Social Intelligence for Cognitive Radios

Nicholas J. Kaminski

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Charles W. Bostian, Chair
Michael S. Hsiao
Allen B. MacKenzie
Jeffery H. Reed
Robert H. Sturges

February 10, 2014
Blacksburg, Virginia

Keywords: Cognitive Radio, Cognitive Engine, Social Language, Social Learning, Behavior Based Approach, LTE Organization

Social Intelligence for Cognitive Radios

Nicholas J. Kaminski

(ABSTRACT)

This dissertation introduces the concept of an artificial society based on the use of an action based social language combined with the behavior-based approach to the construction of multi-agent systems to address the problem of developing decentralized, self-organizing networks that dynamically fit into their environment. In the course of accomplishing this, social language is defined as an efficient method for communicating coordination information among cognitive radios inspired by natural societies. This communication method connects the radios within a network in a way that allows the network to learn in a distributed holistic manner. The behavior-based approach to developing multi-agent systems from the field of robotics provides the framework for developing these learning networks. In this approach several behaviors are used to address the multiple objectives of a cognitive radio society and then combined to achieve emergent properties and behaviors. This work presents a prototype cognitive radio society. This society is implemented, using low complexity hardware, and evaluated. The work does not focus on the development of optimized techniques, but rather the complementary design of techniques and agents to create dynamic, decentralized self-organizing networks.

Grant Information

This work was supported by a Bradley Fellowship from the Bradley Department of Electrical and Computer Engineering at Virginia Tech.

Acknowledgments

What is said in these acknowledgments can not hope to express the proper amount of gratitude. Nevertheless, that which can be expressed here, must be expressed.

I will forever maintain that the best thing about receiving the Bradley Fellowship was the opportunity to work with Charles Bostian. Thank you for taking me on as your student.

To my wife, I couldn't appreciate your vigilant support more. I think we both know I wouldn't be the same without you. Thank you so much for trying to keep me sane.

To my parents, in many ways, my pursuit of this is your fault. All those years have lead me here and I couldn't be happier with the result. Thank you for being you and helping to make me me.

To my past lab mates, in particular Alex and Al, you two left too soon. Alex working on all those projects was a blast and certainly helped propel me along. Al you were always able to introduce me to a new perspective. Thank you both for helping me through those early days.

All figures in this document were created by the author.

Contents

List of Figures

List of Tables

Acronyms

API	application programming interface
AWGN	additive white Gaussian noise
CE	cognitive engine
CR	cognitive radio
DSA	dynamic spectrum access
FPGA	field programmable gate array
FSK	frequency shift keying
GFSK	Gaussian frequency shift keying
GA	genetic algorithm
IQR	inner quartile range
LTE	Long Term Evolution
MAC	medium access control
OOK	on-off keying
PCO	pulse coupled oscillator
pmf	probability mass function
RF	radio frequency
RFIC	radio frequency integrated circuit
RSSI	received signal strength indication
TDD	time division duplex
TDMA	time division multiple access

Chapter 1

Overview and Problem Statement

1.1 Problem Statement

Cognitive Radios are the fusion of synthetic intelligence and flexible radio platforms. This combination was originally envisioned by Mitola as a sort of personal radio assistant [1]. Mitola defined several degrees of cognition that might be exhibited by a cognitive radio (CR). In the fullest expression of his vision, radios would anticipate a user's needs, contact any peers that can help meet these needs, dynamically access open portions of the spectrum, collectively determine a protocol for interaction, and tailor their operation subject to a set of objectives. These highly cognitive radios would automatically handle any of a user's wireless desires. Since Mitola's pivotal work, a great of work has been done in an effort to achieve such highly cognitive devices. An important step along this path is the deployment of self-organizing networks that dynamically fit into their environment without need for human intervention. However, the reality is that current systems have yet to reach this milestone.

Currently, there are two primary ways that CR systems fail to achieve the goal of dynamically self-organizing networks. Most CR systems rely on communicating a great deal of information between nodes. This over sharing of information results in the "faculty meeting" problem [2], i.e., nodes spend all of their efforts "talking" about themselves and sharing their own findings without accomplishing the task at hand. Their desire to share this amount of information requires the imposition of some central agent to broker and control the deluge. For example, DARPA's XG dynamic spectrum access (DSA) program [3] relies on various centralized subsystems to manage the communication and resource usage of nodes. Spectrum brokers [4] are centralized entities that offer similar abilities. Such centralization offers many advantages, however, it also requires a good deal of overhead (both in infrastructure and communication) and it does not scale well as the network size increases.

This work address both of the issues raised above through the building of CR societies. The combination of intelligence and flexibility that defines CRs allow for the construction of self-organizing societies based on the use of efficient social languages. These social languages allow the radios to communicate only the most pertinent information through a combination of deliberate behavior and common understanding. The use of such social languages parallels several societies found in nature, from ants and bees to human beings.

1.2 Introduction of Methods

Building CR societies allows individual CRs to use their own intelligence to adapt to local environments while maintaining societal coherence through the social language. The result of this process is a society made up of distinct radios that is able to adapt and learn as though it were a single organism. Emergent behavior rises from the local interactions and communication of radios that organizes the radios around each other and other radios outside of the society.

Figure 1.1 shows a radio society navigating through frequency and time. In this figure the radio society avoids interference in order to find an open channel. Once in this open channel the society organizes itself in time providing each member with a time slot for communicating application data. The shaded areas represent external interference faced by the radio society. Each radio in the society is represented by its own shape and color. The shapes on the upper plot display the time and frequency of each radio's transmissions for each of its slots. The lower plot shows the throughput and length of each radio's slot.

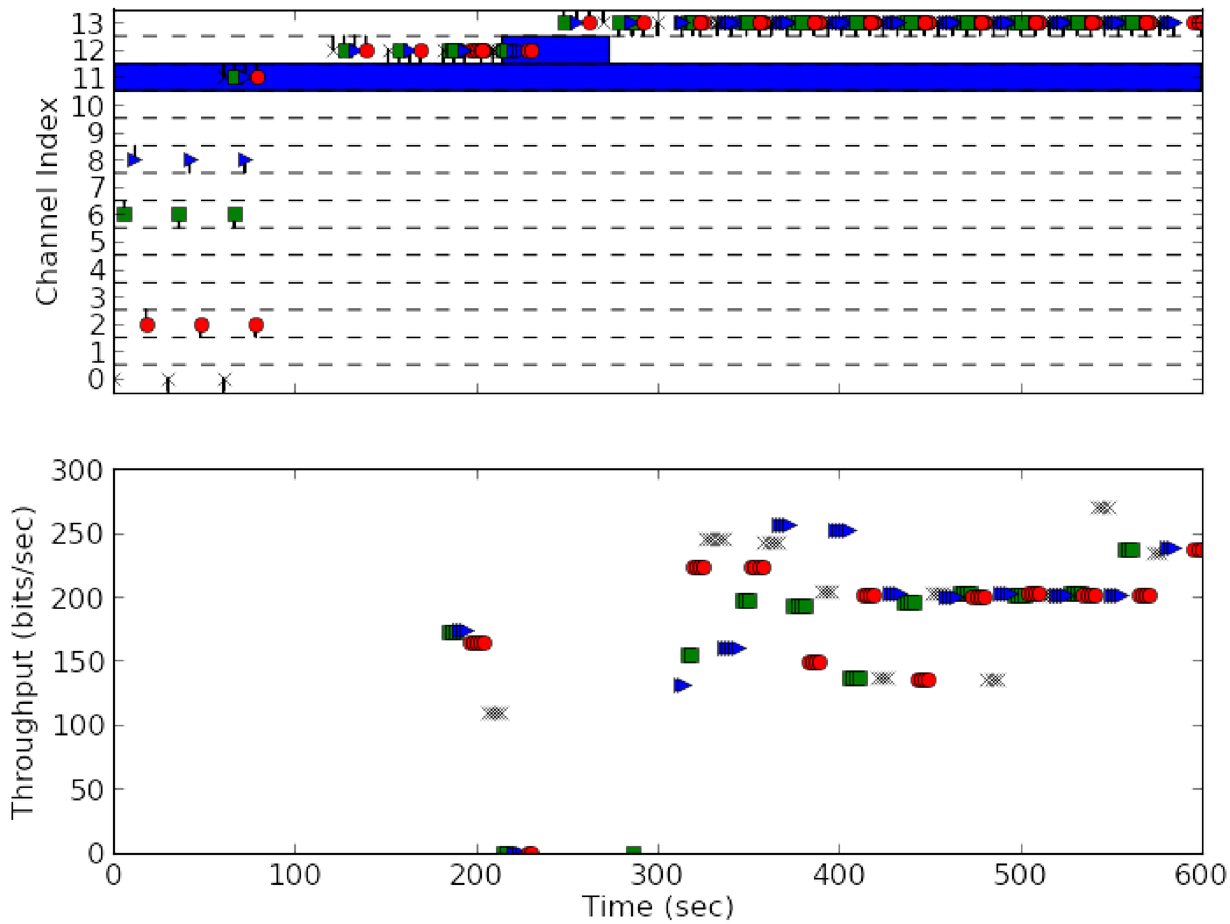


Figure 1.1: Radios Navigating Frequency/Time Space

The society depicted in this graphic is faced with the common CR problem of attempting to coexist with various other radio systems. In this situation, the onus is on the CR system to avoid interference with other networks, while meeting its own needs for spectrum access. To accomplish this spectrum sharing the CR society selects frequencies and transmission slots that fit around the transmissions of non-member radios in a manner that still allows for good communication between member radios.

The society shown in Figure 1.1 is based on the simultaneous time flocking and frequency flocking behaviors. These behaviors use a social language in the form of a single packet beacon to coordinate radios around the obstacles in time and frequency. Each behavior specifies a radios actions based upon its understanding of its local environment. The operation of the society then arises from the accumulated effect of each individual radios behavior. Thus this society is not directly controlled by the flocking behaviors, but rather it is controlled through the designed interaction of the behaviors.

Additionally, the society shown in Figure 1.1 is implemented using small inexpensive radio platforms. This work focuses on these platforms because the scalable nature of the approach will likely have greatest appeal for the purpose of building large networks of inexpensive nodes. The method of constructing CR societies removes the need for powerful computational platforms by focusing on intelligent interactions rather than large amounts of data processing.

1.3 Contributions

This work makes several contributions to the field of CR. These are based on the introduction of the concept of a CR society. A CR society provides a method for developing a CR network based on the principles of decentralization and self-organization. This method employs a novel exploration of natural society principles, culminating in the introduction of social languages, and the unprecedented application of behavior-based system design to CRs. In this way, CR societies offer the development of dynamic networks based on multi-disciplinary research.

The principles of CR societies provide network level control through emergence built upon the interaction of local behaviors. This work demonstrates that minor adjustments in individual radio behavior can result in large differences in overall network behavior. This allows for addressing a number of different applications with only minor differences in individual radios. Additionally, networks that apply the concepts of a CR society have access to social learning. In this form of learning, long observed in several natural societies, but never before implemented, the network as a whole is able to learn and adjust its behavior accordingly. Information gathered in this manner does not reside in any particular radio; rather, it is distributed throughout the entire network, existing in the interactions between radios. This work demonstrates the ability to enable social learning as an emergent property.

In order to examine the attributes of a CR society this work implements a society on low complexity hardware platforms. These inexpensive, simple platforms show that the mechanisms that enable CR societies have low demands on hardware capabilities. In the course of implementing a prototype society, this work develops several behaviors for addressing problems commonly faced by CR. These behaviors are designed to be as simple as possible to control interactions, making them applicable to a variety of situations. Specifically the implementation discussed here demonstrates self-organization methods and techniques that could be applied to organization and scheduling problems in Long Term Evolution (LTE) cellular networks.

1.4 Organization of Work

Chapter 2 discusses the concepts of the social language and societal learning. This chapter introduces several sources of inspiration for the work and explores key aspects of successful societies in nature.

Chapter 3 provides background from robotics on the construction of artificial societies. Robotics, as a field, contributes a great deal of practical methods for developing artificial societies.

Chapter 4 examines aspects of the work specific to the field of radios. This chapter specifies the challenges particular to radios and summarizes related work.

Chapter 5 covers the specific approach to building a self-organizing dynamic CR society. The approach employs techniques from both CR and robotics research in order to achieve the benefits exemplified in natural societies.

Chapter 6 details the approach to the implementation of the approach. This chapter discusses the methods used to realize the necessary components, the hardware platforms running the system, and the environment in which the system was tested.

Chapter 7 describes the process of testing the system and presents the results.

Chapter 8 summarizes the work and provides the conclusions of the dissertation.

Chapter 2

Societies and Communication

2.1 Introduction

Any collection of individuals that works together in the same environment must compete and/or cooperate to utilize some limited pool of resources. It is generally accepted that cooperation yields the most productive utilization of resources for a group and its individuals. Societies form out of these groups to foster this cooperation.

Societies foster cooperation through the mechanisms of communication. This mechanism serves to align the goals of individual members and share the relevant data for coordinating members of the society. Such alignment underpins the existence of several organisms, from ants and bees to human beings. Cooperation in these natural societies provides inspiration for the development of artificial societies.

Fundamentally, CRs are intelligent, flexible radios focused on enabling communications. As such, they are well suited to adapt to their surroundings, naturally have a goal that may be achieve only through some form of cooperation, and must cope with limited resources. The attributes and goals of CRs makes the development of artificial societies a natural fit for the advancement of CRs. The communication mechanisms that drive social interactions would allow for the coordination of CRs without greatly diverting the limited resource of spectrum away from the communication of application data, i.e., the primary purpose of CRs.

The development of CR societies based on efficient communication requires an understanding of the communication methods that hold such a society together. Communicating fundamentally requires messages that can be both observed and understood by the intended recipients of the messages. Every society has particular factors that affect these attributes, and an artificial society of CRs would be no different. Analyzing the communication methods available for frequent, efficient CR coordination is necessary to develop social communication methods.

Once social communication methods are in place, societies can be formed. The communicative connections among members of a society allow the society to take on an intelligence of its own. Specifically, the society as a whole is able to learn information that is not necessarily directly apparent to any individual member, but influences the behavior of all members. In this scenario, the information is stored in between the members themselves, in their interactions. This societal learning removes dependence on any particular agent, preserving performance of the group in the face of the entry or exit of any members.

This chapter expands on these concepts in order to develop a social approach for developing cognitive networks. Section 2.2 examines efficient social mechanisms for the communication of coordination information. Section 2.3 provides an understanding of the components of these mechanisms and analyzes how they may be applied to CRs. Section 2.4 discusses how the society learns once the members are afforded the necessary communication. Section 2.5 concludes the chapter.

2.2 Social Language

A society is built on the cooperation of a collection of peers to achieve some goal. It is well known that effective cooperation requires some degree of communication. Herein lies the challenge for CRs; while these devices are designed to enable efficient communication with other entities, they themselves do not yet have an efficient method of communicating with each other. The problem is rooted in the variety and amount of data that CRs typically share for accomplishing their goals.

The standard method for solving this problem is some form of centralized coordination [3,4]. However, this centralization reduces the scalability of the network and increases its overhead. It is worth noting that many natural societies do not require such centralization; Doerr points out that "there is no 'master fish'" in a school of fish [5]. Instead, natural societies tend to rely on efficient stylized forms of communication that are particular to their group, e.g., the waggle dance of bees [6], and the pheromone trails of ants [7]. This "social transmission of information" is necessary to hold many natural societies together without the need of a central authority [8].

Cognitive radio societies require a social language to shed the weight of centralization. In sociology and linguistics, the term social language is used to refer to the union of direct verbal language and the contextual information that surrounds it [9]. The addition of contextual information removes much of the ambiguity that is otherwise present in language, eliminating the need for other forms of clarifying communication. This contextual information represents an intelligent agent's knowledge of the environment and the society it which it is communicating [10]. The union of contextual information and direct communication allows for an efficient, society specific means of communication. For our purposes, we will take social language to be a form of communication that relies on the combination of contextual information and direct communication that is based on societal rules.

Nature provides many examples of social language as defined above. Bees perform a so called waggle dance to indicate the locale of either a new nesting site or a good food source. In this dance a bee travels a straight path waggling its body for some duration before making a non-waggling return to its starting point in order to repeat the process [11]. This dance has many layers of meaning associated with it. The orientation of the dance is the orientation of the object relative to the sun, with up typically being directly in line with the sun. The duration of the waggling phase indicates the distance to the advertised object. Perhaps most interestingly, the intensity of the waggle reveals the bee's excitement about the object. The nature of the object, either food or new nesting site, is indicated entirely by context. When bees are still living within the hive, the dance indicates the location of food. When bees have exited the hive to form a swarm, the dance communicates information about new nesting locations, altering the purpose of the dance to serve the swarm's goal of finding a new home. Additionally, it is worth noting that this dance serves as the basis for a voting mechanism for the bees, with each dancer attempting to elicit acolytes to the support of a particular strategy. Thus, the waggle dance provides a context-reliant means for the bees to communicate without ambiguity.

Human society also employs social languages in a variety of ways. Due to the increased complexity of human society compared to that of bees, there is no single dominant form of social language among humans, although, humans often communicate using society specific, context dependent methods. Perhaps the clearest example of this is pantomime, in which gestures are used to convey some meaning based upon the context in which they are used. Facial expressions are often used in a similar way, communicating a great deal about a person without the need for speech. However, direct verbal communication can itself be a social language. Catchall words such as "thing", "stuff", or "deal" are often used in a manner that depends on the context and broad social rules. The sentence "Hand me that thing." appears to have no meaning on paper, but in many implemented scenarios the context of the sentence clearly indicates the object of desire and the tone of the sentence combines with social hierarchy to indicate the person being addressed. The complexity of human society and the ingenuity of human beings allow for the fluid and dynamic creation of social languages to fill the needs of the moment, resulting in a myriad of actualizations of the concept.

Social language provides the necessary means for a group of CRs to reach a common understanding of both individual abilities and society goals. According to celebrated scholar Jürgen Habermas, successful societies are based on communicative action, which has the sole purpose of achieving mutual understanding between members [12]. In fact, Habermas claims that reasoning is "thinking codified in language" and that rationality rests in how agents acquire and use knowledge rather than the possession of any particular information [12]. This suggests that social language provides the mechanism necessary for a collection of agents to transform from merely a group to a thinking society. Extending the understanding of individual ability to contribute to society wide goals, allows for the society to take on a cognition of its own; social language is necessary for the transition from a network of intelligent individuals to an intelligent network.

Societies based on communicative action benefit from consensual coordination, i.e., members freely agreeing on goals and plans of action [10]. This is contrast with so called strategic action based groups, in which members focus only on their own goals. Coexistence in a strategic action society is inherently contentious in nature; individuals are forced to create threats or bribes to coerce peers into agreeable behavior. This contention ends up achieving the situation that is the least bad for all individuals involved, typically with the cost of coercion as a detracting factor. Consensual coordination, on the hand, allows for achieving situations that are the most beneficial to society members by aligning the individuals in pursuit of goals. However, such a consensus typically may only be reached by members predisposed to coordination, with the means to achieve common understanding of individual abilities and goals.

Note that while consensual cooperation typically provides more ideal results, it is not always available. As discussed above, such cooperation requires society members with the proper disposition to be provided with a efficient means to communicate. When these conditions are not met, game theory provides a means for analyzing contentious cooperation through the application of economic theory to societies [13]. This contentious cooperation requires only that individuals have known goals that they are pursuing in a reasonable manner. Contentious cooperation provides a viable option in situations where consensual cooperation is unfeasible.

Social languages are distinct from what is more traditionally termed a language in that they provide a society-specific mode of communication enabling the coordinated action of the group. These social languages are not designed to convey the fullest expression of emotion, but rather to relate the basic information that holds a society together. As such, social languages aim to transfer only the information most necessary for coordination to allow for frequent use without great penalty. The bee waggle dance is an excellent example of this concept [6]. Honey bees have established a method of broadcasting pertinent information about food or new nesting sites based on actions suited to bees.

Recall that the purpose of a CR is the transport of application data. To do this CRs must often work together in networks. This means that the societies of CRs have the goal of accessing the spectrum in a manner that supports the transport of data between nodes. Social language provides a means to efficiently communicate coordination information, as opposed to application data, in order to accomplish this goal.

2.2.1 Social Language for Cognitive Radios

Inherently, a social language transfer of information is based on efficiently attaching the information to some action available to the communicator. This action needs to be observable to the communicator's peers and the information needs to be attached in an understandable manner. The information that is transmitted in this way should be useful to any given member of the society, not targeted at any particular agent. Social language is based on using the commonalities of the society to communicate information to all local members of the society.

In the case of cognitive radios, social language allows information transfer through understanding the reasons for a peer's actions. Each CR makes decisions about how to control the communications under its own supervision in order to establish and maintain necessary communication links with its peers based on its understanding of its situation. These decisions are then translated into actions based on currently available radio hardware. In this way the CR's actions always encode its understanding of its situation in some way. Spreading this understanding across the entire society allows peers to achieve the mutual understanding discussed by Habermas in [12]. Thus, achieving the benefits of social language in a CR comes down to decoding the information already encoded into the actions of a CR.

Decoding the information that drives a CR's actions requires a common understanding of the method each node uses to determine its own actions. This is most easily accomplished when nodes share a common method for determining their actions. In this situation each node can gather information from its peers that is useful for determining how it should proceed in order to fit into a society. Each node is able to determine its next action based on an estimate of its peers actions in order to avoid interference.

Social language for CR provides the additional information needed to allow convergence of nodes without requiring the exchange of future plans or detailed sensing information. Each node in a society makes local decisions about its own action and performs this action in a manner that allows for the easy decoding of contextual information associated with using one action instead of another. Each other node then simply observes the information conveying details of the performer's action and uses this information to determine its own action. This results in a direct connection between all the actions of every radio in a society without the need for a central control. Through proper use of this connectivity, the society can converge to a useful operating point in an emergent fashion.

Inherently, this connectivity extends the situational awareness of each node in the society. Since each action encodes the reason for using that particular action and each reason is based, at least partially, on other observed action, observing only a few actions can provide enough information to make far-reaching estimates. While this extended awareness allows for the coordination of several nodes, it must be managed to avoid unwanted feedback loops of influence. That is, if all nodes strongly base their actions on other nodes' actions, any action (whether correct or not) will guide the future actions of nodes that observe it. In a realistic scenario, nodes will take improper actions for a variety of reasons, and the web of awareness must be flexible enough to prevent any action from rippling too far in order to prevent false convergence.

2.3 Components of a Social Language

Fundamentally, there are two parts to the successful use of social language: observability and understanding. A social language must be based on sending messages that may be observed by other members of the society. The various factors that are associated with this concept are elaborated upon in Section 2.3.1. Once the message as been witnessed by the receiving node, it must be understood in a useful fashion by that node. The issues related to this attribute are expounded upon in Section 2.3.2.

2.3.1 Observability

Observability of a social language has two complementary factors: the lucidity of the sender and the discernment of the receiver. The sender's lucidity is related to how easy it is to identify a message as important. The discernment of the receiver, on the other hand, concerns the level of a receiver's ability to identify the details of a particular message. For example, examining the waggle dance of bees, it is clear that the sender of information in this form of social language makes a special effort for its message to be clearly observable. Contrast this with subtle facial expressions during a game of poker. This social language can communicate a great deal, but it requires the receiver to be extremely discerning. Social languages are most successful and efficient when these two factors are matched, with the sender working just hard enough to be observed by the intended receivers.

The concept of observability greatly depends on the details of the actualization of the social language. Naturally, the domain in which a message is sent affects the mechanisms required for observation. For example a fish or robot may be sending messages through its position in space. This requires receivers to be able to discern accurately the important features of the position, i.e., relative position of two fish may be critically important but the absolute position of a fish may not matter at all. Additionally, the complexity of information to be communicated determines the degree of detail required from observation. In the fish scenario, the dominant social language is concerned with simply communicating a fish's position. The sending fish need only exist in physical space and the receiving fish must determine a relative distance to an accuracy of approximately its own width. Bees, on the other hand, wish to communicate the location, through distance and heading, and goodness of an object through their waggle dance. This requires the sender to encode several details into a lucid dance and the receiver to discern the details of heading, length of dance, and intensity of waggle. Note that the domain of the bee's social language is still the physical space that the bee occupies, but includes many more factors than in the case of the fish. The details of what a social language is attempting to convey and the method of conveying its message ultimately determine the requirements on the agents sending and receiving messages.

Note that the domain of communication, itself imposes certain challenges to the observability of a social language. Consider the bee performing a waggle dance, a form of communication used both inside a hive and on the surface of a swarm which has left a hive to start a new colony. When this dance is performed within a hive, there is no artificial lighting, so visual observation of the dance is unlikely. Rather the receiving bees must be in physical contact with the dancing bee or in range of the electric field generated by the dance [14]. As bees do not generate especially intense electric fields and electric fields decay as the range from the source increases, those that wish to receive the signal must be within some maximum range. Additionally, note that this decay over distance makes attaching information to the exact value of field intensity foolhardy. This is perhaps more clear in the case of radios. Radios would have a very difficult time communicating information by observing each other's absolute signal power levels, as the signal power level decays with distance. Instead, radios employ amplitude modulation based on relative power levels within a signal to communicate. Understanding how the transmission of messages varies through time and space is an important factor for social language, just as it is for any form of communication.

Finally, the concept of interference is as important for social language as it is for any communication scheme. A sender can not be considered lucid if it always overlaps its message with some other sender. In fact, such a sender damages the lucidity of all those with which it interferes. Again, as with any other communication this does not defeat the ability to transfer useful data; it simply requires more sophisticated techniques for successful communication [15]. Typically sophisticated techniques for managing communication through interference are not available for social language, as the concept of social language aims to deliver a low cost solution for transferring only pertinent coordination information. Therefore, it is typical to expect interference to result in total data loss. This inevitable loss is typically handled through reliance on the web of situational awareness reinforcing useful messages though the actions of others, e.g., bees that miss an original dancer's message may catch a repeated performance from a mutual peer. Depending on the methods used for implementation a social language, message fidelity could be quantified in a number of ways, just as any other communication method. While this question of how message fidelity and associated metrics affect social language is important, it is not addressed further here.

2.3.2 Understanding

Understanding a social language is the act of determining the useful information encoded within some observed message. Actions inherently encode some amount of data; this is the principle upon which all communication is built. Social language is no different; the difference lies in the data and where it is applied. The data being used in social language typically holds only a small amount of information and it is only relevant for a short period of time. Therefore, the encoding and decoding of the data needs to be as cheap to perform as possible. Additionally, social language is of use to decision making processes; thus the data can be occasionally be directly transferred without explicit encoding and decoding phases at all. The goals and attributes of a social language serve to differentiate it from a traditional general purpose communications mechanism; social language stresses society specialization and efficiency over general utility.

Since the object communicated by social languages is some form of coordination data, the most straightforward social languages are simply deliberate specialized actions. Receiving nodes need only determine why a peer took some action in order to gain the benefits of this communication. For example, a fish might observe a peer move closer to itself. In this situation the fish must examine available information to determine why the peer took this action; perhaps the peer encountered an obstacle or perhaps the peer wishes to mate. This principle holds in more complex social languages, as well. For example, bees need only to decode the reason behind the particulars of a peer's dance to determine the location of an object of interest. Determining the driving stimulus of the communicative actions of social language allows the receiving agents to response appropriately.

Note that the information transferred by social language is for internal consumption of the agents. While it is conceptually helpful to think of the agents as decoding a particular message by asking themselves why a peer took a particular action, this process is not always necessary. In the fish example, it is unlikely that one fish pauses to consider why one of its peers is darting toward it sharply. Rather the fish automatically reacts to the rapidity and direction of the movement by moving away from the peer who has encountered an obstacle. This reaction represents both the understanding of a message and the application of the information gained. Separating the process into encoding, decoding, and acting is really only useful for early examination of a particular social language, where the separation highlights the basis for an interaction and the expressibility of a particular message. In practice, social language is more concerned with encouraging appropriate reactions to behavior than the actual transfer of bits.

2.3.2.1 An Information Theoretic Perspective

The action based communication of social language is well suited to the coordination of CRs, as direct communication for coordination detracts from their primary purpose, i.e., the communication of application data. The society specialization and focus on efficiency of a social language make the cognitive engine (CE), the reasoning center of a CR, the ideal component to handle the encoding and decoding of actions. The CE already determines which actions a CR should take based upon the build in interaction methods. Adding the ability for the actions themselves to carry coordination information achieve the benefits of social language without detracting direct communication resources from the transfer of application data.

Exploring the information that is already encoded into the actions of a CR provides insight into how a social language may be designed for CRs. Since the actions of a CR are determined based upon some stimulus, the actions represent encodings of this stimulus. The action-determining process used by the CR is the process that encodes the stimulus information into the action. Examining the amount of stimulus information that actions contain following this encoding process provides insight into which action-determining processes are well suited for use in an action based social language. Information theory provides the tools necessary for this examination.

Recall that Mitola posited cognitive radios as a combination of artificially constructed intelligence and flexible radios to enable new uses of communications. Mitola borrowed the concept of an OODA loop from Boyd [16] to describe the operation of CRs. This loop is a cycle of observation, orientation, decision, and action, as shown in Figure 2.1. Radios operating in accordance with the OODA loop execute tasks associated with each stage of the cycle. First radios observe their environment, taking in data. Next radios orient themselves according to the information collected, which consists of comprehension of the data through determination of metrics and/or filtering. Once comprehension is carried out, a decision about what a radio should do, based on the understanding gained, is made. The radio than takes action based on its decision.

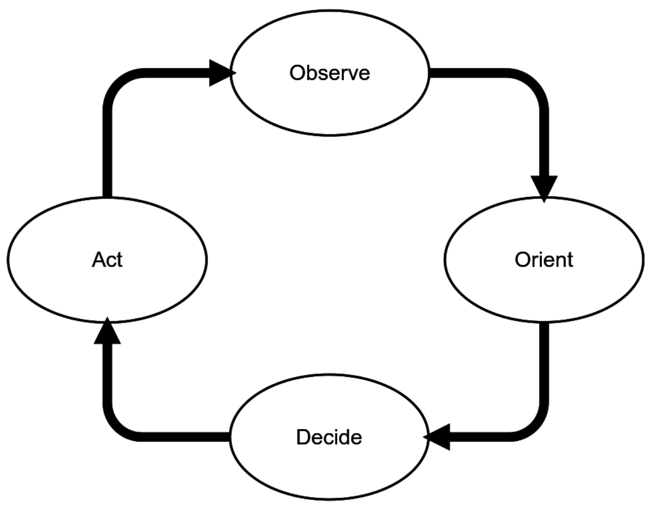


Figure 2.1: OODA Loop

Social language serves to augment the steps of the OODA loop. A social language provides radios with the ability to observe previously unobservable information about their peers. Once these observations are made, radios can orient themselves by decoding the information contained in the social language messages. This information can then be used to decide and act. For example, as a bee searches for food sources, the bee follows an OODA loop augmented with social language. The bee observes a waggle dance by one of its peers. The bee determines the goodness of the food source by comprehending the observed amount of waggle. The bee then decides whether or not to visit this source and acts on that decision. Thus social language is used to encode and transfer information that can add the operation of peers.

