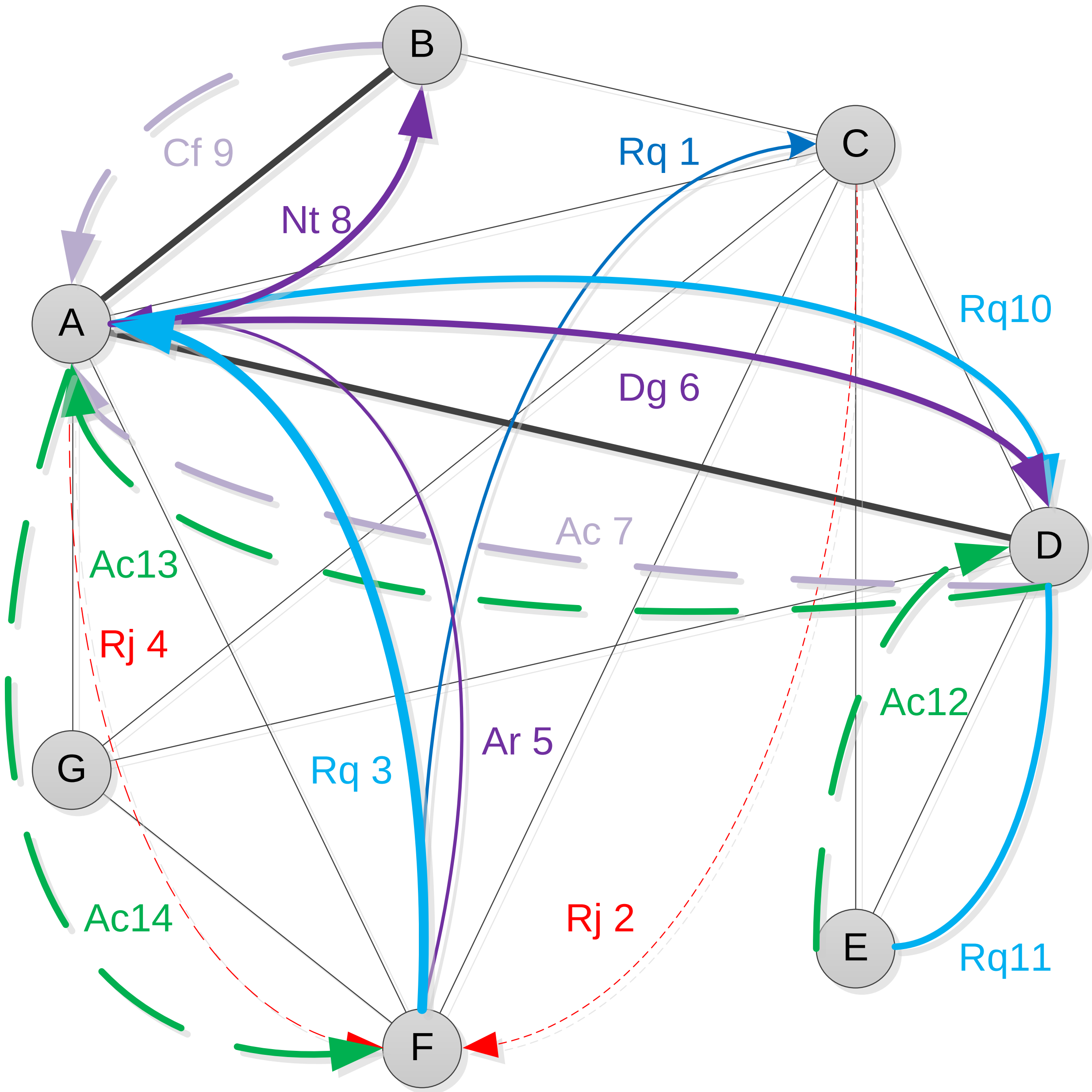

Figure 8.3: Message exchange including downgrade and notify messages

Table 8.3: Messages passed in scenario 1

#	S	R	Message
			$Desire, F, E, VID, 23347, 26965$
1	F	E	$Request, F, E, VID, 23347, 26965, [A, F, C, E]$
2	E	F	$Reject, C, NOCAPACITY$
3	F	A	$Request, F, E, VID, 23347, 26965, [F, A, D, E]$
4	A	F	$Reject, C, NOCAPACITY$
5	F	A	$Argue, HIGHPRIORITY$
6	A	D	$Downgrade, Conn[D, B], HDV, VID, [A, D]$
7	D	A	$Accept, [D, A]$
8	A	B	$Notify, HDV, VID, Conn[D, B], [A, B]$
9	B	A	$Confirm, [B, A]$
10	C	E	$(fwd)Rq, F, E, VID, 23347, 26965, [F, A, D, E]$
11	D	E	$(fwd)Rq, F, E, VID, 23347, 26965, [F, A, D, E]$
12	E	D	$Accept, [E, D, A, F]$
13	E	D	$Accept, [E, D, A, F]$
14	A	F	$Accept, [E, D, A, F]$

Figures 8.4 and 8.5 demonstrate the superior total bits transferred over a simulated day for a sparsely connected network and a densely connected one at a low request rate of 79 requests per hour. Our argumentation based approach yield a clearly higher throughput in most cases. In the case of the densely connected network with a low request rate, the difference is marginal; however, the network is only lightly challenged. In Figures 8.4 and 8.5, the upper set of curves illustrate the effects of a much higher request rate of 180 requests per hour

where the network is challenged in terms of load. In these figures, and many that follow, the simulation was repeated for 100 trials with new seeds for the random generators determining the request generation, the source and destination nodes, the grade of service, the priority, and the length of the requested lease. The requested lease length is drawn from a normal distribution with a mean of 1 hour and a standard deviation of 10 minutes. The mean results are plotted with upper and lower 95% confidence intervals also plotted in dotted lines.

