STABLIZATION IN WIRELESS SENSOR NETWORKS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Hui Cao, B.S., M.S.

* * * * *

The Ohio State University

2008

Dissertation Committee:

Anish Arora, Adviser

David Lee

Prasun Sinha

Approved by

Adviser

Graduate Program in

Computer Science and

Engineering

ABSTRACT

Wireless sensor networks (WSNs) are deployed to cooperatively monitor environmental conditions, where sensor nodes may be subjected to harsh terrains, severe weather, and even physical damage. All of those give rise to state corruption, hardware failures, and noisy measurements etc. In my work, I focus on stabilization of a WSN system, where regardless of the current state of the system, each computation of the system will eventually converge to a legal state in a finite number of steps and henceforth the system will continue to operate as specified.

In my dissertation, I address stabilization in dynamic systems with varying equilibrium, such as (1) pursuit-evasion games in networked environments to minimize capture time (distance) and (2) communication protocol for mobile networks to maximize energy efficiency. In those systems that seek to maintain optimal performance, their legal states must satisfy the desired optimality. Stabilization of these systems thus implies that when the current state of the system becomes suboptimal, they will eventually resume optimal performance. I also introduce feature calibration, where the applicability of calibration is broadened by lifting the calibration problem from the level of sensors to that of sensing applications.

I formulate the maintenance of optimality of performance in dynamical systems in terms of the standard notion of stabilization. I provide three techniques — estimator-based, MinMax controllers that lead to Nash equilibrium, and transformer-based.

Each of these techniques relates to a different aspect of the system, respectively, its input, its controller, and its output. For systems with observable external inputs and computable optimality, stabilization may be achieved by adding a stabilizing input estimator to the system. But environments and external inputs are often unobservable, which makes reestablishing optimality difficult. To overcome this difficulty, I present two alternative methods, one based on a game-theoretic MinMax strategy that leads to Nash equilibrium, and the other based on a feedback control mechanism that adds a stabilizing output transformer to the system. I exemplify these two approaches with a pursuit evasion application and a MAC layer duty cycle adaptation protocol, respectively.

For stabilizing application design, I focus on differential games in networked environments with constrained communication resources leading to delays, losses and finite rates in information updates. I focus on two typical differential games: pursuit-evasion game for target capture and an "asset protection" game. I show the inherent stabilization features of my pursuit strategies, both in terms of implementation as well as the strategies themselves. My pursuer strategy is stabilizing, provided that the sampling period and the delay in obtaining the evader state information updates scale linearly with the pursuer-evader distance.

For stabilizing protocol design, I provide a stabilizing receiver centric MAC protocol-OMAC, where robust asynchronous discovery occurs in parallel with synchronous (receiver centric unicast and broadcast) communication. I identify the fact of receiver dominance in energy consumption and introduce the design paradigm of receiver centricity, which is in contrast to the current sender based MAC layer design. I believe this new paradigm will dominate energy sensitive designs. To minimize the energy

consumption of asynchronous discovery, I design a wakeup schedule that achieves the optimal bound for 3 state (listen, beacon, and sleep) radios. And, to minimize the energy consumption of synchronous communication, I design a distributed stabilizing controller that adapts the duty cycle at each node to correspond to that node's actual communication traffic levels.

Despite recent theory development, methods of calibration that accurately recover signals from biased sensor readings remain limited in their applicability. Acoustic sensors, for instance, which have been popular in low power wireless sensor networks, are difficult to calibrate in this manner, given their significant hardware variability, large dynamic range, sensitivity to battery power level, and complex spatial/temporal environmental variations. We submit that the applicability of calibration is broadened by lifting the calibration problem from the level of sensors to that of sensing applications. We show feasibility of easy, accurate calibration at the level of application-specific features, via an example of recovering the feature of acoustic signal-to-noise ratio (SNR) that is useful in event-detection applications.

This is dedicated to my wife Lanyan Fang, my sons Albert Cao and Brad Cao

ACKNOWLEDGMENTS

To complete my Ph.D. study, it would have been impossible without the help from many great people to provide me guidance, support, and encouragement.

Firstly, I would like to thank my adviser Prof. Anish Arora. It is him who has given me the opportunity to conduct research in exciting sensor networks area in the past five years. I greatly appreciate his insightful advice on my research as well as personal life. I also greatly appreciate his patience in guiding me through the road to high quality research, from defining problems to explaining results.

I am thankful to my dissertation committee members Prof. Prasun Sinha and Prof. David Lee, who have given me valuable suggestions and advises. Prof. Prasun Sinha has guided me through all the networking courses and provided me many helps in research and my career. Prof. David Lee has provided me insight suggestion during our meetings. In addition, Prof. Ten H. Lai, as a member of my candidacy exam committee, has taught me how to clearly define a problem and how to solve it systematically.

It is my fortunate to know Dr. Kenneth W. Parker in early 2004. His insight and experience guided me through my Ph.D. study. We have been working several topics in the past 3 years. All of them are full of joy and excitement, which greatly enrich my Ph.D. experience. Without him, it would be impossible to complete my thesis at this time.

I would also like to acknowledge Dr. Emre Ertin. It is him who guides me through my first publication. I still remember the time when I visited his home frequently to conduct experiments in his backyard. I have owned him a lot for his wisdom and friendship. It is always a great fun to interact with him.

In the passed five years, I have been involved in several projects in DARPA NEST research program. I have benefited from many great scientists and researchers including Dr. William Leal, Dr. Mohamed Gouda, Dr. Ted Herman, Dr. Sandeep Kulkarni, Dr. Mikhail Nesterenko, Dr. Rajiv Ramnath, and Dr. Brian Flanagan.

I would also like to acknowledge the help and friendship from Dr. Hongwei Zhang, Dr. Santosh Kumar, Dr. Sandip Bapat, Dr. Vinayak Naik, Lifeng Sang, Vinodkrishnan Kulathumani, Mukundan Sridharan, Prabal Dutta, Murat Demirbas, Taewoo Kwon, Wenjie Zeng, and Jing Li.

Finally, I would like to thank my wife Lanyan Fang, who has been alway available to help me during my Ph.D study, with her passion and encouragement. My life has been filled with joy because of our twin sons, Albert Cao and Brad Cao.

VITA

June 12, 1976	Born - Shandong, China.
1998	B.S. Automation, Department of Automation Shenyang University of Technology.
2001	M.S. Control Theory and Engineering, Department of Automation Tsinghua University.
August - December 2006	Research Intern Eaton Corporation Innovation Center, WI, USA.
September - December 2007	Research Intern Siemens Research Corporation, NJ, USA.
2003 - present	Graduate Teaching and Research Associate, The Ohio State University.

PUBLICATIONS

Research Publications

Amitabha Ghosh, Luis Pereira, Ting Yan, and Hui Cao "Modeling Wireless Sensor Network Architectures using AADL". *4th European Congress ERTS Embedded Real Time Software (ERTS)*, January, 2008.

Hui Cao and Anish Arora "Stabilization in Dynamic Systems with Varying Equilibrium". *9th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, November, 2007.

Hui Cao, Ting Yan, Luis R. Pereira, Sujit R. Das, and Bruce Lewis "Using AADL to Analyze and Design Embedded System". *Embedded System Design Magazine (CMP)*, August, 2007.

Hui Cao, Kenneth W. Parker, and Anish Arora "O-MAC: A Receiver Centric Power Management Protocol". *14th IEEE International Conference on Network Protocols (ICNP)*, November, 2006.

Hui Cao, Emre Ertin, Vinodkrishnan Kulathumani, Mukundan Sridharan, and Anish Arora "Differential Games in Large Scale Sensor Actuator Networks". *5th International Conference on Information Processing in Sensor Networks (IPSN)*, April, 2006.

Emre Ertin, Anish Arora, Rajiv Ramnath, Mikhail Nesterenko Vinayak Naik, Sandip Bapat, Vinod Kulathumani, Mukundan Sridharan, Hongwei Zhang, and Hui Cao "Kansei: A Testbed for Sensing at Scale". *5th International Conference on Information Processing in Sensor Networks (IPSN)*, April, 2006.

Anish Arora, Rajiv Ramnath, Emre Ertin, Prasun Sinha, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Hui Cao, Mukundan Sridhara, Santosh Kumar, Nick Seddon, Chris Anderson, Ted Herman, Nishank Trivedi, Chen Zhang, Mohamed Gouda, Young-Ri Choi, Mikhail Nesterenko, Romil Shah, Sandeep Kulkarni, Mahesh Aramugam, Limin Wang, David Culler, Prabal Dutta, Cory Sharp, Gille Tolle, Mike Grimmer, Bill Ferriera, and Ken Parker. "ExScal: Elements of an Extreme Scale Wireless Sensor Networks". *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2005.

Anish Arora, Rajiv Ramnath, Prasun Sinha, Emre Ertin, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Mukundan Sridharan, Santosh Kumar, Hui Cao, Nick Seddon, Chris Anderson, Ted Herman, Chen Zhang, Nishank Trivedi, Mohamed G. Gouda, Young-ri Choi, Mikhail Nesterenko, Romil Shah, Sandeep S. Kulkarni, Mahesh Aramugam, Limin Wang, David E. Culler, Prabal Dutta, Cory Sharp, Gilman Tolle, Mike Grimmer, Bill Ferriera, Kenneth Parker. "Project ExScal". *International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, 2005.