Figure 2.2 displays how the CE, pictured as the brain of a CR, encodes information into the actions that a CR takes. The CE determines what action to take based on two separate sources of information: environment information and built in information. The environment information is information that a CE gains from sensing the environment in some way. The built in information comes from the design of the CE. This information represents any factors that limit or control the operation of a CE. For example, CEs generally select an action from a limited pool of options; the bounds on this option pool constitutes information built into the CE by its designer. Both of the environmental information and the built in information play a role in determining the final action of the CR.

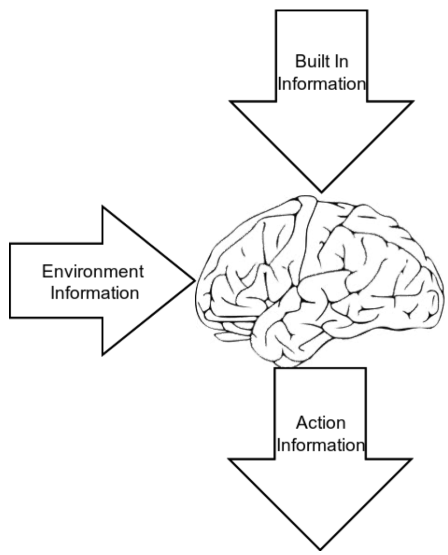


Figure 2.2: Encoding of Information Into Action

The most straightforward way of understanding the information contained in an action is to view the action as a random variable. The amount of information, in bits, that any random variable contains is the minimum number of bits that are required to specify the value of that random variable. For example a flipped coin can take on either heads or tails; expressing heads as 1 and tails as 0 allows the specification of the value of a coin with one bit, so the value of a flipped coin contains one bit of information. In the same way one can determine a bit based mapping of the actions that a CR might take in order to examine the information that an action contains.

Understanding the source of the information contained in an action, rather than simply the amount, is of primary importance to social language. Since the goal of designing social language for CRs is to communicate coordination information through the actions of the radios, understanding the source of the information contained in the actions provides insight into which CEs are well-suited for social language communications. Note that achieving this understanding requires examination of the roles of the information used by a CE in the selection of an action. Recall that, in the example of the coin, the number of available values determined the number of bits required to describe the value and therefore the amount of information in a coin flip. In the same way the number of options available for a given action determines the amount of information contained in that action. Since the pool of options for any action is determined by the built in information of a CE, this sort of information determines the amount of information an action may contain. However, the influence of built in information does not end there. Rather the process for selecting an action may depend on some internally generated information, such as an internal random number generator. In this case the value of the random number generator contributes to the selection of the action. Thus the built in information related to the number generator is in some way reflected in the action selected. In a similar manner environment information is typically used in some way to help select actions and is therefore represented in the selected action.

Given this understanding of how various sources of information contribute to the information in an action, a modification of the process may be made to allow for social language communications. Recall that the built in information is intrinsic to the selection process employed by the CE, both determining the amount of information contained in an action and contributing to the selection of the action directly. This information is tied to the operation of action selection. The environment information, on the other hand, represents the insertion of external information into the action determining process. This information can be anything that aids in the selection of a particular action. The built in information may be viewed as the information related to the encoding process and the environment information may be looked on as the message that is encoded. Thus if instead of a sensor reading, the environment information represents the object of social language communication, the proportion of environment information represented in the selected action indicates the proportion of social language information that action-determination method may communicate.

Given this model for the encoding of information into actions, Information theory provides the tools required to gain insight into social language design. Traditionally, the tool for examining the information content of some random variable is entropy. The entropy provides the number of bits that are required, on average, to express the value of the random variable. This measure, denoted as $H(x)$ where x is the random variable involved, is based on the probability of the variable assuming any particular value with the general result that the more random the variable, the more bits required to represent it. Assuming a discrete random variable (a necessary assumption for computer usage), the entropy is also a measure of the uniformity, i.e.; a variable with a uniform probability mass function (pmf) has the highest possible entropy. Additionally a maximum bound for the entropy of any given variable is easily obtained through the examination of a variable with the same possible values but a uniform pmf.

The common way to examine the informational relationship between variables is the mutual information. Mathematically, the mutual information between two signals, A and B, is given as:

$$I(A;B) = H(A) - H(A|B) \tag{2.1}$$

Conceptually, this is the number of bits required on average to describe signal A, minus the number bits that are still required to describe A once B is known. More simply, the mutual information is the number of bits of information that B contains about A. Note that the mutual information is symmetric, therefore the amount of information that A contains about B is the same as the amount of information that B contains about A [17]. This measure can be very powerful, but is not well suited for comparison purposes, as the magnitude is not bounded to any particular interval.

The uncertainty coefficient serves as a much better measure for comparing informational relationships and thus is the most important measure here. This measure normalizes the mutual information by the entropy of one of the two signals. Mathematically, this is written as:

$$U(A;B) = \frac{I(A;B)}{H(A)} \tag{2.2}$$

The uncertainty coefficient gives the portion of bits about A that may be predicted using knowledge about B. Conceptually, this provides the proportion of information in A that comes from B. This measure is limited to values between 0 and 1, providing a clear measure of the accuracy with which signal A (the reason for selecting a particular action) can be predicted given knowledge of signal B (the action in question). This measure therefore gives the capacity of an action to hold social language information.

Testing A simplified test scenario is used to examine the utility of social language based on the measures introduced above. This scenario examines the degree to which the information in an action comes from environment information rather than built in information. To enable this examination three cognitive radios have been created. Each CR chooses an action based solely on a measurement of the noise present in the environment. These actions are cataloged along with the noise measures that proceed them. Once the cataloging is concluded, the recorded actions and noise are used to calculate the measures of interest discussed above. The CR takes action by selecting a power level and modulation scheme. The noise is assumed to be purely additive white Gaussian noise (AWGN).

The goal of this testing is to examine the proportion of action information that comes from external information for various different CE techniques. Recall the exact nature of the external information is not important; rather, these tests examine the degree to which this information is represented after the encoding process that constitutes action selection. This testing will provide an indication of the CE methods that are best suited to encoding and decoding social language information. The degree to which external information is represented in the action measured in terms of the ability to predict the external information given the action, i.e., the uncertainty coefficient. Note that the tests conducted simply reveal trends for a selection of CE techniques and do not aim to cover the nuances of optimal CR strategies.

Each of the three CEs utilize a different technique to determine an action. The first CE, depicted in Figure 2.3, generates a random number which it then uses to select an action from a table of acceptable actions without considering the noise. The second CE, shown in Figure 2.4, uses the noise to select an action from a table. This CE follows a rule based or expert system strategy. The third CE, displayed in Figure 2.5, uses a genetic algorithm (GA) to select its action. The GA employed by this CE randomly generates a population of candidate actions and evolves them based on a fitness function. This fitness function calculates a weighted combination of the transmit power required, the estimated bit rate based on the sensed noise level, and the modulation index of each candidate solution. The highest fitness solutions from each generation are used to generate the next generation through the mechanisms of crossover, in which two parent candidates exchanges portions of the solution to form new solutions, and mutation, in which portions of candidate solutions are randomly changed. Once a certain number of generations have been run, the GA returns the highest fitness candidate solution as the action to be taken. This method of action determination uses the environment information for the creation of the fitness landscape on which candidate solutions are judged. All CEs choose actions from the same pool of options.

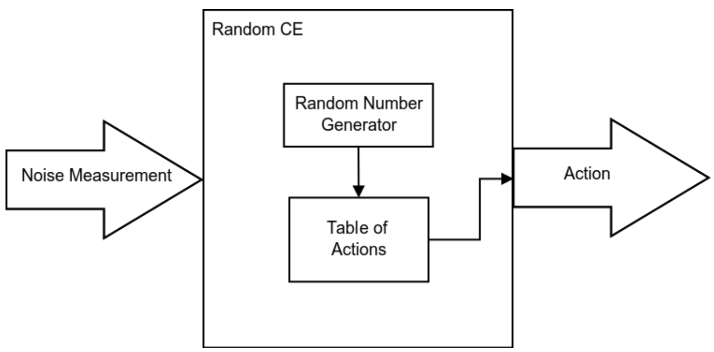


Figure 2.3: Random CE

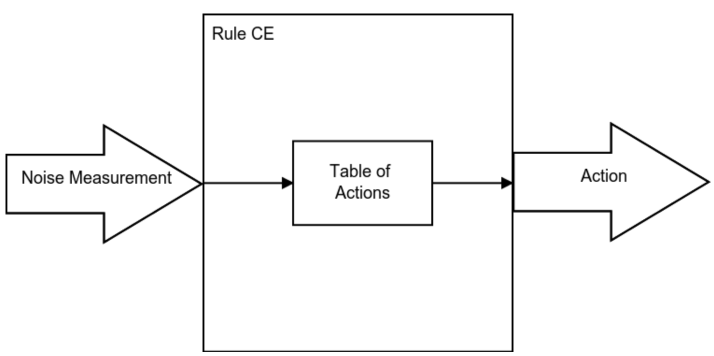


Figure 2.4: Rule Base CE

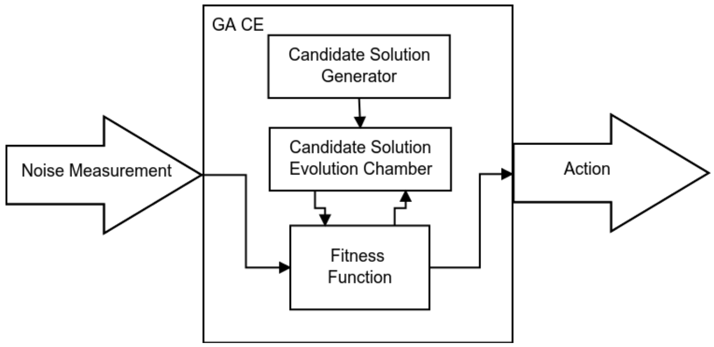


Figure 2.5: GA CE

Results The results of the testing provide a clear indication of the trends in the impact of an action-determining technique on the proportion of action information that reflects the external information. The various methods for selecting an action apply the environment information in different ways, which results in different degrees of representation in the selected action. The testing reveals how the various action selection methods' application of built in knowledge affects the connection between the selected action and the environment information.

The results of these tests are given in the tables below. The tables each relate results concerned with a particular type of action. Table 2.1 covers the total action, consisting of both the power setting and the modulation selection. Table 2.2 focuses on only the power setting portion of the action. Table 2.3 focuses on only the modulation selection portion of the action. Examining the total action and each of the sub-actions separately reveals aspects of the influence of environment information on the action that are otherwise obscured. Each table shows the total information contained in its action type (in the form of the action entropy), the mutual information between the action and the noise, and the uncertainty coefficient relating the action and the noise. Each of these figures are calculated from the recorded noise values and actions. In this testing the uncertainty coefficient is calculated by normalizing the mutual information between the action and the noise by the information contained in the action.

Table 2.1: Total Action Results under 100,000 Tests

CE Under Test	Action Entropy	Mutual Information	Uncertainty Coefficient
Random	2.708010	0.023096	0.008529
GA	1.610803	0.166126	0.103133
Look Up	2.085064	2.085064	1.000000

Table 2.2: Power Sub-Action Results under 100,000 Tests

CE Under Test	Power Entropy	Mutual Information	Uncertainty Coefficient
Random	1.609423	0.006733	0.004183
GA	0.565302	0.117135	0.207208
Look Up	1.099828	1.099828	1.000000

Table 2.3: Modulation Sub-Action Results under 100,000 Tests

CE Under Test	Modulation Entropy	Mutual Information	Uncertainty Coefficient
Random	1.098610	0.003165	0.002881
GA	1.046020	0.031977	0.030570
Look Up	1.097648	1.097648	1.000000

Recall that the uncertainty coefficient provides the proportion of information in the action that comes from the environment information, given knowledge of available action options. Additionally, recall that the information in the action represents both the built information used when selecting the action and the environment information. This concept is made clear in the results. In the case of the random CE the environment information was not used at all in the determination of the action and therefore not represented in the selected action (as indicated by a very low uncertainty coefficient). Contrast this to the rule based CE, in which the action is completely determined by the environment information. In this case, the environment information is the only information represented in the selected action; no built in information was used in selecting an action. The built in information of the rule based CE influences the action selection process in a different way. Note that the action entropies for the rule based CE and the random CE are not the same, even though both use the same pool of options for actions. This results from the difference in the probability of obtaining certain actions. The random CE has a uniform probably of any given action, resulting in the highest possible entropy for the action pool used. The rule based CE, on the other hand, selects its actions as a result of the sensed noise, which follows a non-uniform Gaussian probability distribution. This results in some actions being less likely than others and a lower action entropy. Thus while the uncertainty coefficient reflects the fact that the rule based CE selects an action based on the environment information, the difference in action entropy reveals details about the built in information used in selection.

The GA CE provides an interesting case in between the other two CE extremes. This CE clearly reflects both environment information and built in information directly in the selection of an action. Note that this does not imply that that either the environment information or the built in information is not fully encoded into the action, but rather that the other kind of information is represented as well. Recall that the GA randomly generates and evolves a collection of candidate solutions based on a fitness landscape that involves the environment information as well as other information. Thus the selected action is the product of the random number generation, the random evolutions, and the other calculated attributes in addition to the calculation that involved the environment information. All of these additional built in sources of information contribute to the final selection of the action, as reflected in a low uncertainty coefficient. Note that this in no way precludes the decoding of the environment information from the action; rather it implies that additional information is required for accurate decoding.

A low uncertainty coefficient signals the need for more complex social language implementations due to the need for additional decoding information. The variety in social language complexity based on the need for additional information is clear when considering ants and humans, both of which utilize social language to underpin their societies. Ants are less intelligent than humans; they tend to use comparatively direct methods for determining what to do. The ant action based social language therefore relies on an action selection method where the message of the language completely determines the action taken. Decoding their social language is therefore very simple; one must just test for the intensity of particular chemicals. Humans, on the other hand, are highly intelligent and tend to consider several sources of information to make any given decision. Human action based social languages, body language for example, involve actions that are selected based on several sources of information, e.g., comfort level as well as message to be sent. Thus, decoding human social language requires a great deal more knowledge; a person tends to understand the body language of a person who they are familiar with but won't necessarily pick up the nuances of a stranger's message. As such human beings have to rely on models of their peers to decode the complex mappings [18], resulting in a more complex system for social language.

Note that the proportion of action information representing external information may not be the same for all actions, despite a common action determining method. Examining the uncertainty coefficient for the individualized actions reveals a telling point. Note that the coefficient is much larger for the power sub action than for the modulation sub action. This suggests that this particular GA more strongly relies on the environment information to select a power level than the modulation setting. This situation can result from the specific implementation of the algorithm, such as a bias in altering power more than modulation during evolution, or the fitness functions used for the algorithm. This suggests that decoding the environment information from the power of the selected action requires less additional information than decoding the same information from the modulation. Note that the lower uncertainty coefficient of the total action versus the power sub-action does not imply that decoding environment information is harder simply because the modulation is known. Rather, it suggests that the environment information is reflected in a smaller proportion of the action in the case of the total action than in the case of the power sub-action.

The results presented here provide a number of suggestions for the design of an action-based social languages for CRs. Chief among these is the notion that rule based determination of actions to be used for social language will require the smallest amount of additional information for decoding. Rule based social languages simply require all members of a society to have the same rule set to decode a particular action, whereas a social language based on a GA would require both a common knowledge of the fitness function to be used as well as communication of the states of internal random number generators. Additionally, the results suggest that in non-rule based social languages, some parts of the message could require more information to decode than others. This suggests that multi-layered social languages, where certain members of a society receive more detailed information based on their greater knowledge, are possible.

The information theory analysis conducted has provided insight into the sort of decision process that leads to the best understanding of the reasons behind a radio's behavior. Determining these reasons allows for communication through action-based social languages. It is clear that rule-based methods, with mutually exclusive rules, provide the most direct connection between a radio's actions and its reasons. The behavior of radios controlled with rule-based methods point unequivocally to particular input stimuli. While this method does not necessarily allow for behavior optimization at the individual radio level, it does provide the most straightforward approach for an initial development of a CR society. As is discussed below, employing social language allows for learning on the society-wide or network level.

2.4 Societal Learning

Societies of individuals with connected awareness exhibit the ability to learn as a group. In this societal learning, the collective transforms itself in a manner that stores the information to be learned. The state of the society as whole represents the storage of information learned through interacting with the environment. In this way the entire society learns as though it was a single organism.

Such super-organism learning is actually fairly common. Seeley describes the concept as it applies to swarms of bees which are seeking a new nesting site:

> ...there is no central decider who posses synoptic knowledge or exceptional intelligence and directs everyone else to the best course of action. Instead in both swarms and brains, the decision-making process is broadly diffused among an ensemble of relatively simple information-processing units, each of which possesses only a tiny fraction of the total pool of information used to make a collective judgment. ...These similarities point to general principles for building a sophisticated cognitive unit out of far simpler parts. [11]

Social language provides the connectivity necessary for the process of societal learning. As the society interacts with its surroundings, the strength of the connection between assorted nodes varies based on their communications through social language. For example, bees in a swarm looking for a new home interact with peers through the waggle dance. This interaction influences other bees' decision making process, which then results in bees either agreeing with the initial bee and reinforcing that bee's declarations or disagreeing and offering alternatives. In this way the language of the waggle dance provides the conduit that connects the individual reasoning elements of the bees. The bees use this conduit to reinforce or subdue the firings of other bees in a manner analogous to the action of neurons in a brain.

The connections between the individuals in a swarm result in a transformation of the whole system. Bees travel out from the swarm and bring back information about the goodness of a potential site. Their campaigning draws other bees to independently assess the site. The result of this is an emergent learning of appropriate nesting sites at the swarm level. No individual bee tallies up the number of its peers that vote for a particular site, rather the voting process simply influences bees to investigate individually. Determination of the swarm's overall assessment of a site must be made on the swarm level by determining the proportion of bees interested in a particular location. In this way, the number of bees that a swarm devotes to investigating a location stores information about the swarm's evaluation of that site. Once a critical mass of bees displays great interest in a new nesting site, the swarm, as a whole, has determined that the new site would make an appropriate home. Selection of a new site signals that the swarm has learned the local landscape of potential homes and made a decision by weighing its options before choosing the one best suited to its needs.

Artificial neural networks provide a technique for learning based entirely on varying the strength of influence between neurons that each determine their own action through the consideration of all connected peers [19]. Societies perform the same process in a slightly more sophisticated manner. The interactions between members of a society is not typically as simple as the accumulation and repeating of signals, as is usually the case in artificial neural networks. Rather the interaction between peers in a social group is more dependent on the details of the interaction and the society. In fact, the nature of these interactions makes societies extremely complex to model. However, there is no denying that the fundamental principles that underpin artificial neural networks are exhibited by societies.

Societal learning, just as social language, is focused on advancing the state of the society, not on the explicit storage and retrieval of arbitrary information. Much like social language the processes of societal learning are simply the actions and interactions of the society members. Each member influences the actions of those around it, until the society learns how best to interact with its environment. During this learning phase, the society as a whole probes its environment to discover the pertinent information it desires and stores this information in the interactions of its members. For example bees may discover a new nesting site not fit for their needs. Rather than having to continually re-examine this site, the bees collectively remember its unfit qualities through the interactions of bees directing new searches elsewhere.

The society must be taken as a whole to witness societal learning. Examining the action of a single member, bee or radio, does not reveal what the society knows. Rather it is through the examination of how the group influences its members that the learning is revealed. As bees discover prime nesting sites, the group influences several members to examine these sites, revealing that the society has learned both the location of a good site and the local fitness landscape as it pertains to nesting sites. The bees make their final observation by optimization over the nesting fitness landscape of the terrain in their range. To do this they must first learn this landscape; no outside force provides it to them. When the whole group makes the decision about a new nesting site, it is clear that that the society has learned the necessary information about its local fitness landscape.

2.5 Conclusion

This chapter has developed a social approach that provides CRs with much needed means for efficient coordination and societal learning. The concept of social language, as developed in Section 2.2, provides an efficient means of coordination for the implementation of CR networks. The components of social language, as examined in Section 2.3, highlight the major factors that must be considered when implementing social languages for CRs. Finally, the societal learning abilities, explored in Section 2.4, enable a truly distributed intelligence for CR networks.

Chapter 3

The Robotics Approach to Building Radio Societies

3.1 Introduction

The work done in cooperative mobile robotics provides several tools that have yet to be applied to field of cognitive radio. These tools offer a framework for developing multi-agent systems, insights into common issues for such systems, and methods for their evaluation. This chapter provides an overview of the most pertinent work from the large body of literature for cooperative mobile robotics in Section 3.2. It then goes on to present an examination of the challenges of applying robotic approaches to the problem of developing CR societies in Section 3.3. Finally conclusions are discussed in Section 3.4.

3.2 Robotic Societies

Roboticists, specifically those working in the sub-field of cooperative robotics, have been constructing collections of artificial agents for several years. In this time they have developed several methods for constructing societies that work together to accomplish some goal. While these techniques are focused on solving problems in the robot domain of operation, they offer significant insight into the challenges and approaches of developing artificial societies. Note that the robotics research does not concern itself with the development of social languages; this is because robotic societies are not primarily focused on the creation of communication structures. Nevertheless, the lessons learned from this field can greatly aid the production of CR societies.

Over the work done in the field of robotics, behavior-based approaches are most significant to the development of CR societies. These approaches employ the simultaneous use of several goal oriented behaviors [20], making them well-suited to handle multi-objective problems, such as those commonly faced by CRs. Additionally, there has been a fair amount of work on developing a general approach to implementing behavior-based systems that extends the approach beyond the domain of robotics. These systems provide a strong foundation for artificial CR societies.

There is a large body of literature concerning the development of cooperative multi-robot systems. The majority of this literature falls into a field often referred to as cooperative mobile robotics. Commonly cited reasons for interest in multi-robot system include the reduced costs associated with using several cheap robots instead of a single complex one, the need for additional entities to accomplish a task, and increased performance [21, 22]. The field of cooperative mobile robotics, traces its linage back to the construction of some of the first autonomous robots [23] and remains active today. Much of the work that concerns the development of artificial CR societies was conducted in the period of 1986 to 1995. This period saw an explosion of interest in the field and development of several of the strategies most applicable to the problems faced by cognitive radios

Cao et. al., likely provide the most in depth survey of cooperative mobile robotics for the period of interest [21]. This survey makes the distinction between collective behavior, behavior that occurs whenever more than one agent is present, and cooperative behavior, that requires the agents to actually cooperate, which is defined as increases in the total system utility through some "mechanism of cooperation." The authors go on to identify the five major research focus areas for cooperative mobile robotics as: group architecture, resource conflict resolution, cooperation origin, learning mechanisms, and geometric problems. Group architecture concerns techniques for providing the infrastructure upon which system behavior may be implemented, which is the area most useful for CR societies. Resource conflict resolution focuses on providing methods to access the same physical resources. Researching cooperation origins explores methods for bringing about cooperation without explicitly building it into the system. The area of learning examines methods to leverage artificial learning techniques in multi-robot systems. Finally, geometric problems are those that relate to the issues faced by several robots attempting to work in the same two- or three-dimensional world.

The area focusing on group architecture is the most applicable to CR problems because it is the most general area of research in the field. Providing the infrastructure for system behavior addresses several questions that are common to both robotics and radio research. These include designing centralized or decentralized architectures, enabling cooperation between hetero- and homogeneous agents, determining methods for inter-agent communication, and creating methods for agents to model each other. Decentralized architectures are widely recognized as having several general advantages over centralized architectures [24–26]. The question of heterogeneous versus homogeneous mixes of agents is not of particular interest in this work and so is not discussed further here. Agent communication and modeling provide a variety of insights that are further discussed below.

While communication in multi-robot systems does not apply the concept of social language, there are several lessons available. The notions that underpin the concepts developed in the previous chapter are echoed in in the work of cooperative robotics. Consider the following quote from [27].

> Communication is often considered a deliberate act, but state communication is not necessarily "intentional" since information can be relayed by passive observation. The sender does not necessarily explicitly broadcast its state, but allows others to observe it. In nature this type of communication is demonstrated when animal changes its posture or external appearance, such as a dog raising its hackles or exhibiting flight behavior in response to fear.

Note that in this quotation Balch and Arkin use the term broadcast to mean the generation, encoding, and transmission of a message through traditional communication mechanisms. Following the discussion of the prior chapter, every example they gave is the broadcast of a social language message. Balch and Arkin are referencing the technique of encoding information into actions without fully exploring the approach. Rather they simply apply the approach commonly found in nature as an effective means of agent coordination.

Although robotic systems do not define communications in the terms of social language, the concept frequently arises. A sign board approach is used in [28] and a diffusion based protocol is applied in [29]. The benefits of efficient communication are made clear throughout the field of cooperative robotics. In fact, Ray presents testing that suggests any communication greatly enhances cooperation of agents, although efficient communication certainly comes at a lower cost [30]. Finally, it is worth noting that while several interesting approaches for multi-robot communications exist [31–34], the specific goals of CR require a different focus for implementation.

Agent modeling is a multifaceted topic, however the impact on inter-robot cooperation is clear. Agent modeling aligns with communication to provide more effective cooperation, typically through more effective communication [21]. Note that modeling goes beyond the observation based communication to provide a representation that allows for one agent to make inferences about another. Robot methods of modeling tend to utilize several domain specific sensors and methods that are beyond the interest of this work.

Of primary interesting to the work, however, is the method by which coordination is enabled on top of the group architecture. While robotics provides several methods for accomplishing this, the behavior-based approach is the best fit for the requirements of CR. This approach of employing several goal oriented behaviors simultaneously was first introduced by Brooks in [35]. The behavior-based approach to implementing artificial systems more closely resembles the natural brain's ability to juggle several objectives than most other artificial intelligence approaches [36]. As such, the approach matches the multi-objective needs of a CR [37].

While Brooks introduced the behavior-based approach for multi-robot systems, his student

Mataric expanded the approach to its fully utility. Over a series of works [24,38–40] Mataric fleshed out the behavior-based approach in order to solve the "problem of distributing a task over a collection of homogeneous mobile robots." The core of the methodology is the use and combination of so called "basis behaviors." These behavioral primitives represent a minimal set of control laws that allow for the "structuring, synthesizing, and analyzing" of a system's overall behavior [24]. A set of basic behaviors is defined to be both necessary, in that all behaviors in the set are necessary for system operation, and sufficient, in that no behavior in the set is reproduced either by any other behavior in use or through the combination of other behaviors in the set. Mataric uses this basic behavior concept to build up systems in which cooperation is an emergent property.

Specifically, Mataric's work concerns the operation of several robots in a two-dimensional plane [24]. The robots rely on a basic behavior set consisting of safe-wandering, following, dispersion, aggregation, and homing in order to accomplish tasks such as flocking and foraging. Safe-wandering allows robots to move through the environment while avoiding obstacles. Following enables one agent to move behind another. Dispersion spreads the agents throughout a space and aggregation groups them. Homing allows the robots to move toward some specified target or direction. Flocking requires robots to stay within a specified distance from each other while moving toward a goal. Foraging tasks robots with collectively searching a space with obstacles in order to find and retrieve some target.

The goals of flocking and foraging provide examples for Mataric's two methods of basic behaviors [24]. Since flocking requires maintaining a distance from peers and moving toward a target while avoiding obstacles, this goal can be achieved through a combination of the dispersion, aggregation, homing, and safe-wandering basic behaviors. Since each basic behavior is formulated such that it outputs a new velocity vector, the combination is simply a weighted addition of the outputs of each basic behavior, where the weights specify the relative influence of each basic behavior. Foraging, on the hand, is slightly more complex. This goal requires robots to search through a space interspersed with obstacles for a cache of targets. Once found the targets are to be returned to some home location. Completing this task requires the robots to safe-wander through space, avoid peers that have yet to find the cache, follow peers that have found the cache, and move toward targets. However, unlike flocking, a robot completing the forage task doesn't apply all its behaviors simultaneously; rather it employs a single behavior appropriate to its current phase of operation. The robots change behaviors based on external stimuli, e.g., whether or not they have found a target.

Note that the two methods of behavior combination are not mutually exclusive. Rather they represent the two mechanisms for combining behaviors, either simultaneously or in phases. Additionally, the combination mechanisms are not limited to basic behaviors; compound behaviors may be combined using the same mechanisms. In this way Mataric has developed a behavioral algebra that allows for the construction of a myriad of complex behaviors from a single basic set.

While there is no universal method for selecting basic behaviors for a given task, Mataric offers several criteria for selection [24]. Specifically Mataric presents seven criteria for selecting behaviors for decentralized multi-agent systems: simplicity, locality, correctness, stability, repeatability, robustness, and scalability. As the behaviors develop a system wide emergence when employed on several agents, they should be as simple as possible to avoid unwanted emergence. To maintain decentralization, the behaviors must be based on locally available information. Behaviors should stably produce the desired outcome in a repeatable fashion without requiring optimal performance of hardware. Finally, behaviors should scale as the system increases in size.

Mataric also provides a discussion on the evaluation of multi-agent systems. Mataric notes that the interactions present in the system — between agents and other agents or agents and the environment — depend on a plethora of implementation level details [41]. Additionally, the state of the any given agent depends on the state of all other agents in any system that allows for any form of interaction. Such inter-connectivity makes system level models intractable even for simple systems. As such the only option for system evaluation is the observation of the system performing as desired. To accomplish this in behavior-based systems, confirmation of basic behaviors as well as their combinations proves the systems' operation. This bias toward empirical confirmation through implementation echos the attitude of many researchers who build multi-robot systems [21, 26, 27, 30, 31, 42].

Note that while Mataric makes a large contribute to multi-agent system design, her work is not a complete solution. Specifically, Mataric does not directly address questions of transferring information between peers. That is, several of the behaviors or combinations require knowledge from a peer, e.g., whether the peer is holding a target item or the current motion of the peer. While these elements could be determined through the use of advanced sensors, Mataric uses low bit rate radio transmissions to communicate this information between peers. Additionally, note that this information could be shared through the use of a social language, employing less sophisticated sensors and understanding of a peer's actions to determine the necessary information.

Roboticists have also done a great of deal of research on the topic of social intelligence. However, this term is not applied in the same manner as it used in this work. Dautenhahn provides a good definition of the concept of social intelligence as it applies to robotics in [43]. The potential for the use of robots as artificial assistants to human beings makes the interaction between artificial agents and humans an active area of research. The methods of interaction employed by the artificial agents represent their social intelligence. There is a large body of literature on the topic, with [44] and [45] providing in depth surveys. While this work is certainly of great importance to any artificial agent that is designed to interact with humans, this topic lies beyond the scope of the work of this dissertation.

3.3 Robotics for Radios

As discussed in the previous section, the field of cooperative mobile robotics has a great deal to offer CRs. Specifically, the behavior-based approach for multi-robot systems offers a method for implementing multi-objective oriented complex behaviors in a network of CRs. The approaches provided by Mataric in particular offer an excellent foundation for CR systems.

There are number of strong parallels between the radio and robot domains that make the application of robotics work appealing. As discussed, both domains must often face tasks that feature multiple objectives. Additionally, both CRs and robots must navigate spaces interspersed with obstacles. Robots navigate physical space that features physical obstacles, whereas CR navigate through time and frequency with obstacles in the form of interferers. On some the most fundamental levels the tasks of robots and CRs are quite similar.