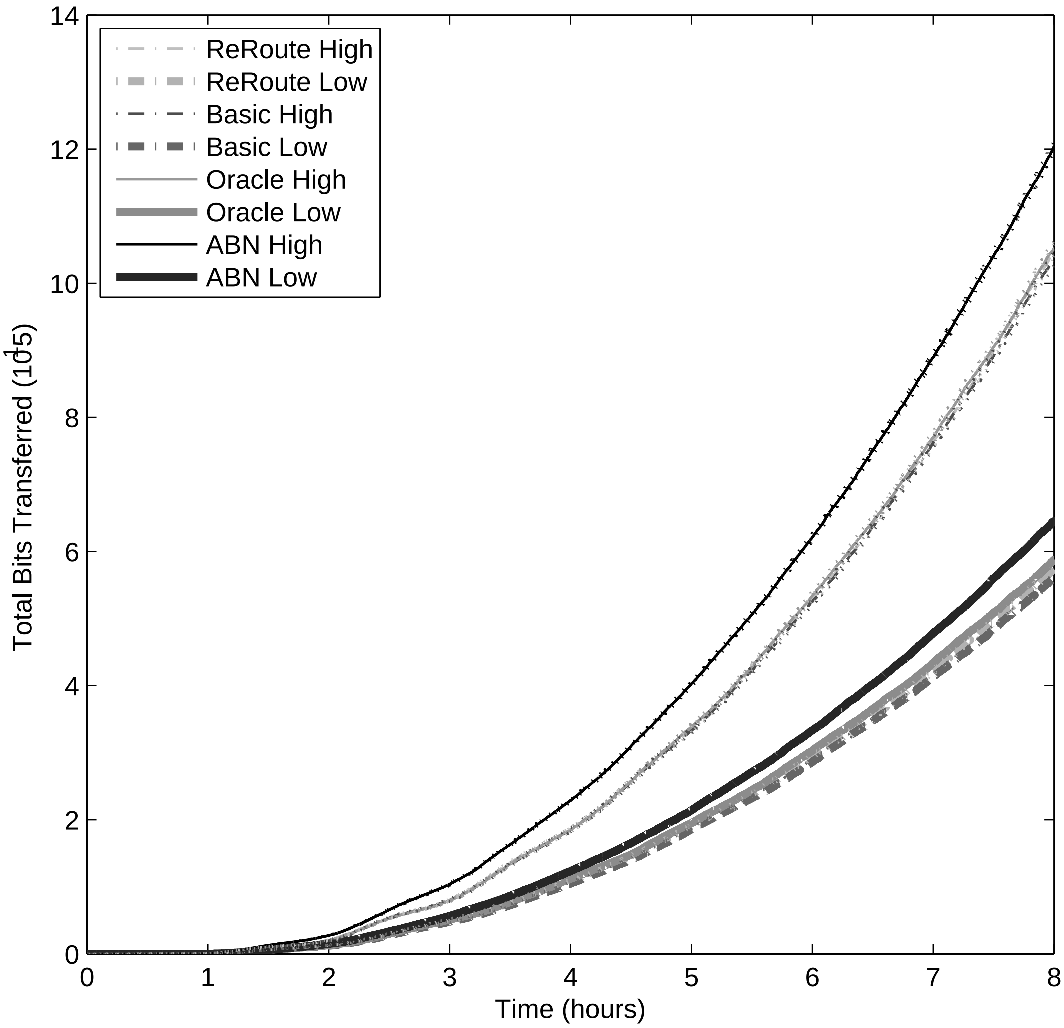

Figure 8.4: Total Bits Transferred With Low Request Rate Sparsely Connected Network