Anish Arora, Prabal Dutta, Sandip Bapat, Vinod Kulathumani, Hongwei Zhang, Vinayak Naik, Vineet Mittal, Hui Cao, Murat Demirbas, Mohamed Gouda, Young-Ri Choi, Ted Herman, Sandeep Kulkarni, U. Arumugam, Mikhail Nesterenko, A. Vora, and M. Miyashita. "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking". *Computer Networks (Elsevier)*, 46(5):605-634, December, 2004.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in:

Computer Networking	Prof. Anish Arora
	Prof. Ten-hwang Lai
	Prof. David Lee
	Prof. Prasun Sinha
Statistics	Prof. Douglas E. Critchlow
Control theory	Prof. Kevin Passino

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Wireless sensor networks (WSN) have emerged as key technologies that enable a broad spectrum of new applications. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance [2, 6]. Currently, wireless sensor networks are being applied in many civilian application areas, including environment and habitat monitoring, health care applications, home automation, and traffic control.

WSN consists of spatially distributed autonomous devices capable of sensing, processing, and wireless communication, to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Sensor nodes may be subjected to harsh terrains, severe weather, and even physical damage. All of those factors give rise to several types of faults, such as state corruption, message loss, message delay, and noisy measurements. Either one of them may break down the system. To deal with those faults, I focus on approaches that achieve fault tolerant via stabilization. In other words, network

protocols and application schemes are designed so that even if the system deviates from optimal or normal state, it can still convergence to the optimal or normal state.

Stabilization in WSN is a challenging task, because each individual node is limited in its sensing, processing, communication capabilities, and battery capacity [21]:

- **limited sensing:** Sensors on network nodes have limited sensing range, typically only monitor information in vicinity. Their sensing abilities are also influenced by hardware differences, environmental variations such as temperature, humidity, vibration or even wind. Consequently, sensor readings on individual nodes are inaccurate and even create false positive or false negative detections.

- **limited processing:** Sensor nodes usually have limited computation power and memory, which limit the amount of sophistication that can be built on individual nodes. In addition, faults such as corruption variables, miss fired tasks may occur during the execution.

- **constrained communication:** Wireless radios used in those systems usually have limited data rate and low reliability. Wireless link quality is also dynamic due to factors such as interference and changes in environment.

- **limited battery:** Sensor nodes are usually battery-powered, and it is hard to replace them after deployment. To save energy, nodes must sleep or partially sleep when necessary. Typically, radio is duty-cycled because it is the major source of power consumption. The design of stabilization must also consider the case where radios are frequently turned off.

In this dissertation, I investigate approaches to provide stabilization for dynamic system with varying equilibrium, and address the stabilization problems in application

layer and MAC layer. Specifically, in application layer I focus on application strategies under communication constrains, such as information update rate, network delay, and message loss; in MAC layer I focus on power management protocols that achieve high energy efficiency while engaging the property of stabilization despite mobility, link dynamics, and clock variations. I also design auto-calibration methods for sensor nodes to compensate the hardware variants among them.

1.1.1 Application strategies

Sensor network technology has enabled new surveillance systems [2, 6], where sensor nodes equipped with processing and communication capabilities can collaboratively detect, classify and track targets of interest over a large area. We investigate two representative pursuit-evasion games. One is a classical pursuit-evasion game for target capture [43, 39], where pursuers try to catch the evaders as soon as possible; the other is called "asset protection game"(also called Lifeline Game) where pursuers try to protect a linear target by intercepting the evaders as far as possible from the target. These games have practical applications in real world applications and the techniques introduced for these games can be generalized to a wide variety of differential games.

Target track information obtained by local processing of sensor information needs to be routed to mobile agents through multi-hop communication links, which results in delays, message losses and random arrival times of the packets carrying track information. When designing those systems, application designer must consider these communication constrains, to make application strategies robust. On the other side,

network designer must satisfy the requirement of application. Understanding the relationship between application and network is necessary to design a reliable surveillance system.

Furthermore, those networks are usually deployed in harsh environments, state information may be corrupted, and even worse some network nodes are destroyed, which also necessitates the stabilization of application strategies [14] [13].

In addition, evader may change its strategy intentionally or be lack of enough information. Pursuit strategies must be designed to guarantee a minimal payoff even if evader deviates from rational strategies.

In this work, we concentrate on the formulation of optimal pursuit control strategies despite network effects, such as non-periodic track updates, message loss and delays, to derive optimal strategies, bounds on their information requirements and the scaling properties of these bounds. Those strategies can also stabilize to optimal state even state information is corrupted. In addition, pursuer can still achieve a minimal payoff irrespective of how the evader changes its strategies.

1.1.2 Energy efficient protocols

Energy is a fundamental bottleneck of wireless sensor networks. Radio communication is the dominant power consumption in all the components [19]. To extend the lifetime of sensor networks from less than one week with full wakeup to one year or several years, power management must be designed to achieve a duty cycle less than 1%. In practice there are very few examples of deployments that achieve this level of performance. Synchronous MACs can achieve this level of performance, but have not been widely adopted largely because of the system implications for discovery

and bootstrapping, which are neccessary procedures to stabilize to synchronous communications from asynchronous communications. In a low duty cycle network, two neighboring nodes that are out of sync may never communicate because their wakeup schedules may never overlap. The following conditions may drive synchronous network out of sync:

- Link quality dynamics: In [35], multiple schedules (time zones) have been shown to coexist consistently on a 50 Mica2Dot motes network running S-MAC. Without neighbor discovery and stabilization to a global time, partitioned networks never converge to a synchronous schedule.

- Clock variations: In sensor networks, clock hardwares used on network nodes have higher skew around $50PPM$ (up to 0.01% error between pair of nodes), which requires time synchronization every $10s$ minutes. In addtion, environmental factors such as temperature and humidity also affect clock skew. Therefore, time synchronization needs to be maintained across the network continuously. However, time synchoronization cannot solely depend on synchronous communications for a duty cycled network.

- Mobility: It also posts great challenges to synchronous MACs. Mobile networks need to continuously add and subtract nodes and also to partition and recombine.

Other than discovery and bootstrapping, duty cycle adaptation is a key consideration for mobile network MACs. Because the point of designing a low duty cycle system is typically for energy efficiency, adapting the duty cycle according to the network traffic is key. If the duty cycle is lower than required, higher collision or

sender buffer overflow can happen; if the duty cycle is higher than required, energy is wasted on idle listening. However, duty cycle adaptation may introduce instability into system, where duty cycle may never converge to optimal operating points when traffic varies frequently. The adaptation mechanism must be design to guarantee stabilization despite dynamic traffic.

In summary, our goal of this work is to design energy efficient MAC protocols that achieve duty cycle that is lower than 1%, while can still stabilize to optimal synchrounous state despite mobility, link dynamics, hardware variations, and dynamic traffic.

1.1.3 Feature calibration

Despite recent theory development, methods of calibration that accurately recover signals from biased sensor readings remain limited in their applicability. Acoustic sensors, for instance, which have been popular in low power wireless sensor networks, are difficult to calibrate in this manner, given their significant hardware variability, large dynamic range, sensitivity to battery power level, and complex spatial/temporal environmental variations. The goal of this work is to design a simple, accurate calibration procedure at the level of application-specific features, via an example of recovering the feature of acoustic signal-to-noise ratio (SNR) that is useful in event-detection applications.

1.2 Contributions

1.2.1 Stabilization of systems with varying equilibrium

System design often explores optimality of performance. What is optimal is, however, often not predefined or static in most cases, because it is affected by the

context of operation, such as the environment or external system inputs. We formulate the maintenance of optimality of performance in dynamical systems in terms of the standard notion of stabilization. For systems with observable external inputs and computable optimality, stabilization may be achieved by adding a stabilizing input estimator to the system. But environments and external inputs are often unobservable. To overcome this difficulty, we present two alternative methods, one based on a game-theoretic MinMax strategy that leads to Nash equilibrium, and the other based on a feedback control mechanism that adds a stabilizing output transformer to the system. We exemplify these two approaches with a pursuit-evasion application and a MAC layer duty cycle adaptation protocol, respectively.

1.2.2 Stabilization of application strategies

For both the target capture game and the asset protection game, we have proved the following properties hold:

- **Optimal pursuit strategy under perfect information:** the best pursuer strategy is to move toward a location that is decided by current pursuer and evader locations and their speed ratio. At the mean time, the best evader strategy is to move to that location also.

- **Sampling rate requirements of the optimal pursuit strategy:** Under perfect information, the global state is available to the pursuer at all times. This is an unrealistic assumption for a sensor network implementation where the information can be provided only at discrete time intervals. We have proved the sampling rate is inversely proportional to the relative distance between the pursuer and evader in order to maintain optimality of pursuer strategy.

The result is particularly important for sensor network implementations using resource constrained nodes, because it informs how the information data rate can be reduced based on the state of the game so as to conserve the energy and bandwidth resources of the network.

- **Effect of message losses:** To guarantee the optimal evader capture, the information must be updated before the pursuer reaches a critical location defined in [14]. In the presence of message losses, the pursuer needs to issue multiple queries within a sampling period and adjust the frequency of its queries according to the state of the game. To minimize the frequency of the queries, it suffices that the network communication protocol scale to provide higher reliability as the distance between the pursuer and evader decreases. We have shown this lower bound frequency depends on the distance between the pursuer and evader.