On the other hand, the differences between the robot and radio domains make a direct application of robot techniques unfeasible. These differences are summarized in Table 3.1. Robots tend to move through their space as a byproduct of their goal, e.g., foraging robots are attempting to move targets from one location to another. Even goals that are formation based exist primarily to serve another purpose, e.g., research into formation control has a broader ultimate goal. Contrast this to radios that exist in their space as an end in and of itself. This concept rests on the question of what it means for a radio to exist in a portion of spectrum. Note that while a radio is not communicating information it does not occupy any spectrum and that while it is communicating the amount of spectrum occupied is directly related to the rate of communication. Ideally radios would not move at all in their space, they would simply exist there, communicating data. This difference of purpose leads to an even more important difference between the robot and radio domains, the concept of observability. A robot is observable in the physical domain by default; other robots may sense that robot without special effort on the part of the robot being sensed. A radio on the other hand is invisible to peers, or at least very difficult to observe, unless transmitting. This results in several problems that exist for CRs but not robots. Finally, it is worth noting the persistent observability of a robot comes at the cost of making continuous paths through its space. Radios have no such restriction in time-frequency space; a radio can "teleport" from one frequency to another without being present in any frequency in between. These differences in focus, observability and fundamental challenges mean the specific problems and solutions of one domain do not carry over to the other.

Table 3.1: Differences between Robot and Radio Domains

	Radio Domain	Robot Domain
Purpose of Occupying Space	End goal of operation	Means to an end
Observability	Only during transmissions	Always
Motion Restrictions	Discontinuous motion allowed	Only continuous motion allowed

In so far that CRs and robots both represent intelligent agents, with common classes of problems, general approaches may serve both domains. However, the specific differences between each domain requires the tailoring of any approach to the domain in which it is implemented. Thus the parallels between the challenges faced by robots and radios make a behavior-based approach to building multi-agent systems in either domain appealing, but the differences between the same challenges requires such an approach to be fitted to each domain individually.

3.4 Conclusion

Cooperative mobile robotics offers the proven tools for the structuring, synthesizing, and analyzing of multi-agent systems. While the work does not name them as such, these systems certainly qualify as artificial societies, as discussed in Chapter 2. The behavior-based approach for multi-robot systems, in particular, provides techniques of great use to the construction of CR societies. The work done in robotics combined with the understanding of societies, social language, and societal learning presented previous constitute the toolbox needed to construct successful CR societies.

Note that the use of the tools presented here requires an understanding of both their capabilities and the goals for their use. As discussed in Section 3.3, there are several domain differences that represent potential pitfalls in the application of robotic techniques to the world of radio. As such, careful handling is require for a successful utilization of robotic approaches for radio radio problems. The following chapters present methods to avoid pitfalls and successfully apply the tools described here to the building of CR societies.

Chapter 4

Medium Access Control for a Society of Cognitive Radios

4.1 Introduction

This work addresses the problem of creating a decentralized self-organizing network that dynamically adapts to its environment without disturbing any pre-existing networks. In pursuit of this goal the concept of a CRs society — a group of CRs that fosters coordination through efficient communication based on the encoding of information into actions — has been defined and discussed in Chapter 2. Additionally, tools from cooperative mobile robotics for constructing such a society using the simultaneous operation of several goal driven behaviors and evaluation techniques based on the implementation and observation of systems have been examined in Chapter 3. This chapter examines available tools from the field of cognitive radio and discusses the radio-specific focus of the work.

4.2 Importance of the MAC Layer

It is worth noting that the problem at hand is not an optimization problem. Rather, the problem centers on the issue of allowing several radios to access the spectrum cooperatively without restricting the dynamic properties of the system through an over emphasis on information sharing. Since the role of the medium access control (MAC) layer in a radio system "is to coordinate the access to the channel so that information gets through from a source to a destination," [46] this layer is the obvious choice for the work of this dissertation.

The initial purpose of abstracting the functions of a radio into layers was to allow for the independent development of the myriad abilities require to run a communication network [47]. Such independence allows for the parallel development of all aspects of a radio network. Thus, it is the opinion of the author that understanding the role of an approach in terms of the existing radio stack has the significant advantage of allowing the joint application of several proven techniques to problems.

In this vein, it worth noting that a great deal of work on the topic of physical layer optimization for CR has already been conducted. Notable examples of this work include [37,48]. Applying radio layer abstraction to such work and the techniques for coordinating CR societies developed here allows for the coexistence of both approaches. Specifically, the GA technique that Rondeau demonstrated can optimize physical layer parameters while the mechanisms for CR societies determine appropriate times and frequencies for the use of such parameters. Note that such a coexistence would certainly require the sharing of information between the entities; however, such sharing does not indicate a blending of responsibilities. Information sharing between layers in the cognitive stack is akin to the passing of data between layers in a traditional stack; given an appropriate application programming interface (API), the abstraction of functions is not disturbed.

In order to maintain the abstraction of responsibility in the radio stack, along with the benefits discussed above, the work of coordinating a CR focuses on the responsibilities of the MAC. As such, the work provides mechanisms for CRs to determine appropriate strategies for accessing spectrum in a dynamic, decentralized manner that avoids interfering with other radio systems. This focus allows the goals of an CR society to be stated as follows: determine an appropriate time for the transmission of application data and determine an appropriate frequency for this transmission. Note that the term appropriate is used to imply that the transmission of application should not interfere with any other transmissions, whether they are from members of the same society or not, and that data should be transmitted in a manner that allows for it to eventually reach its destination, i.e., transmissions on channels without receivers are not beneficial. Additionally, note that the system must allow for transmission by several other agents, i.e., it must be cooperative and scalable.

4.3 Overview of Related Work

Cognitive radio has been an extremely active field since the introduction of the concept in 2000 [1]. As such it has generated a very large body of literature. In this wealth of references, there are several systems that apply bio-inspired techniques to CR, e.g., [49–55]. Within this group there are several examples of applying concepts from natural societies to CR resource allocation, e.g., [51, 53, 55]. The authors of [51] present a purely mathematical examination of applying flocking mechanisms to the problem of organizing nodes in frequency and time. A simulation based study of swarm based methods for spectrum auctioning is presented [55]. A method for time synchronization to add MAC protocols is studied through simulation in [53]. Of these, [53] comes the closest to enabling a dynamic self-organizing network of CRs. However, this work relies on the creation of local centralization through synchronization to a so called "master node."

There are several noteworthy examples of systems that create contentious cooperation within networks and with radios outside of the network. Game theory provides the most common method of achieving this form of cooperation. A survey of such works is provided in [56] and notable approaches including [57] and [58]. These works offer impressive methods for systems that foster cooperation in which CRs need only consider directly pursuing their own goals. Game theory implicitly requires a level of rationality for its operation. Mataric warns that in situated multi-agents rationality may not always be available, "due to incomplete or nonexistent world models and models of other agents, inconsistent reinforcement, noise, and uncertainty" [41]. While there are methods for addressing these concerns, the work presented focuses on building consensual cooperation into societies explicitly.

Interestingly, despite the large body of literature there are relatively few examples of MAC focused CR techniques. Of the examples that exist, several fall victim to limitations of over sharing information and/or centralization. For example, the authors of [59] discount decentralized architectures due to the increased need for messaging. The authors of [60] claim that time-slotted architectures for decentralized systems are unfeasible due to the need for time synchronization. There are few examples of CR MAC protocols work toward the benefits of a dynamically self-organizing decentralized network. Several of these use approaches that rely on capabilities that are not typically available to situated CRs. For example, both [61] and [62] employ techniques based on variations of the Markov decision model which require CRs to have knowledge of the probabilities with which channels will be occupied. The authors of [62], on the other hand, make the common assumption that a clear control channel be available for coordinating CRs. Additionally, there several examples that do exist do not fully embrace the concept of a dynamic self-organizing decentralized network. For example, the authors of [63] present a dynamic MAC switching protocol that allows for the coordination of radios through the use of a global control plane of communication but does not allow for local decision making. The work presented in this dissertation overcomes most or all of the limitations that plague other approaches.

The body of literature does provide some tools and insight, though. Chief among these is likely the pivotal work of Doerr, Sicker, and Grunwald in [5]. This work provides one of the earlier examinations of applying the lessons of natural societies, specifically that of fish, to the problems of CR. It presents a CR network that does not require any central authority or peer communication and in the course of doing so offers an example of what natural societies have to offer the field. However, this approach has not been implemented on real radios. As such, the approach does not account for realistic constraints on sensor and computational performance.

A promising tool, initially presented for wireless sensor networks, is the DESYNC algorithm first published in [64]. This algorithm, inspired by the actions of fireflies, provides means of spreading radios within a time period. The algorithm breaks time into periods of a specified length. Each radio keeps track of time independently from all other radios, starting its own clock when it first comes online. The algorithm is concerned specifically with the progress of a radio through periods, which is referred to as the radio's phase. For example, if the radio is halfway through a period at time t, it is referred to as having a phase of 0.5 at that time. When the phase of the radio reaches 1 the radio fires a beacon and resets its phase to 0. Each radio observes its local phase when it receives a beacon from a peer. The goal of the algorithm is to spread the firings all radios such that the maximum possible time exists between the firing of every node.

The DESYNC algorithm is best understood as the arranging of beads on a ring. The time period in which nodes fire beacons segments time and can be envisioned as a ring. The phase of a radio's firing can then be represented as the position of a bead along this ring, with the 12 o'clock position representing both a phase of 1 and a phase of 0. Radios move along the ring in a clockwise direction, with their position corresponding to their phase at a particular moment in time. Figure 4.1 shows such a visualization of the initial time configuration of radios A, B, C, and D, just as radio A fires. Figure 4.2 shows a representation of the same collection of radios with an optimal configuration of firings, again just as radio A fires.

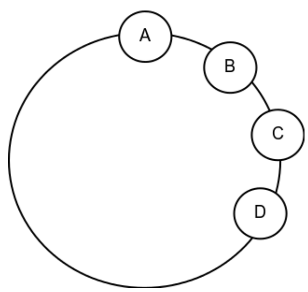


Figure 4.1: Initial Phase Spacing

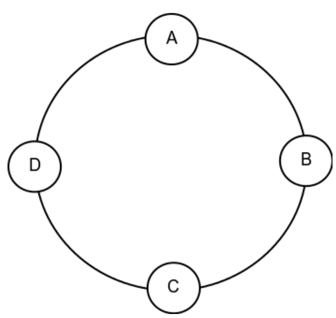


Figure 4.2: Final Phase Spacing

To achieve its goals, the DESYNC algorithm employs a method of updating the firing times based on the local observations of the radios. Specifically each node updates its firing time to be at the midpoint of its closest phase neighbors, i.e., the radios that fire directly before and after the radio contribute to updating its firing time. Consider the initial phase configuration shown in Figure 4.1. Radio D has phase neighbors radio C, its prior phase neighbor, and radio A, its next phase neighbor. The longer delay between radio D's own firing and that of radio A than between radio C's firing and radio D's firing would indicate that radio D should wait longer than one period's length to fire its next beacon. By waiting longer than one period to fire, radio D shifts its phase closer to radio A's position and farther from radio C's. Radios may move clockwise on the phase ring by waiting longer than a period to fire, move counter-clockwise by waiting less than a period to fire, and remain stationary by waiting one period to fire next. Figure 4.3 shows a visualization of the move that radio D would take from the initial position shown in Figure 4.1. Proofs to show that such moves will allow the system to converge from Figure 4.1 to Figure 4.2 are given in [64–66].

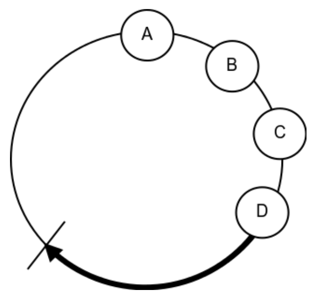


Figure 4.3: Radio D's Move

The authors of [64] go on to state that DESYNC can serve as the underpinning of time division multiple access (TDMA) based MAC protocol as well. The starting phase of the slots corresponds to the midpoint between the radio's firing and the firing immediately before its own and the ending corresponds to the midpoint between the radio's firing and the firing immediately following. Organized in this manner, the slots do not overlap, converge towards a fair distribution among all active radios, and allow for communication before reaching convergence.

There are several papers that discuss the DESYNC algorithm [64–68]. Each of these confirms DESYNCs operation theoretically for single-hop networks. Extensions to the algorithm for multi-hop networks are presented in [67] and [68]. The algorithm was implemented in a centralized architecture in [64], with several peer radios and a base station. The peer radios desynchronized themselves in order to communicate with the base station using TDMA. This specific algorithm has not been applied to CR or used for peer-to-peer communication in a completely decentralized network.

The DESYNC algorithm has several features that make it well suited for use in a CR society. The algorithm's focus on local observation facilitates decentralized use. However, as presented in [64], DESYNC is not designed for use as a behavior in a larger system. Its the complexity can lead to undesired interactions. Thus, for the purposes of this work simplifications are necessary. Supporting aspects of the algorithm, such as the entry procedure presented in [64], were not considered for this work. I have adopted only those parts of [64] that allow for the spreading of radios in a period and the establishment of slots. These components of the DESYNC algorithm are integrated with other behaviors in the full systems.

To the best of the author's knowledge, several of the approaches presented in this work are completely novel to the field of cognitive radio. The concept of social language allowing for the encoding of information into actions represents a new technique for the coordination of CRs. The lessons about multi-agent system design from cooperative robotics have not yet been applied to CR problems. These approaches represent new methods for tackling the problems of CR.

4.4 Conclusion

The chapter has examined the radio-specific focus of this work, presented context from the current body of literature, and collected CR related tools. Section 4.2 explains how this work best fits into a radio network at the MAC layer and restates the goals in terms of MAC protocol responsibilities. Focusing on providing a MAC layer solution to the problem of coordinating a decentralized network allows for the use of other proven approaches in handling the other problems of a radio network. Section 4.3 provides context for this work from CR literature and explains radio specific tools for building CR societies. This work offers several novel benefits and methods to the field of cognitive radio.

Chapter 5

Building CR Societies

5.1 Introduction

So far, we have examined several tools to aid in the construction of societies of CRs. The examination of the construction of artificial societies presented in Chapter 2 indicated the factors that allow for efficient intra-society communication through a social language. The exploration of robotics techniques for developing multi-agent systems in Chapter 3 provided the behavior-based method of handling multiple objectives. The review of relevant work in CRs discussed in Chapter 4 offered the foundation for decentralized coordination of radios in time.

This chapter discusses how all of these tools may be combined to enable the development of CR societies. The social language provides an efficient mechanism for radios to reveal pertinent information without over sharing and detracting from the goal of application data transfer. This concept is discussed further in Section 5.2. The multi-behavior approach provides a framework for handling the various objectives of a CR society. This framework informs Section 5.3. The tools for time coordination aid in tackling one of the CR society objectives. This tool is discussed in Section 5.4.2.

5.2 Social Language

Communication among peers enables societies of CRs. As previously explored, in Chapter 2, social language provides an efficient means of providing this communication through utilization of context information and society specific mechanisms. Additionally, recall that Chapter 2 explored methods for encoding relevant information into actions. Finally, recall that Chapter 4 discusses the value of implementing a MAC layer focused approach for CR societies. Thus it is clear that the social language for CR societies should be based on communication through MAC layer actions in order to avoid diverting resources from direct communication.

The issues of radio observability greatly affect CR social language. The MAC layer is focused on determining appropriate access times and frequencies for accessing the spectrum. However, radios can not necessarily observe a peer's decision about how to access the spectrum. Rather CRs require a more explicit method of communicating the information about their intended spectrum that does not require registering with a central agent or greatly detracting from the transmission of application data.

The waggle dance of bees provides an analogy for what is required from a CR social language [11]. Bees need to communicate the location and goodness of some object, which is unobservable to bees at a hive. To solve this problem bees employ an explicit action to communicate this information to their peers. However, note that the bees also use their waggle dance for the purpose of determining a hive wide consensus through voting, which shapes the implementation and use of their social language.

CRs need their own waggle dance, with a slightly different focus. Specifically, CRs need an action-based communication method that can disseminate otherwise unobservable information to peers. CRs are less concerned with a voting mechanism analogous to that used to make decisions about new hive locations. Instead CRs need a mechanism to coordinate their sizes and positions in frequency-time. Note that a unique attribute of CRs is their ability to change their perceived size, since a CR's size in frequency-time refers to amount of spectrum occupied and the amount of time that spectrum is occupied. Thus the problem of CR social language is the problem of communicating the information necessary for radios to fill available space without overlapping their peers.

Serving all of the needs of a CR social language is a simple as sending a beacon. For our purposes, a beacon consists of a single packet, broadcast throughout a society. There are several examples of the power of a beacon for efficiently communicating information, but perhaps of the most applicable example is presented in the introduction of the DESYNC algorithm [64]. This algorithm uses a beacon to communicate a radio's size and position in time without requiring receiving radios to do anything more than receive the beacon and decode its header. Additionally, a beacon, as discussed here, provides a concise message that allows a social language in which recipients react in an appropriate manner without an inordinate amount of decoding. Specifically, in the case of DESYNC the decoding amounts to adjusting the firing time of the next beacon. This beacon based language also naturally communicates a radio's position in frequency.

Note that using beacons for social language strikes a balance between observability and the goals of CRs. Since radios will only recognize beacons as messages, CRs are effectively only visible while sending beacons. Transmissions for sending application data can not necessarily be differentiated from transmission by radios outside of the society, which are considered to be obstacles. This short period of message observability allows the CRs to spend the majority of their efforts on completing their primary goals and limits interference that may be caused by social language use. However, these benefits come at the price of radios potentially missing the brief messages. The potential for missing messages means that the society must be tolerant of such blind spots. This fault tolerance can be achieved through the interconnectiveness fostered in a society, meaning that there is a threshold for the maximum number of messages that be missed.

For CR societies a beacon based social language provides an efficient means of communicating a radio's position and timing to its peers. This beacon is not directed at any specific recipient, but rather is designed to allow broadcasting information throughout an entire society. That being said, for the beacon to efficiently communicate the channel of the CR, it is only sent on the frequency that the CR intends to use for application transmissions. To maintain up to date information within a society the beacon is sent once every T seconds, where T is referred to as the beacon period. Additionally, recall that in Chapter 2 testing showed that rule based encoding of information into actions allowed for the easiest use of the information by receiving agents. Thus, the beacon time and frequency of a CR is determined through rules to keep the efficiency of the social language high. This mechanism allows the radios to communicate all the information they need to coordinate themselves in time and frequency.

5.3 Applying Multiple Behaviors

As discussed in Chapter 3, the field of robotics provides the framework of the behavior-based design of multi-agent systems. This approach to designing multi-agent systems is based on selecting a basic set of behaviors, common to every agent, that can be combined to achieve the desired overall system performance. Note that behaviors are combined within a given agent to form compound behaviors and among a number of agents to form emergent behaviors. This multiplicity of combination inherently requires basic behaviors to be as simple as possible to avoid any unwanted interactions. Thus, basic behaviors are selected as simple solutions to individual, single-objective, problems and then combined.

While this approach does not include social language when it is presented in robotics, it does offer advantages for the use of social language. Since every member of a society has the same behaviors, they all react to stimuli in a similar way. This allows each individual in a society to understand the reasons for its peers actions without requiring a separate model for a peer's decision making process. The simplicity of the behaviors used also eases the complexity of social languages. As explained in Chapter 2, action based social languages are easiest to use when actions are determined in a straightforward manner. However, this simplicity is degraded when the interaction of several behaviors must be considered, thus social languages are best suited to communicating through actions that are influenced by a smaller number of behaviors.

5.4 Time Flocking

Time flocking is a behavior, composed of several basic behaviors, aimed at organizing the transmissions of radios in time. The coordination of the size and position of CRs employs the beacon based social language discussed in Section 5.2. Time flocking is also a common behavior, meaning that every radio in the society follows the same behavioral rules. Removing the need for behavior specialization allows for both easy understanding of the social language information and scalability. This coordination allows the radios employ the complex, composite behavior associated with the actions of a dynamic self-organizing decentralized network in time.

Accomplishing the time goals of CRs comes down to developing behaviors to address each time goal and combining them. Recall that the behavior-based approach to multi-agent systems tackles multi-objective problems by decomposing the problem into into sub-problems that can be addressed by individual behaviors and then combining these behaviors. In the case of CRs, the time goals can be stated as determining an appropriate window of time for the transmission of application data. Here appropriate specifically means two things: first, the transmissions must not interfere with other radios in the society and second, the transmissions must not interfere with radios outside of the society. Decomposing the goals in this way highlights the different approaches to handling them. Interference with peers within the society can be handled through the cooperation of neighbors, built into societies through communication and complementary behaviors. Interference with external radios must be handled in a different manner. Here, we refrain from making any assumptions about the capabilities of external radios that may be the subjects of interference. This strategy allows for operation alongside a wide variety of external systems, avoiding any dependence on abilities that may not exist. Note that while this assumption allows for the most general use, it is not necessary when information about external entities is readily available. Additionally, note that a lack of such information limits the options for interference control and makes avoiding interference the most straightforward approach.

Time flocking for CR societies arises from the combination of avoidance and cooperation behaviors. This compound behavior provides a method for addressing the two separate objectives of controlling intra- and extra-society interference. Avoidance behaviors are common in the field of CR and normally take the form of sensing prior to transmission. The avoidance behavior used in this work is described in Section 5.4.1. Addressing intra-society has the option to employ more complicated methods, involving communication among peers and the complementary design of MAC layers for peer radios. The behavior employed here is covered in Section 5.4.2. The combination of these is discussed in Section 5.4.3.

5.4.1 Avoidance

Avoidance in time is simply a case of not transmitting while another source is transmitting. Failing at this goal can take one of two forms, either the CR starts transmitting while another radio is transmitting or another radio starts transmitting while the CR is already transmitting. Avoiding extra-society interference then has two components. Firstly, one must check for existing transmissions prior to transmitting. Secondly, transmissions must be kept short to reduce the possibility of less capable systems starting transmissions during CR transmission or shorten delays for systems that detect transmission but not change frequency. This goals can be met by enforcing a sensing period prior to any transmission and limiting transmissions to single short packets.

Note that this strategy provides the desired interference elimination at the cost of throughput. This is a necessary trade off to accomplished the desired dynamic behavior. Balancing this trade off can be formulated as an optimization problem for specific scenarios, however, this does not provide good general purpose utility due to variations of systems.

5.4.2 Cooperation

There are numerous options for handling intra-society interference through cooperation. Recall the criteria for selecting basic behaviors given in [24]: simplicity, locality, correctness, stability, repeatability, robustness, and scalability. Thankfully, the DESYNC algorithm already provides a tool to accomplish time-oriented sharing in a manner that meets all of the above criteria. This method provides the necessary self-organization of radios in time. This algorithm was discussed in Chapter 4.

Note that this cooperation behavior must simultaneously determine the time position and size of application transmissions. For example, DESYNC provides the position by adjusting the timing of beacon messages and the size by determining the start and end of transmission slots. Note that the use of TDMA style slots is not required by this behavior, but rather it provides a clear indication of appropriate times for application transmissions. Such clear indications help to determine whether or not the possibility of intra-society application interface exists. They also allow for preventing bad situations that result from faults. For example, a modification to DESYNC that resets transmission slot times when a beacon is received during a radio's active slot provides for recovery from faults in time cooperation.

5.4.3 Time Combinations

Combining the cooperation and avoidance behaviors discussed above provides the total time-flocking behavior of CR societies. While the exact methods for combining behaviors discussed in [24] do not necessarily directly apply to CR societies, they provide guidance. Mataric discussed combining robot behaviors on the basis of either the weighted sum of target velocities or stimulus induced behavior switching. The concept of weighted sums of individual targets generated by behaviors does not necessarily apply to CRs, because of the wide variety in parameters managed by CRs. Such a variety of parameters, each with its own focus or implication, typically does not allow for combining all the behaviors in a CR as the weighted sum of signal parameters. Instead, several CR behaviors tends to affect the same sort of parameters. For example, both the behaviors of avoidance and cooperation simultaneously affect the timing of transmissions, but in slightly different ways. Avoidance affects the timing of individual transmissions, while cooperation affects the timing and overall length of the slot for application transmissions. Combining avoidance and cooperation provides the time-flocking behavior required for CR societies.

5.5 Frequency Flocking

Frequency flocking is the frequency complement to time flocking. It is a composite, common behavior aimed at dynamically organizing a CR society in frequency. Recall that time flocking controls both the position and size of the CRs in time. Frequency flocking only controls the position of the CRs in frequency. This is because the spectral size of transmission depends on physical layer characteristics, which are beyond the control of MAC layer solutions.

Just as in time flocking, the frequency goals of a CR society have two major components. First, the CRs in a society need a means of finding one another. Ideally, the entire society would operate on the same frequency, which allows for the social language to reach, and therefore help coordinate, the most possible radios. Second, the society needs a mechanism to find new open frequencies when a particular channel becomes congested, either through the presence of too many CRs or from the appearance of a outside radio system. Each of this sub-goals require the CRs to acquire knowledge of peer positions in frequency and the state of a channel, as being either clear or congested. Tackling the goal of keeping the society together in frequency is discussed in Section 5.5.1 and handling the need to find new channels is covered in Section 5.5.2. A method of learning to aid these behaviors is discussed in Section 5.5.3.

5.5.1 Aggregation

Aggregation is the behavior that keeps radios together in a society. Since the CRs that are members of a society have efficient methods for sharing the spectrum, grouping societies provides the highest amount of coordination. Therefore whenever possible the society should be operating together on a single frequency. To accomplish this aggregation provides the behavior required for CRs to group themselves in frequency.

This behavior comes down to a search technique for finding peer radios. Recall that the social language for CRs is based on sending a beacon that communicates the timing of CR's slot and its frequency of operation. Specifically, it worth noting that this social language makes CRs identifiable to its peers for instances rather than long durations in time. Any method of searching for peers must take account of this limited observability. Since each radio sends a beacon once every beacon period, each channel must be searched for the duration of one beacon period. This search time period, referred to as the dwell time, prevents radios from missing the presence of peers.

To allow for a bounded search, the range of allowed channels is specified for each radio. A search order then specifies the order in which these channels are searched for peers. Note that the search order of channels must be determined in manner that minimizes the occurrence of search cycles, where radios continue to miss each other. For an example of a search cycle consider a case in which two radios A and B are on channels 2 and 4. If the search pattern for each of this radios simply indicated that they should always check the next higher channel, the radios would never find one another. Rather A would check channel 2 while B checks channel 4, then A would check 3 and B would check 5 and so on. Rather, the radios always search the channel at the center of the specified range first and expand out from there. Specifically, the radios will search the center, then one above the center, then the center again, then one below the center, then the center again, then two above the center, and so on. This focus on search the center of their range gives the radios a rendezvous point, shifts the society toward the middle of their range, allowing for future alterations in either direction, and prevents search cycles. Thus the search order combines with the dwell time in order to provide a frequency aggregation behavior.

5.5.2 Dispersion

Dispersion in frequency is much simpler than aggregation. This behavior simply moves the CR out of congested channels. To accomplish this CRs need only to determine whether their current channel is congested, through sensing, and change channels if it is. Since finding other radios is not the priority of this behavior, the factors of dwell time and search order are not important here. Rather the CRs simply move to the next channel in the available range and attempt to operate there. This method provides CRs with a straightforward method of handling congestion.

5.5.3 Frequency Learning

The frequency flocking of CRs can be improved by allowing the radios to remember trends in frequency use. Since the radios have initial knowledge of the channels in which they may operate, the radios can use this to keep track of which channels tend to be open and which tend to be occupied. Specifically, each channel is assigned a visitation probability,

p. When the radio behavior indicates that a radio should visit a particular channel, there is a $1-p$ that the channel is skipped in favor of the next channel in line to be visited. Initially the visitation probability is one for every channel, but is decremented by value d whenever a radio must leave a channel due to congestion. This naturally drives radios away from channels that are over used. However, since the usage of channels changes with time, visitation probabilities are incremented by a small value i every n beacons to allow radios to forget stale information. This light weight learning allows radios to adapt to the frequency environment over time without requiring unrealistic knowledge, such as the transmission probability of external systems, or demanding too many computational resources.

5.5.4 Frequency Combination

Combining the behaviors of frequency aggregation, frequency dispersion, and frequency learning is fairly straightforward. Aggregation and dispersion are triggered at different times. Aggregation behaviors come into effect when a CR finds itself alone in a channel and dispersion is activated when a CR is in a congested channel. Frequency learning is slightly different in that it does not directly affect the frequency of operation; rather, it supports the other frequency behaviors. As such, frequency learning is active continuously in the background of the other behaviors. Thus the combinations of the behaviors are able to provide a total frequency flocking behavior.

5.6 Combination Flocking

The combination of time flocking and frequency flocking provides the overall behavior of the network. These two composite behaviors both operate simultaneously, controlling various aspects of the network. The important consideration in this combination is the interaction of the behaviors involved. Even though behaviors may affect different aspects of the system their interactions must be considered. These additional interactions arise from the complexities of radio systems versus robotic systems, i.e., radio systems don't control their behavior primarily through a single parameter. Specifically consider time avoidance and frequency dispersion, each of which requires sensing. Additionally, note that the sensing required for time avoidance provides the information needed in frequency dispersion. That is, if a CR is prevented from transmitting due to sensing an occupied channel prior with the time avoidance behavior, it's an indication of a congested channel. Thus the operation of one behavior can support another. However, consider the interaction between time avoidance and frequency aggregation. Frequency aggregation determines the presence of other radios by receiving their beacons. However, if CRs are prevented from sending beacons due to the time avoidance behavior, their peers might falsely activate the aggregation behavior. To prevent this false activation, aggregation should be delayed until dispersion can activate properly. Simply delaying allows dispersion to act before CRs falsely determine the need for aggregation activation. Consideration of these interactions allows for the proper combination of CR behaviors.

5.7 Methodology

So far this work has provided a development of the method for constructing CR societies. This method began with a definition of the problem to be solved. This allowed for the identification of the primary objectives that must addressed to solve the problem at hand. These primary objectives provide the factors that must addressed by behaviors. Recall from Chapter 3 that behaviors need to be simple approaches to solving individual aspects of some objective. Given the behaviors necessary to address the problem at hand, a social language can be determined to support the behaviors. Finally these behaviors and language can be implemented according to the selected hardware platform, as will be discussed in Chapter 6.

Figure 5.1 briefly summarizes the method for developing CR. The mechanisms of social language are used in conjunction with the behavior-based approach to multi-agent system design to provide flexible networks of CRs. Combining these techniques provides both connectivity among peers and controlled multi-objective emergence. The attributes of social language allow the radios to overcome the challenges of their domain while avoiding undue amounts of coordination communication. Additionally, domain specific tools must be incorporated in any behavior based system.