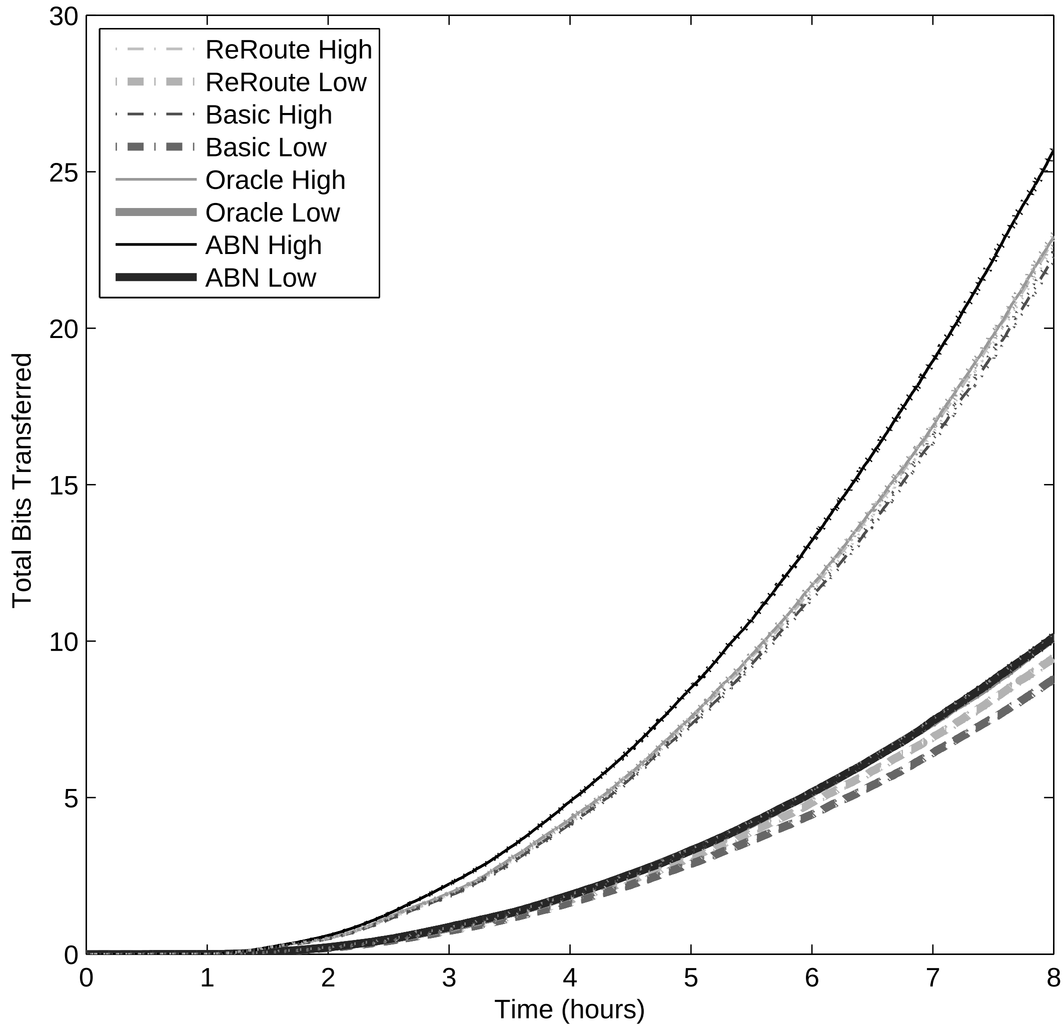

Figure 8.5: Total Bits Transferred With Low Request Rate Densely Connected Network

We also characterized the number of active connections supported over time. The scenario generator randomly creates desires for connections at one of three different grades with durations around a mean of one hour. In this and subsequent plots, the mean for 100 randomized trials is plotted for each case, with 95% confidence limits plotted with dashed lines on either side of the mean. With a review of Figures 8.6 and 8.7, one can see that the argumentation approach supports more connections than any of the other techniques at higher request rates. This is accomplished mainly through the downgrade process that is essential to satisfying new priority connections, while preserving existing connections' ability to communicate, albeit at a low rate. It is also clear that the system produces consistent results under a variety of trials.

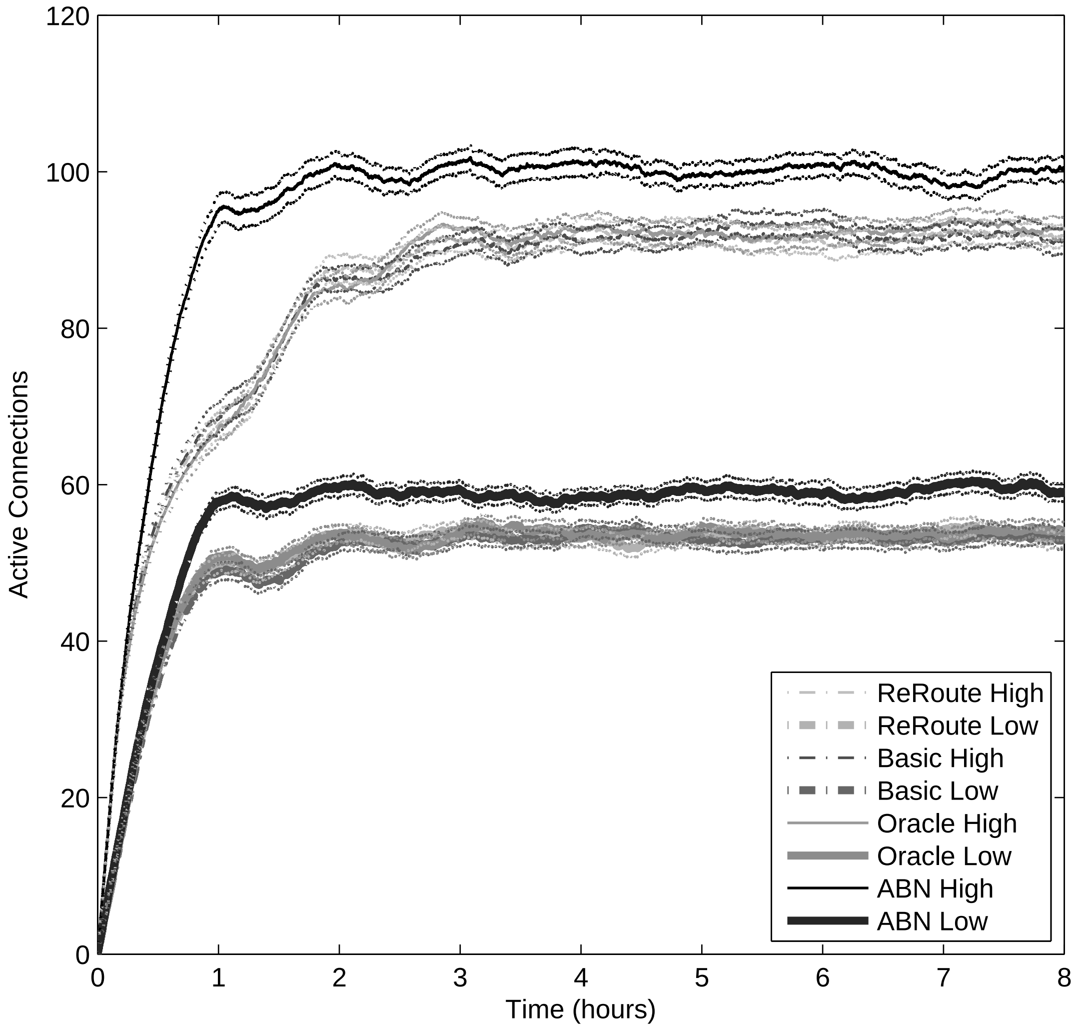

Figure 8.6: Total Active Connections With High Request Rate Sparsely Connected Network

Figures 8.8 and 8.9 illustrate the failure rate for the various models at a request rate of 79 request/hour and 180 requests/hour with both sparse and dense connectivity. As expected, the argumentation approach has the lowest failure rate. This is due to the ability to downgrade connections in order to squeeze more into the finite capacity. Densely connected networks suffer fewer failures than sparsely connected ones since there are more options to explore. It is interesting to note that the ReRouting agent performs worse than the basic agent. Even though re-routing may cause a single proposal to be accepted, it actually loads more nodes, thereby making future proposals more likely to run into capacity constraints. Perhaps in actual operation, the greater number of hops might mean a shorter distance between each hop. This would enable a higher order modulation to maintain a constant signal to noise ratio at the same power, leading to a narrower bandwidth. Ultimately, the channel could support more connections. Our model does not account for these details. Since the Oracle model has perfect knowledge of the link capacities throughout the network, it never fails during a negotiation. Instead, at high loads, it fails at the very beginning in trying to create a proposal from the intention, because it cannot find an available route.