- **Effect of packet delay:** The evader location information needs to be routed from the local fusion center to the pursuer through wireless multiple hop links. The multi-hop communication imposes non-negligible delays on the evader state information. We assume the network is time synchronized and the packets are timestamped at the source so that the pursuer will be able to calculate the delay of the packets it received. We have proved that an optimal Nash equilibrium exists if delay is linearly proportional to distance between the pursuer and evader.

Stabilization of the pursuer strategy:

We have shown that Nash equilibrium can still hold despite communication constraints in both the target capture game and the asset protection game. In this section, we discuss the stabilization properties of the pursuer strategy in the presence of state corruption as well as change in evader strategy.

- **State corruption:** The optimal pursuer strategy is based on the latest evader location information, and is thus independent of history information. Even if state information is corrupted, the pursuer should continue to query the latest evader location and move according to its optimal strategy. After it receives the correct evader location information, Nash equilibrium is reestablished.

- **Change in evader strategy:** Every min-max equilibrium strategy enjoys the guarantee of a minimal payoff—regardless of what its opponent chooses to do. Our pursuer strategy thus has a sort of stabilization property, in the sense that irrespective of how the evader changes its strategy, the pursuer strategy is guaranteed to achieve a minimal payoff, i.e., catch distance or catch time. Following the min-max strategy is the evolutionary stable choice for the pursuer.

1.2.3 Stabilization of MAC protocols

Receiver centric power management design

We introduce the concept of receiver-centric power management protocols, where receivers are scheduled to wake up in different slots. Ideally, only one receiver wakes up at a slot and a potential sender only need to wakeup to transmit in that slot; idle listening is avoided. This is different from the sender based design that current MAC layer protocols have assumed. In the sender centric design, the sender wakes up all the potential receivers during the transmission even if the message is unicast. The

key feature of receiver-centric protocol is that it avoids the vast majority of collisions by staggering or scheduling receiver on times rather than staggering or scheduling transmission times. We have shown:

- By modeling several popular MAC layer protocols, we derive bounds on performance for energy efficiency of those non-receiver centric protocols. In particular, we analyze four abstract models, Synchronous Blinking (e.g. TMAC [49], S-MAC [52]), Long Preamble (e.g. B-MAC [41], XMAC [54]), Structured Time-Spreading (also called Asynchronous Wake-Up), and Random Time Spreading. These results strongly suggest that scheduling the receiver so as to minimize (or eliminate) the potential for interference (or collisions) could be from 10 fold to 100 fold more efficient than current practice [15].

- We provide two receiver based scheduling techniques. One is centralized deterministic scheduling (Staggered On), the other is decentralized pseudo-random scheduling (Pseudo-randomized Staggered On). Surprisingly, the decentralized pseudo-random scheduling achieves only slightly lower energy efficiency compared with the global scheduling. Both of the receiver based scheduling techniques show orders of magnitude improvement over current transmitter based scheduling protocols.

Energy efficient protocol design for mobile networks

We design a new MAC protocol, called O-MAC, to create virtual MAC services that achieve duty cycles on the order of 1% for mobile networks [12], while guarantee stabilization despite link dynamics, clock variations, and mobility. Specifically, we design an energy efficient protocol that provides robust asynchronous discovery and

synchronous (unicast and broadcast) communication in parallel. Discovery, unicasts, and broadcasts each have different wakeup time intervals that either do not overlap with the others or that overlap with low probability; this is realized via pseudorandom slot selection [12].

Energy efficient asynchronous discovery

Many researchers have observed that synchronous MACs achieve significantly higher energy efficiencies than the theoretical limits for asynchronous MACs. Yet the common practice is to use asynchronous MACs. This is in large part due to the stabilization problem created by synchronous communication. Stabilizing a network requires discovery of these nodes and node/time synchronization to reestablish synchronous communication. For an always-on sensor network, neighborhood discovery is not an issue, since packet transmission can be overheard by neighboring nodes. However, in a low duty cycle mobile network, two neighboring nodes may never communicate because their wakeup schedules never overlap. Even worse, for any wakeup schedule that is based on time synchronization, unsynchronized nodes may be lost forever because synchronized nodes and unsynchronized nodes are mutually unaware of each other. We conclude that a synchronous protocol must coexist with energy efficient neighbor discovery in mobile, dynamic low duty cycle networks.

We formulate the problem of optimal neighbor discovery for a duty cycled network, and provide a class of optimal wakeup schedules that achieve neighbor discovery with minimum energy. In contrast, to use the optimal two states schedule proposed by [47] and [55], a node has to randomly select to beacon or listen during wakeup. Our 3-state schedules consume $1/\sqrt{2}$ of energy required by other approaches to achieve neighbor discovery in a discovery frame deterministically.

Duty cycle adaptation

Different traffics require different duty cycles to achieve optimal energy efficiency. In other words, nodes must adapt their duty cycle according to traffic changes. To achieve the optimal operating point, duty cycle adaptation algorithm must also consider stabilization as a necessary property in dynamic traffic.

We provide a duty cycle adaptation mechanism that is based on feedback from channel utilization and collisions. In addition, we identify a metric, the Activity Ratio, using which we transform the duty cycle optimization problem into a fixed point control problem. We have refined our algorithm so that it guarantees stabilization (as proven via the Lyapunov method.)

1.2.4 Feature calibration

We submit that the applicability of calibration is broadened by lifting the calibration problem from the level of sensors to that of sensing applications. We show feasibility of easy, accurate calibration at the level of application-specific features, via an example of recovering the feature of acoustic signal-to-noise ratio (SNR) that is useful in event-detection applications. By easy, we mean there is an efficient, purely local, and stimulus-free procedure for recovering SNR (that compares measured variances for multiple randomly chosen sensitivities, effected via acoustic sensor hardware support); unlike extant calibration methods, the procedure does not need to rely on any synchronization among nodes, long-term correlation between their respective environments, or assumptions about training events. And by accurate, we mean the procedure yields low error in SNR estimation. We provide experimental validation

of the difficulty of directly calibrating acoustic signals and the accuracy of our SNR calibration procedure.

1.3 Organization of this thesis

The rest of this thesis is organized as follows.

Chapter 2 introduces three methods to achieve stabilization for dynamic systems with varying equilibrium.

Chapter 3 describes the details of the networked differential games in surveillance systems. We presents two pursuit-evasion games: target capture and asset protection.

Chapter 4 introduces a new design philosophy for power management protocols: receiver centric. We also describe the challenges to design a stabilizing receiver centric protocol.

Chapter 5 presents O-MAC, a receiver centric stabilizing MAC layer protocol for mobile networks.

Chapter 6 defines a new type of calibration - feature calibration and provides one method to calibrate acoustic SNR feature.

Chapter 7 summaries the results of the dissertation and concludes with future work.

CHAPTER 2

STABILIZATION IN DYNAMIC SYSTEMS WITH VARYING EQUILIBRIUM

In this chapter, we formulate the maintenance of optimality of performance in dynamical systems in terms of the standard notion of stabilization. For systems with observable external inputs and computable optimality, stabilization may be achieved by adding a stabilizing input estimator to the system. But environments and external inputs are often unobservable. To overcome this difficulty, we present two alternative methods, one based on a game-theoretic MinMax strategy that leads to Nash equilibrium, and the other based on a feedback control mechanism that adds a stabilizing output transformer to the system. We exemplify these two approaches with a pursuit-evasion application and a MAC layer duty cycle adaptation protocol, respectively.

2.1 Introduction

System design often explores optimality of systems, as measured in terms of some performance metric(s). However, these optimal states are not predefined or static

in many systems because they are affected—or even decided—by the context of operation, such as the environment or external system inputs. When operating context changes, the equilibrium state(s) at which optimal performance is achieved also change, which requires the system to reestablish equilibrium continuously. By way of example, fault occurrences, environmental parameter changes, and new user inputs/traffics can each leave the system in a suboptimal state, as the optimal system state is often a single equilibrium point or a narrow region of points. These considerations motivate the importance of designing the property of stabilization in systems with varying equilibrium.

The standard notion of the stabilization of a system implies that regardless of the current state of the system, each computation of the system will eventually converge to a legal state in a finite number of steps and henceforth the system will continue to operate as specified. In the context of systems that seek to maintain optimal performance, their legal states must satisfy the desired optimality. Stabilization of these systems thus implies that when the current state of the system becomes suboptimal, for instance as a result of change in operating context, they will eventually resume optimal performance.

Designing stabilization to ensure optimality is however not an easy task. We attribute this to two facts: (a) Dynamically varying equilibrium: Equilibrium is often not determined by system itself, it is also affected by the operating context. Thus, when operating context changes, equilibrium can also vary. (b) Difficulty of detecting optimality: To determine optimality, all possible outcomes may need to be compared. When several equilibria coexist in a system, local detection of optimality can often leave the system in a local maximum (or minimum). In addition, optimality detection

is sensitive to noise. An output spike introduced by noise may be falsely regarded as a maximum.

Our goal in this work is to investigate new system design methodologies that exploit stabilization techniques to maintain optimality despite changes in the context of system operation.

2.1.1 Summary of our results

We study classes of stabilization problems for systems that achieve optimality. Based on a general system model, we provide three techniques that achieve stabilization.

1. For systems with observable external inputs, we add a stabilizing estimator to the system, using which the system infers the external inputs. Since optimality is determined by external inputs and system, when the system is controllable, we can achieve optimal performance by incorporating the estimated values of the external inputs.