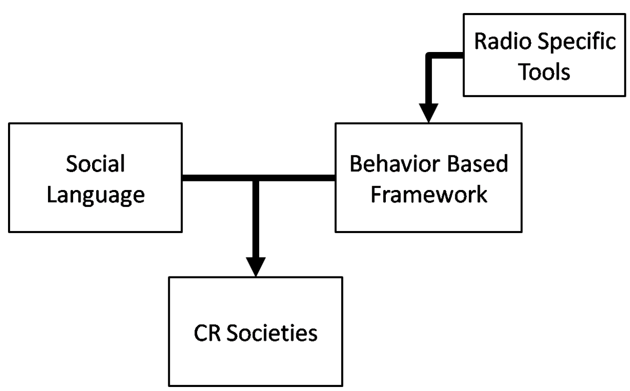


Figure 5.1: Approach to developing CR Societies

Applying social intelligence to a problem is matter of following a design process. First the problem must clearly stated. CR societies provide practical solutions through emergent behaviors based on interactions both internal and external to radios involved. As such the problem at hand must be clearly understood. In this work, social intelligence is applied to the problem of developing self-organizing decentralized networks that can benefit from social interactions. Once the problem is stated and understood, the arena in which the problem is to be addressed must be defined. The scope of the solutions provides a clear indication of the actions that may be taken and the responsibilities that must be handled. For example, this work addresses the problem on the MAC layer, with the result that the solution must handle the time and frequency size and position of transmissions. Given the arena of solution, the problem must be decomposed according to the parameters that must be set. Understanding the problem in terms of parameters to be set gives an indication of the behaviors that are required. In this work radio transmissions must be organized in time such that they do not overlap with external interference and each radio is provided with an opportunity to transmit application data. Additionally, the radios must be organized in frequency such that they cluster in frequency, exit congested channels, and learn channel conditions over time. After the problem has been decomposed according to the parameters, a social language must be developed to communicate important information among peers. Recall that Chapter 2 provides insight into appropriate mechanisms for social language communication. A beacon that communicates a radio's position in time and frequency provides the social language used here. Next basic behaviors that address each sub-objective in each parameter must be determined. Recall from Chapter 3 that these behaviors need to be as simple as possible to reduce unwanted interactions. Additionally, the basic behaviors should be based on local observations and the information made available through the social language. Here the time avoidance and time cooperation behaviors provide the abilities to prevent transmissions into interference and determination of slots for radios. The frequency aggregation, frequency dispersion, and frequency learning allow the radios to cluster in frequency, exit unfit channels, and learning channel conditions, respectively. The only remaining task is then to implement the behaviors according to capabilities of the hardware platform. The next chapter discusses how this was done in this work.

Figure 5.2 shows the problem decomposition for this work. At the top is the total system behavior, the decentralized self-organizing network of CRs. As discussed in Chapter 4, this is primarily a MAC layer problem. Defining this domain suggests that behaviors to control the two parameters of time and frequency are required. In time behaviors are needed to prevent interference and provide cooperation among peers. In frequency behaviors are needed to group radios, allow them to escape interference, and learn their environment. Each of these behaviors provides the simplest solution to their individual goals based on local information and the beacon based social language.

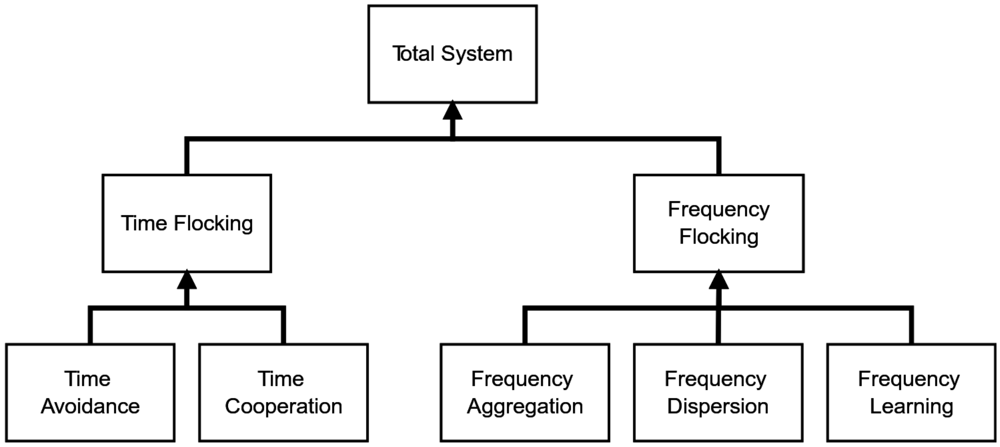


Figure 5.2: Diagram of Behaviors

This work provides the first development of a CR society. Thus, this work focus on examining the attributes of CR societies and a proof of concept implementation. Given the initial introduction of the concepts presented here, future research can go on to develop a more general methodology for CR deployments. Note however, that the application focus of CR societies means that particular approaches and metrics for every society will be different.

5.8 Conclusion

This chapter has discussed the methods for constructing CR societies with the tools gathered thus far. Specifically, the behavior based approach for designing robotic systems offers the framework for handling the various objectives of a CR society. This framework is supported by the efficient communication of action based social languages and tools for the time coordination of radios. The method of combining these tools presented here provides a novel approach to designing networks of CRs based on knowledge gained from several diverse fields.

Chapter 6

Implementation of a Prototype Cognitive Radio Society

6.1 Introduction

This chapter discusses the implementation of the behavior-based approach to developing CR societies discussed in Chapter 5. The implementation presented here is a prototype of a CR society, focused on providing a proof-of-concept, highlighting significant features of the work. As such, each aspect of the implementation is designed to exhibit the features of a dynamic self-organizing society of CRs. Section 6.2 discusses the hardware platform selected for this purpose. Section 6.3 provides details about the implementation environment and tools used for development. Section 6.4 covers the implementation details of the behaviors themselves. Section 6.5 describes the system architecture used for implementation and discusses some of the computational aspects of the system. Section 6.6 examines the limitations imposed by the design choices. Finally, Section 6.7 summarizes and concludes the chapter.

The prototype implementation presented here uses low-cost radios and computational platforms. This means that the individual radios face stringent limitations in terms of hardware capabilities. These limitations result in timing jitter, reconfiguration costs, and random computational delays. As will be show during evaluation, the methods used to develop CR societies are robust against these limitations, although these factors certainly impact the design of the system.

6.2 Implementation Platform

The development of a dynamic self-organizing network of CRs, here termed a CR society, focuses on enabling coordination through local behaviors and lightweight efficient communication. The use of local behaviors limits the demands on the sophistication of each individual CR and helps to provide scalability. Efficient communication continues the theme of limited sophistication and provides a means to coordinate the development of emergent behavior. Such mechanisms allow for the use of low sophistication CRs.

The use of low sophistication CRs provides an option for the proliferation of self-organizing CRs. As such, testing the development of CR societies on inexpensive, minimally intricate radios is of key importance for determining the value of the approach. Therefore, the implementation platform has been selected for its inexpensive and modestly complex nature. The implementation helps to validate the approach for use in large networks of inexpensive radios.

The implementation hardware for this work is the combination of a Beagleboard-xM computational platform [69] and a Hope-RF RFM22B radio front end [70] named SKIRL. This combination of hardware was first proposed for CR by Young [71]. The system offers a flexible radio frequency integrated circuit (RFIC) supported by computational power analogous to the average smart phone. Young points out that the use of a RFIC instead of the more common software defined radio approach improves the radio performance and decreases cost by removing the reliance on powerful field programmable gate arrays (FPGAs). While this does come at the cost of reduced options at the physical layer, such a system is ideal for the work conducted here. Table 6.1 summarizes the capabilities of the SKIRL platform.

Table 6.1: Summary of Hardware Capabilities

Frequency Range	240 - 930 MHz
Modulations	frequency shift keying (FSK), Gaussian frequency shift keying (GFSK), on-off keying (OOK)
Maximum Power Output	13 dBm
Minimum Receiver Sensitivity	-121 dBm
Sensing Bandwidth	100 kHz

The SKIRL platform provides the hardware and minimal software for testing CR societies. The Beagleboard-xM runs a minimalist form of the Ubuntu Linux operating system, which allows for direct interaction with the system through ethernet and use of several different programming languages. The computational part of the platform interacts with the radio front end by setting a collection of registers and a Python module exists to streamline the process. The platform can access and sense only a single channel at a time, and it requires a finite reconfiguration time for any change. Specifically, sensing information is provided in the form of received signal strength indication (RSSI) measures for the current channel. Data is sent by adding to a transmission buffer on the RFM22B, setting all necessary physical layer parameters (typically modulation and bit rate), and sending the transmit command to the front end. Data is received into a received buffer and held there, until it is read out. The details of controlling the SKIRL platform are discussed in Young's dissertation [72]. Figure 6.1 provides a picture of the SKIRL platform with a US nickel for scale. The entire platform costs approximately $ 250.00 USD.

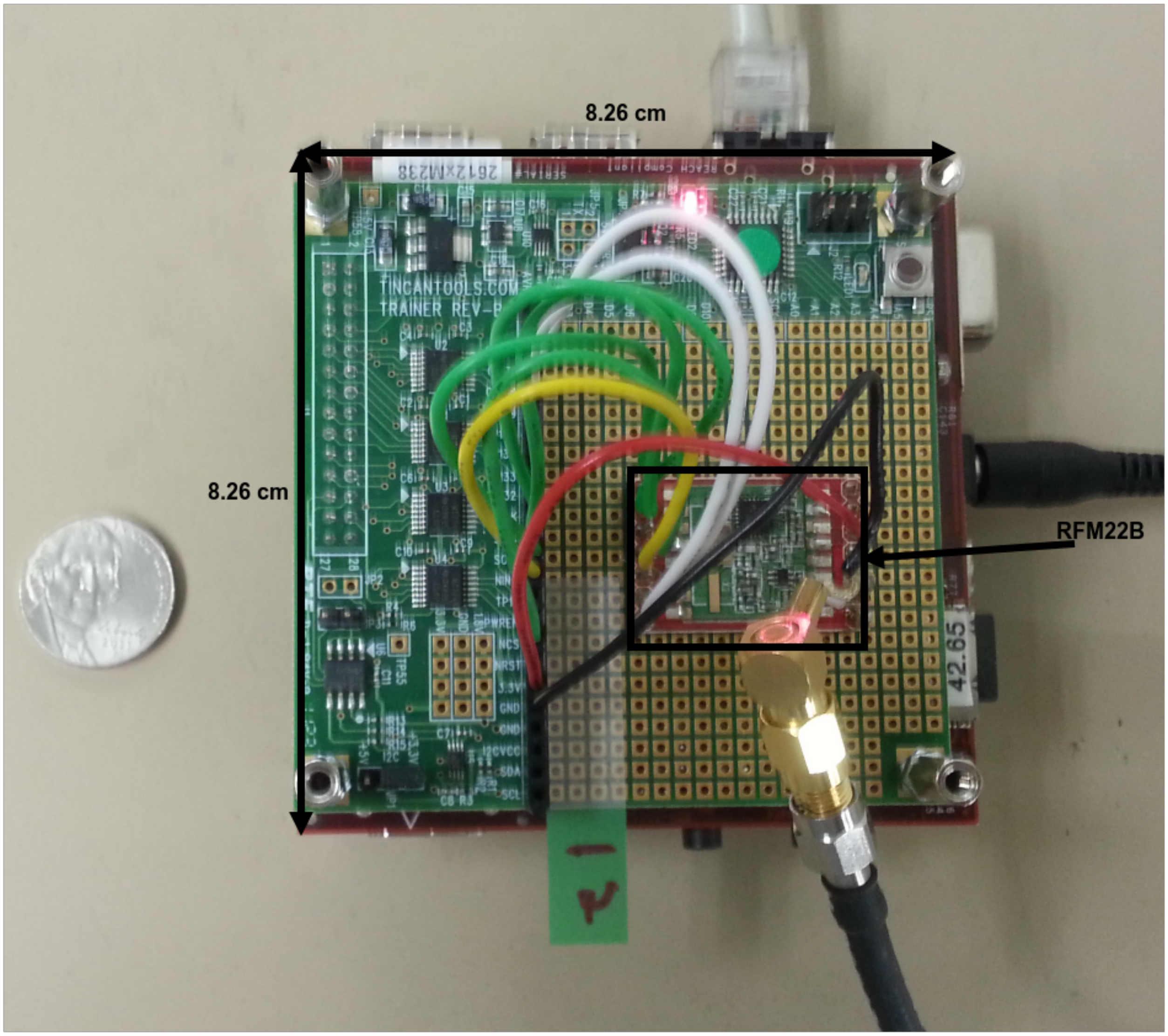


Figure 6.1: Picture of SKIRL Platform

As is the case with any hardware based implementation, the SKIRL platform informs the parameters used in the development of CR societies. The RFM22B employed on the SKIRL platform is optimized for use in the 433 and 915 MHz bands, thus all testing is conducted around 433 MHz. Specifically 21 channels with center frequencies between 430.25 MHz and 437.75 MHz have been defined. The centers of adjacent channels are spaced 750 kHz apart.

Since the work focuses on MAC layer control, the same physical layer parameters are used for all transmissions. Specifically, GFSK modulation with a bit rate of 4.8 kbps and a power of 17 dBm was used for the transmission of every signal.

Each transmission consisted of packets, 64 bytes in length. The structure of this packet is summarized in Table 6.2. The first 14 bytes of the packet constitute the packer header, which includes the packet type, a local time stamp, the sender, and the intended recipient of the packet. Packets may be firefly beacons, application data, or acknowledgment packets. Note that the time stamp information is not used by receiving radios and is present simply for debugging purposes. The remaining 50 bytes of the packet are application data. No error checking is applied.

Table 6.2: Packet Structure

<table>
<tr><th>Byte</th><th>Component</th><th>Field</th></tr>
<tr><td>0</td><td rowspan="14">Header</td><td rowspan="3">Packet Number</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td rowspan="8">Time Stamp</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td>Destination</td></tr>
<tr><td>12</td><td>Source</td></tr>
<tr><td>13</td><td>Packet Type</td></tr>
<tr><td>14</td><td rowspan="3">Payload</td><td rowspan="3">Data</td></tr>
<tr><td>...</td></tr>
<tr><td>64</td></tr>
</table>

6.3 Implementation Setting

As this work aims to provide a proof of concept for the CR societies discussed herein, four CRs were used. Each radio was assigned a unique name for identification. The names are fuhr, gillies, joliat, and howe. These radios were arranged in a 2.82 m by 3.58 m space as shown in Figure 6.2. Also shown in this figure are antennas for a HP 8594E spectrum analyzer and two HP 8648C signal generators. The spectrum analyzer provided a means of verifying the positions of CRs in frequency and the signal generators simulated external networks. The system was implemented in a controlled lab environment, free from any unwanted signals.

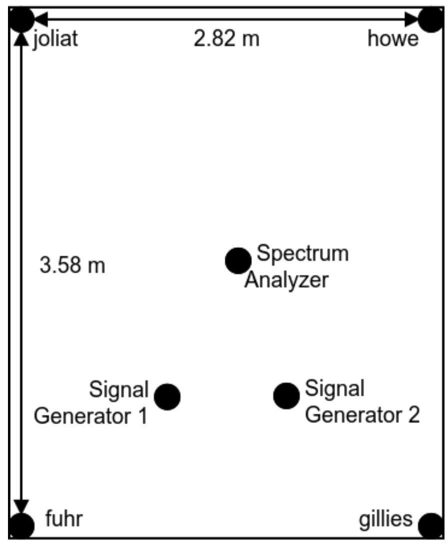


Figure 6.2: Arrangement of Radios

A logging system allows for the tracking of radio behavior more closely than can be achieved with the spectrum analyzer alone. As the CRs operate, they generate log messages with either high or low priority. Low priority messages provide detailed information useful for debugging and development purposes. High priority messages, on the other hand, are those most useful for determining system behavior and tracking interaction with other radios. All messages are saved into a local log file and high priority messages are sent to a central logging machine through ethernet. The purpose of this central machine is only to collect log information in a effort to observe interactions between radios; no control or synchronization messages are sent out from this machine. Instead, the central logging computer collects all messages into a signal file in the order that they arrived. Figure 6.3 provides a screen shot of the logs being collected as they are received by the central logging computer. This logging system provides tools necessary for development.

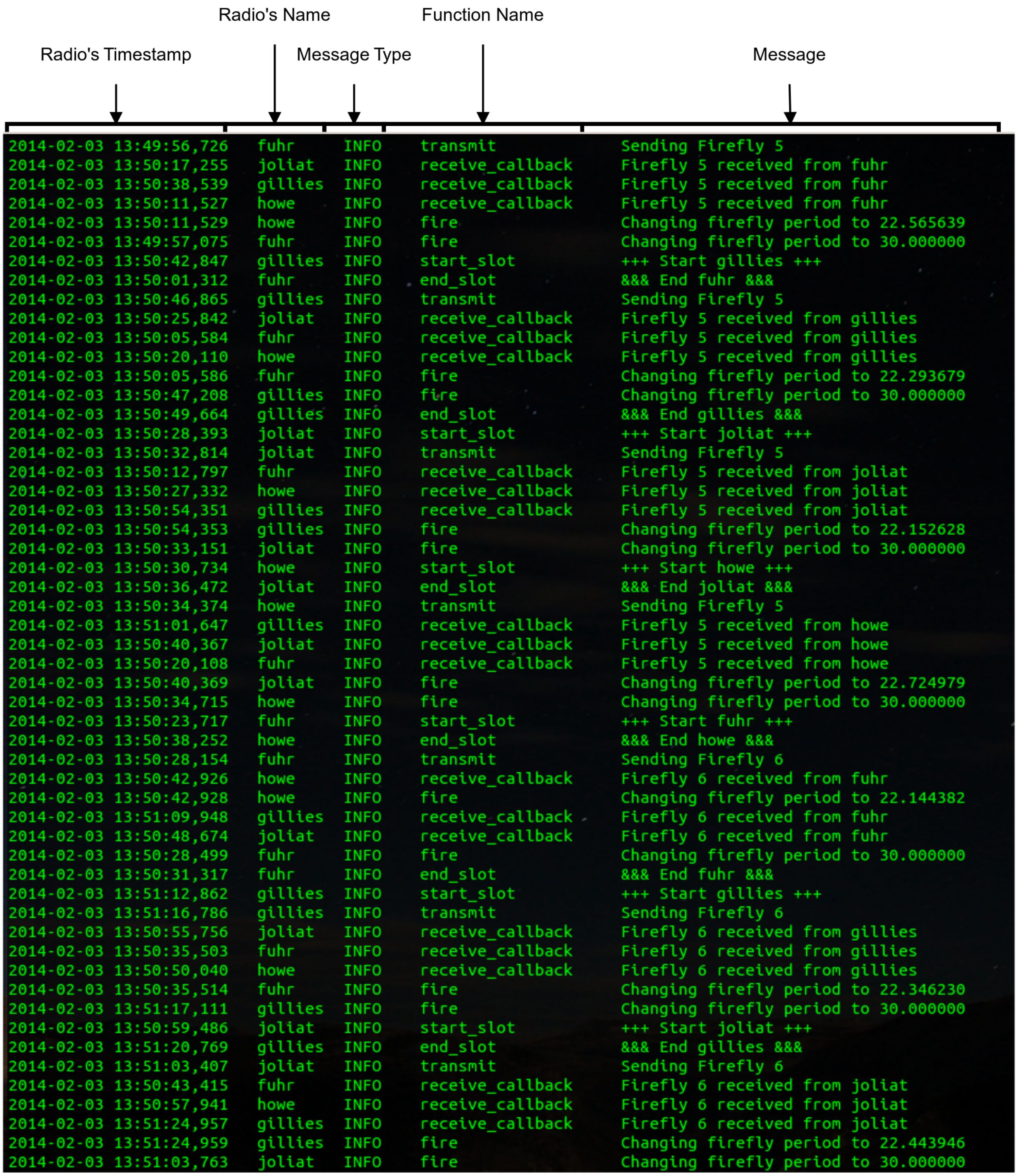


Figure 6.3: Screen Shot of the Logging

6.4 Implementing Behaviors

This section provides the details of the implementation for each behavior discussed in Chapter 5. Recall that these behaviors follow the behavior-based approach to multi-agent system design from robotics. As such each behavior provides a simple, local solution to a specific problem faced by a CR society. Note that these behaviors do not provide the only solutions available, rather they were selected and designed for the purpose of implementing a prototypical CR society on the SKIRL platform. Each behavior was therefore implemented according to the capabilities of the SKIRL platform.

6.4.1 Time Avoidance

The purpose of time avoidance is to eliminate interference with external entities. Recall that Chapter 5 discussed two methods for accomplishing this: sensing prior to transmission and short transmissions. On the SKIRL platform sensing is RSSI based. Thus, prior to any transmission, channels were monitored for signals above a specified RSSI threshold for a specified sense time. For my purposes, the channel intended for use is monitored for 0.1 seconds for signals with RSSI greater than -22.5 dBm. These parameters were empirically determined to accurately sense the presence of other transmissions. If a signal with RSSI above the threshold is sensed, the system immediately determines that the channel is busy and prevents transmission; otherwise the system waits the full sensing time before declaring the channel clear for transmissions.

6.4.2 Time Cooperation

Time cooperation is the main time behavior of the system. This behavior is tasked with determining an appropriate position and size for application data transmission. Recall that in the context of this system, the size of a transmission in time refers to the time slot in which such a transmission is allowed and not necessarily the duration of time that the channel is actually in use, which depends on the amount of data to be sent and the physical parameters used. This definition for size implies a TDMA approach to time management, which, as discussed in the previous chapter, has the benefit of clearly specifying who is authorized to use the channel.

The DESYNC algorithm [64] provides the foundation on which the time cooperation behavior is built. Recall from Chapter 4 that this behavior is based on spreading the transmissions of radios in time. As such this is the first behavior to employ the beacon based social language. Additionally, recall that the DESYNC algorithm split the time into periods of a specified length. These periods are analogous to the beacon periods discussed in Chapter 5. The DESYNC algorithm provides the means to adjust these periods for the maximum possible distance in time between adjacent firings and for defining slots based on beacon firing. Note that the beacon used here is a form of social language that communicates a radio's position, and as we will see later indicates its size, in time, even though the authors of [64] do not recognize it as such. The implementation of these methods and various modifications that allow for combination with other behaviors is discussed below.

To examine the DESYNC algorithm, we will first focus on the spreading of beacons, termed desynchronization by the authors of [64]. Initially, let's consider a single radio in order to understand the mechanism for splitting time into rounds. When started our radio is given a value for its beacon period, T, of 10 seconds, for example. This means that every 10 seconds, from the activation of the radio onward, this radio will fire a beacon. Each firing signals the end of a time round of the radio. The percentage progress of a radio through a round is referred to as the current phase $\phi(t)$ of the radio. Note that this term is borrowed from the study of pulse coupled oscillators (PCOs) [73] and that phase is a dimensionless quantity. For example, if the radio has 5 seconds until the next firing, it has a phase of 0.5 and if the radio has 2.5 seconds until the next firing, it has a phase of 0.75. Phase may be calculated as $(t_{current} - t_{start})/T$, where $t_{current}$ is the current time as given by a local running clock and t_{start} is the time when a round began. When radios reach a phase of 1 they fire a beacon and restart their timer, indicating a reset of the phase to 0.

Figure 6.4 provides a visualization of the phase of radios. Phase advances in the clockwise direction. This circular depiction of phase provides a visual representation of a moment in time. Note that the DESYNC algorithm splits time into sequential periods; a single period is sufficient to depict any moment in time. In this diagram the 12 o'clock position represents the beginning of a time period, this is the position that corresponds to both a phase of 1 and a phase of 0. As time passes, radios move around the circle in a clockwise direction. The radios first reach position A which corresponds to a phase of 0.25 and then proceed to position B, or a phase of 0.5. Note that each diagram only depicts a single moment in time and displays the various phases of the radios show. Thus, Figure 6.4 shows two radios, A and B, just at the moment when radio A has a phase of 0.25 and radio B has a phase of 0.5.

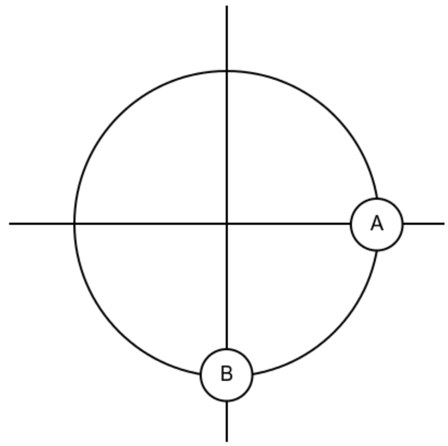
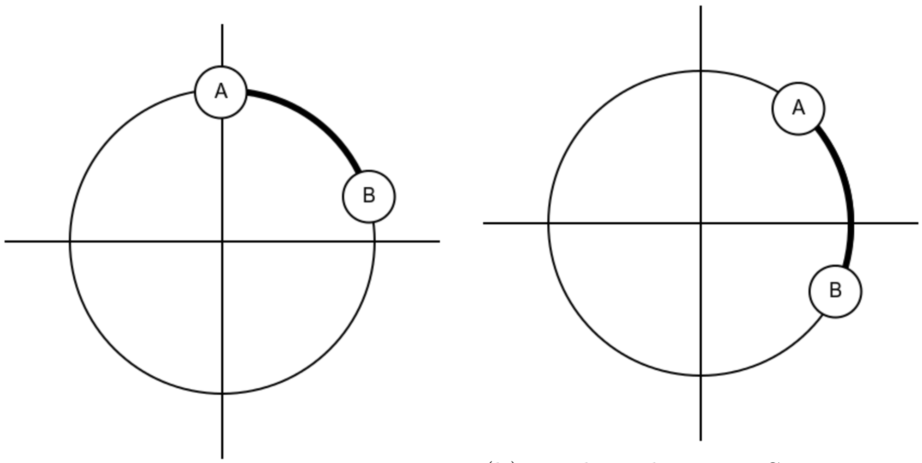


Figure 6.4: Visualization of Radio Phase

Given this system for understanding the phase of a radio within a given time period, let us now consider the difference of phase between two radios. At this point, it should be noted that the internal clocks of radios are not synchronized, so no radio has access to the phase information of another radio. Instead, radios have access to the phase difference between themselves and their peers. Imagine two radios A and B, each firing beacons every T seconds. Without loss of generality, assume that radio A fires a beacon first. Radio B can determine the phase difference between itself and radio A, $\Delta_{BA}(t)$, at that moment in time simply by noting its own phase at that moment. This is because $\Delta_{BA}(t)$ is given as the difference $\phi_B(t) - \phi_A(t)$, however when radio A fires its phase is reset to zero. Note that strictly the phase difference varies with time due to adjustments and clock drift, however clock drift is typically small, such that phase differences are fairly static between deliberate adjustments.

Figure 6.5 shows the principles of determining phase difference. Figure 6.5a shows the situation just after A fires and Figure 6.5b depicts a point later in time. Note that at each of these time points the phase difference is the same. However, consider the process of radio B determining this difference at each point. Just after radio A fires, radio B has a wealth of knowledge. Firstly, since radio A just fired radio B knows that A will be resetting its phase to 0 and radio B knows its own phase. This allows radio B to easily calculate the phase difference between itself and radio A. At a later point, on the other hand, radio B does not have all necessary information. Specifically while radio B knows its own phase, it does not have access to radio A's phase. Thus radio B can easily determine the phase difference between itself and radio A just after radio A fires, but not at other points in time.



(a) Radio Phases Just After A fires

(b) Radio Phase at Some Later Time

Figure 6.5: Phase Difference Mechanics

Given the concepts of phase and phase difference, along with a mechanism for a node to determine phase difference through observation, the mechanism for determining a new phase may now be discussed. To proceed, let us first clarify symbols. For a group of n radios the phase of the i^{th} radio is given by $\phi_i(t)$. All radios are assumed to fire once every T seconds. The term round will refer to a cycle in which every radio fires once. For notational convenience, let use assume that radios are numbered such that $i < j$ implies that radio i fires after radio j in a given round. Finally, the phase difference $\Delta_i(t)$ will refer to the difference $\phi_i(t) - \phi_{i-1}(t)$. The concept of phase neighbors provides the final ingredient for determining a new phase for a radio. Note that any radio i is phase adjacent to both radio $i - 1\, mod\, n$ (its next neighbor) and radio $i + 1\, mod\, n$ (its previous phase neighbor). Recall from prior discuss that any radio i has access to both $\Delta_i(t)$ and $\Delta_{i+1}(t)$ through observation of its peers. Then the determination of new phase simply requires applying the local goal of maximizing the time distance between adjacent firings to the local situation. Specifically, radio i wants its next firing to be the mid-point of its two phase neighbors. Equation 6.1 shows how this midpoint may be determined with local information.

$$\begin{aligned}
\phi_{mid}(t) =& \frac{1}{2}\left(\phi_{i+1}(t) + \phi_{i-1}(t)\right) && (6.1)\\
=& \frac{1}{2}\left[\left(\phi_{i+1}(t) - \phi_i(t)\right) - \left(\phi_i - \phi_{i-1}(t)\right)\right] + \phi_i(t) && (6.2)\\
=& \phi_i(t) + \frac{1}{2}\left(\Delta_{i+1}(t) - \Delta_i(t)\right) && (6.3)
\end{aligned}$$

Figure 6.6 provides a visual depiction of the benefits of adjusting one's phase to midpoint of its phase neighbors. In this figure radio B has just determined the phase difference between itself and radio A, after previously determining the phase difference between itself and radio C. From this information radio B knows that it is closer in phase to radio A than radio C. Radio B then uses the process discussed above to determine the midpoint between radios A and C to use as its new target phase. This midpoint provides the greatest distance between radio B and its phase neighbors, without changing the firing order.

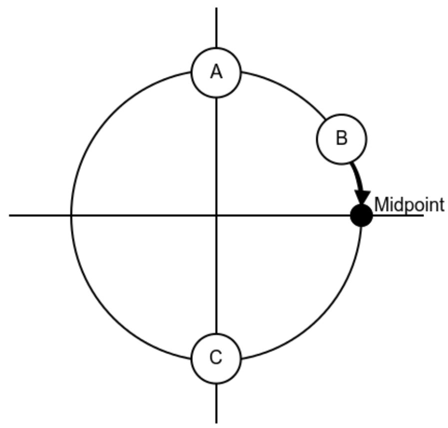


Figure 6.6: Midpoint of Phase

Given a new goal phase, achieving the phase for the next round is a matter of altering beacon timing. Recall that radios fire beacons once every T seconds, which corresponds to always firing with a phase of 1. When a radio needs to adjust its phase it simply adjusts the time before firing its next beacon to be T' instead of T. If $T' < T$ the radio reduces its phase and if $T' > T$ the radio increases its phase. After a phase adjustment is made, radios return to waiting T seconds to fire the next beacon. The adjustment period T' is easily determined from a target phase ϕ as ϕT.