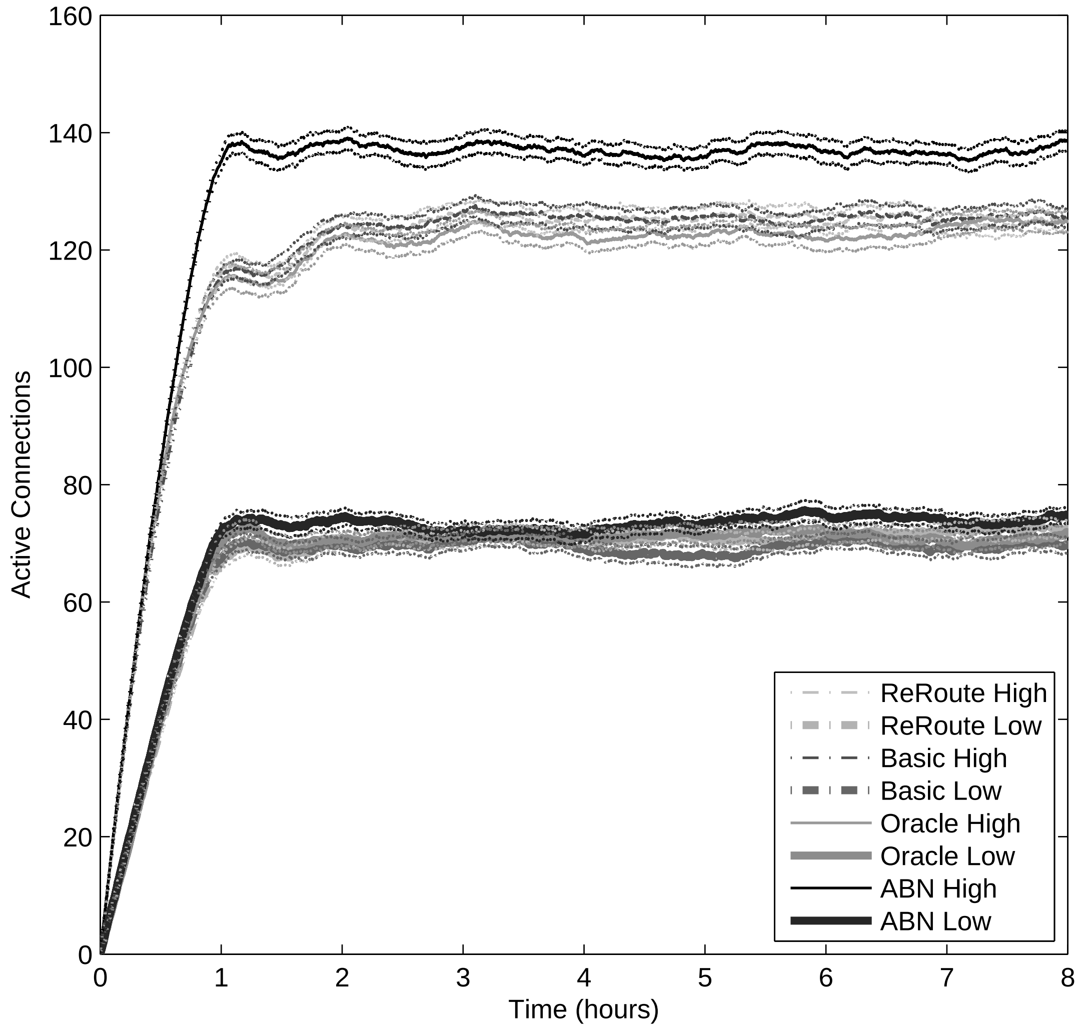

Figure 8.7: Total Active Connections With High Request Rate Densely Connected Network