2. For systems with unobservable external inputs, we suggest the design of MinMax controller strategies. We prove whenever a MinMax strategy leads to Nash equilibrium, stabilization (to the Nash equilibrium) is achieved. We illustrate this technique via an optimal catch time pursuit-evasion application where the design of a MinMax strategy for a pursuer guarantees stabilization irrespective of the evader's choice (or change) of strategy.

3. When outputs are observable, we suggest the design of a feedback control module. To eschew the difficulty of optimality detection, we add a stabilizing transformer to the system, using which an optimization problem is transformed into a

fixed point control problem. We illustrate this technique via an optimal energy-efficiency duty cycle MAC design that uses a feedback control algorithm to guarantee stabilization.

2.1.2 Related work

Reactive systems [36] (also called open systems) are systems whose role is to maintain an ongoing interaction with their environment, as opposed to calculating a final value upon termination. The literature on stabilization has considered reactive systems in a number of ways, of which we recall a few. In [27], an adaptive program is defined as a program that changes its behavior based on current state of environment. Operators that compose adaptive programs are developed in that work for both sequential and distributed program classes. A formal definition of stabilization in the presence of changes in operating context and general classes of faults is given in [4], in terms of closure and convergence. In this paper, we also use closure and convergence properties to define stabilizing optimality. [33] focuses on the adaptive stabilization of reactive distributed protocols; it shows that general reactive systems can be implemented in an adaptive way, i.e., the recovery time of stabilizing protocols can be proportional to the number of faults.

In [5], both termination and stabilization are investigated in message-passing systems relative to external input. In [10], stabilization of majority consensus is presented. [9] focuses on mutual exclusion. Both [5] [10] and [9] provide solutions for specific reactive systems. Control theories such as Lyapunov Theory are applied in [40], [18] to explore its application in stabilization. Some new classes of stabilization

such as Self-organizing [1] [20] and Selfish stabilization [17] are related with game theory approaches.

Compared with current work, we may identify two distinguishing features of our study: (a) Optimality: the set of legal states, which the system should converge upon starting from an arbitrary state, are characterized by one or more optimal properties of interest. (b) Reactivity: as the external inputs changes continuously, the corresponding equilibrium states can vary continuously as well.

2.1.3 System model

As shown in Figure 2.1, our system model consists of four major components: system controller, internal subsystem, external input, and faults.

- **Controller:** The system controller is the component that manages the computation of the system. If protocols or algorithms are added to the system to achieve stabilization, their execution is controlled by this component.

- **Internal subsystem:** The internal subsystem accepts commands or data from controller to change its behavior. However, its outputs may also be affected by when external inputs change and/or faults occur. The distinction between internal subsystem and controller is that internal subsystem is governed by its inherent mechanism. In control theory, internal subsystem is called "plant".

- **External inputs:** Based on their influence on the equilibrium states, we choose to classify the operating context into two parts: external inputs and faults. External inputs are defined as that part of the operating context that can directly impact system equilibrium. If external inputs are known, they together with controller, uniquely decide the equilibrium of system.

- **Faults:** In contrast to external inputs, faults affect system equilibrium arbitrarily or transiently. In this paper, we assume that when faults occur the net effect is to perturb the system into a potentially arbitrary state. The goal of stabilization then is to subsequently ensure that continued computation of the system will converge to a legal (i.e., optimal) state eventually (and ideally in a timely manner).

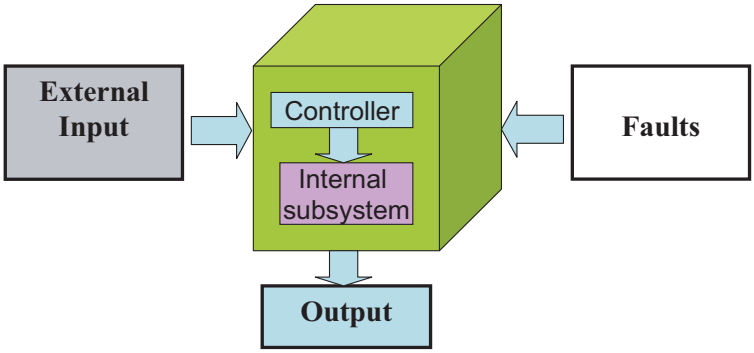

Figure 2.1: The system model

In summary, our model defines several key components that affect system equilibrium. Many systems such as control system can be generalized by this model.

Notations. We will use the following notations in the rest of the paper.

i: External inputs	I: The set of all possible external inputs
c: Controller value	C: The set of all possible controller values
y: Outputs	Y: The set of all possible outputs

A steady state means a stable condition that does not change over time or in which change in one direction is continually balanced by change in another. We assume

that in steady states, external inputs and the controller value determine the outputs through a system function, f, that is, $y = f(i, c)$.

Definition 1 *We say that the system is in an optimal state for a given choice of external inputs $i \in I$ and controller value c iff the system output $f(i, c) = \max\{f(i, c')|(c' \in C)\}$.*

Note that we use max to indicate optimization. However, min and other types optimization can be achieved through a general system function, f.

Definition 2 *We say that the system has computable optimality iff there exists a known function O such that for any choice of external inputs $i \in I$, the system output $f(i, O(i)) = \max\{f(i, c')|c' \in C\}$.*

Definition 3 *We say that external inputs are observable iff they can be measured directly or inferred from outputs.*

Definition 4 *We say that a system has stabilizing optimality iff it satisfies two conditions:*

- *Closure: if the system state is optimal, it will remain optimal, unless faults occur or external input changes.*
- *Convergence: upon starting from an arbitrary system state, every computation of the system will eventually converge to an optimal state.*

In the following sections, we will illustrate three techniques that focus on external inputs, controller, and output respectively to achieve stabilizing optimality.

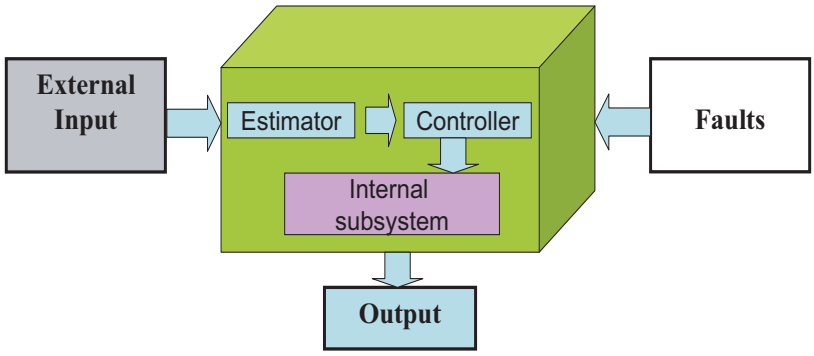

Figure 2.2: System stabilization with observable inputs

2.2 Stabilizing optimality via a stabilizing estimator

If external inputs are observable, an estimator can be added into the system, see Figure 2.2. This estimator would execute as an independent program and output a value that estimates the external inputs, by filtering noises in measurements of the input. Examples of estimator include Kalman filters, Wiener filters, and maximum likelihood estimators (MLE). If system has computable optimality as Definition 2, the controller can determine the equilibrium through the known function O, using the estimated value as opposed to the ideal external inputs. The estimator must however be self-stabilizing, to deal with situations when the estimators state itself is corrupted by fault occurrence.

Lemma 1 *If the added estimator is stabilizing and system has computable optimality, the system has stabilizing optimality.*

Using this approach, the stabilizing optimality of a system is achieved through continuous self-stabilizing estimation. (a) Closure: when a system is in equilibrium, if it has computable optimality and its estimation is correct, the system function O

will compute the same equilibrium. (b) Convergence: If the system is in an arbitrary state, continued computation of the estimator would eventually lead to the external inputs being estimated correctly, and the system function O will the re-establish the equilibrium.

However, the technique of adding stabilizing estimators is prone to several vulnerabilities. Firstly, the estimator must closely follow the dynamics of external inputs. Inaccurate or severely delayed estimation may produce suboptimal output. In addition, ensuring that the estimator is stabilizing can be nontrivial especially if it depends on system history which may also get corrupted. The implementation of the estimator should also be stabilizing, i.e., implementations of the estimator may introduce their own internal states. When faults happen, those internal states may be corrupted, and the estimator must recover from these as well. Furthermore, calculating the function $c^* = O(i)$ may not be trivial in all applications. Next, we will provide two techniques that avoid those shortcomings.

2.3 Stabilizing optimality via MinMax strategies

In game theory, the payoff for a player depends on the choices made by other players. We may model the interaction among external inputs, controller, and faults as players in a game, as shown in Figure 2.3. When external inputs and faults are unpredictable, the game becomes a non-cooperative one. In non-cooperative games, simultaneous actions from other players (external inputs and faults) are unobservable. Therefore, maintaining optimal performance based on an estimator is infeasible.