Note that the authors of [64] suggest tempering each phase adjustment to help convergence. Since phase adjustments affect the next beacon fired, but are determined from the most recent observations, adjustments inherently contain a delay of approximately T seconds between the determination of a new phase and acting on that determination. This delay can degrade overall performance since all radios are able to adjust their phase in a given round. To avoid such degradation, the authors of [64] suggest setting target phases as shown in Equation 6.4. In this equation, ϕ_{target} is the new, tempered target phase, $\phi_{current}$ is the most recently used phase, and ϕ_{mid} is the mid-point of phase neighbors as discussed above. The parameter α is the tempering factor with value between 0 and 1 and typically close to 1.

$$\phi_{target}(t) = (1 - \alpha)\phi_{current}(t) + \alpha\phi_{mid}(t). \tag{6.4}$$

Phase adjustments as discussed above allows radios to spread through out in a time period and provide the basis for determining slots. Radios simply start their slots at the midpoint of their prior neighbor's phase and their own firing. Slots are then ended at the midpoint of their next neighbor's phase and their own firing. Note this organization of slots is designed to keep a radio's firing within its own slot to prevent one radio's beacon from interfering with another radio's application transmissions. Determining these midpoints again applies the knowledge of phase differences gained through use of the beacon social language. Radios have access to $\Delta_i(t)$ by noting their phase when the next phase neighbor fires. Likewise $-\Delta_{i+1}(t)$ is determined by a radio noting its phase when its prior phase neighbor fires. The phase associated with the beginning of a slot can then be determined as shown in Equation 6.5 and similarly for the end in Equation 6.7. Recall that radios fire at a phase of 1 and immediately reset their phase to 0; thus, either can be used for the phase of radio's firing Equation 6.5 uses a phase of 1 to ensure slots begin prior to a radio firing, while Equation 6.7 uses a phase of 0 to ensure ending after the firing. Once target phases for the start and end of a slot are determined the times to wait can be determined in a similar manner to the wait times for beacons.

$$\begin{aligned}\phi_{start}(t) =& \frac{1}{2}\left(1 + \phi_{i+1}(t)\right) && (6.5)\\ =& \frac{1}{2}\left(1 + \Delta_{i+1}(t) + \phi_i(t)\right) && (6.6)\end{aligned}$$

$$\begin{aligned}\phi_{end}(t) =& \frac{1}{2}\left(0 + \phi_{i-1}(t)\right) && (6.7)\\ =& \frac{1}{2}\left(\phi_i(t) - \Delta_i(t)\right) && (6.8)\end{aligned}$$

Figure 6.7 depicts the transmission slot of radio A. Note that the slot start point is at the midpoint of radio A's firing and radio C's phase and the end is at the midpoint of radio A's firing and radio B's phase. Additionally, note that the slot contains the firing of radio A. These properties allow the radios to split the time into slots based on the firings of peers.

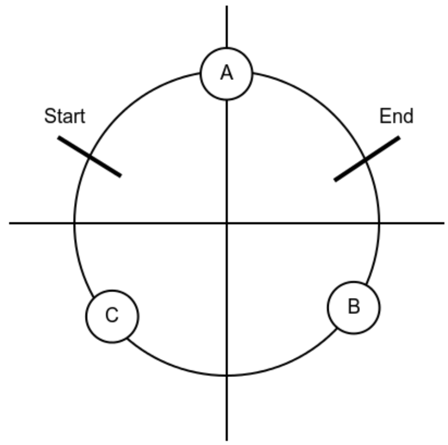


Figure 6.7: Depiction of Slots as Radio A Fires

The authors of [64] suggest a mechanism for a new radio to join an existing network of radios. This method employs an interrupt packet to signal the owner of the current slot that the new radio would like to join the network. This interrupt packet adds a packet to the algorithm with the goal ensuring that new beacons will be heard in an existing network.

The above describes the DESYNC algorithm as presented in [64], with the added term of social language. This algorithm provides a means to organized radios in time. However, the DESYNC algorithm is not fit for direct use as a behavior in a CR society.

This deficiency arises because the DESYNC algorithm does not include mechanisms to cope with the interactions that arise from multiple behaviors. Since both the time avoidance and time cooperation behaviors control the timing of transmissions, the interactions between these behavior must be considered. Specifically, because the time avoidance behavior either delays the transmission of packets or prevents their transmission entirely, it degrades the performance of the DESYNC algorithm. Additionally, note that when the avoidance behavior prevents the sending of a timing beacon, that radio effectively exits the system. Unfortunately, the interrupt packet that would then be used for the radio to rejoin the system could then prevent additional beacons. This could lead to a cycle of forcing radios to exit the network and greatly damage the cooperation. To prevent issues that arise from the interaction of behaviors, the situation must be carefully considered. For example, the issue in this case arises from the additional sensing enforced by the avoidance behavior. Note that such additional sensing also removes the need for the interrupt packet; if radios spend additional time sensing, missing the beacon of a new radio is less likely. Thus interrupt beacons can safely be removed from the system. To prevent the overlapping of slots that can arise from this scenario, radios reset their slots if a beacon is received during their slot.

Figure 6.8 depicts a scenario in which interrupt packets are problematic. This figure shows the moment in time just as radio A is preparing to fire. Radio A, employing the avoidance behavior, is sensing the channel to determine whether a transmission is safe. First, we will consider the outcome if radio C sends an interrupt packet. With this approach radio C would send an interrupt packet that would cause radio A to believe the channel is busy and refrain from firing. Rather than waiting an indeterminate amount of time for the channel to clear, radio A simply does not fire and behaves as if it as exited the system. In this scenario, radio C's entry has forced radio A out of the group. Additionally note that when radio A rejoins the group it has the possibility of forcing another radio out. Thus the interrupt packet mechanic provides a means for cascading errors. Contrast this to a scenario that discards the interrupt packet. In this new scenario, radio C simply joins the system by first simply receiving any packets that other radios send. This allows radio C to determine an appropriate place for itself in time. However, the other radios in the group are unaware of radio C's presence until it fires for the first time and have likely already split up their time period into slots for each radio. Thus when radio C fires for the first time, it will fire in the slot of another radio. To allow for radios to reconsider their time organization, the radio whose slot has been interrupted ends its slot immediately. This allows radios to join the group with only the penalty of a portion of a slot and gives radios the ability to reorganize.

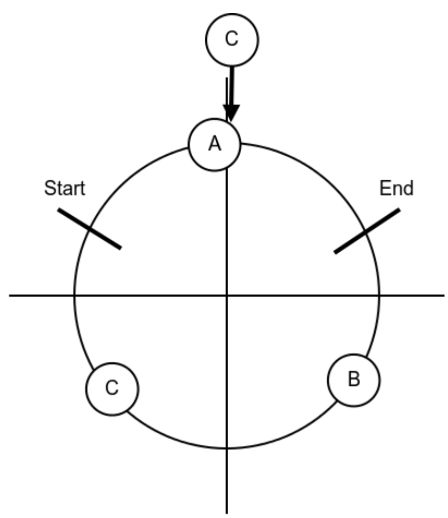


Figure 6.8: Scenario for Resetting Phase

6.4.3 Frequency Aggregation

Frequency aggregation provides the mechanisms necessary to keep a society of CRs together in frequency. Clustering a radio society in frequency allows its members to make the best use of their abilities. Specifically, keeping radios together allows the time behaviors to coordinate the radios and make effective use of the spectrum. Note that for radios to group in frequency, they must be able to recognize their peers.

Given the limitations of the SKIRL platform peer recognition is not simply a matter of sensing. Peer recognition requires platforms to discern identifying details about peers that allow for differentiating them from other entities in the environment. Note that in radio frequency (RF) domains, such identifying details are typically not easily captured. Additionally note that the SKIRL platform is limited to providing RSSI measurements for sensing. Therefore peer recognition through sensing alone is not available with the SKIRL platform.

Instead the system relies on the beacon based social language for peer identification. Members of CR societies are able to recognize peers through reception of beacons. In this case, recognition through language requires radios both to be on the same channel at the time that one radio fires. Thus the frequency aggregation behavior is designed around accomplishing this.

As discussed in Chapter 5, frequency aggregation employs a frequency search pattern and a channel dwell time to accomplish its goals. The frequency search pattern is constructed to provide radios a defined rendezvous channel while still allowing radios to find peers on other channels. The dwell time holds the searching radio on each channel for one beacon period to avoid changing channels before a peer sends a beacon to make its presence known. Note that this dwell time assumes that each radio has a different phase at any given time.

The final factor of the frequency aggregation behavior is its activation conditions. These specify when a radio should begin searching for peers. For this purpose each radio keeps track of active peers on its current channel. This is accomplished by adding radios to a list when a beacon is received from them. Radios are removed from the list if two periods past without receiving a beacon. When the list is empty, the radio assumes that it is alone in the channel. If a beacon is received from a radio already in the list, the count for that radio is reset. This prevents the radios from falsely determining an empty channel if beacons have been prevented due to time avoidance, but does not overly delay appropriate reactions to empty channels.

6.4.4 Frequency Dispersion

Frequency dispersion allows radios to leave channels that are too congested to use. Recall that the time avoidance behavior prevents radio transmissions if a channel is busy. In isolation this could have the effect of a radio waiting an indeterminate time to send a beacon in a highly congested channel. Clearly, this is not ideal. Thus the frequency disperse method selects a new channel for use when a channel becomes overly congested. The system simply moves to the next channel in terms of frequency when frequency dispersion is activated.

Changing frequency has a cost. If radios change frequency at the first sign of interference, societies would rapidly become fragmented. Additionally, reconfiguration is not an instant operation, and further delays the transmission of application data. Therefore, instead of changing frequencies at the first prevented transmission, CRs wait until two consecutive beacons have been prevented. For my purposes, this method allows radios to effectively differentiate truly congested channels from minor interference.

6.4.5 Frequency Learning

Frequency learning allows the CRs to improve their channel selection over time. Recall that both frequency aggregation and frequency dispersion use deterministic methods to select the next channel of operation. Such determinism reduces the challenges of coordinating radios in a society, however, it limits the ability of radios to adapt to a variety of situations. Frequency learning provides that ability.

Frequency learning supports the other frequency behaviors by adjusting the probability of selecting a particular channel. As discussed in Chapter 5 each channel is assigned a visitation probability, p, that is incremented and decremented based on a radio's individual experiences. This visitation probability is the probability that a particular channel is actually visited when selected by one of the other frequency behaviors. Stated differently $1 - p$ is the probability that a channel is skipped and the radio instead moves to the next channel it would otherwise visit. The visitation probability for a channel is reduced by d when radios determine that channel is congested and employ dispersion to leave that channel. The visitation probability is incremented by a small amount i after n beacons to allow radios to forget about stale congestion information.

Herein lies the problem. While frequency learning allows CRs to adapt to their operating environment over time, this adaptation comes at the cost of determinism. The amount of determinism that remains depends on the parameters d, i, and n. The values of these parameters must depend on the hardware's capabilities to handle non-deterministic behavior. Specifically, the sensing capabilities of the hardware determine how well peers motions in frequency may be determined in cases when they can not be predicted. As discussed the sensing capabilities of the SKIRL platform are limited. Thus any non-determinism must be reduced as much as possible.

Thankfully, non-determinism can be effectively removed by setting d and i to 1 and n to 3. This parameter combination has the effect of completely ignoring channels that have determined to be congested for three beacon rounds before forgetting about the congestion, three beacon periods later. This steers radios away from congested channels for three beacon rounds, which provides a balance between adaptability to the environment and management of non-determinism.

6.5 System Implementation

The behaviors of CRs in an artificial society, as discussed above, were implemented in Python. The system architecture is shown in Figure 6.9. Each radio runs its own independent instance of the system shown. The interface to the system provides command line based control and parameter setting to facilitate testing and development. The behavior control module manages the behaviors of the system. The logging subsystem provides the logging capabilities discussed above. The timing subsystem manages the timing of beacons and slot startings and endings. Finally the radio subsystem handles interfacing with the radio hardware.

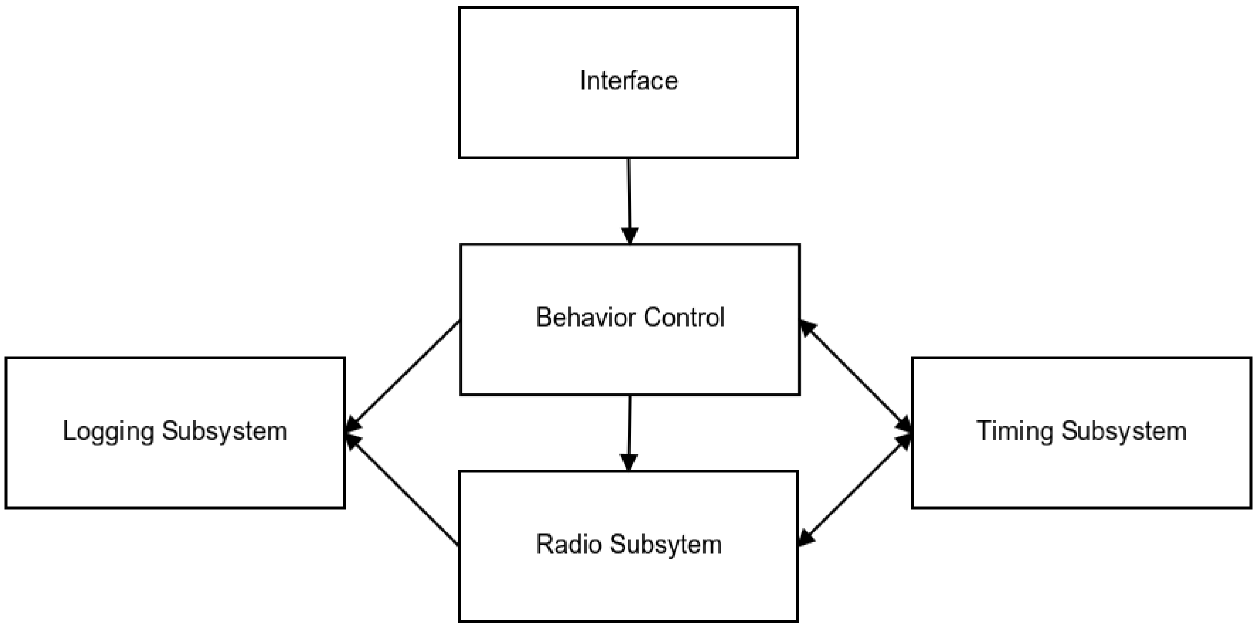


Figure 6.9: System Architecture

The system was implemented as a multi-threaded application. This model allows for the parallel operation of multiple behaviors and increases the responsiveness of the system. Note that special consideration must be made to maintain such responsiveness through use of interrupt based control architecture instead of polling mechanisms. Additionally, threads have been designed with the limited computational power of the platform in mind. All of these considerations allow for successful implementation.

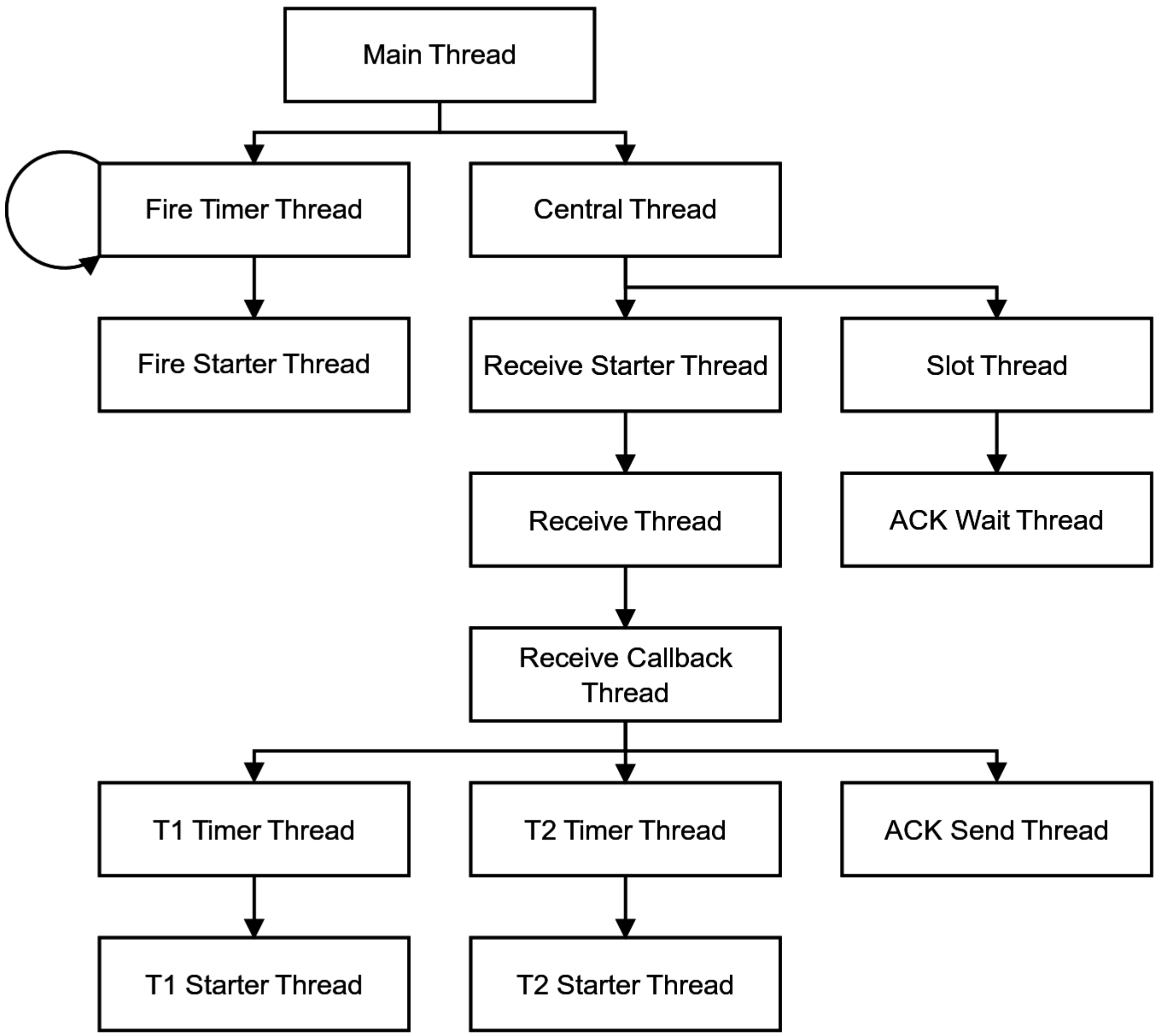


Figure 6.10: System Threads

Figure 6.10 provides an overview of the threads used to realize the architecture shown in Figure 6.9. In this figure, threads spawn those below them and are connected to them. The three timer threads (Fire Timer, T1 Timer, and T2 Timer) wait a specified period of time prior to starting their children threads. These timer threads are the most time critical threads and thus must be as streamlined as possible. The Receive Starter Thread ensures that the radio can receive a beacon at any point. Recall that the SKIRL platform only has a single radio front end and can not transmit and receive at the same time. Thus the Receive Starter Thread exists to keep the radio in receive mode unless actively transmitting. It accomplishes this task through management of receive threads that exist purely to read data from the receive buffer, appropriately time stamp the data, and pass it into the system for comprehension. This comprehension is chiefly handled by the Receive Callback Thread. This thread houses the calculations needed for the time cooperation behavior and spawns helper threads as needed. The T1 and T2 Timer threads are the helper threads that start and end the slot at appropriate times. This slot starting and ending takes the form of activating or deactivating the Slot Thread, which transmits application data while active.

6.6 Implementation Limitations

As discussed throughout, the platform used for implementation affected several design decisions. The platform and those decisions provided several limitations, as well. These limitations are primarily tied to the issue of observability. For the purpose of this prototype implementation, a single hop-network was used, which removes several complicating factors. However, this decision requires each node to be able to observe the transmissions of all of its peers. Thus the geographical size of the network is limited by the frequency and power selected. A slightly more subtle aspect of the observability of peers is tied to the time necessary to take actions. Since the SKIRL platform only has a single front end, radios can not receive and transmit at the same time. Additionally, note that a finite time is required to transition from transmitting to receiving. This means that beacons can not be arbitrarily close together. Primarily this places a minimum requirement on the length of beacon rounds. For the prototype implementation discussed here, this minimum is about 5 seconds per radio. Note that this limitation is tied to limited hardware capabilities.

6.7 Conclusion

This chapter has discussed the implementation of a prototype CR society. The details of this implementation have been covered, including the hardware used, the setting of the implementation, specific behavior details, computational approach, and limitations. The role of each of these factors in the implementation of a CR society has been considered.

This implementation serves to demonstrated the effectiveness of CR societies. The selected hardware platform for implementation displaying the ability of the approach to provide solutions for low complexity systems. The combination of behaviors tailored to solve individual problems based on the available hardware highlights the flexibility to adjust methods to suit available capabilities.

Chapter 7

Evaluation and Analysis

7.1 Introduction

This chapter presents the method used to evaluate the prototype implementation discussed in the previous chapter. This method is based on the approach stressed by Mataric in [24], which examines each behavior in turn before considering compound behaviors. The operation of each behavior discussed in Chapter 6 is evaluated below before examining the emergent properties that arise from interactions between behaviors.

7.2 Evaluation Approach

Recall from Chapter 3 that the work done in robotics shows that evaluation of multi-agent systems provides a non-trivial problem. The mutual dependencies among agents in such systems lead to a complexity that is intractable for situated systems. This is especially true for behavior-based approaches due to the additional complexity that arises from interactions between behaviors within an agent. Recall from Chapter 6 that such interactions required careful consideration to prevent recurring faults.

In order to handle the myriad complexities, roboticists have focused on the evaluation of behavior-based systems through implementation. Implementation provides the only available method to accurately capture all the relevant interactions that occur in realistic environments. Mataric provides a framework for the implementation based evaluation of behavior-based systems. While this framework stresses the importance to exposing agents to realistic noise and interference, it also notes that the environments for such testing must be well controlled in order to understand the operation of the system. Additionally, Mataric suggests the operation of individual behaviors should be confirmed, as much as possible, before any combinations are assessed [24].

As the prototype implementation of a CR society is built upon behavior-based agents, my evaluation of this CR society follows the method suggested by Mataric. That is, each behavior is considered to confirm its operation before compound behaviors are examined. Note that in some cases this strict isolation can not be maintained, as one behavior is used to activate another; in these cases the behaviors must be considered together. Thankfully, the activating behavior is fairly straightforward. The activation of frequency dispersion through consideration of beacons suppressed by time avoidance provides an example of such a case. The evaluation of each behavior primarily focuses on determining correct operation. Thus, the specific evaluation technique and metrics of interest vary for each behavior.

The non-synchronized local clocks of the radios presents a problem to the examination of system behaviors. Determining the time difference between the actions of radios requires a common time. In order to determine this common time, shared events are necessary. Specifically, when a radio transmits a beacon, other radios receiving the beacon provide events that are shared between radios. For the purpose of evaluating the system, the time difference between every two radios is calculated for each beacon sent and received. These differences are then averaged for each radio pair in a given test. This provides the means to shift all events to a common time without altering their relative spacings.

The purpose of this work is the examination of the emergent behavior that arises from the interactions of behaviors and radios. Recall that such interactions occur at the MAC layer in order to preserve the ability to work with other techniques. As such this work is not concerned with the optimization of individual behaviors or even individual radios. Thus the evaluation carried out here focuses on the self-organization and emergent properties and does not attempt to demonstrate individual cognition.

7.3 Time Avoidance

The purpose of the time avoidance behavior is to prevent transmission in busy channels. As discussed in Chapter 6, the prototype system accomplishes this by sensing the channel for 0.1 seconds prior to any transmission. If the RSSI is ever greater than -22.5 dBm the channel is determined to be busy; otherwise the channel is assumed to be clear. Verifying the operation of this behavior is a matter of ensuring that transmissions do not occur when the channel is busy.

This behavior was tested using a single CR and a signal generator. The signal generator served as the source of external interference. The signal generator was human controlled and transmitted with a power of 10 dBm. As the time avoidance behavior applies both to application transmissions and beacon transmissions, either is appropriate for testing. Thus, the radio transmitted 5 beacons with a period of 5 seconds for time avoidance testing.

Two separate interference profiles were used for testing this behavior. Each scenario was tested both with the time avoidance behavior disabled and with it enabled to provide a comparison. In the first the CR was faced with a constant level of interference for 30 seconds. This represents channels with heavy interference. In the second the interference took the form of 7 second long pulses, centered every ten seconds. This situation provides a more intermittent interference. Each scenario was tested without time avoidance and with time avoidance.

Figures 7.1 and 7.2 depict the tests conducted without using the time avoidance behavior. The shaded regions in each figure display the periods of interference and the deltas indicate the positions of beacon transmissions. Clearly the CR in this test completely ignores the interference and transmits beacons without regard for other transmissions. This result exemplifies what the time avoidance behavior is designed to prevent.

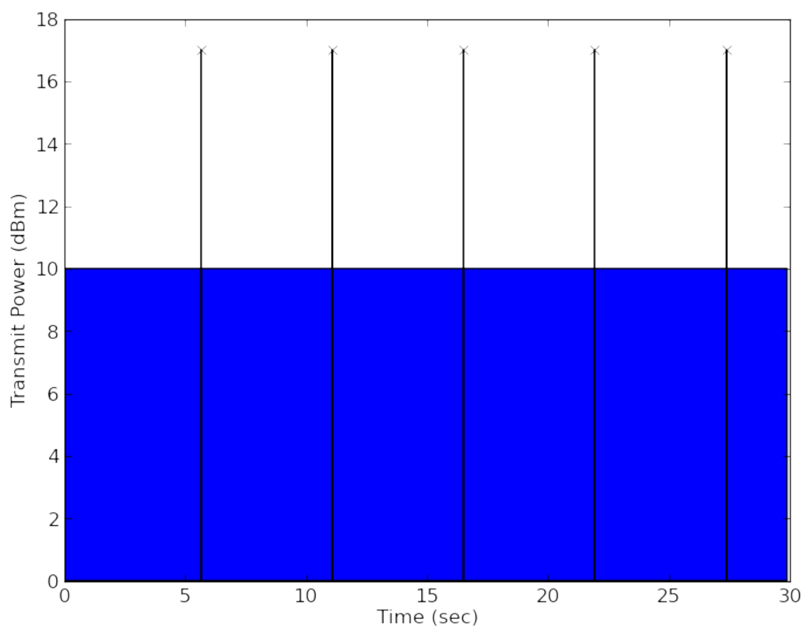


Figure 7.1: Continuous Inference without Time Avoidance

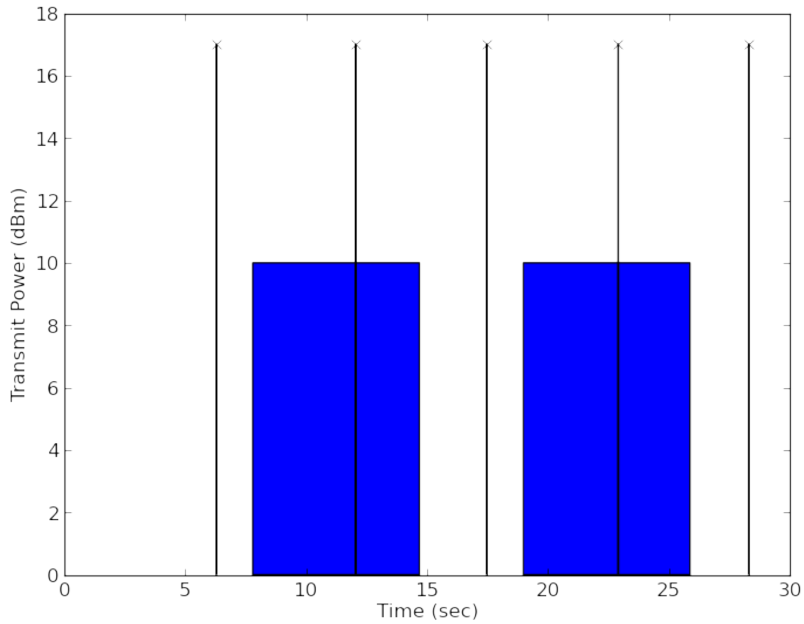


Figure 7.2: Intermittent Inference without Time Avoidance

Figures 7.3 and 7.4 presents the tests conducted using the time avoidance behavior. Again the shading indicates interference and the deltas indicate beacon firings. Note that with the time avoidance behavior, the CR is able to detect and avoid interference. Notice that this ability does not require interference to be continuous.

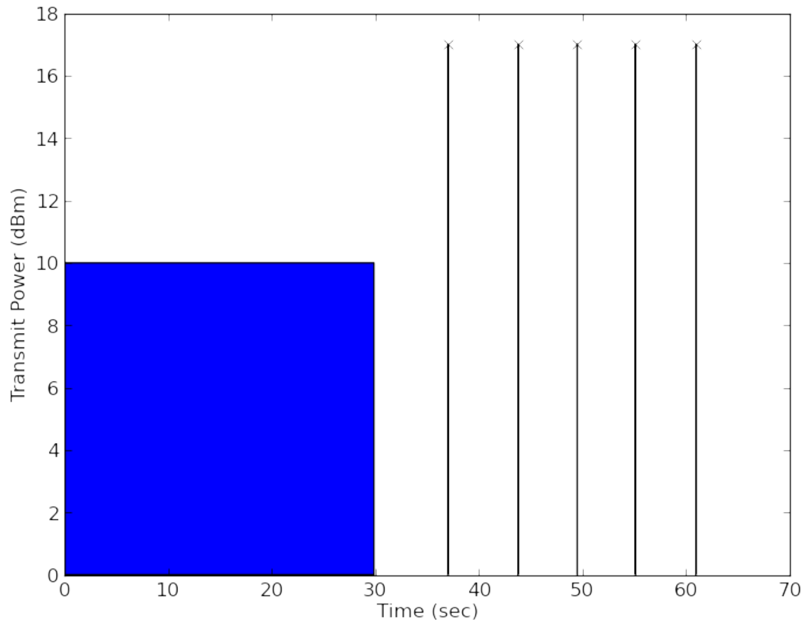


Figure 7.3: Continuous Inference with Time Avoidance

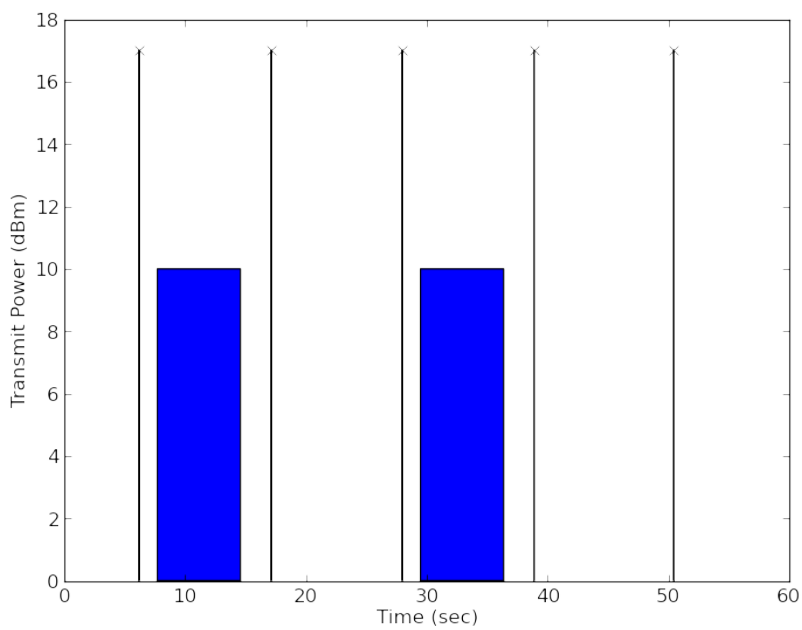


Figure 7.4: Intermittent Inference with Time Avoidance

7.4 Time Flocking

Time cooperation is the most complex time behavior. This complexity primarily arises from the interconnections inherent to the radios. Connections exist in the form of the dependencies between the firing times of the radios, as well as the interactions between actions within a single radio. Additionally, note that while the time cooperation behavior is based on the DESYNC algorithm, this behavior has been tailored according to the need for time avoidance. This dependence makes testing the time cooperation in isolation unfeasible. Thus, the two behaviors, time avoidance and time cooperation, were tested together as the compound time flocking behavior.