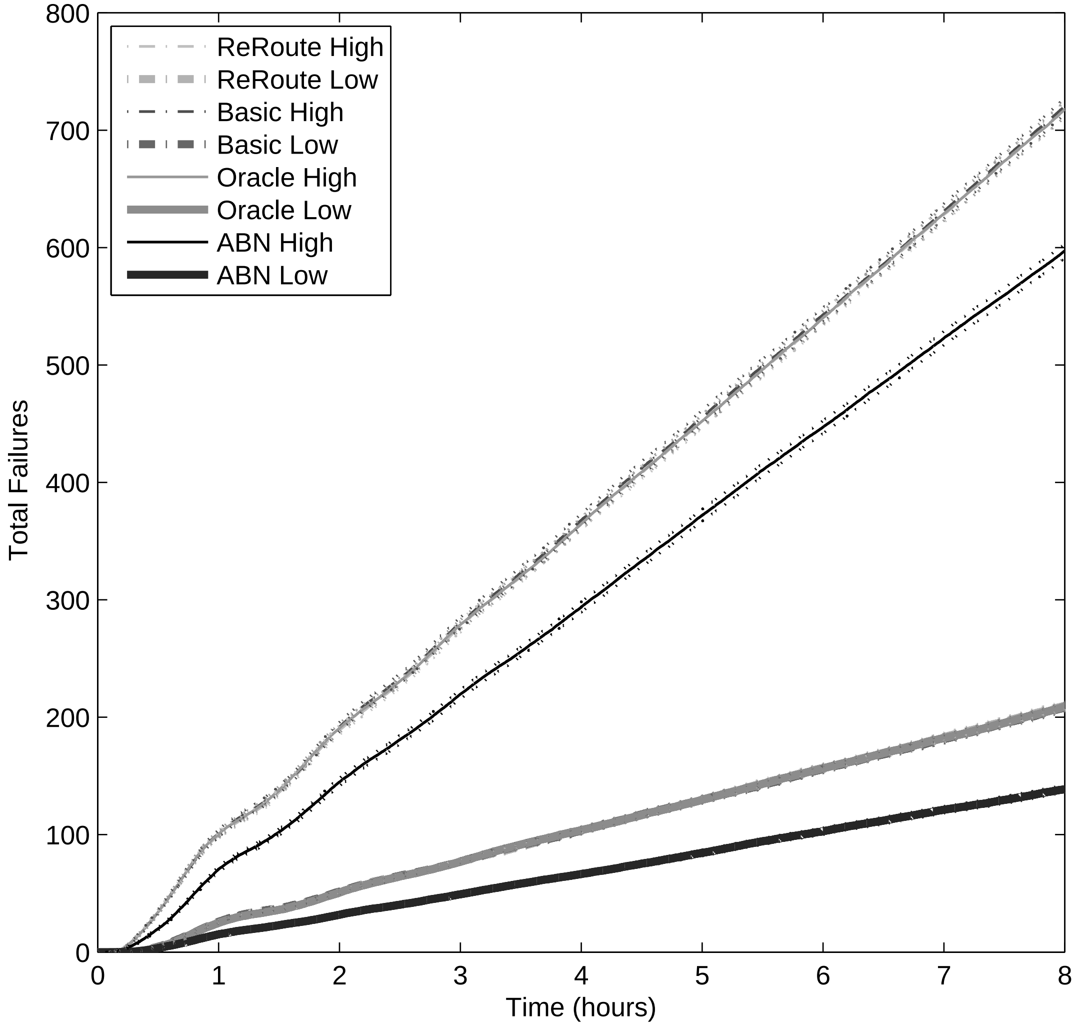

Figure 8.8: Negotiation Failures in Sparsely Connected Network

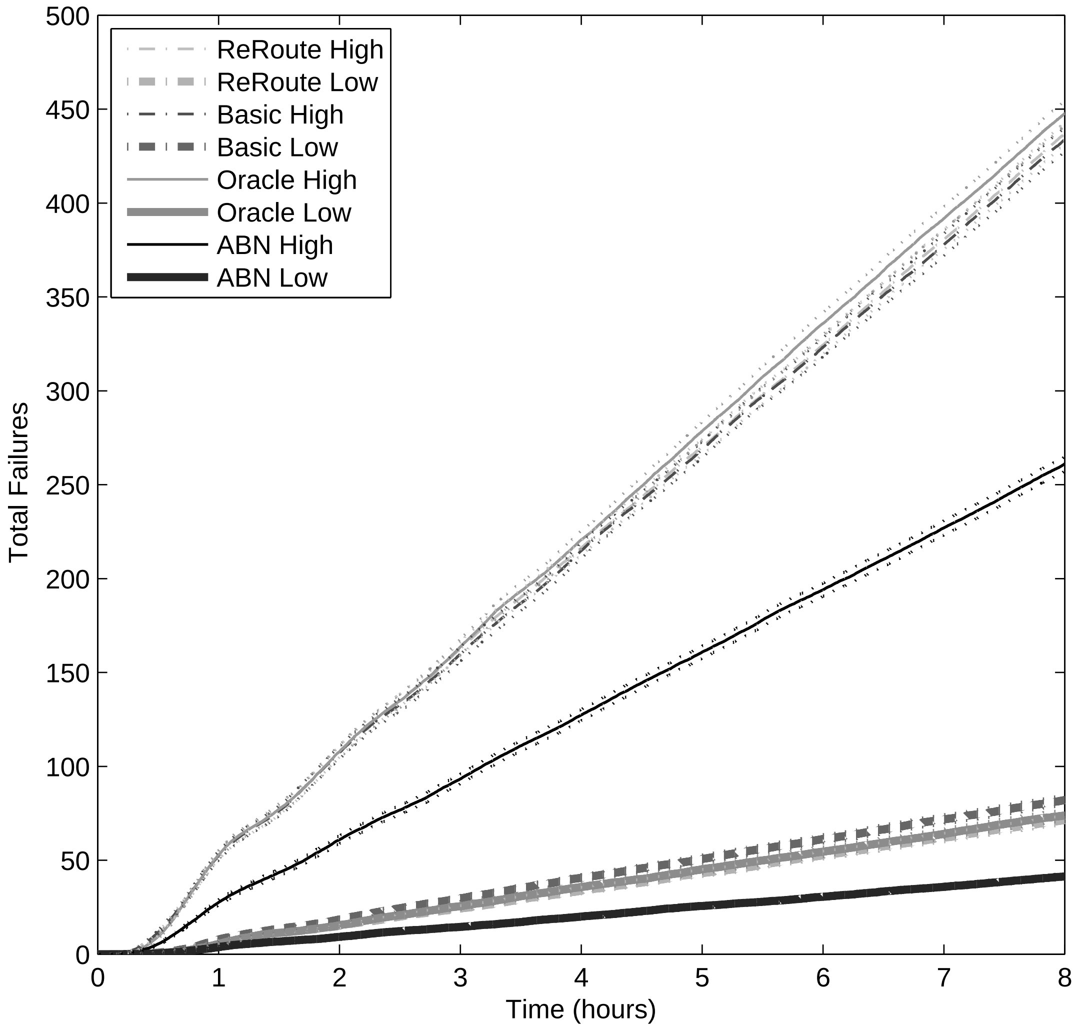

Figure 8.9: Negotiation Failures in Densely Connected Network

One of the concerns in negotiations is the overhead and latency induced by the need to negotiate. We estimate this by examining the number of messages required for each negotiation model. As seen in Figure 8.10, there is only a small penalty in using the more powerful argumentation approach. Recalling our statements in Section 7, these predictions are likely to be pessimistic compared to an operational system. Here, we have plotted the confidence intervals using error bars for clarity.