To deal with unobservable simultaneous actions from external inputs and faults, our technique is to design a controller based on MinMax strategy. MinMax is a

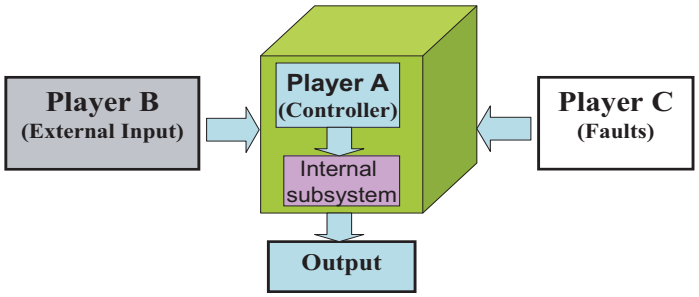

Figure 2.3: System model in terms of game theory

method in decision theory for minimizing the maximum possible loss. It can alternatively be thought of as maximizing the minimum gain (MaxMin). In our system model, the MinMax strategy for the controller would minimize the maximum loss introduced by external inputs and faults. This is a conservative approach, but stabilization can be achieved when MinMax strategies lead to Nash equilibrium, as explained below. Nash equilibrium is a solution concept for a game where a player has no gain by changing its own strategy unilaterally. Specifically, if no player can benefit by changing its strategy while the other players keep theirs unchanged, then the current set of strategies and the corresponding payoffs constitute a Nash equilibrium.

Theorem 1 *When a Controller adopts a MinMax strategy, the system has stabilizing optimality iff the MinMax strategy leads to Nash equilibrium.*

Proof: We combine the effects of external inputs and faults as one external player input e, where $e \in E$, E is the set of all possible inputs. The external player input e and controller value determine outputs through the system function f, that is, $y = f(e, c)$. Without loss of generality, we assume that the MinMax strategy of the

Controller is $\max_{c \in C} \{\min_{e \in E} f(e, c)\}$, while the corresponding strategy of external player is $\min_{e \in E} \{\max_{c \in C} f(e, c)\}$

1. If the MinMax strategy leads to Nash equilibrium, by the definition of Nash equilibrium, $\exists(e^*, c^*)$, such that for $\forall e \in E, c \in C, f(e^*, c) \leq f(e^*, c^*) \leq f(e, c^*)$. (a) Closure: when $(e, c) = (e^*, c^*)$, since Controller adopts a MinMax strategy, it will output c^*. For external players, they have no incentive to change to other strategies, because $f(e^*, c^*) \leq f(e, c^*)$. (b) Convergence: when $(e, c) \neq (e^*, c^*)$, since Controller adopts a MinMax strategy, it will output c^*. For external players, the best option is to get the strategy to output e^*, because when $f(e^*, c^*) \leq f(e, c^*)$, $\min_{e \in E} f(e, c^*) = f(e^*, c^*)$. Therefore, the system stabilizes.

2. If the MinMax strategy stabilize to a state with outputs (e', c'), according to definition of MinMax strategies, $\min_{e \in E} f(e, c') = f(e', c') = \max_{c \in C} f(e', c)$. Because of convergence, we have $\forall e \in E, c \in C, f(e', c) \leq f(e', c') \quad \wedge \quad f(e', c') \leq f(e, c')$. Therefore, (e', c') is a Nash equilibrium.

■

MinMax strategy and Nash equilibrium are two different solution concepts in game theory. MinMax strategy cannot guarantee Nash equilibrium, while Nash equilibrium is not necessarily derived from MinMax strategy. As far as we know, Theorem 1 is the first result to link them together via stabilization.

2.3.1 Case study: Pursuit-Evasion Game (PEG)

We illustrate the MinMax strategy technique via a target capture game, one application based on our 1000+ nodes Exscal project in DARPA-NEST program [6][14].

In this game, the pursuer (controller) tries to catch the evader (external input) as soon as possible, while the evader tries to prolong the time taken to be caught (catch time) T_c, the output of this system. This is a zero-sum game. Zero-sum describes a situation in which a player's gain or loss is exactly balanced by the losses or gains of the other player(s).

Model: We denote the current time as t. The list of state variables in this game is the following:

- $L_p(t) = (x_p, y_p)$: pursuer location, $\quad L_e(t) = (x_e, y_e)$: evader location
- V_p: the maximal pursuer speed, $\quad V_e$: the maximal evader speed

When evader is caught at time t_c, the catch time is calculated by: $T_c = t_c - t$. The payoff for pursuer in this game, $\mathcal{J}_p$, is defined as: $\mathcal{J}_p = -T_c$, while the payoff for evader in this game, $\mathcal{J}_e$, is defined as : $\mathcal{J}_e = T_c$. We define the distance between the pursuer and the evader as: $dist(t) = \|L_p(t) - L_e(t)\| = \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2}$.

MinMax strategies: Every Δt time interval, both the pursuer and the evader should respectively move to the location based on their own MinMax and MaxMin strategy. To design a MinMax strategy for the pursuer, we should consider the best actions of an evader. Figure 2.4 describes the pursuer strategy (proof details can be found in [13] or Chapter 3). Literally, the pursuer strategy is to move towards to evader with full speed in the next time interval $[t, t + \Delta t]$.

Stabilization of strategies: The MinMax strategy of pursuer provides robustness against uncertainty of evader strategy:

Input: (x_e, y_e)
Output: $(x_p(t+\Delta t), y_p(t+\Delta t))$
Parameter: V_p, V_e
Internal state: (x_p, y_p)
$x_p(t+\Delta t) = x_p + \frac{x_e - x_p}{dist(t)} \cdot V_p \Delta t$
$y_p(t+\Delta t) = y_p + \frac{y_e - y_p}{dist(t)} \cdot V_p \Delta t$

(a) Pursuer strategy

Input: (x_p, y_p)
Output: $(x_e(t+\Delta t), y_e(t+\Delta t))$
Parameter: V_p, V_e
Internal state: (x_e, y_e)
$x_e(t+\Delta t) = x_e + \frac{x_e - x_p}{dist(t)} \cdot V_e \Delta t$
$y_e(t+\Delta t) = y_e + \frac{y_e - y_p}{dist(t)} \cdot V_e \Delta t$

(b) Evader strategy

Figure 2.4: Pursuer strategy and Evader strategy

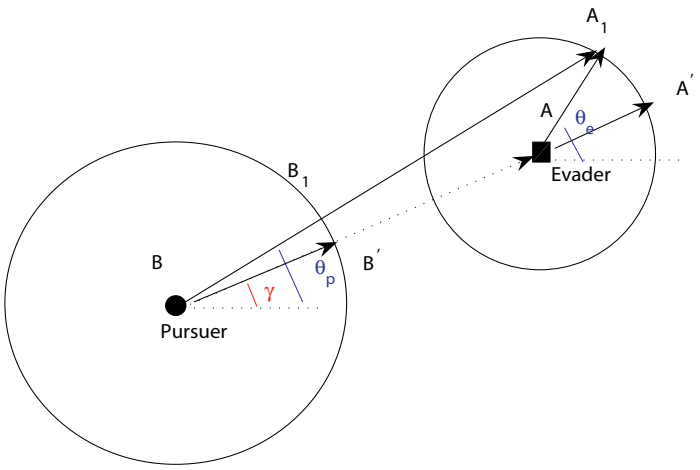

Figure 2.5: Pursuit-evasion game for target capture

Theorem 2 *In PEG, if the pursuer follows the actions described by Figure 2.4, the following inequality holds even if the evader changes its strategy.*

$$T_c(t) \geq T_c(t+\Delta t) + \Delta t, \qquad T_c(t) = \frac{dist(t)}{V_p - V_e}$$

Proof: As shown in Figure 2.5, after time interval Δt, the pursuer will move to B'. In this case, the best option for evader is to move to A', because only in this case does the following equation holds: $T_c(t) = T_c(t+\Delta t) + \Delta t$. Otherwise, assuming

the evader moves to any location A_1 other than A', then by triangle inequality: $\overline{BA} + \overline{AA'} > \overline{BA_1} \Rightarrow \overline{BA} + \overline{AA'} > \overline{B_1A_1} + \overline{BB_1}$. In other words, $dist(t) > dist(t + \Delta t) + (V_p - V_e)\Delta t \Rightarrow T_c(t) > T_c(t + \Delta t) + \Delta t$.

■

Similarly, the following inequality also holds:

Theorem 3 *In PEG, if the evader follows the actions described by Figure 2.4, the following inequality holds, even if the pursuer changes its strategy.*

$$T_c(t) \leq T_c(t + \Delta t) + \Delta t, \qquad T_c(t) = \frac{dist(t)}{V_p - V_e}$$

The pursuer strategy and the evader strategy form a Nash equilibrium (as we will prove next) that is based on MinMax strategy, thus they achieve stabilizing optimality of the game.

Theorem 4 *In PEG, if the pursuer follows the MinMax strategy described by Figure 2.4, the system has stabilizing optimality.*

Proof: Firstly, we prove that the MinMax strategies of the pursuer and the evader lead to Nash equilibrium. We denote the possible strategies of pursuer and evader as $(a_p, a_e), a_p \in P, a_e \in E$, and the actions in Figure 2.4 as (a_p^*, a_e^*) separately. From Theorem 2, we have: $\min_{a_e \in E} T_c(a_p^*, a_e) = T_c(t) = T_c(a_p^*, a_e^*)$. From Theorem 3, we have: $\max_{a_p \in P} T_c(a_p, a_e^*) = T_c(t) = T_c(a_p^*, a_e^*)$. Therefore, $T_c(a_p, a_e^*) \leq T_c(a_p^*, a_e^*) \leq T_c(a_p, a_e^*)$. The MinMax strategies of the pursuer and the evader thus lead to Nash equilibrium. From Theorem 1, Nash equilibrium leads to the stabilization of system.