Recall that the time cooperation behavior organizes radios within a time period and provides them with non-overlapping slots that utilize the full time period. When this is combined with the time avoidance behavior the result provides radios a method of organizing themselves while avoiding transmissions into a busy channel. While we will see that even this seemingly simple combination of behaviors can result in various emergent properties, testing first focused on simply verifying the radios' ability to organize themselves.

Three metrics were used in order to examine the time organization. The first of these is provided by the authors of [64], who initially proposed the DESYNC algorithm. This metric is the so called average DESYNC error. This is the average deviation from the optimal spacing for a given round. Equation 7.1 provides the mathematical form of this metric. In this equation, n is the number of radios involved, $\overrightarrow{\Delta}^{(k)}$ is the "vector" of radio spacing for a given round k, and $\overrightarrow{\Delta}^{*}$ is the vector of optimal radio spacings T/n, where T is the beacon length. Note that both spacing vectors are of length n because they include all spaces between radios i and $i+1$ as well as the spacing between radio $n-1$ and radio 0. These vectors are actually simple collections of differences, written compactly in vector notation and do not have magnitudes and directions in the usual sense. This metric therefore provides the system's distance from the optimal spacing of radios. The average DESYNC error has units of seconds.

$$error = \frac{1}{n}\|\overrightarrow{\Delta}^{(k)} - \overrightarrow{\Delta}^{*}\| \tag{7.1}$$

The second metric is the average phase deviation for a given round. This metric indicates the average adjustment radios make to their phase for a given round. Equation 7.2 shows the mathematical form for this deviation. Here, n is again the number of radios, and $\overrightarrow{\phi_{tar}}^{(k)}$ is the vector of target phases for the radios in a given round k, as set following the process discussed in Chapter 6. Again, $\overrightarrow{\phi_{tar}}^{(k)}$ is a vector is sense discussed above. Note that the target phase of 1 indicates that the radios will keep the same phase between two firings. Therefore the average phase deviation reveals the distance of the system from a stable organization. The average phase deviation is a dimensionless quantity.

$$dev = \frac{1}{n} \| \overrightarrow{\phi_{tar}}^{(k)} - 1 \| \tag{7.2}$$

The final metric used for examination of the time organization is the average throughput of the radios in a given round. Since the purpose of a radio system is ultimately the transfer of application data, the utility of that system eventually rests on its ability to transfer data. As such the throughput provides an indication of the usefulness of CR societies. Note that because physical layer adaptation is not a focus of this work, the precise value of achieved throughput is less important than its indication that each radio had a clear opportunity to send a message. For this work, all sent packets, as opposed to only acknowledged packets, contributed to the presented throughput. However, it is worth noting that acknowledgments do increase the throughput of the system by preventing radios from waiting for acknowledgments during their slot. Specifically, in order to apply a penalty for missed acknowledges, indicating a packet collision of some kind, the radios were forced to wait for 2 seconds after missing acknowledgments. Throughput is expressed in units of bits per second.

Two scenarios were used with the metrics above to examine the time organization capabilities of the radios. These scenarios are summarized in Table 7.1 The first of these scenarios started radios 2.5 seconds apart on the same channel without external interference. The radios had a beacon period of 30 seconds.These radios then organized themselves in time by communicating through their beacon based social language. The delay between the start commands provided the only initial connection between radios; each radio independently initialized its system and made adjustments based upon locally available information. Due to the highly threaded nature of the implementation, the initialization process does not take exactly the same amount of time for each radio as it depends on the internal thread handling variables of each platform. Thus while the radios were started 2.5 seconds apart, this does not necessarily provide the initial spacing of their beacons. Once started, radios employed the strategies outlined in Chapter 6 to communicate their own time position and size to peers and to use information from their peers to adjust themselves. Each radio sent 20 beacons in each test before shutting down to limit the collected data to manageable amounts.

Table 7.1: Time Organization Scenarios

	# Radios Involved	Beacon Period	Initial Spacing	Interference
Scenario 1	4	30 sec	2.5 sec	None
Scenario 2	4	30 sec	4.5 sec	None

All four radios were tested in this first scenario five times. The metrics discussed above were collected for each of the five tests in order to extend trends. Figures 7.5, 7.6, and 7.8 present the DESYNC error, phase deviation, and throughput results, respectively. Each of these is shown in a given interaction round and interaction rounds are numbered starting with 1. Recall that each radio fires once during an interaction round. Additionally note that the metrics described above are not necessarily defined over every interaction round. Specifically the average phase deviation is the change in phase between an interaction round and the one proceeding it; therefore this metric is not available for a single interaction round.

Figure 7.5 displays the average DESYNC errors for each of the five tests of scenario one with a different color and symbol. These errors show the distance of the system from its optimal time distribution in each interaction round. Thus five symbols, representing the errors from each of the five tests, are shown for each round. Recall that an interaction round consists of one beacon firing from each radio within the network. Also, note that the average DESYNC error is defined above as the L1-norm of the difference between the actual spacing vector and the optimal spacing vector for a given round. Displaying all the results in this way highlights the variations between tests that result from the hardware limitations of the implementation platform.

Examining the average DESYNC error for scenario 1 it clear that the system converges as intended. Although the radios started over 2 seconds away from their ideal time positions, they were typically able to coordinate themselves to achieve a very small error (on the order of 100 milliseconds) by the tenth interaction round. Additionally note that this small error was maintained from then onward, with only small deviations. In particular notice the deviations just after 5 and 10 seconds and just before 20 seconds. Note that in each of these instances one test shows a deviation from the steady error reduction of the other tests. Random errors occur in individual tests as a result of a combination of the hardware limitations, particularly the computational limitations, and the use of old knowledge of peer time position. Recall from Chapter 6 that radios base their positions in the next round upon information from the current round and that all radios make adjustments each round. This has the result that when a radio is delayed in sending a beacon, due primarily to thread lag, this error affects the delayed radio as well as all those that made assumptions about that radio. The overall effect of this on the system is a jump in error for rounds immediately following delays. This thread lag is caused by the thread switching required by the execution of several threads on a single core embedded processor. However, note that the system is able to recover from these sorts of errors. Additionally note that these errors are reduced in magnitude as the system convergences.

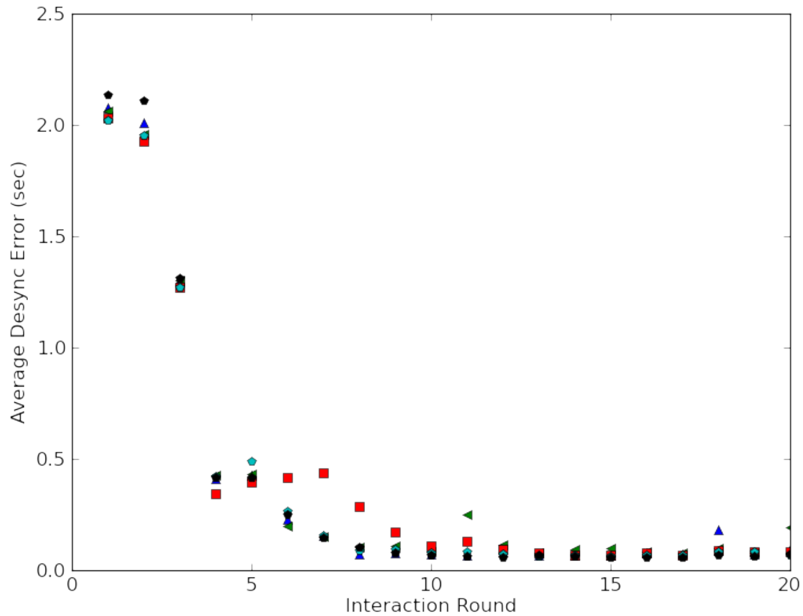


Figure 7.5: Average DESYNC Error for Scenario 1

The average phase deviation is shown in Figure 7.6. This figure follows the same conventions used in Figure 7.9. That is the results for each of the five tests conducted are shown individually with different colors and symbols. Figure 7.6 shows the average magnitude of phase adjustments in each round. Specifically, this figure displays how much radios alter their time position relative to the previous round on average. Note that the radios begin with moves that alter their phase as much as 6%, but, after this large jump, they quickly converge to much smaller alterations. This initial jump comes from the fact that not all radios adjust their phase between round 1 and round 2. The first radio to fire in round 1 does not observe any radios firing prior to its own beacon transmission. Therefore the first radio does not have enough information to make an adjustment and simply maintains it phase, resulting in a phase deviation of 0. As the average phase deviation provides the distance that the overall system jumps in phase between rounds, this lack of change initially reduces the overall deviation. Beyond this initial round, all radios make adjustments in each round. Note that this changing results in an echoing of the DESYNC error results. This arises because the same limitations that cause a DESYNC errors result in the need for small alterations from round to round.

The average phase deviation provides the average assumption error of radios in the system. This assumption error arises from the method for updating a radio's phase. Recall that radios base the calculation of their new phase based upon finding the midpoint of their most recent observations about their peer's phases. Updating in this way results in radios greedily pursuing their optimal phase position based on the assumption that their peers will keep the same phase in the next round. However, since all radios are allowed to update each round, this assumption is not typically correct. The degree to which the assumption that a peer will not change its position is incorrect is the amount that peer changed its phase, its phase deviation. Thanks to the repeated recalculations of time position and size, there is not an explicit penalty for altering one's time characteristics. Rather radios are penalized when they incorrectly assume that their peers will maintain a phase from round to round. Thus the phase deviation provides the penalty due to radio adjustments in each round. Note that as the system moves toward convergence these assumption errors are smaller and smaller, resulting in decreasing obstacles to final convergence.

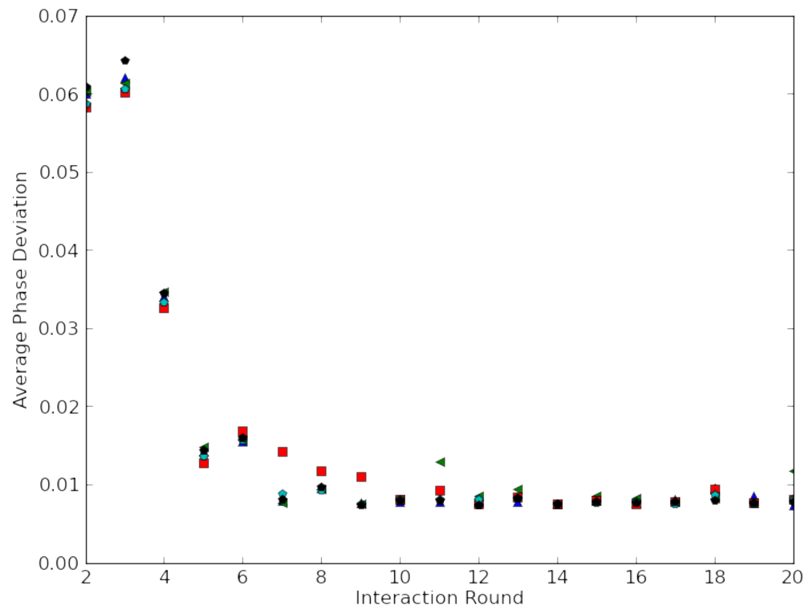


Figure 7.6: Average Phase Deviation for Scenario 1

Figure 7.7 provides an explanation of the method for visualizing throughput. The throughput is determined for each radio in each round by allowing the radio to transmit application data continuously during its slot and dividing the number of bits sent by the duration of the slot. Note that because each test involved four radios, this results in 20 throughput measurements for each interaction round. Rather than showing all twenty results for each round, box plots show statistics for the throughput achieved by the radios in each round. The red line within the boxes shows the median throughput for all radios, across all tests, for the round. The top of the box shows the 3rd quartile of the throughputs. The bottom of the box shows the first quartile. The upper whisker denotes the position of the largest data point within 1.5 multiplied by the inner quartile range (IQR) plus the 3rd quartile, where the IQR is the difference of the 3rd and 1st quartiles. The lower whisker indicates the position of the smallest data point within 1.5 multiplied by the IQR subtracted from the 1st quartile. The whiskers of the boxes show the actual data points where as the boxes show statistical measures, which occasionally results in whiskers being displayed within boxes.

Finally, outlining data points are shown as crosses. This method of displaying throughput provides information about the typical throughput levels that radios can expect as a result of their progress toward convergence. Note that because all radios are involved, smaller spreads indicate an even distribution of throughput among the radios. This indicates that the radios are able to share the available resources most effectively.

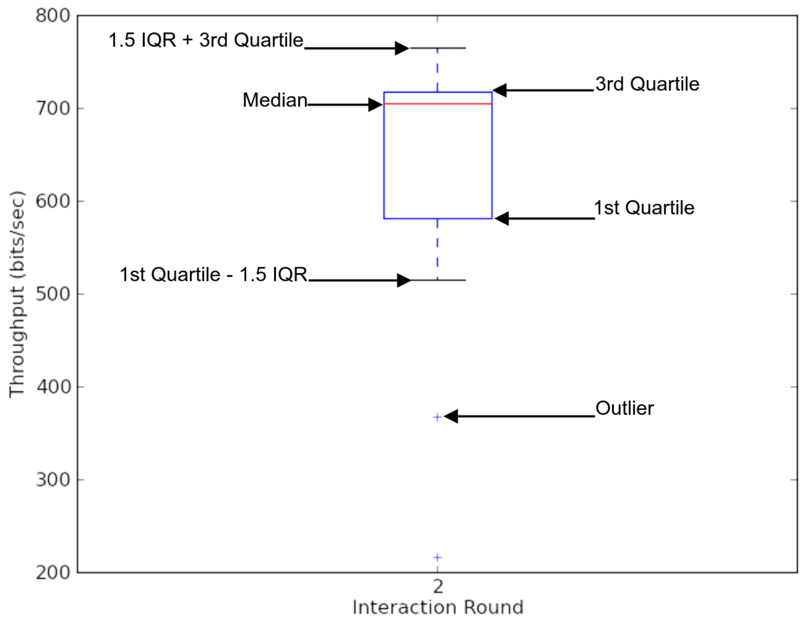


Figure 7.7: Explanation of Throughput Notation

The throughput, shown in Figure 7.8, provides an indication of the utility of time organization. While the throughput is affected by a number of variables, e.g., physical layer parameters and acknowledgments, it does provide an indication of the radio's ability to transfer application data. Figure 7.8 clearly shows that the radios are able to organize themselves in time and evenly divide periods into slots for the transfer of application data. From the perspective of the time flocking behavior, the primary factor controlling the ability to transfer application data is the length of time each radio is allowed to transmit application data. Note that this throughput follows a loose inverse relationship with the DESYNC error and the phase deviation, indicating that as the system reaches steady operation, more opportunity is available for application transmissions and this opportunity is more evenly shared among the radios. Note that distributing time equally among radios provides the greatest opportunity overall to transmit application data, leading to the highest average throughput [64]. Additionally the reduced assumption error that comes with reduced phase deviation allows for more accurate division of time into slots, which helps to prevent collisions that result in missed packets and/or acknowledgments, leading to lower throughput. Note that the small errors resulting from hardware limitations do not greatly affect the typical activated throughput of the radios. However, small slots are possible during convergence.

These outliers result from one radio experiencing a shorten slot during a round, this occurrence is atypical and does not preclude convergence. Recall that radios define their slots based on peer observation from the previous round. Thus radios begin the transmission of application data in round 2. Finally, the increased spread in the final round is due to radios shutting down during this round, as they send their 20th beacon.

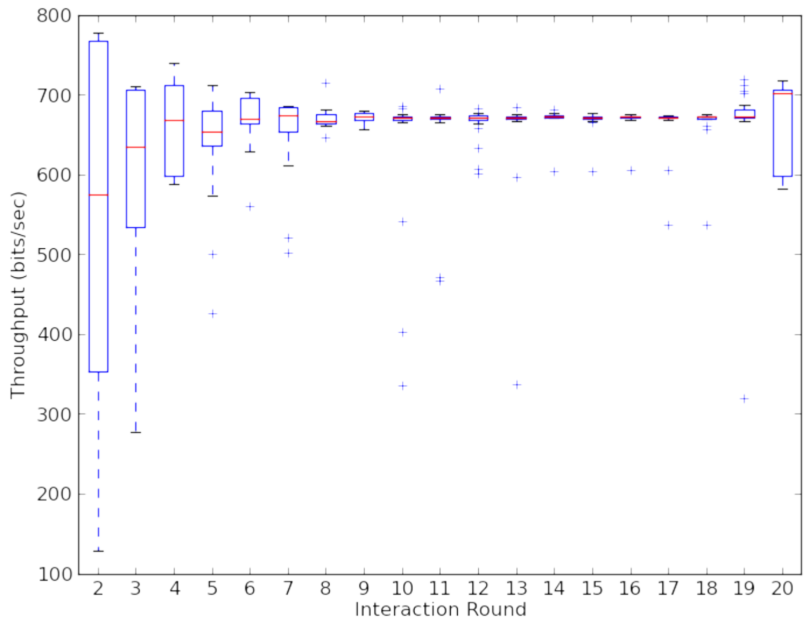


Figure 7.8: Average Throughput for Scenario 1

The second scenario for examining time organization repeats the conditions of the first scenario, with altered initial spacing. This scenario was also tested with five tests. Again four radios were started in a channel without interference, except this time the radios were started 4.5 seconds apart, which is closer to their ideal spacing. This testing in this scenario is displayed using the same methods as above, with various symbols or box plots as appropriate. Recall that the first scenario displayed the ability of the radios to organize themselves in time, but it also showed indications that the combination of old information and limited computational power causes sub-optimal system performance. Specifically the radio's assumptions about their peers and their lack of ability to perfectly achieve their desired time position and size causes adjustments that cascade through the system. Examining a scenario where the radios start closer to their optimal spacing allows for the closer examination of these factors.

Figure 7.9 displays the average DESYNC error for scenario two. Note that random thread delays affected these tests as they did for the tests of scenario 1. The trends of these tests are also the same, but the random events observed here display useful lessons. Firstly, the radios are only starting approximately 1 second away from their optimal spacing, as shown by the initial value for DESYNC error, which alters the vertical axis scaling. Secondly notice that large spike in error experienced by the test shown with a red square late in the tests. This spike results from a combination of system attributes. Specifically, this spike is the manifestation of the jitter in the system, as discussed above. Recall that smaller jitter spikes are present in Figure 7.5, and are caused by computational delay when actually sending a beacon. The spike seen is of an extremely large magnitude, resulting from a long delay in execution. While these sort of errors are rare, they are certainly possible with limited capability hardware. Importantly, note that this error does not defeat the convergence of the system. The radios in the system have made calculations based on assumptions of maintaining non-delayed firing phases, so only the delayed radio is affected in the round of delay. In the following rounds radios are able to readjust. This readjustment allows for quicker convergence because the radios are already in convergence at the time of the spike. This spike draws other radios away from convergence for a single round. Recall that the step size of radios in a single round is purposefully limited by the α parameter, as discussed in Chapter 6.

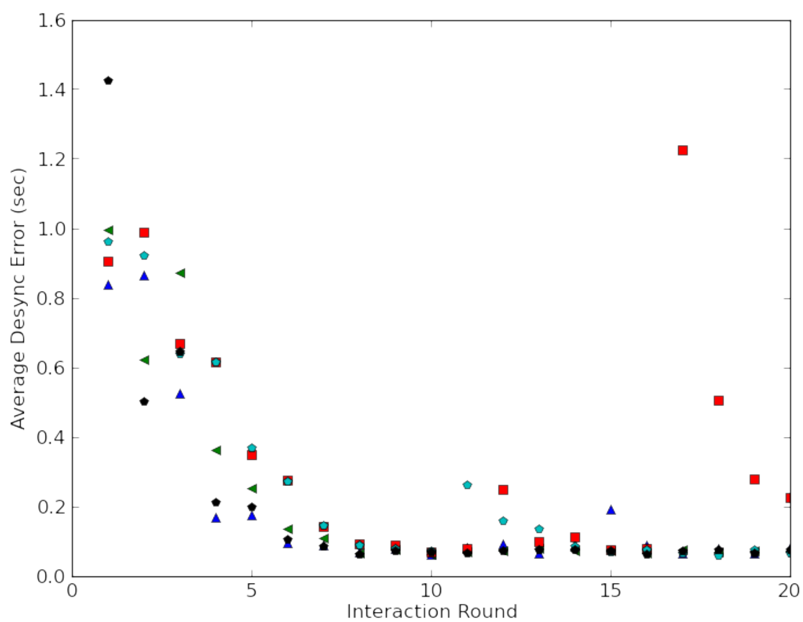


Figure 7.9: Average DESYNC Error for Scenario 2

The phase deviation for scenario two, shown in Figure 7.10, again reiterates the trends present in the DESYNC error. Note the large late spike is again obvious. Here, the magnitude of the phase alterations is of particular interest, as the radios make some of the largest alterations due to the later error. This late delay error causes large assumption errors, related entirely to the magnitude of the delay. Additionally note that the round following the initial spike also exhibits large assumption errors. This is again related to the delayed radio but in this case the error arises because the delayed radio does not expect the non-delayed radios to change, as it is not necessarily aware of its own delay. Rather the delayed radio is only made aware of its own error by observing how the rest of the network changes. While recognizing one's own errors could certainly be built into the radios, social language already provides a method for radios to detect such errors. Additionally the robust methodology of the network allows for the recovery from these assumption errors.

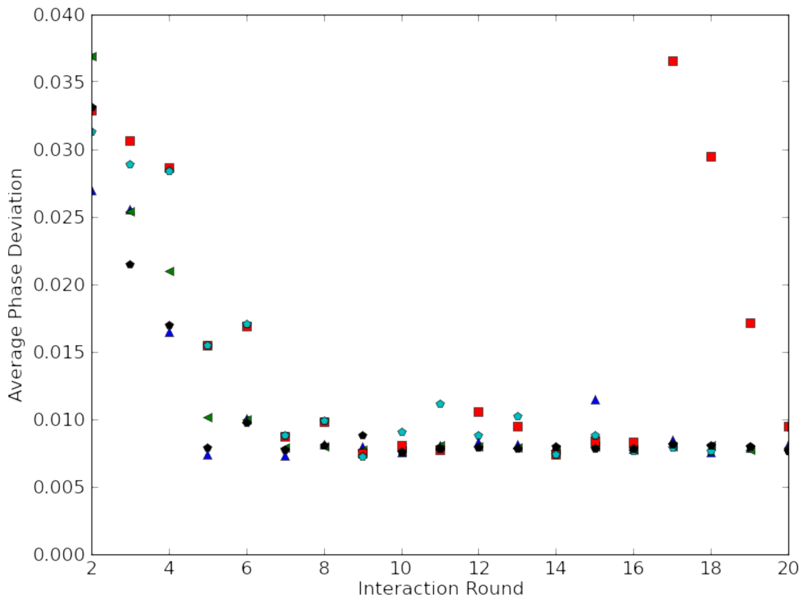


Figure 7.10: Average Phase Deviation for Scenario 2

The throughput for scenario two, shown in Figure 7.11, indicates the impact of errors on the utility of the system. This information is shown using the same format as Figure 7.7. Note that the median throughput exhibits less variation than seen above due to the start closer to convergence. However, the spread of throughputs throughout is larger than seen in the prior scenario. This is related to the increased instances of random computational delays. Statistically, the largest deviations are within 25% of the median throughput for any given round, with most within 12.5%. Note that even in round 17, the round which experienced the large error spike, deviations are within 10% of the median and in round 18 deviations are within 25%. Importantly note that round 19 restricts deviations back to 10%. This indicates that even in the cases of large, rare errors the system is apply to provide opportunity for the transmission of application data and quickly recover to previously experience levels.

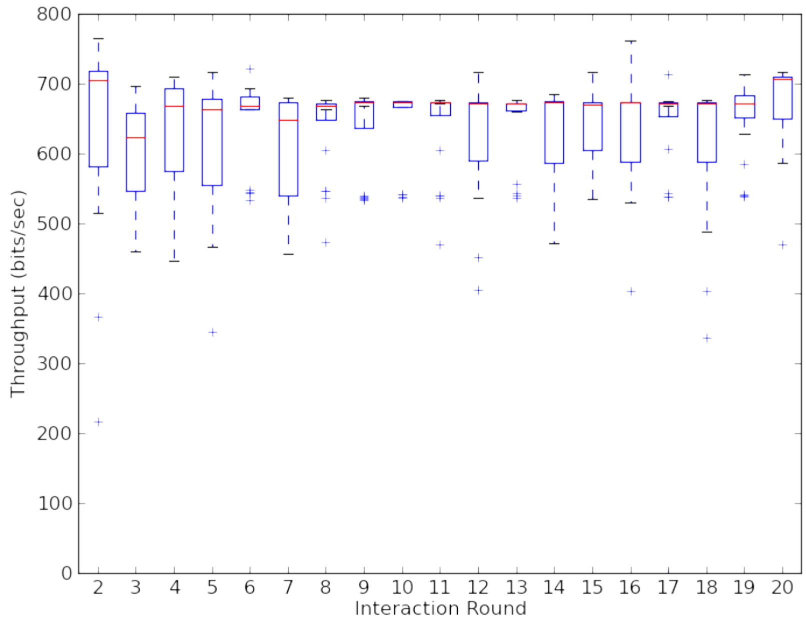


Figure 7.11: Average Throughput for Scenario 2

The tests discussed above confirmed that the system is able to achieve convergence, in spite of hardware limitations, for groups of radios where all radios are activated in the same round. This indicates that a system of radios is able to successfully organize themselves in time when peers can communicate through their social language. However, there are additional scenarios that need to be addressed. Specifically, the entry scenario, in which a CR attempts to join an existing group, is of particular interest.

The entry scenario examines the ability of the system to allow radios to join a preexisting group. Recall that this scenario was discussed in Chapter 6 and required special modification based on the need for time avoidance in the overall system. Testing this scenario consisted of allowing three radios, fuhr, gillies, and joliat, to organize themselves in time and then starting a fourth radio, howe, at some later time later so that it may join the other radios and carve out a place for itself. Each radio fires 23 packets, with a beacon period of 30 seconds, prior to shutting down. This number of packets allows observation of the pertinent radio behavior, but limits the data to reasonable levels for visualization.

A typical test of this scenario is show in Figure 7.12. This figure shows the time organization of the radios in the upper portion and the associated throughputs in the lower portion. Specially, the upper section shows the slots of the radios, as collections of a radio's individual symbol, and the firing of radios, as a vertical line topped by a radio's symbol. The bottom portion provides also shows the slot length as a line of a radio's symbol, but provides the throughput averaged over that slot as well. The radio's symbols are provided in Table 7.2. This method of displaying radio behavior provides the clearest indication of the evolution of a radio society in time.

Table 7.2: Radio Symbols

fuhr	Black X
gillies	Green Square
joliat	Blue Triangle
howe	Red Circle

Figure 7.12 displays the start of fuhr, gillies, and joliat and the establishment of their slots prior to howe's entry. This entry occurs at roughly 110 seconds, as indicated. After its entry, howe listens to the beacons of its peers, gathering information about their firings. Our entrant then fires its first beacon at roughly 140 seconds, just before joliat fires its own beacon. As shown, this cuts joliat's slot short, as intended and allows for all the radios to adjust to the new radio in the system. The radios then proceed to organize themselves, with consideration of their new peer, according to the results shown above. Notice that the radios do not necessarily reach their final convergence before shutdown begins. This test does not focus on convergence because its purpose is to maintain visibility of the entry sequence; however it is still clear that the radios are able to organize themselves in time once a new radio joins an existing system.

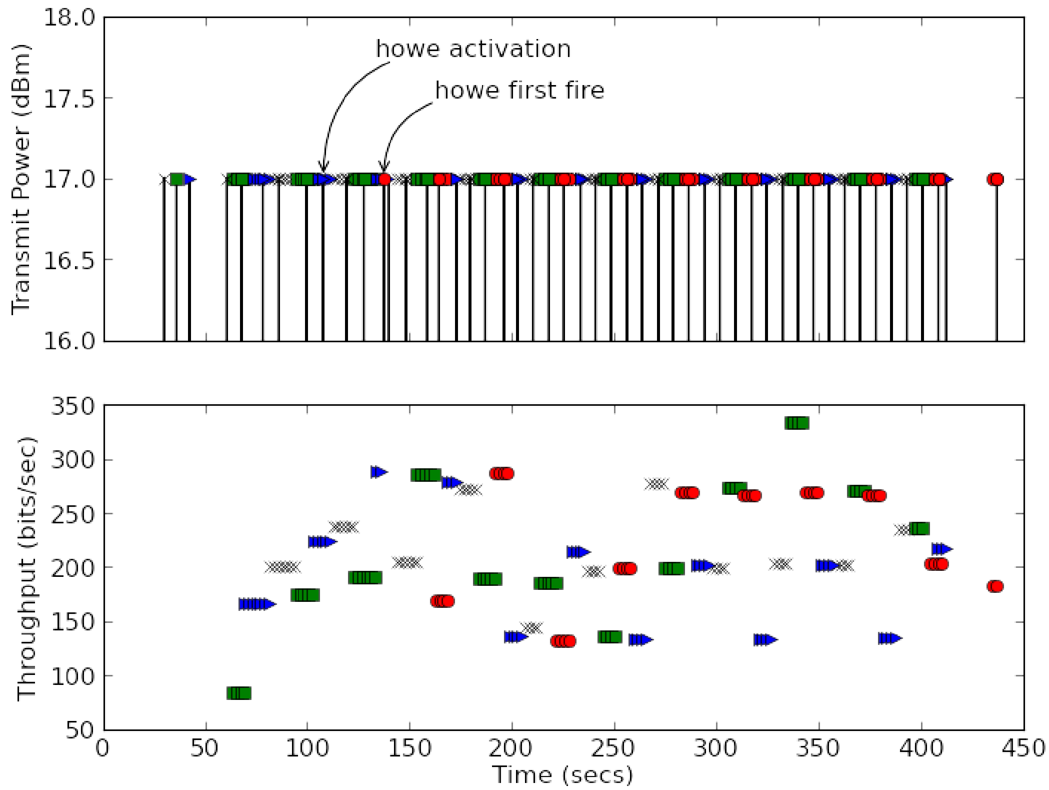


Figure 7.12: Entry Scenario

The mechanisms that allow for radio entry are the primary focus for these tests. Note that implementation's focus on maintaining the ability to receive information allowed joliat to receive howe's beacon and end its own slot to allow for necessary time position adjustments. In this situation joliat ended its slot in recognition that its assumptions about the firing positions of its peers were incorrect. This recognition and response to an assumption error large enough to allow for a slot to overlap the firing of a neighbor gives the system a chance to recover from errors. While the beacons were not close enough together to trigger the time avoidance behavior, the system's focus on listening for social language messages and appropriately responding to those messages allows for new radios to join existing groups, as intended.