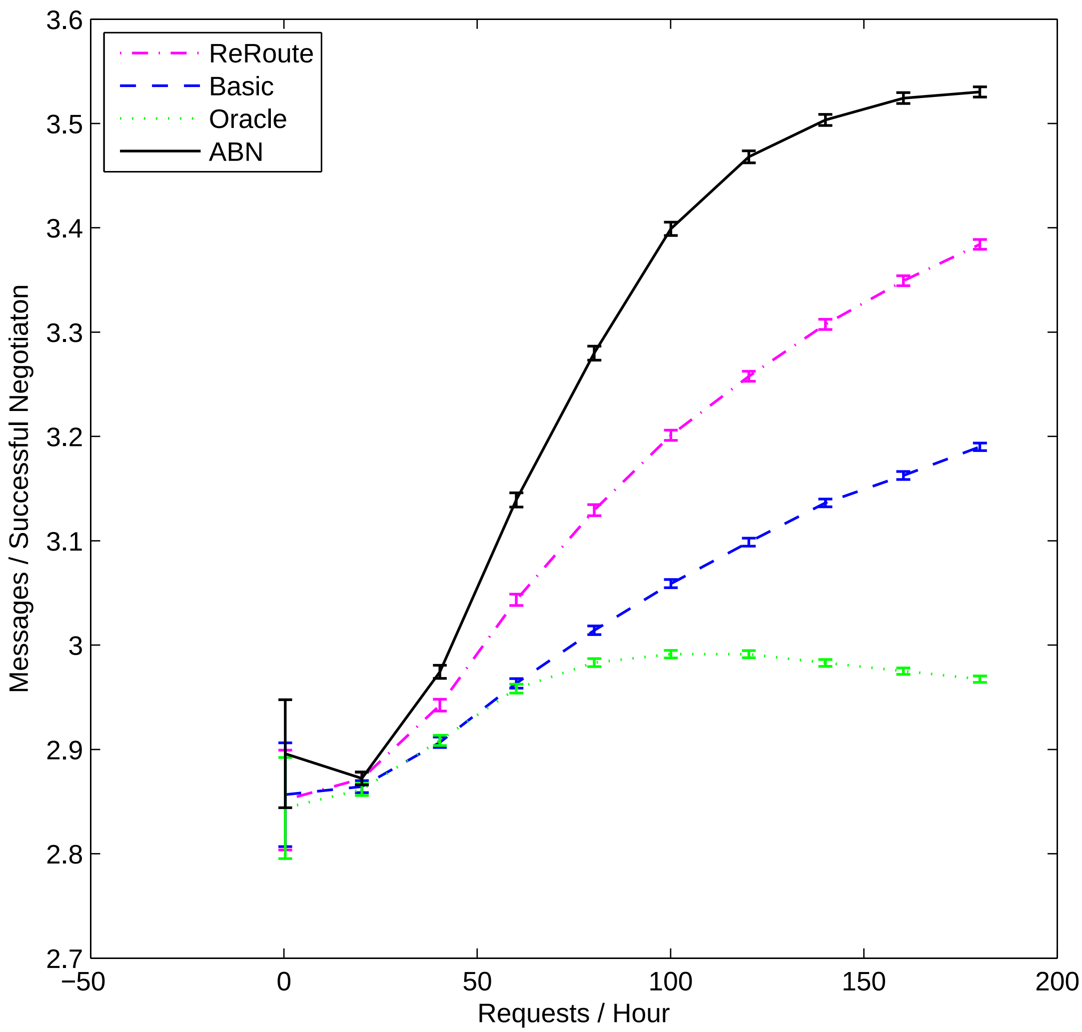

Figure 8.10: Number of Messages per Successful Negotiation

In conclusion, we have presented an illustrated scenario that explores the ability of our negotiation framework to handle a network with many existing connection and a high rate of new requests. This example studies four different negotiating agents with unique strategies and rules. The more complex argumentation based negotiation agent performs better than the others in terms of total bits transferred during the simulation period, the total number of active connections, and in minimizing the number of failed requests. It accomplishes this with highly constrained rules and with a small number of messages. The negotiation system appears to be stable even as existing connections expire. The improvement is enabled by the concept of renegotiating the terms of existing connect leases, and is facilitated by the negotiation framework.

We have now shown how our framework reaches deals in two significantly different networks with different rules and negotiation protocols. This demonstrates that our framework is generic and can be used for many applications. In fact, careful observation reveals that the second negotiation model could even be used in wired networks to allocate limited pipe capacity for streaming multi-media. Also noteworthy is that we achieve superior results with a small number of messages. This opens the door to use this technique at the physical layer. Finally, multi-agent argumentation based negotiation does not necessarily have to involve running on multiple radios. We can envision running this system as a scheduler on a base station controller. Each mobile radio would be represented by an agent negotiating on its behalf, but running on the base station with only minimal messages between the agent and the radio node. This could allow for much more complex negotiations, even when time is a premium.