■

Stabilization of implementation of strategies: When the strategy is implemented as a program, which may introduce additional states, to ensure the stabilization of the system, we must consider the stabilization of the implementation. In our case study, the state information–location of pursuer and evader can be corrupted. The optimal pursuer strategy is however based solely on the latest location information, and is thus independent of history information. If the state information is corrupted, the pursuer should continue to query for the latest location and move according to its optimal strategy. After it receives the correct location information, Nash equilibrium is reestablished. In other words, it is straightforward to implement this strategy as a program that is stabilizing.

2.3.2 Discussion

When MinMax strategies do not suffice to derive Nash equilibrium, mixed strategies can be applied. A mixed strategy is to choose randomly between different strategies based on calculated weighted possibilities. The celebrated Minmax theorem states that a solution for mixed MinMax strategies for two players always exists and the solution is always a Nash equilibrium. Therefore, by using mixed MinMax strategies, we can always obtain stabilizing solutions. We suggest that MinMax is probably the best available strategy for a wide range of zero-sum games. MinMax has been applied in many applications to deal with uncertainties. In [50] the MinMax approach to the design of systems that are robust with respect to modeling uncertainties is studied, and the efficacy of the methods proposed for a general game is validated for the case of problems of matched filtering, Wiener filtering, quadratic detection, and output

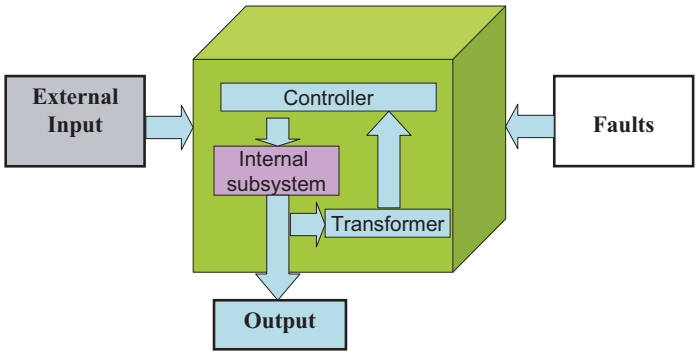

Figure 2.6: System stabilization via feedback control

energy filtering. We also applied MinMax to a much more complex application–asset protection game in [14] and Chapter 3.

2.4 Stabilizing optimality via feedback control

When external inputs are difficult to estimate or are unobservable, but outputs are measurable, one approach to designing system optimality is via output feedback as shown in Figure 2.6. There are, however, two major issues with this approach: (a) Difficulty of determining equilibrium: As we noted before, equilibrium is not predefined in many cases. It may therefore necessitate the use of optimization procedures. These procedures may however be of high complexity or be error-prone when equilibrium varies. (b) Stability of feedback loop: This may be difficult to achieve owing to delay and uncertainty of feedback loop. Uncertainty is inherent when the system model is inaccurate or when faults occur.

To eschew the difficulty of finding equilibrium, we add a stabilizing **Output Transformer** to the feedback loop. An output transformer is a function from outputs into optimality measurements (OM). It is found in several cases that the Input-OM

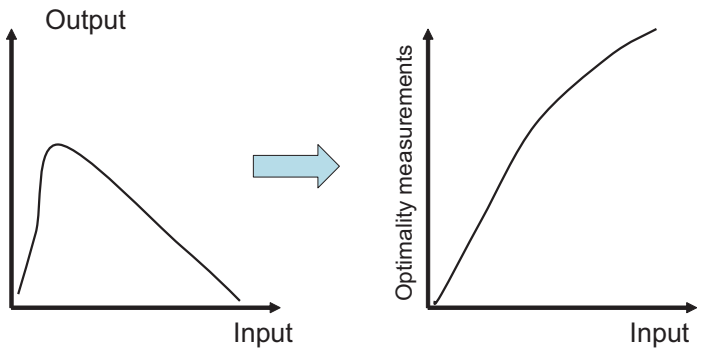

Figure 2.7: Transform output into optimality measurements

relationship is much simpler than Input-Output relationship, as is illustrated in Figure 2.7, which makes the design of stable feedback control based on OM simpler than that based on outputs.

The design of an output transformer depends on the particular application, but the following two conditions are sufficient for its use:

1. **Monotonicity:** The input-OM function is monotonic (increasing or decreasing). This condition deals with the possibility that when output is not optimal for a given input, the output by itself may not suffice to decide whether the input is less than the optimal input or more than the optimal input. Monotonicity of the Input-OM function provides definite feedback to the input.

2. **Uniqueness:** When optimality is obtained, OM is a fixed value or in a value in a narrow region, and is independent of the external inputs. This property provides robustness against varying external inputs that affect equilibrium.

Theorem 5 *If a stabilizing output transformer satisfies* **Monotonicity** *and* **Uniqueness**, *the system has stabilizing optimality.*

Proof: Let M be the Input-OM function, and $m = M(i)$ where i is the input and m is the OM. Let the fixed OM value where optimality is obtained be denoted by m^*.

Firstly, we assume that $M(i)$ is an increasing function. The simple feedback control algorithm shown in following stabilizes to the fixed value m^* (The algorithm stabilizes to a narrow region as presented in Figure 2.12).

$$\textbf{if } (m > m^*) \quad i = i - \Delta i;$$
$$\textbf{else if } (m < m^*) \quad i = i + \Delta i;$$

The following conditions are satisfied: (a) Closure: When the system satisfies optimality, $m = m^*$. In this case, the control algorithm leaves the input i unchanged, and so the output also stays unchanged. When $m = m^*$ is obtained, the system satisfies optimality which, by Uniqueness, is independent of the input. Therefore, optimality is closed. (b) Convergence: when the system state is not optimal, $m \neq m^*$. The control algorithm keeps changing the input i until $m = m^*$ eventually.
In the case that $M(i)$ is a decreasing function, similar control algorithms can be designed. Thus, the system can be stabilized.

■

We emphasize that Monotonicity and Uniqueness are not necessary conditions, so other forms of output transformers may also exist for achieving stabilizing optimality. Their control algorithm would likely be more complex.

2.4.1 Case study: duty cycle adaptation

We illustrate the stabilizing output transformer technique via a protocol for duty cycle adaptation. Network longevity is a key requirement for battery powered wireless

sensor network. This suggests that node radios must be scheduled to switch off most of the time, i.e., to achieve as low a duty cycle as possible while still accommodating the network traffic. Analytical results [15] indicate that different traffics require different duty cycles to achieve optimal energy efficiency. The goal of duty cycle adaptation then is to provide sufficient but minimum duty cycle for accommodating varying traffic.

Matching the duty cycle of the system to the load is a challenge problem and achieving its stability is even more difficult. If the duty cycle is lower than required, higher collision or sender buffer overflow can happen; if the duty cycle is higher than required, energy is wasted on idle listening. Changing the duty cycle may change the link reliability and thus the routing structure, which changes the traffic. However, traffic may affect duty cycle in return. Therefore, oscillation may happen in this control loop. Stabilizing duty cycle adaptation is a critical requirement for low duty cycle systems.

For the sake of presentation, let us consider a 6 node wireless sensor network example to illustrate our design. (More complex network deployment and traffic pattern can be found in [12] and Chapter 5.) All of the nodes are within communication range of each other. All 5 senders transmit with the same rate randomly when receiver is up as shown in Figure 2.8.

Transformer design: The list of state variables in this protocol is the following:

- $D(t)$: Controller value: the receiver duty cycle at time t
- $I_i(t) = I(t)$: External input: traffic from sender i at time t

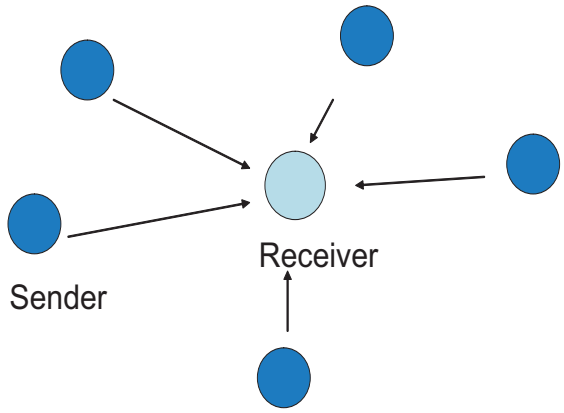

Figure 2.8: A wireless network with 5 senders and 1 receiver

- $P_i(t) = P(t)$: Input to the internal system: the probability of sender i transmitting when receiver is up at time t
- $O(t)$: output traffic at time t, note that it includes only the goodput.
- $C(t)$: output collisions at time t

The input to the internal system, $P_i(t)$, is: $P_i(t) = \frac{I_i(t)}{D(t)} = \frac{I(t)}{D(t)} = P(t)$. Therefore, the output traffic is: $O(t) = 5 \cdot P(t) \cdot (1 - P(t))^4 \cdot D(t)$. Energy efficiency is defined as the ratio of output traffic and receiver duty cycle: $E_e(t) = \frac{O(t)}{D(t)} = 5 \cdot P(t) \cdot (1 - P(t))^4$.