In fact, the listening aspect of time avoidance merely contributes to the mechanisms that allow new nodes to join the system. To examine this, the priorities of the system were slightly altered. Recall that time avoidance, as discussed in Chapter 6 mandates that radios listen to the channel before any transmissions, including application transmissions. In addition, to ensure reception, radios listen for acknowledgments after transmitting application messages. To accomplish this, a CR keeps its front end in receive mode until just before transmitting and then immediately returns to receive mode. The result of this action is that the radios have a focus on listening for the entry of new peers or possible interference. This individual focus results in an emergent sharing that allow for new nodes to enter an existing group, as shown above.

Altering the individual focus of the radios, even slightly, changes the emergent properties of the overall system. For example, if the radios are not required to reenter receive mode for acknowledgments directly following each transmission, they send application messages more aggressively, no longer hampered by the delays of starting receive, processing acknowledgment packets, and stopping receive prior to transmission. Instead the radios can stop receiving, and then transmit throughout their entire slot. This obvious benefit of higher throughput comes at the price of individual focus on listening and therefore the emergent time sharing.

Figure 7.13 displays the results of a brief test that exemplifies the effect of altering individual radio focus on overall system behavior. This test was conducted with the same conditions as that for Figure 7.12, except for the alterations to application data sending policy as discussed above. Note that the alterations do not affect the time avoidance behavior. Thus, in this case while howe still enters the system at 120 seconds, it is prevented from firing until joliat fires its final beacon, just before joliat's shut down. The individual radio focus of maximally transmitting application data during its slot results in radios forming a defensive phalanx in time. Note that the radios that start earlier (fuhr, gillies, and joliat) still organize themselves in time. Its only when howe attempts to join that the difference is evident.

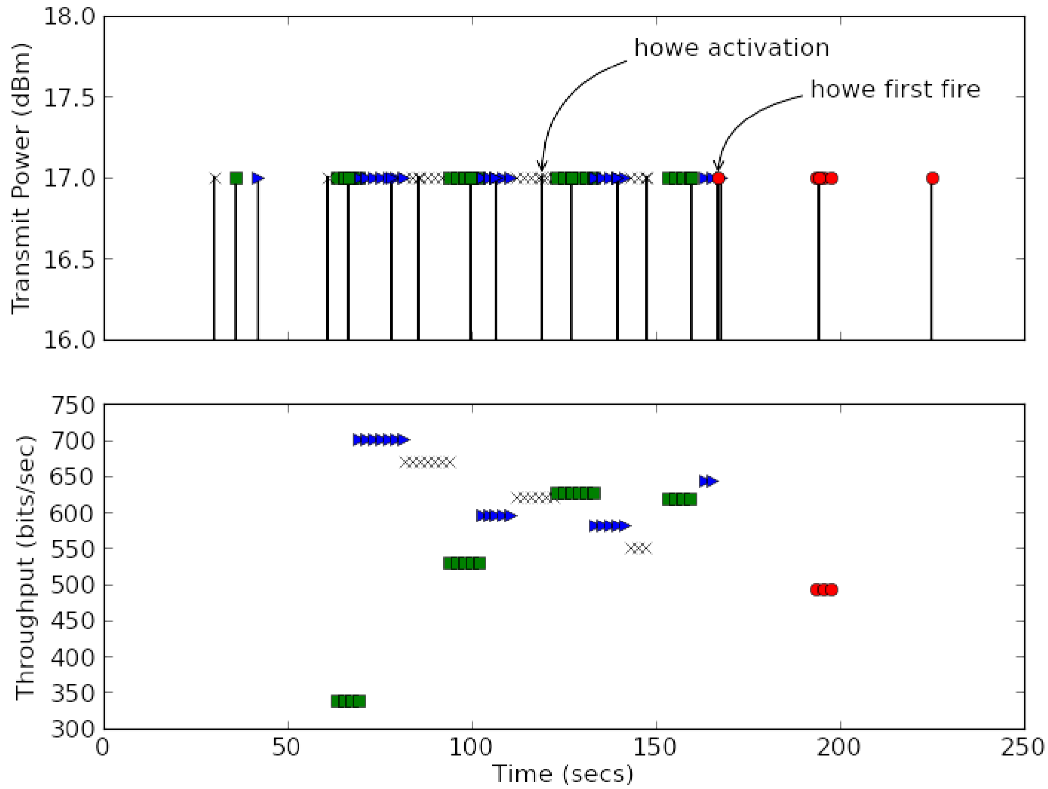


Figure 7.13: Lock Out Scenario

Focusing on the mechanics that lend to this time phalanx, the combination of the aggressive slot usage and the time avoidance behavior cause this particular emergence. Since the radios don't establish slots prior to determining their neighbor's positions, the initial time organization is not affected. Instead once the radios form slots they aggressively use them. This means that a new radio, attempting to join the system will be prevented from firing its own beacon because the channel is already busy with application data, as shown in Figure 7.13. Note that this alteration provides an indication of the emergence that results from the interactions between the behavior of various radios.

The final aspect of time flocking that must be examined is the effect of interference on the ability of the a society to organize itself in time. In order to examine this four radios were tasked with organizing themselves in time while faced with various interference profiles. Three interference profiles were applied during these tests. First the radio society faced interference for the first 75 seconds of its operation. Next the radio society was interfered with for the duration of a full round after they had established a time division that allowed for application transmissions. Finally the beacon of a single radio was interfered with after the radios had established a time organization.

Figure 7.14 displays the behavior of the radios when faced with the first interference profile.

Note that the time avoidance behavior of the radios prevents them from transmitting during the initial period of interference. Since the radios do not transmit beacons, no social language based coordination can take place and the radios simply delay the start of their organization in time. This straightforward situation serves to confirm the expected operation of the time avoidance behavior. Since time cooperation has not yet determined slots for the radios, time avoidance simply dominates the radios' behavior and suppresses beacons.

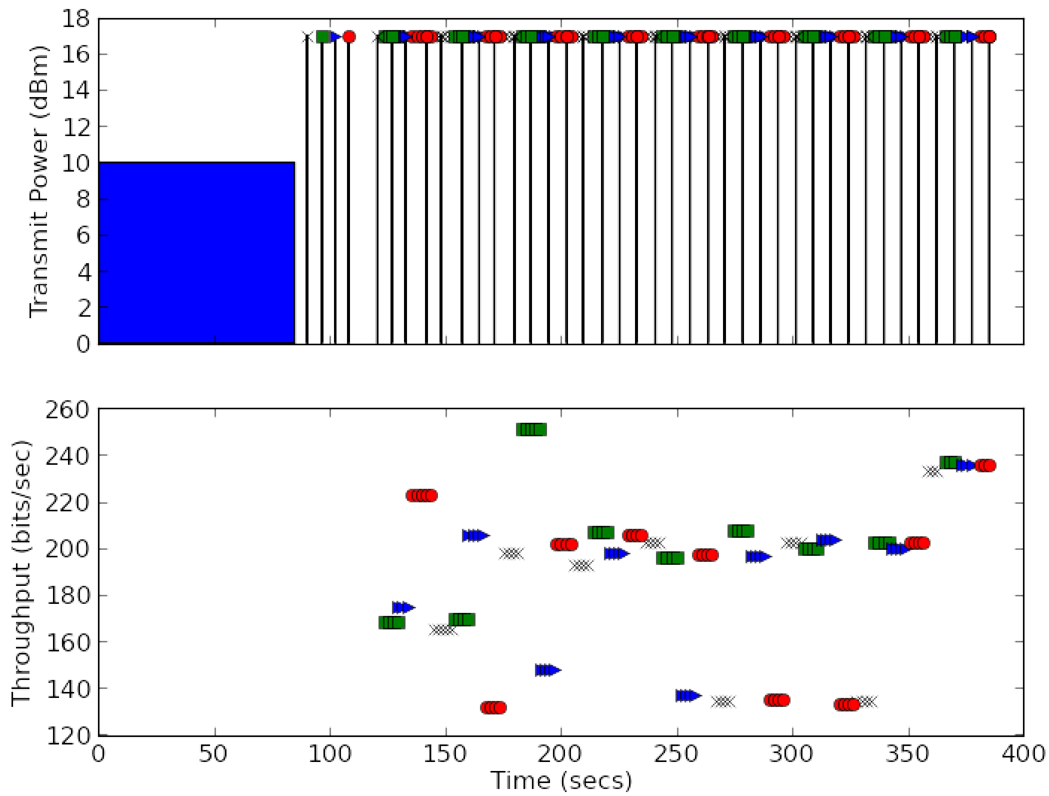


Figure 7.14: Scenario 4: Constant Interference

Figure 7.14 presents the results when radios are faced with one round long interference after establishing a time organization. Here the radios are effectively reset. Notice that the slots of the radios are still active during the interference, but the throughput for this time is zero for each radio because time avoidance is preventing application transmissions. Recall that the radios determine their time position and size based on the reception of neighbors' beacons. Since the interference prevents the transmissions of neighbor beacons, the radios are unable to determine appropriate slots and thus must reorganize themselves once the interference ends. Notice, however, that radios are able to recover after a setup round and return to normal operation once the interference is gone. Finally, if this interference were longer, the behavior would not noticeably change, i.e., once the radios are forced to reorganize themselves in time, waiting longer to perform this reorganization does not tangibly change the system's behavior when only the time flocking behavior is active.

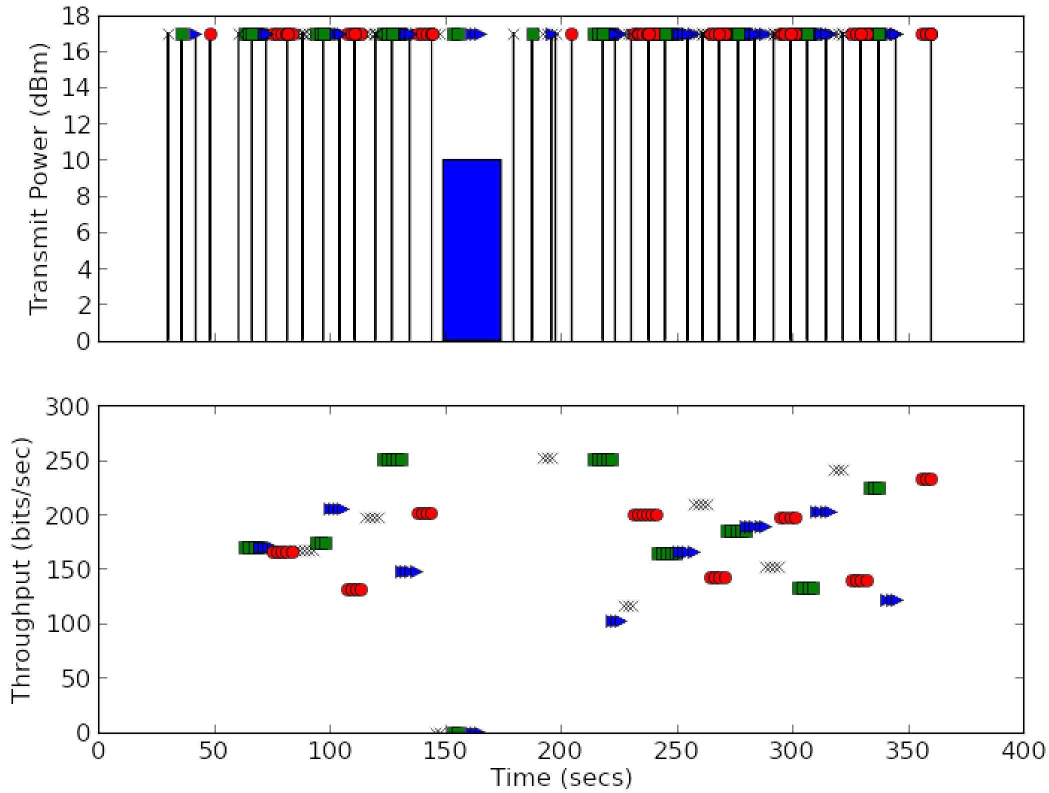


Figure 7.15: Scenario 4: 1 Round of Interference

Figure 7.15 depicts system behavior when one radio is interfered with after a time organization has been established. This scenario is similar to the entry scenario, however here the situation is easier. Notice that the interference has prevented gillies from transmitting application data and its beacon after the group has established slots for each radio. This causes gillies to effectively exit this system. Once this happens the other radios expand in time to absorb the empty space left from gillies leaving. Recall from Chapter 6, that the factor α limits the degree to which this space can be fully absorbed. This limiting is especially helpful when gillies fires its next beacon one round after it was forced to exit. Specifically note that in one rounds' time only gillies' neighbors, fuhr and joliat, are affected by gillies' absence. Both of these radios are pulled into the void left by gillies; note their close spacing directly before 200 seconds. This motion in time would allow fuhr and joliat to expand their own slots by the time left by gillies. This additional time would later be shared with howe as the system reorganized itself. However, gillies reenters the system before such reorganization can take place. Specifically gillies enters the system after joliat's beacon, as joliat moved toward fuhr to account for new space, but during joliat's slot. Thus, as discussed above, joliat ends its slot in recognition of the error in assuming that gillies had truly exited the group, and the system proceeds. Thus the memory of gillies in the initial organization of the system maintains a space, albeit with different phase neighbors, for gillies even one round after gillies exits the system.

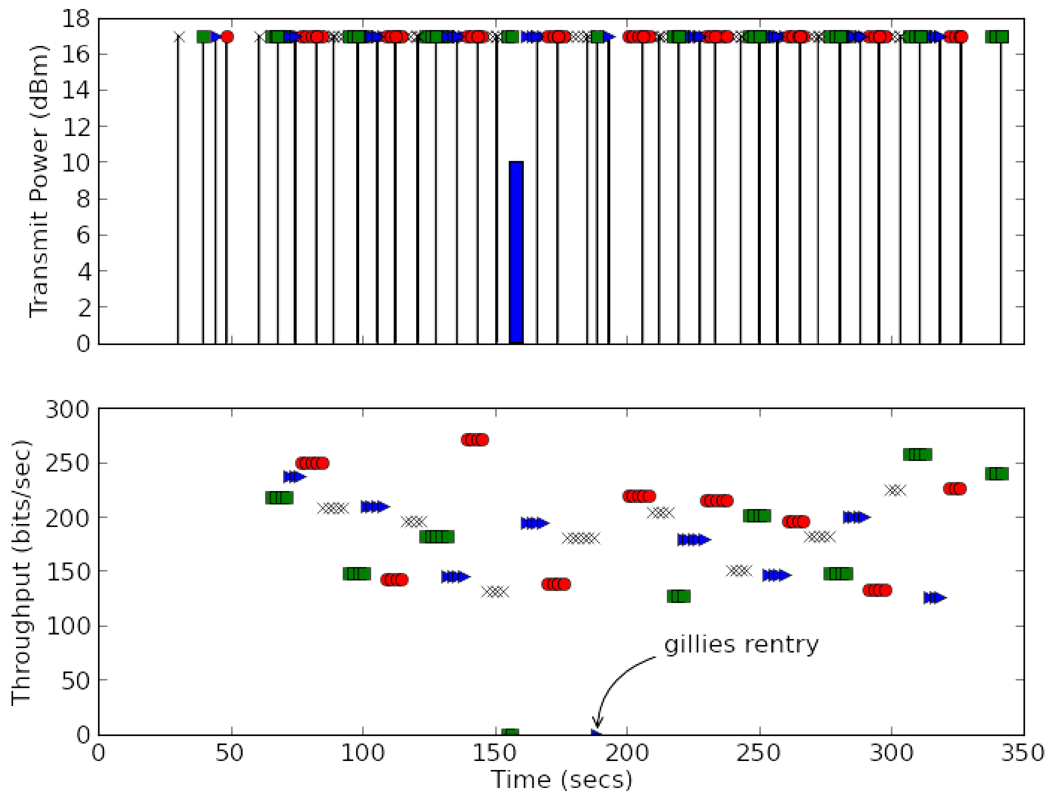


Figure 7.16: Scenario 4: 1 Beacon of Interference

Interference with a single radio in a system demonstrates the system's abilities for social learning. In this time flocking system the time relative positions and sizes represent the knowledge that the system has gained over time about how to organize radios in time. Notice that this knowledge exists in the form of the spacing between radios, rather than within an radios in particular. Additionally, note that the system is able to make use of this information even when a radio exits the system, i.e., the system remembers how to organize four radios in time even after gillies exits the system. However, note that the system will not remember this information forever and immediately started taking action to forget information about four radios in favor of information about three radios once gillies exited the system. Additionally, note that if all radio radios exit the system, no information is maintained, as there are no interactions in which the information may be stored, as seen in the case interference that lasts for a full round.

The above tests have examined the time flocking behavior from a number of angles to confirm its operation. The importance of behavior interactions and the effect of individual action on the emergent nature of the system have been examined. Additionally, the social learning enabled by the mechanism of time flocking has been discussed. Time flocking has been shown to provide a CR the ability to flexibly organize itself in time.

7.5 Frequency Aggregation

Frequency aggregation is meant to group radios in frequency. Recall that this behavior is based on a search with a dwell time through channels. The search pattern focuses on bringing together radios in a so called rendezvous channel by using the social language to make radios observable.

Examining the operation of this behavior simply requires ensuring that the radios search through channels as expected and ensuring the ability of radios to group in frequency. These attributes provide the necessary abilities to radios in a society.

Figure 7.17 displays the search pattern and dwell of the frequency aggregation behavior by allowing fuhr to attempt to find its peers. In this figure, fuhr's entry into a particular channel is marked by fuhr's symbol and fuhr's beacons are marked by a vertical line topped with fuhr's symbol. Notice that, to ensure that peers are not missed, fuhr stays in each channel for two beacon periods. Additionally note that fuhr returns to the rendezvous channel between other channels to increase the chance of meeting peers there. The search pattern would loop through all 21 channels before continuing, if fuhr were allowed to operate long enough to do so. Table 7.3 provides the frequencies for each channel. Finally note that the initial two beacons are sent on channel 10 simply because this has been set as fuhr's initial channel and not necessarily because of frequency aggregation.

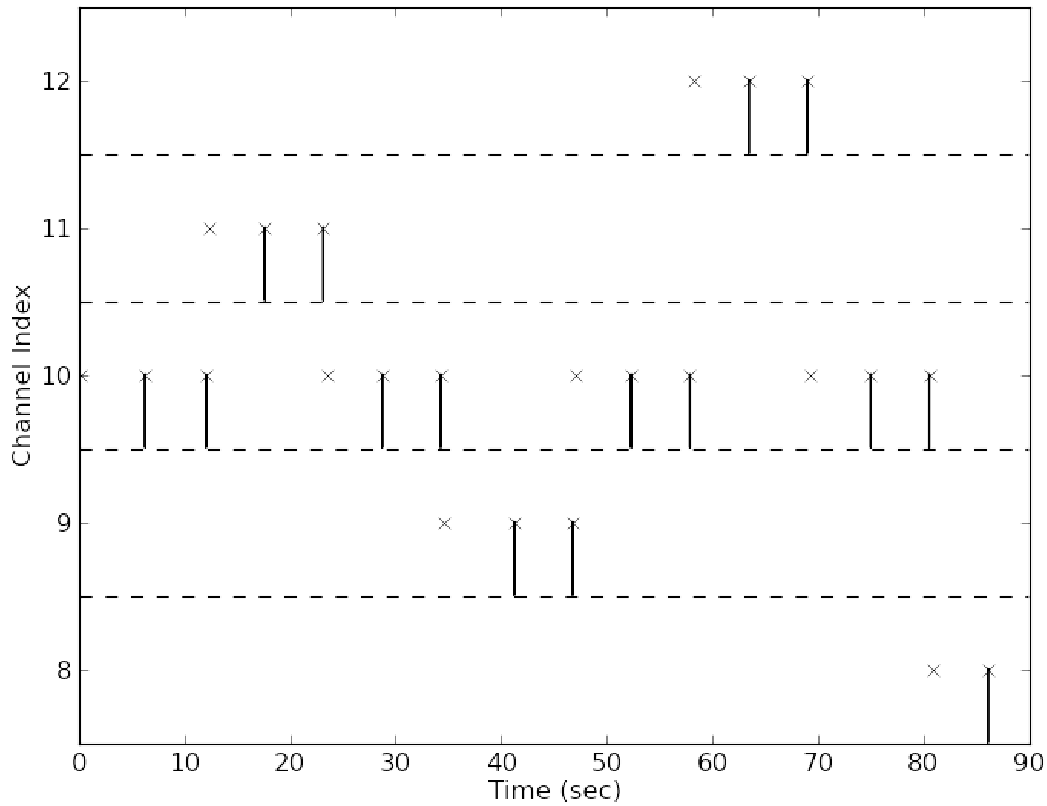


Figure 7.17: Frequency Aggregation Search

Table 7.3: Channel Frequencies

Channel Index	Center Frequency (MHz)
0	426.5
1	427.25
2	428.0
3	428.75
4	429.5
5	430.25
6	431.0
7	431.75
8	432.5
9	433.25
10	434.0
11	434.75
12	435.5
13	436.25
14	437.0
15	437.75
16	438.5
17	439.25
18	440.0
19	440.75
20	441.5

Figure 7.18 shows how frequency aggregation groups radios in frequency. Note that this figure displays actions in the same manner as Figure 7.17. Here four radios are initially spread across several channels. These radios apply the frequency aggregation behavior in order to group together on channel 11 and once grouped they no longer have need to search for peers. Notice that these radios did not group on the intended rendezvous channel, channel 10, because they found one another before reaching that point in the search pattern. Frequency aggregation, therefore provides a straightforward method of grouping radios.

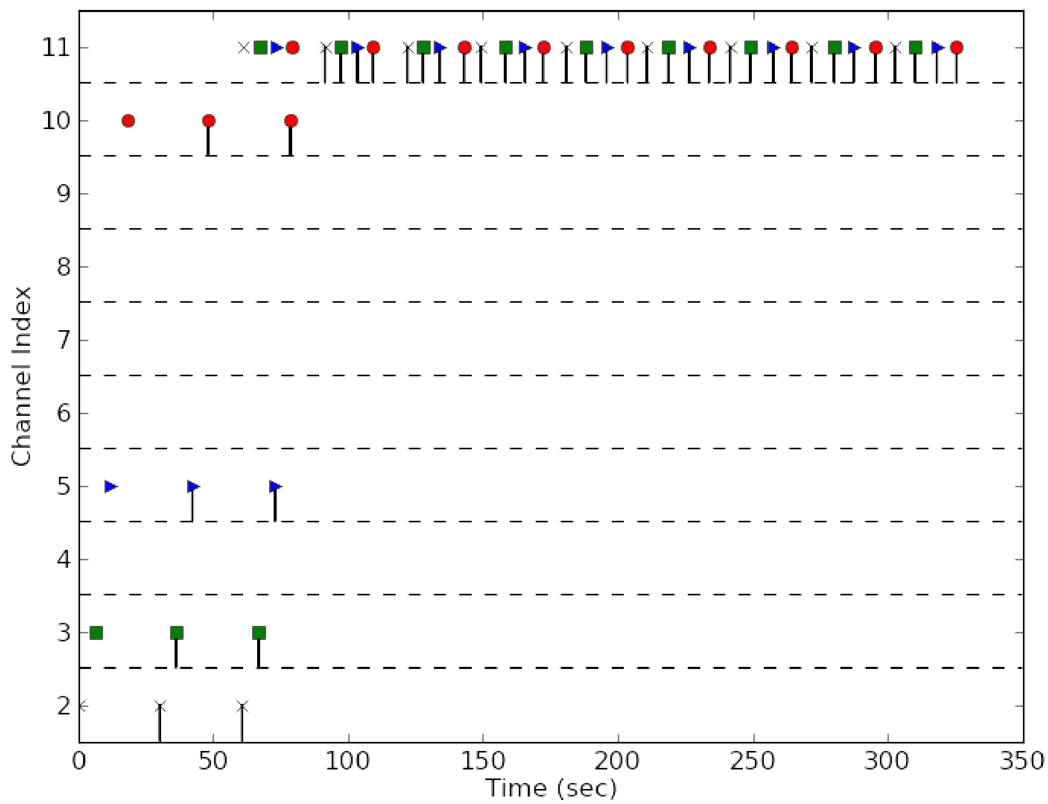


Figure 7.18: Frequency Aggregation Testing

7.6 Frequency Dispersion

Frequency dispersion is the complement to frequency aggregation. Where aggregation groups radios in frequency, dispersion provides a means for radios to leave congested channels. Note that while this behavior could spread radios in frequency when too many members of the same society are attempting to use the same channel, the behavior's primary purpose in the prototype implementation is leaving channels that are experiencing external interference. As such, this behavior is very straightforward in that radios simply move to the next higher channel when experiencing interference that lasts for two beacon periods. As show above CR societies can benefit from societal learning in cases when all society members are subject to interference, and thus it is better to stay on a particular channel to benefit from this. However, when a radio experiences interference that lasts for two beacon periods, it assumes that all members of society have been forced out of the system and there is no longer any social memory. In these cases it is better for radios to simply move to a new channel in order to avoid continuing interference.

Figure 7.19 depicts the behavior of radios experiencing long term interference. In this situation each radio independently makes the assumption that no members of its society are still active and there is no benefit to remaining on the current channel. Additionally the radios assume that the interference will continue long enough on their current channel that changing to a new channel is worth the reconfiguration cost. Thus each radio simply moves to the next channel. Experiencing no interference on their new channel, the radios do not change again.

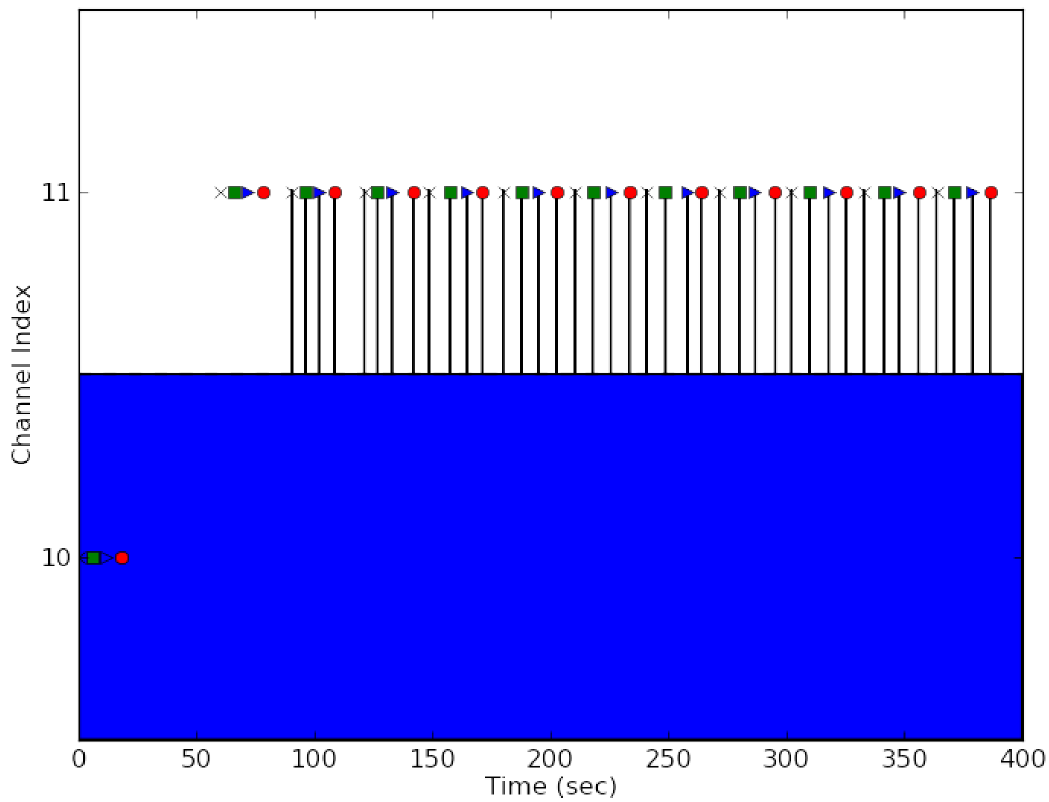


Figure 7.19: Frequency Dispersion with Constant Interference

Figure 7.20 presents the behavior of radios under the influence of shorter term interference. In this case the radios only experience interference that lasts for a single round. Here each radio determines changing channels is not a good decision. They base this on the built in knowledge that they have time flocking behaviors that can handle interference. As shown above, time flocking can apply social learning to recover from some interference. Frequency dispersion therefore avoids activation unless it is confident that learning based recovery is unavailable. Recall that each radio makes decisions based on local information and does not know the current time spacing between itself and its peers, as all radios adjust their positions independently; thus the radio has no way of knowing that a full round has experienced interference until two of its own beacons have consecutively been suppressed.

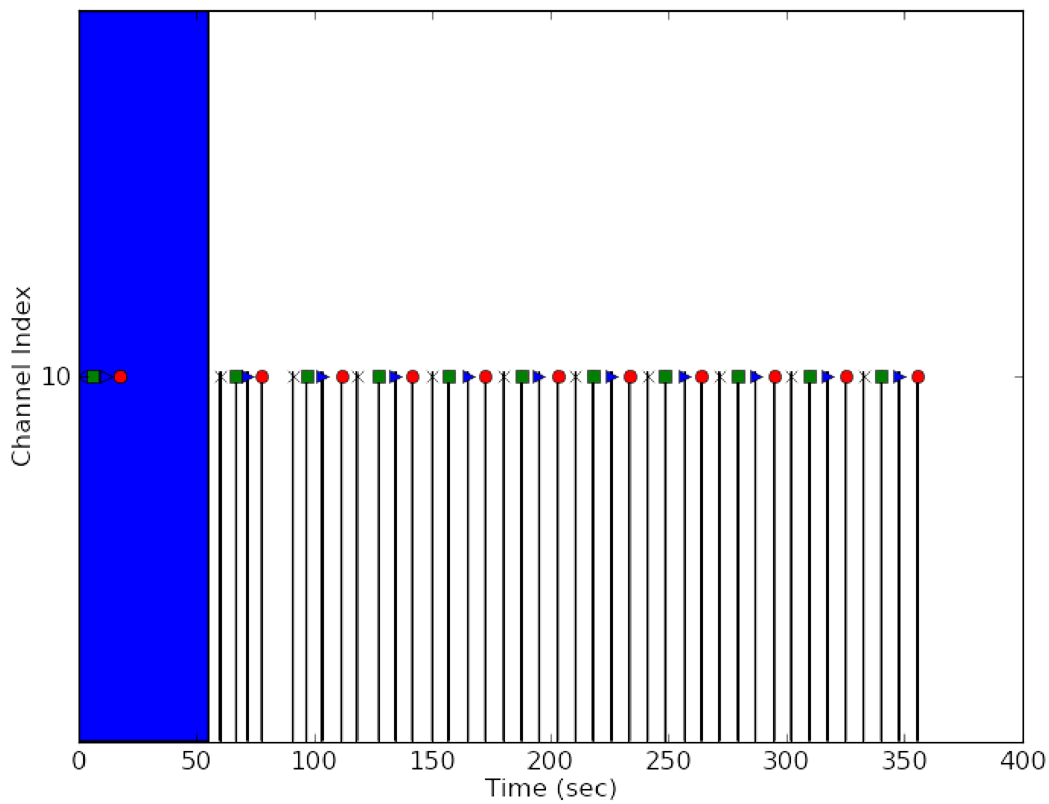


Figure 7.20: Frequency Dispersion with 1 Round of Interference

While frequency flocking is not a sophisticated behavior, it displays some important attributes. Recall that the focus of the prototype implementation of CR societies is an examination of the methods involved. Thus, while more sophisticated methods of determining that a radio should change frequency are available, they obfuscate the important interactions between frequency dispersion and the learning that occurs inherently in time flocking. We will see that the simple methods for channel selection serve to highlight the important aspects of interaction with frequency aggregation below. Thus the simplicity of the frequency dispersion behavior allows for examination of the benefits that may be gained from combining behaviors.