CHAPTER 9: CONCLUSION AND FUTURE WORK

We have demonstrated the feasibility and characterized the performance of our Frequency Parallel Blind Link Rendezvous algorithm. It is designed to be highly compatible with modern adaptable radio hardware and waveforms. It's operating parameters can be adjusted according to the application to conservatively respect Primary Users in avoiding unintended interference. We designed and analyzed two different detectors. We also related its performance in a noisy environment to conventions frequency hopping sequences and discussed some of the tradeoffs between the two technologies. We have implemented this algorithm in a detailed Matlab simulator and also in a GNURadio software defined radio platform. The algorithm itself is generic and there are many features that can be adjusted to fit a particular application. Future research on this topic could involve optimizing the design of the probe signals, creating and analyzing detectors customized to the specific probe signal, and determining the best way to ramp the signal amplitude and duration. It will also be illuminating to design and deploy implementations of the algorithm on hardware such as FPGAs, cell array processors, multi-core DSPs, or graphics processor units. Finally, with actual hardware implementations, one could characterize the power consumption versus scanning rate in particular, and other design parameters in general.

We created a negotiating framework based upon multi-agent systems with argumentation. This framework was used to develop two very different networks. We demonstrated that even with a tight constraint on the negotiating options, improvements in total active connections and bit rate are achieved, while reducing the number of failures in extremely stressed network loading situations. A Java based simulator aids in the characterization of the system and provides a solid code base from which to develop a run-time fielded system.

Now that the general feasibility and utility of this approach is established, the rules and protocol can be optimized for various real-world applications. These applications will likely include a more complex set of rules, and therefore, more complex arguments. A significant advance would be to include effective arguments in proposals as they are first offered. This will reduce the number of messages required to reach a deal in more complex negotiations. The challenge will be to choose the best argument amongst possibly conflicting ones. Another advancement could include the reasoning of priority arguments based upon more abstract facts. One example we briefly considered involves an earthquake disaster response scenario where a gas company dispatcher request might receive elevated priority, for example over a fire department dispatcher, when the system can reason that repairing pipelines may reduce the risk of fires. This will require an ontology covering the relevant facts and a reasoner operating over those facts. Security will be an important issue in most networks. The system should be able to authenticate nodes issuing requests and ensure that those making priority assertions really have that priority. Some of the work involving game theory in cognitive radio can be used to create a system of rewards and punishments to encourage truth telling and cooperation.

When considering real world implementations, one will have to consider handling timeouts through a roll back procedure. While the system is designed to be robust to timeouts, this capability will need to be implemented for a practical system. Practical systems will also require nodes to determine the capacity of their links. This will involve relating requested data rates to RF parameters through Shannon's Capacity Theorem and channel sounding. With this, they will b able to negotiate over RF parameters. This leads to a naturally heterogeneous network where each node and each link will have a different capacity, that may change over time. One can also consider heterogeneous capabilities of each node in terms of supported modulation types, output power, and other performance parameters.

This will lead to new argument facts informing other nodes of its capabilities. Request can then be more precisely tailored to the target, increasing the likelihood of success. These more complicated negotiating models in turn prompt more consideration of what arguments to send and how to resolve conflicts in the knowledge base, and when to expire stale facts. Finally, many fielded systems involve mobility. This puts greater stresses on the efficiency and speed of the negotiating system and makes managing the timeliness of data more critical.

LIST OF REFERENCES

[1] "Auction 78 aws-1 and broadbandpcs." `http://www.gnu.org/software/gnuradio/`. [Online; access 18-March-2012].

[2] P. Kolodzy, "Spectrum policy task force report," Tech. Rep. ET Docket No. 02-135, FCC, November 2002.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201–230, February 2005.

[4] B. Horine and D. Turgut, "Link rendezvous protocol for cognitive radio networks," in *IEEE DySPAN*, April 2007.

[5] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. SAMS, 2001.

[6] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier fdma for uplink wireless transmission," *IEEE Vehicular Technology Magazine*, vol. 1, pp. 30–38, September 2006.

[7] "Wimax forum." `www.wimaxforum.org`. [Online; accessed 19-October-2012].

[8] "The ieee 802.16 working group on broadband wireless access standards." `www.ieee802.org/16/`. [Online; accessed 19-October-2012].

[9] "3gppp." `www.3gpp.org`. [Online; accessed 19-October-2012].

[10] "Ieee 802.11." `www.ieee802.org/11`. [Online; accessed 19-October-2012].

[11] "Dvb-t." `www.dvb.org`. [Online; accessed 19-October-2012].

[12] G. Ganesan and Y. G. Li, "Cooperative spectrum sensing in cognitive radio networks," in *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 137–143, November 2005.

[13] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 131–136, November 2005.

[14] J. Polson, "Cognitive radio: The technologies," in *Cognitive Radio Technologies* (B. Fette, ed.), ch. 4, pp. 153–155, Elsevier, 2006.

[15] K. Balachandran and J. H. Kang, "Neighbor discovery with dynamic spectrum access in adhoc networks," in *IEEE 63rd Vehicular Technology Conference*, vol. 2, pp. 512–517, 2006.

[16] C. Han, J. Wang, and S. Li, "A spectrum exchange mechanism in cognitive radio contexts," in *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2006.

[17] O. Holland, A. Attar, N. Olaziregi, N. Sattari, and A. H. Aghvami, "A universal resource awareness channel for cognitive radio," in *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2006.

[18] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary signatures for rendezvous in ofdm-based dynamic spectrum access networks," in *IEEE DySPAN*, April 2007.

[19] J. Zhao, H. Zheng, and G. H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 259–268, November 2005.

[20] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation isssues in spectrum sensing for cognitive radios," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 772–776, 2004.

[21] A. E. Leu, K. Steadman, M. McHentry, and J. Bates, "Ultra sensitive tv detector measurements," in *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 30–36, November 2005.

[22] H. Tang, "Some physical layer issues of wide-band cognitive radio systems," in *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 151–159, November 2005.

[23] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks IEEE DySPAN*, 2008.

[24] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 216–227, February 2011.

[25] M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "Dimsumnet: New directions in wireless networking using coordinated dynamic spectrum access," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 78–85, June 2005.

[26] H. Lin, Z. Lin, X. Chu, and Y.-W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *GREENCOM-CPSCOM '10 Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, 2010.

[27] Z. Lin, H. Lin, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *2011 Preceedings IEEE INFOCOM*, 2011.