As shown in Figure 2.9, when $P(t)$ increases, energy efficiency increases before reaching a maximum, and then decreases thereafter. We define **activity ratio** as the optimality measurement of this case study. **Activity ratio** is measured through output, and is defined as ratio of the total activity versus receiver duty cycle.

$$A(t) = \frac{O(t) + C(t)}{D(t)} = 1 - (1 - P(t))^5$$

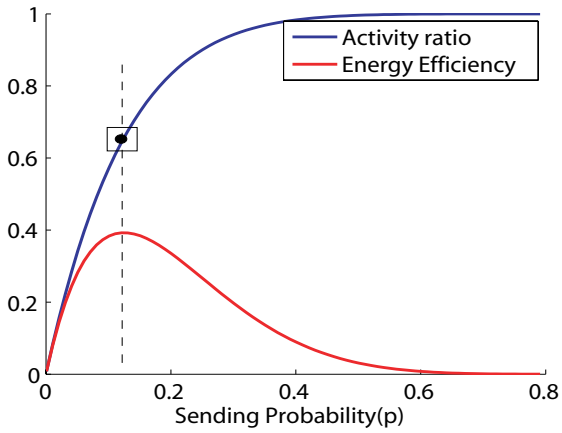

Figure 2.9: Activity ratio & energy efficiency as a function of traffic level

Note that activity ratio is equal to the probability of the channel being non-idle. By using the activity ratio as the optimality measurement [12], we have transformed an optimization problem into a fixed point feedback control problem, as shown in Figure 2.9.

Basic feedback protocol algorithm: Before we present the algorithm, we introduce a Proposition firstly [12].

Proposition 1 *When activity ratio converges to within a small region* $[A_{min}, A_{max}]$*, optimal energy efficiency is obtained.*

In this section, we focus on providing a feedback control mechanism to ensure the activity ratio converges to within a small region $[A_{min}, A_{max}]$, wherein the optimal duty cycle is obtained.

Figure 2.10 presents a simple generic control protocol by using Multiple-Increasing-Multiple-Decreasing (MIMD). Let:

d_r be the receiver duty cycle, A_{max} be the maximum activity ratio,

Input: $r_a, d_r(k)$ **Output:** $d_r(k+1)$ **Parameter:** $A_{min}, A_{max}, \alpha, \beta$
if $(r_a > A_{max}) \Rightarrow d_r(k+1) = d_r(k) + d_r(k) * \alpha$;
else if $(r_a < A_{min}) \Rightarrow d_r(k+1) = d_r(k) - d_r(k) * \beta$;

Figure 2.10: Basic adaptive non-stabilizing duty cycle protocol

A_{min} be the minimum activity ratio, α be duty cycle increasing rate,

β be duty cycle decreasing rate;

Although the invariant for this simple program is $r_a \in [A_{min}, A_{max}]$, the program cannot guarantee stabilization, as we show in the next section.

Stabilization of Feedback Control Protocol: Although MIMD achieves better convergence and energy efficiency, the method does not converge. Unless α and β are chosen carefully, it is possible that the activity ratio r_a may oscillate from below A_{min} to above A_{max}, as is shown in Figure 2.11.

To prevent oscillation, we add stabilization into the basic feedback control protocol, as shown in Figure 2.12. The main idea is that when a transition from $r_a > A_{max}$ to $r_a < A_{min}$ or from $r_a < A_{min}$ to $r_a > A_{max}$ happens, the rate of receiver duty cycle change decreases. In this protocol, the variable $lstate$ is used to indicate different states:

INC: duty cycle is increasing DEC: duty cycle is decreasing

$OVER$: transitioning from $< A_{min}$ to $> A_{max}$ NOR: steady state

LOW: transitioning from $> A_{max}$ to $< A_{min}$

Variables δ_i and δ_d denote the step size (rate) of increasing or decreasing duty cycle. This protocol provides guarantee of stabilization:

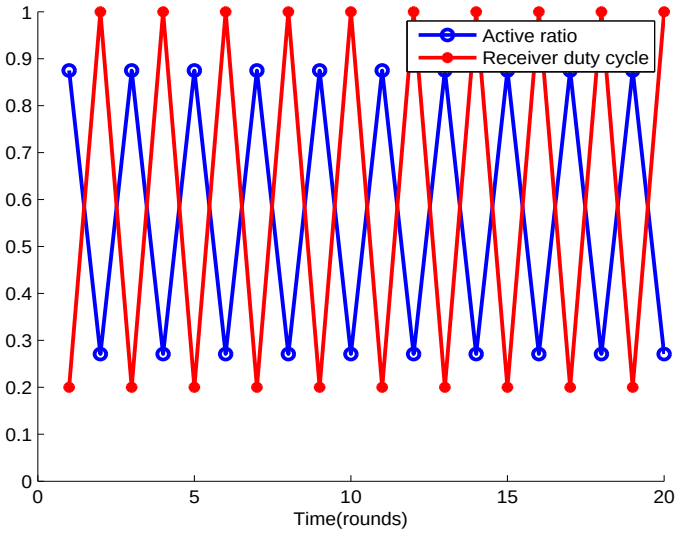

Figure 2.11: Oscillation in the basic adaptive duty cycle algorithm

Input: $r_a, d_r(k)$
Output: $d_r(k+1)$
Parameter: $A_{min}, A_{max}, \alpha, \beta$
State: $lstate, \delta_i, \delta_d$
if $(r_a > A_{max})$
  if $(lstate = INC) \| (lstate = NOR)$
    $d_r(k+1) = d_r(k) + d_r(k) * \alpha;$
    $\delta_i = d_r(k+1) - d_r(k);$
  if $(lstate = DEC) \| (lstate = OVER)$
    $\delta_d = \delta_d/2;$
    $d_r(k+1) = d_r(k) + \delta_d;$
    $lstate = OVER;$
  if $(lstate = LOW)$
    $\delta_i = \delta_i/2;$
    $d_r(k+1) = d_r(k) + \delta_i;$
else if $(r_a < A_{min})$
  if$(lstate = DEC)$
    $d_r(k+1) = d_r(k) - d_r(k) * \beta;$
    $\delta_d = d_r(k) - d_r(k+1);$
  if $(lstate = OVER)$
    $\delta_d = \delta_d/2;$
    $d_r(k+1) = d_r(k) - \delta_d;$
  if $(lstate = OVER)$
    $\delta_d = \delta_d/2;$
    $d_r(k+1) = d_r(k) - \delta_d;$
  if $(lstate = INC) \| (lstate = LOW)$
    $\delta_i = \delta_i/2;$
    $d_r(k+1) = d_r(k) - d_r(k) * \delta_i;$
    $lstate = LOW;$
  $lstate = DEC;$
else
  $lstate = NOR$

Figure 2.12: The stabilizing optimality protocol for adaptive duty cycle

Theorem 6 *When the incoming traffic is steady, the adaptive duty cycle protocol described in Figure 2.12 stabilizes to an activity ratio in* $[A_{min}..A_{max}]$*, by which optimal efficiency is obtained.*

Proof: The invariant for this program is $r_a \in [A_{min}, A_{max}]$, the program is closed. Next, we will prove its convergence.

For a network with η senders, every node transmits with a certain duty cycle d_s^i. Note the duty cycle of different senders may be different, so we use a vector D_s to represent: $D_s = [d_s^1, d_s^2, ..., d_s^\eta]$. The relationship between the receiver activity ratio r_a and the receiver duty cycle d_r can be expressed as a function: $r_a = f(D_s, d_r)$. When D_s is fixed, function $f(D_s, d_r)$ is a decreasing function. In other words, when receiver duty cycle increases, activity ratio decreases when incoming traffic is fixed. For instance, in the case of random traffic, $r_a = f(D_s, d_r) = 1 - (1 - p_i)^\eta$, where $p_i = \max\{1, \frac{d_s^i}{d_r}\}$. We use Lyapunov theorem to prove convergence. Let the Lyapunov function be: $E_l = \min\{\|r_a - A_{min}\|, \|A_{max} - r_a\|\}$. When no transition happens, either $\|r_a - A_{min}\|$ or $\|A_{max} - r_a\|$ is a decreasing function, guaranteed by the decreasing function $r_a = f(D_s, d_r)$. When a transition happens, the protocol guarantees that an infinitely smaller step size of duty cycle is either added or subtracted. Function E_l is still a decreasing function. By the well known Lyapunov theorem, this protocol stabilizes to a static point. Given the continuity of the function $r_a = f(D_s, d_r)$, the activity ratio stabilizes to a point in $A_{min}..A_{max}$.

■

The algorithm described in Figure 2.12 is generic, so it can be applied in any system to achieve stabilization, when a transformer as described in Theorem 5 is added into system.

2.5 Conclusion

In this work, we formulated optimality maintenance in dynamical system in terms of the standard notion of stabilization. We focused on three techniques — estimator-based, MinMax controllers that lead to Nash equilibrium, and transformer-based — for stabilization of dynamical systems. Each of these techniques relates to a different aspect of the system, respectively, its input, its controller, and its output.

One advantage of the formulation in terms of stabilization is the appreciation (in Lemma 1, Theorem 1, and Theorem 5) that the components designed for dynamical systems to maintain optimality should themselves be stabilizing. Likewise, the concrete implementations of these components should be stabilizing.

We illustrated the MinMax and transformer based techniques via case studies. Although these examples do include distributed computing, and indeed the latter can achieve stabilization in a network by independent local stabilization of its nodes, it is apparent that several advanced methods studied in the theory of stabilization for composition of stabilizing components can be exploited to deepen these techniques. As such, we find that these techniques deserve to be substantially further studied by the community.

CHAPTER 3

MINIMAX EQUILIBRIUM OF NETWORKED DIFFERENTIAL GAMES

In this chapter, We focused on two typical differential games: pursuit-evasion game for target capture and an "asset protection" game. We formulate optimal pursuit control strategies in the presence of network effects, assuming that target track information has been established locally in the sensor network. We adapt ideas from the theory of differential games to networked games—including ones involving non-periodic track updates, message losses, and message delays—to derive optimal strategies, bounds on the information requirements, and scaling properties of these bounds. We show the inherent stabilization features of our pursuit strategies, both in terms of implementation as well as the strategies themselves.