7.7 Frequency Flocking

Frequency flocking is the composite behavior built as the combination of frequency aggregation, frequency dispersion, and frequency learning. While frequency aggregation and frequency dispersion were discussed on their own above, frequency learning may only really be examined in combination with the other behaviors. This is because frequency learning is a support behavior that adjusts the selection of channels in frequency dispersion and frequency aggregation. Note that this behavior is not combined with either of the other frequency behaviors in order to maintain their straightforward nature, the value of which has already been seen with frequency dispersion. Thus frequency learning is examined here through its impact on the other frequency behaviors.

At this point its is necessary to examine the interaction between frequency dispersion and frequency aggregation. Recall that aggregation occurs when no beacons are present for two beacon periods. However, in the case that a system experiences interference for two beacons, it will not have have received beacons for two periods as well. In these cases dispersion is always allowed to act prior to aggregation. This policy maintains the different focuses of these frequency changing behaviors, which allow for appropriate action to be taken in each situation, (empty channels and congested channels). The prototype implementation presented here does not take advantage of the different activation situation, rather, the separation primarily serves to maintain clarity.

Frequency learning can be examined by observing radio behavior when frequency dispersion and frequency aggregation are activated in sequence. Figure 7.21 displays fuhr taking such behavior. Recall that frequency learning prevents radios from returning to channels that have been learned to be congested. Channels are considered congested when frequency dispersion is activated. Thus when fuhr leaves channel 10 with frequency dispersion, it determines the channel to be congested. It next activates frequency aggregation, which returns it to channel 11. On the next activation of frequency aggregation fuhr would typically travel to channel 10, but instead it travels to channel 9. This is because it has already learned that channel 10 is congested. Thus frequency learning provides the means for a radio to tailor its frequency selection process over time.

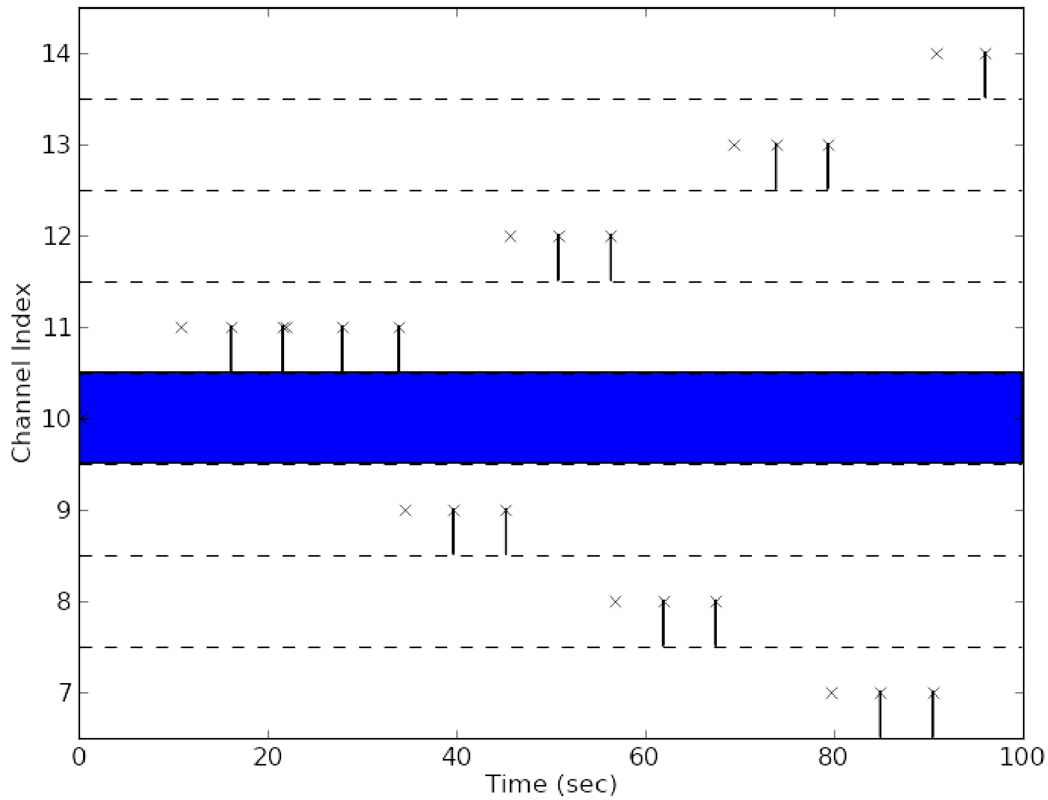


Figure 7.21: Frequency Learning

All together frequency flocking provides the means for a society of radios to navigate through frequency. While this it does not directly control the transmission of application data, frequency flocking supports these transmissions by determining appropriate frequencies for them to take place. Note that this work focuses on sharing channels in order to benefit from time flocking behaviors, but this is not strictly necessary in CR societies. Rather frequency flocking focuses on supporting application data transfer in a manner conducive to overall system goals.

Figure 7.22 provides an example of frequency grouping. In this scenario the individual radios are initially spread throughout channels. In their attempt to find and coordinate with one another they are faced with interference. Appropriate reaction to this interference allows the radios to find each other and coordinate themselves.

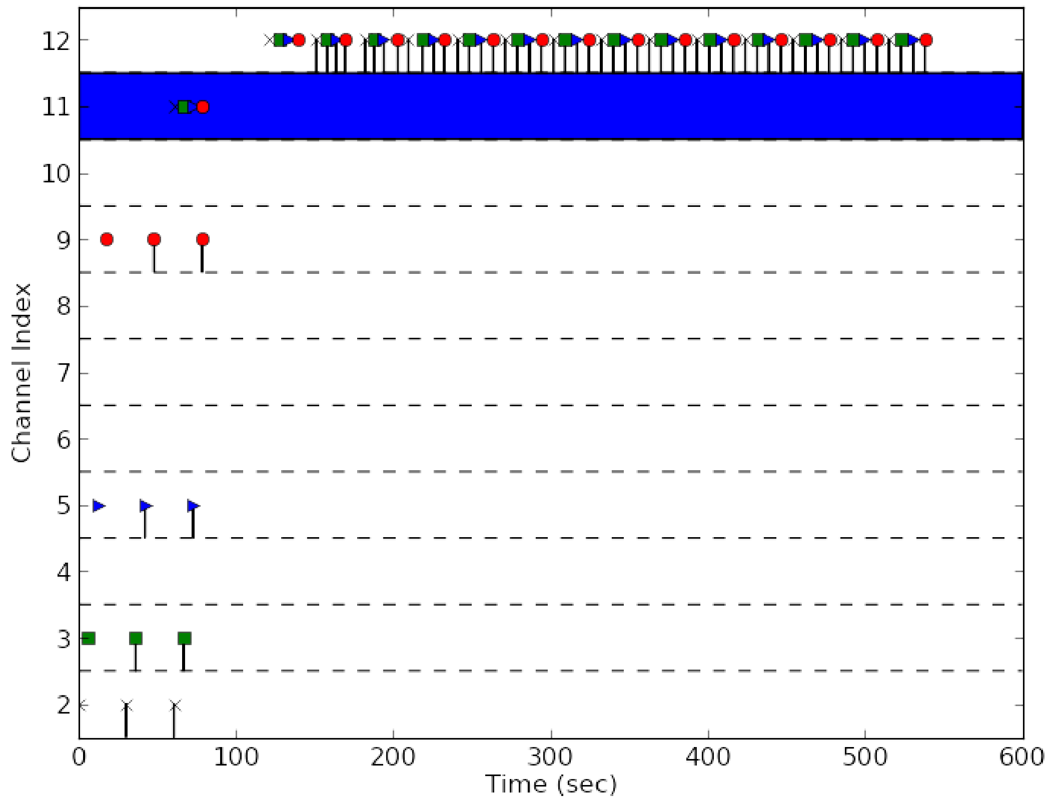


Figure 7.22: Frequency Flocking Scenario 1

7.8 Total System

The total system provides the means to develop self-organizing, decentralized networks through the coordination of radios in time and frequency. This coordination arises from the combinations of the behaviors discussed above. As discussed throughout, both the interactions of radios and the interactions of behaviors within a single radio contribute to the final operation of the system.

Examination of the total system is simply matter of reviewing system behaviors to ensure behavior contribute as expected. This examination first reviews the ability of the radios to find one another in frequency and establish a time organization. Next radios were tasked with performing the same task in the presence of interference.

Figure 7.23 displays the radios finding one another in frequency and establishing a time organization that allows for transmitting application data. This figure combines the information shown in the time flocking figures and that shown in the frequency flocking figures. Here the slots of a radio are displayed by its symbol with no lines. The beacons of a radio are indicated by a vertical line below a radio's symbol and frequency changes are depicted by a vertical line above a radio's symbol. Throughput is determined and displayed in the same manner as above. Figure 7.23 display the radios ability to gather to single frequency and establish a division of time conducive for application transmissions once there. Figure 7.24 shows the society of CRs gathering in frequency, avoiding interference. Each of these tests demonstrates the successful combination of the behaviors discussed above.

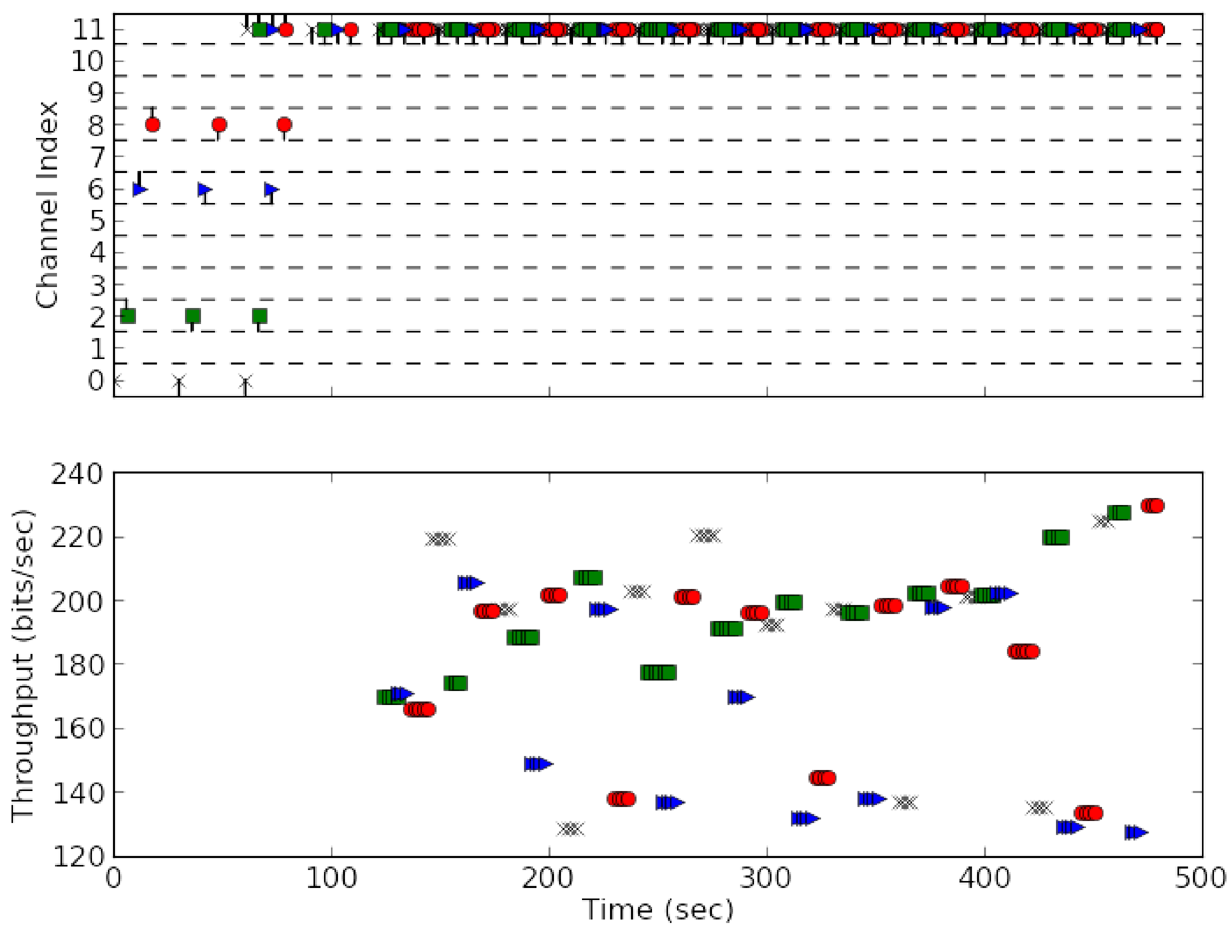


Figure 7.23: Total System Testing without Interference

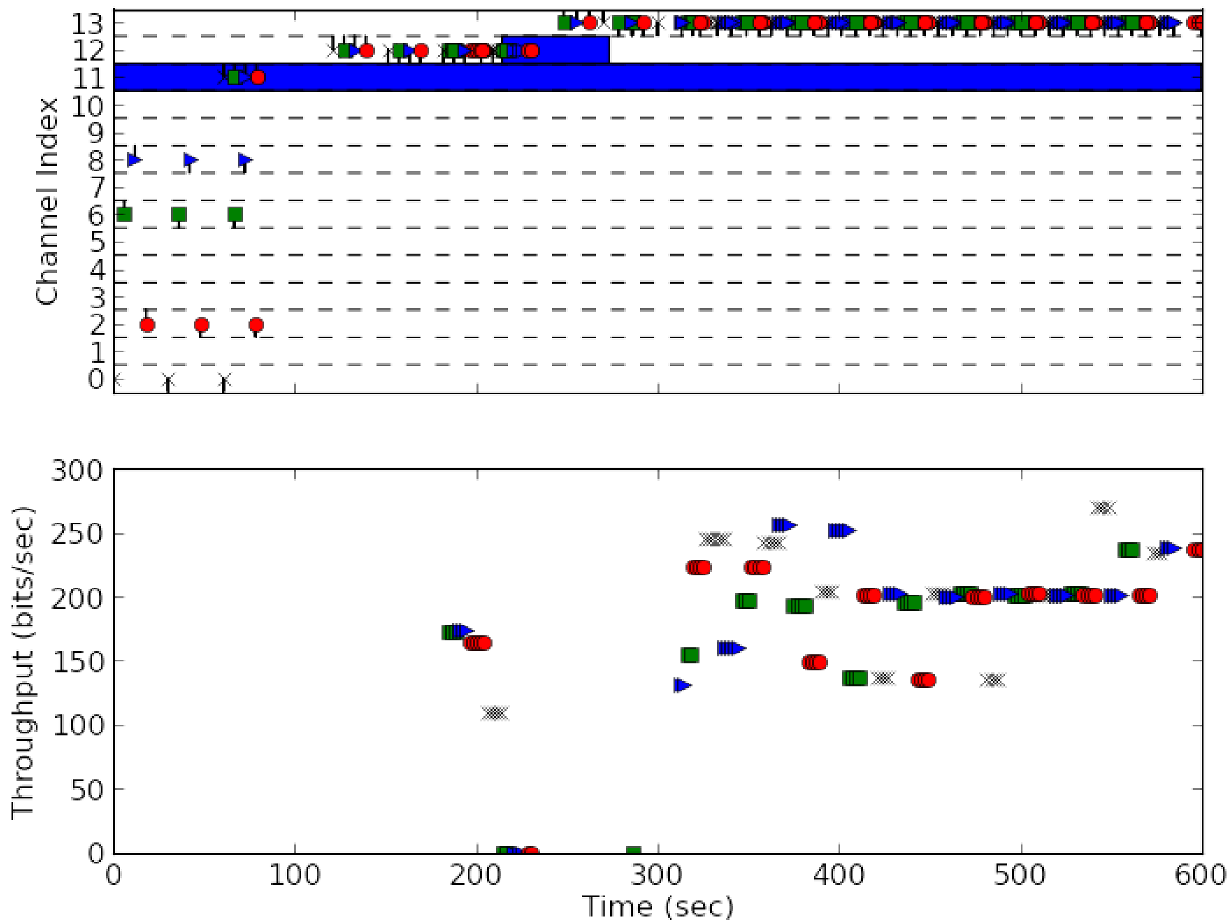


Figure 7.24: Total System Testing with Interference

7.9 Conclusion

This chapter has presented the evaluation of the prototype CR society presented in Chapter 6. Evaluation examined behaviors individually before exploring the combinations of these behaviors and their implications. The combinations of behaviors provide the emergent properties of the systems, including the enabling of societal learning.

Chapter 8

Conclusion

This dissertation has introduced the concept of an artificial society based on the use of an action based social language combined with the behavior-based approach to the construction of multi-agent systems to address the problem of developing decentralized, self-organizing networks that dynamically fit into their environment. The use of social language in these societies efficiently connects members to allow for coordination and societal learning, as discussed in Chapter 2. The behavior-based method to developing multi-agent systems from the field of robotics, covered in Chapter 3, provides a framework for the development and combination of basic behaviors that address individual concerns of a group of agents. The context of CR research, addressed in Chapter 4, offers proven methods to address particular problems faced by CRs. The method for developing CR societies, introduced in Chapter 5, combines these tools and techniques into a unified approach to providing MAC layer solutions for CRs. A prototype implementation, described in Chapter 6, offers a proof of concept and quantifies the CR society approach to networks. Evaluation of this implementation, presented in Chapter 7, explores the benefits of the methodology by examining the interactions of behaviors.

The work covered in this dissertation has focused on providing an approach to developing decentralized CR networks. As such the work has not dwelt on the development of optimized techniques, but rather the complementary design of techniques and agents to serve a particular purpose. The approach presented is based on research from a number of fields that study the interactions of intelligent agents. The work introduces a new method of CR, based on the use of efficient coordination communication, that allows for dynamic networks that learns as a whole through the local interactions and decision making of radios.

The method presented here combines several basic behaviors in order to achieve its goals. The behaviors each address a particular need of the network. The time avoidance behavior allows the radios to avoid interfering with external systems. The time cooperation provides CRs with a decentralized method of dividing time in to slots for the transmission of application data in such a way that stores information in the interactions between radios. The frequency aggregation behavior groups the radios in frequency to allow for coordination. The frequency dispersion behavior provides a method for radios to detect and vacate congested channels. The frequency learning behavior adapts the operations of the radios to their environment over time. Combining all of these behaviors provides the radios with the means to dynamically organize themselves in a such a manner that the network as a whole learns appropriate placements for radios in time and frequency.

Implementation of the methods discussed here was carried out using low complexity CR platforms. The approach of this work does not require sophisticated capabilities from either the computational platform or the radio front end. Instead the development of CR societies focuses on using readily available capabilites in an intelligent manner. The result of this is that inexpensive, low capability CRs are able to dynamically self-organize without support from more powerful platforms.

The work presented here introduces the mechanisms necessary for CRs to take advantage of social learning. This type of learning stores knowledge in the interactions of radios, i.e., their state within the system. The effect of this is the distribution of knowledge throughout the network, mirroring the techniques employed by bees when searching for a new nesting site [11]. While knowledge distributed in this way persists only as long as the radios retain their state within the network, i.e., the network continuously updates its knowledge based on the current radios and their situations. Additionally, note that this information depends on the interactions between the radios, rather than the internal knowledge of the radios and therefore radios can not store this information on their own. Rather social learning allows networks as a whole to learn and adapt to their environment.

The prototype implementation of a CR society presented here demonstrated social learning. Despite the fact the that individual radios used in the network had limited capabilities, the network as a whole was able to learn an appropriate organization for the radios and remember this organization. Note that no individual radio held this information; rather it was spread throughout the network and held in the interactions between nodes. The network was able to apply social language to determine appropriate network behavior changes that persisted in the face of interruption.

The abilities of CR societies presented here have application to several current problems, most notably LTE cellular systems. LTE systems must often organize several elements in a cooperative manner, whether a proliferation of femtocells or base stations. Applying the sort of decentralized self-organizing abilities exhibited here would benefit both femtocells and base stations. Specifically, base stations operating in the time division duplex (TDD) paradigm could greatly benefit from the principles of CR societies. These base stations must organize their transmission cooperatively with other base stations in their area, while also dynamically adapting uplink and downlink time slots according to current traffic. Although the prototype implementation presented here may not directly address these concerns, it demonstrates that the principles of social language and CR societies are appropriate for these types of problems.

Bibliography

[1] J. Mitola, III, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio," Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.

[2] C. W. Bostian, "The promise of cognitive radio for communications and remote sensing for critical infrastructure, disaster, safety, and risk management," in *Radio Science Meeting (USNC-URSI NRSM), 2013 US National Committee of URSI National*, 2013, pp. 1–1.

[3] M. McHenry *et al.*, "Xg dsa radio system," in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, 2008, pp. 1–11.

[4] F. Ge *et al.*, "A cooperative sensing based spectrum broker for dynamic spectrum access," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, 2009, pp. 1–7.

[5] C. Doerr *et al.*, "What a cognitive radio network could learn from a school of fish," *ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, pp. 1–10, 2007.

[6] T. D. Seeley, *Wisdom of the Hive: The Social Physiology of Honey Bee Colonies.* Harvard University Press, 2009.

[7] E. O. Wilson and M. Pavan, "Glanduar sources and specificity of some chemical releasers of social behavior in dolichoderine ants," *Psyche: A Journal of Entomology*, no. 4, pp. 70–76, 1959.

[8] A. Strandburg-Peshkin *et al.*, "Visual sensory networks and effective information transfter in animal groups," *Current Biology*, vol. 23, no. 17, pp. 709–711, 2013.

[9] (2013) Social Language Use (Pragmatics). American Speech-Language-Hearing Association. [Online]. Available: http://www.asha.org/public/speech/development/pragmatics.htm

[10] (2011, September) Jürgen Habermas. Metaphysics Research Lab, Stanford University. [Online]. Available: http://plato.stanford.edu/entries/habermas/

[11] T. D. Seeley, *Honeybee Democracy*. Princeton University Press, 2010.

[12] J. Habermas, *The Theory of Communicative Action*. Beacon Press, 1985, vol. 1.

[13] J. O. Neel, "Analysis and design of cognitive radio networks and distributed radio resource management algorithms," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2006.

[14] U. Greggers *et al.*, "Reception and learning of electric fields in bees," *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1759, 2013. [Online]. Available: http://rspb.royalsocietypublishing.org/content/280/1759/20130528.abstract

[15] R. Learned *et al.*, "Cognitive coexistence: A throughput study of mud-ehanced opportunistic spectrum access," in *Asilomar Conference on Signals, Systems, and Computers*, 2013, pp. 1–8.

[16] J. R. Boyd, *Destruction and Creation*. U.S. Army Command and General Staff College, 1987.

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.

[18] D. Hofstadter, *I am a Strange Loop*. Basic Books, 2007.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1998.

[20] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[21] Y. U. Cao *et al.*, "Cooperative mobile robotics: antecedents and directions," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, 1995, pp. 226–234 vol.1.

[22] S. Souissi *et al.*, "Distributed algorithms for cooperative mobile robots: A survey," in *Networking and Computing (ICNC), 2011 Second International Conference on*, 2011, pp. 364–371.

[23] G. W. Walter, "An imitation of life," *Scientific American*, vol. May, pp. 42–45, 1950.

[24] M. J. Mataric, "Interaction and intelligent behavior," Ph.D. dissertation, Massachusetts Institute of Technology, 1994.

[25] R. Beckers *et al.*, "From local actions to global tasks: Stigmergy and collective robotics," in *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic, Volume 1, Volume 2 Prerational Intelligence: Interdisciplinary Perspectives on the Behavior of Natural and Artificial Systems, Volume 3*, ser. Studies in Cognitive Systems, H. Cruse *et al.*, Eds. Springer Netherlands, 2000, vol. 26.

[26] R. C. Arkin, "Cooperation without communication: Multiagent schema-based robot navigation," *Journal of Robotic Systems*, vol. 9, no. 3, pp. 351–364, 1992. [Online]. Available: http://dx.doi.org/10.1002/rob.4620090304

[27] T. Balch and R. C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00735341

[28] J. Wang and G. Beni, "Distributed computing problems in cellular robotic systems," in *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, 1990, pp. 819–826 vol.2.

[29] T. Arai *et al.*, "Information diffusion by local communication of multiple mobile robots," in *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, 1993, pp. 535–540 vol.4.

[30] A. A. Ray, "Cooperative robotics using wireless communication," Master's thesis, Auburn University, 2005.

[31] M. A. Lewis and G. A. Bekey, "The behavioral self-organization of nanorobots using local rules," in *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, vol. 2, 1992, pp. 1333–1338.

[32] H. Yamaguchi, "Adaptive formation control for distributed autonomous mobile robot groups," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3, 1997, pp. 2300–2305 vol.3.

[33] D. W. Gage, "How to communicate with zillions of robots," in *In Proceedings of SPIE Mobile Robots VIII*, 1993, pp. 250–257.

[34] A. Drogoul and J. Ferber, "From tom thumb to the dockers: Some experiments with foraging robots," in *In Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* MIT Press, 1992, pp. 451–459.

[35] R. A. Brooks, "A robust layered control system for a mobile robot," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1985.

[36] ——, "Intelligence without reason," in *COMPUTERS AND THOUGHT, IJCAI-91.* Morgan Kaufmann, 1991, pp. 569–595.

[37] T. W. Rondeau, “Application of artifical intelligence to wireless communications,” Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2007.

[38] M. J. Mataric, “Minimizing complexity in controlling a mobile robot population,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 830–835 vol.1.

[39] ——, “Designing emergent behaviors: From local interactions to collective intelligence,” in *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, 1993, pp. 432–441.

[40] ——, “Kin recognition, similarity, and group behavior,” in *in ‘Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 1993, pp. 705–710.

[41] ——, “Issues and approaches in the design of collective autonomous agents,” *Robotics and Autonomous Systems*, vol. 16, no. 24, pp. 321 – 331, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0921889095000534

[42] S. Hackwood and G. Beni, “Self-organizing sensors by deterministic annealing,” in *Intelligent Robots and Systems ’91. ’Intelligence for Mechanical Systems, Proceedings IROS ’91. IEEE/RSJ International Workshop on*, 1991, pp. 1177–1183 vol.3.

[43] K. Dautenhahn, “Getting to know each other - artificial social intelligence for autonomous robots,” *Robotics and Autonomous Systems*, vol. 16, pp. 333–356, 1995.

[44] ——, “The art of designing socially intelligent agents - science, fiction and the human in the loop,” *Applied Artificial Intelligence Journal, Special Issue on Socially Intelligent Agents*, vol. 12, pp. 12–7, 1998.

[45] T. Fong *et al.*, “A survey of socially interactive robots,” 2002.

[46] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, 2nd ed. McGraw-Hill, 2004.

[47] L. W. Couch, *Digital and Analog Communication Systems*, 2nd ed. Macmillan Publishing Company, 1987.

[48] A. R. Young *et al.*, “Csere (cognitive system enabling radio evolution): A modular and user-friendly cognitive engine,” in *Dynamic Spectrum Access Networks (DYSPAN), 2012 IEEE International Symposium on*, 2012, pp. 59–67.

[49] O. B. Akan, “Bio-inspired and nano-scale molecular communication: From evolution to evaluation,” in *Performance Evaluation of Computer Telecommunication Systems, 2009. SPECTS 2009. International Symposium on*, vol. 41, July 2009, pp. vi–vii.

[50] M. A. Hoque and X. Hong, "Biostar: A bio-inspired stable routing for cognitive radio networks," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Jan 2012, pp. 402–406.

[51] P. Di Lorenzo and S. Barbarossa, "A bio-inspired swarming algorithm for decentralized access in cognitive radio," *Signal Processing, IEEE Transactions on*, vol. 59, no. 12, pp. 6160–6174, Dec 2011.

[52] K. Wu *et al.*, "Bio-inspired multi-user beamforming for qoe provisioning in cognitive radio networks," in *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, Sept 2012, pp. 173–177.

[53] N. Pari *et al.*, "Bio-inspired time synchronization for cognitive radio ad hoc networks," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec 2012, pp. 980–985.

[54] Q. He and P. Zhang, "Dynamic channel assignment using ant colony optimization for cognitive radio networks," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, Sept 2012, pp. 1–5.

[55] T. Renk *et al.*, "Bio-inspired algorithms for dynamic resource allocation in cognitive wireless networks," in *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, Aug 2007, pp. 351–356.

[56] L. A. DaSilva *et al.*, "Game theory in wireless networks," *Communications Magazine, IEEE*, vol. 49, no. 8, pp. 110–111, August 2011.

[57] J. O. Neel *et al.*, "Convergence of cognitive radio networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 4, March 2004, pp. 2250–2255 Vol.4.

[58] R. W. Thomas *et al.*, "Joint power and channel minimization in topology control: A cognitive network approach," in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 6538–6543.

[59] I. F. Akyildiz *et al.*, "A survey on spectrum management in cognitive radio networks," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 40–48, 2008.

[60] C. Cormio and K. R. Chowdhury, "A survey on {MAC} protocols for cognitive radio networks," *Ad Hoc Networks*, vol. 7, no. 7, pp. 1315 – 1329, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870509000043

[61] L. Lai *et al.*, "Cognitive medium access: Exploration, exploitation, and competition," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 2, pp. 239–253, Feb 2011.

[62] Q. Zhao *et al.*, "Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A pomdp framework," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 3, pp. 589–600, April 2007.

[63] K. C. Huang and D. Raychaudhuri, "Mac protocol adaptation in cognitive radio networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, 2011, pp. 245–250.

[64] J. Degesys *et al.*, "Desync: Self-organizing desynchronization and tdma on wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, 2007, pp. 11–20.

[65] R. Pagliari *et al.*, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 4, pp. 564–575, 2010.

[66] A. Patel *et al.*, "Desynchronization: The theory of self-organizing algorithms for round-robin scheduling," in *Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on*, 2007, pp. 87–96.

[67] J. Degesys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, 2008, pp. 129–138.

[68] C. Mühlberger and R. Kolla, "Extended desynchronization for multi-hop topologies," University of Wuerzburg, Department of Computer Science, Technical Report 460, July 2009.

[69] (2012, dec) Beagleboard-xm. [Online]. Available: http://beagleboard.org/Products/BeagleBoard-xM

[70] (2012, jun) Rfm22b fsk transceiver. [Online]. Available: http://www.hoperf.com/rf/fsk_module/RFM22B.htm

[71] A. R. Young and C. W. Bostian, "Simple and low-cost platforms for cognitive radio experiments [application notes]," *Microwave Magazine, IEEE*, vol. 14, no. 1, pp. 146–157, Jan 2013.

[72] A. R. Young, "Unified multi-domain decision making: Cognitive radio and autonomous vehicle convergence," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2013.

[73] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, Dec 1990.