[28] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, 2012. Early Access.

[29] D. Pu, A. M. Wyglinski, and M. McLernon, "An analysis of frequency rendezvous for decentralized dynamic spectrum access," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1652–1658, May 2010.

[30] "Gnu radio." `http://www.gnu.org/software/gnuradio/`. [Online; access 10-March-2012].

[31] B. Horine and D. Turgut, "Performance analysis of link rendezvous protocol for cognitive radio networks," in *CrownCom*, pp. 503–507, 2007.

[32] J. Mitola III, *Cognitive Radio Architecture - The Engineering Foundations of Radio XML*. John Wiley & Sons, Inc., 2006.

[33] J. Mitola III, *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology, 2000.

[34] T. W. Rondeau, C. J. Rieser, B. Le, and C. W. Bostian, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios," in *SDR Forum Technical Conference*, 2004.

[35] C. J. Rieser, *Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic ALgorithms for Secure and Robust Wireless Communications and Networking*. PhD thesis, Virginia Polytechnic Institute and State University, 2004.

[36] J. Neel, R. Buehrer, B. Reed, and R. Gilles, "Game theoretic analysis of a network of cognitive radios," in *45th Midwest Symposium on Circuits and Systems MWSCAS-2002*, 2002.

[37] J. Neel, J. H. Reed, and R. P. Gilles, "The role of game theory in the analysis of software radio networks," in *SDR Forum Technical Conference*, 2002.

[38] G. Denker, D. Elenius, R. Senanayake, M.-O. Stehr, and D. Wilkins, "A policy engine for spectrum sharing," in *IEEE DySPAN 2007*, April 2007.

[39] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, pp. 169–185, 2002.

[40] W. D. Ross, ed., *Topics*. Oxford, UK: Clarendon Press, 1928. (Original work published c. 350 BCE).

[41] M. Wooldridge, *An Introduction to Multiagent Systems*. John Wiley & Sons, Ltd, 2002.

[42] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg, "Argumentation-based negotiation," in *The Knowledge Engineering Review*, vol. 18, pp. 343–375, Cambridge University Press, Dec 2003.

[43] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and -person games," *Aritificial Intelligence*, vol. 77, pp. 321–357, September 1995.

[44] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, "A framework for argumentation-based negotiation," in *Intelligent Agents IV: Agent Theories, Architectures, and Languages: 4th* (M. P. Singh, A. Rao, and M. J. Wooldridge, eds.), vol. 1365 of *Lecture Notes in Artificial Intelligence*, pp. 117–192, Springer-Verlag, July 1997.

[45] S. Parsons, C. Sierra, and N. Jennings, "Agents that reason and negotiate by arguing," *Journal of Logic and Computation*, vol. 8, no. 3, pp. 261–292, 1998.

[46] F. Giunchiglia and L. Serafini, "Multi-language hierarchical logics or: How we can do without modal logics," *Aritificial Intelligence*, vol. 65, no. 1, pp. 29–70, 1994.

[47] L. Amgoud, S. Parsons, and N. Maudet, "Arguments, dialogue, and negotiation," in *Proceedings of the 14th European Conference on Artificial Intelligence* (W. Horn, ed.), (Amsterdam), IOS Press, 2000.

[48] S. Parsons, M. Wooldridge, and L. Amgoud, "Properties and complexity of some formal inter-agent dialogues," *Journal of Logic and Computation*, vol. 13, no. 3, pp. 347–376, 2003.

[49] D. N. Walton and E. C. W. Krabbe, *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, 1995.

[50] "Resource ReSerVation protocol (RSVP) – version 1 functional specification," RFC 2205, Internet Engineering Task Force, Sept. 1997.

[51] "A resource reservation protocol (rsvp) extension for the reduction of bandwidth of a reservation flow," RFC 4495, Internet Engineering Task Force, May 2006.

[52] L. Hanzo II and R. Tafazolli, "A survey of qos routing solutions for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 9, pp. 50–70, 2007.

[53] J. Pitt, P. Venkataram, and A. Mamdani, "Qos management in manets using norm-governed agent societies," in *Engineering Societies in the Agents World VI* (O. Dikenelli, M.-P. Gleizes, and A. Ricci, eds.), vol. 3963 of *Lecture Notes in Computer Science*, pp. 221–240, Springer Berlin / Heidelberg, 2006. 10.1007/11759683_14.

[54] S. S. Manvi, M. S. Kakkasageri, and J. Pitt, "Multiagent based information dissemination in vehicular ad hoc networks," *Mobile Information Systems*, vol. 5, no. 4, pp. 363–389, 2009.

[55] A. Förster, "Machine learning techniques applied to wireless networks: Guide and survey," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007.

[56] J. Liu and V. Issarny, "Enhanced reputation mechanism for mobile ad hoc networks," in *Second International Conference on Trust Management: iTrust*, 2004.

[57] L. Gan, J. Liu, and X. Jin, "Agent-based, energy efficient routing in sensor networks," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.

[58] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing.* Prentice-Hall, 3rd ed., 1996. Principles, Algorithms, and Applications.

[59] J. F. Chicharo and M. T. Kilani, "A sliding goertzel algorithm," *Signal Processing*, vol. 52, pp. 283–297, August 1996.

[60] E. Jacobsen and R. Lyons, "The sliding dft," *IEEE Signal Processing Magazine*, vol. 20, pp. 74–80, March 2003.

[61] S. Parsons and P. McBurney, "Argumentation-based dialogues for agent coordination," *Group Decision and negotiation*, vol. 12, pp. 415–439, 2003.

[62] L. Amgoud and S. Vesic, "A formal analysis of the role of argumentation in negotiation dialogues," *Journal of Logic and Computation*, 2011.

[63] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, “Routing protocols in ad hoc networks: A survey,” *Computer Networks*, vol. 55, pp. 3032–3080, September 2011.