3.1 Introduction

Sensor network technology has enabled new surveillance systems [2, 6], where sensor nodes equipped with processing and communication capabilities can collaboratively detect, classify and track targets of interest over a large area. These surveillance systems make it viable to use the state information collected through the sensor network to guide mobile agents to achieve surveillance goals such as target capture

and asset protection. A sensor network surveillance system has the advantage of giving the mobile agents access to the global information so that they can optimize their motion for pursuit tasks, as opposed to resource-intensive search and map building tasks. That said, using sensor networks to implement "active" surveillance strategies introduces new challenges as well. Target track information obtained by local processing of sensor information needs to be routed to mobile agents through multi-hop communication links, which results in delays, message losses and random arrival times of the packets carrying track information. In addition, those sensor networks are usually deployed in harsh environments, state information may be corrupted, which also necessitates the stabilization of strategies.

In previous work, Schenato et. al. [43] studied a pursuit-evasion game application using sensor networks. They consider a detailed system model with periodic time updates and present models of vehicle dynamics and uncertainty in track information. Sensor network measurements are assumed to be fused at local stations to produce track information [39]. Evader assignment and pursuer control strategy is calculated at the base station and then communicated to the pursuer agents. Network effects in communicating this information to the pursuer agents and communicating pursuer locations back to the base station are not considered. Within this framework, they derive a series of algorithms to coordinate the pursuers so as to minimize the time-to-capture of all evaders.

In this work, we concentrate on the formulation of optimal pursuit control strategies despite network effects. We assume target track information has been established through local fusion of sensor data. This track information is communicated through the multi-hop wireless network infrastructure to pursuer agent, which calculates an

optimal pursuit strategy based on evader's state and its own state. We adapt ideas from theory of differential games to networked games in the presence of non-periodic track updates, message loss and delays to derive optimal strategies, bounds on their information requirements and the scaling properties of these bounds. We also consider the stabilization issues in the design and implementation of these pursuit strategies. In summary, we show:

1. Pursuer agents should dictate the information refresh rate based on the requirements of the pursuit strategies.

2. Network delays and update periods should scale linearly with the pursuer-evader distance to guarantee the existence of optimal min-max pursuit strategies leading to Nash equilibria.

3. If those derived communication conditions are satisfied, the pursuit strategies do not need to change even if the evader strategy is chosen otherwise or if the state of the network is transiently perturbed.

Differential games entail the study of dynamic interactions between rational agents with conflicting interests [8]. The theory of differential games combines solution concepts of game theory with control theory formalism to formulate optimal feedback strategies for the players. Pursuit-evasion games are natural applications of the theory of differential games and are extensively studied by Isaacs in his seminal work [30]. In the literature, pursuit-evasion games are traditionally modeled as continuous-time perfect information games where the players have access to the global state of the game at all times without delays. In contrast, in this paper, we study the optimal

strategies for pursuit using a communication-constrained network structure. We investigate two representative pursuit-evasion games. One is a classical pursuit-evasion game for target capture, where pursuers try to catch the evader as soon as possible; the other is called "asset protection game"(also called Lifeline Game) where pursuers try to protect a linear target by intercepting the evaders as far as possible from the target. These games have practical applications in real world applications and the techniques introduced in this paper can be generalized to a wide variety of differential games.

The asset protection game in sensor network was first investigated in [14], by formulating a novel min-max equilibrium concept for networked games with delay and discrete time updates. The proposed equilibrium concept considers an omniscient opponent with compete access to state information without delays that can maximally exploit the delays and the inter-sample periods in the information updates. [16] later extended the model by combining an n-hop disk model abstraction of a sensor network to model delay and packet loss. They computed a probabilistic barrier that splits the state space of the game into an escape zone and a capture zone.

In this chapter, we concentrate on traditional pursuit-evasion target capture game and the asset protection game, by considering discrete time updates and communication constraints. In addition, we also discuss the stabilization of pursuer strategies in the presence of suboptimal evader strategy and state corruption.

The rest of this paper is organized as follows. In Section 3.2, we introduce the pursuit-evasion game for target capture. In Section 3.3, we introduce the pursuit-evasion game for asset protection. In both sections, we first introduce the game

model and review the optimal min-max strategies, then we derive the optimal strategies under network communication constraints, and also lower bounds on network performance requirements. Next section, we discuss the stabilization issues in these strategies. Finally, we conclude with the results of experimental studies and extensions of our results.

3.2 Pursuit-Evasion Game for target capture

3.2.1 Problem definition

We first consider a game between two players: a single pursuer and a single evader as shown in Figure 3.1. (For many n pursuer – n evader games, the min-max solution can be reduced to n two player games, by first solving the combinatorial problem of optimal pairing using the value function of the two player game. We discuss the extension to multiple pursuer and evader games in Section 3.6.2.) In this target capture game, the pursuer tries to catch the evader as soon as possible, while the evader tries to avoid being caught or to prolong time to being caught. The state of the game is determined by the two dimensional coordinates of the pursuer and evader, $x = \{x_p, y_p, x_e, y_e\}$. We assume that each player travels at constant speed v_p and v_e and controls the direction of its motion, denoted by θ_p and θ_e. There are no obstacles in the environment to constrain the movement of players. Players employ feedback control strategies $(u_p(x(t)), u_e(x(t)))$ which determine their direction of motion given the current state. The state space can be reduced to two dimensions by defining relative coordinates, $x_r = x_e - x_p$ and $y_r = y_e - y_p$. The state vector x evolves according to:

$$\dot{x} = \frac{\partial}{\partial t} \begin{bmatrix} x_r \\ y_r \end{bmatrix} = f(x, \theta_p, \theta_e) = \begin{bmatrix} v_e \cos(\theta_e) - v_p \cos(\theta_p) \\ v_e \sin(\theta_e) - v_p \sin(\theta_p) \end{bmatrix}$$

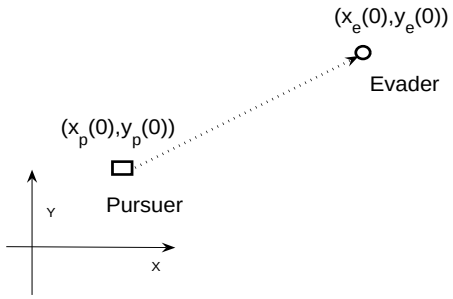

Figure 3.1: The pursuit-evasion game for target capture game

A catch is said to happen when $x_r^2 + y_r^2 < r^2$, where r is the catch radius. In the following, we consider the limiting case of $r \to 0$. (The effect of finite catch radius is discussed in Section 3.6.) Starting from the initial condition x_0 and time 0, if the control strategies $(u_p(x), u_e(x))$ satisfy the catch condition at time T then the payoff is given by $\mathcal{J}(u_p, u_e, x_0) = T$. T is the time when evader is caught. The game is zero-sum, so the pursuer's goal is to minimize $\mathcal{J}$ whereas the evader's goal is to maximize $\mathcal{J}$. Min-max optimal feedback strategies $u_p^*(x), u_e^*(x)$ are defined by the saddle condition:

$$\mathcal{J}_{u_p}(u_p, u_e^*, x_0) \leq \mathcal{J}(u_p^*, u_e^*, x_0) \leq \mathcal{J}_{u_e}(u_p^*, u_e, x_0) \tag{3.1}$$

We also note that the min-max optimal strategy pair $(u_p^*(x), u_e^*(x))$ is also the Nash equilibrium [38] for this zero-sum game, where none of the players have an incentive to change their strategy unilaterally given the rival is maintaining its strategy.

For each initial condition x_0, the value of the game is defined as $V(x_0) = \mathcal{J}(u_p^*, u_e^*, x_0)$. The value function is uniquely defined irrespective of the number of min-max strategy pairs that satisfy the saddle point property in 3.1. In this paper, we limit our discussion to initial states x_0 with finite positive value $V(x_0)$ and to games where the speed of the pursuer is greater than the speed of the evader.

3.2.2 Optimal pursuit under perfect information

The value function and the associated optimal strategies for the game defined in Section 3.2.1 can be derived using the Isaac conditions, a form of Hamilton-Jacobi-Bellman equations of optimality. Here we choose to present geometric solutions to provide intuition for the pursuit-evasion game under network effects.

Theorem 7 *If the ratio of the pursuer speed v_p to the the evader speed v_e, α, is larger than 1, then the min-max optimal strategy for the evader and pursuers is given by:*

$$\theta_e(x_0) = \gamma, \qquad \theta_p(x_0) = \gamma \tag{3.2}$$

where $\gamma = \tan^{-1}(\frac{y_r}{x_r})$ and $V(x_0) = \frac{\sqrt{x_r^2+y_r^2}}{(\alpha-1)*v_e}$. Equivalently, the pursuer moves toward the evader directly until catching the evader, while evader moves in the same direction to prolong the catching time.

Proof: Given the current location of the evader and pursuer, the set of points that the evader can reach before the pursuer is given by the well known Appolonius circle. The min-max optimal strategies for both the pursuer and the evader are to go directly to the boundary point.

As shown in the Figure 3.2, the current pursuer location is B and the current evader location is A. For any time interval dt, the maximum distance of pursuer and

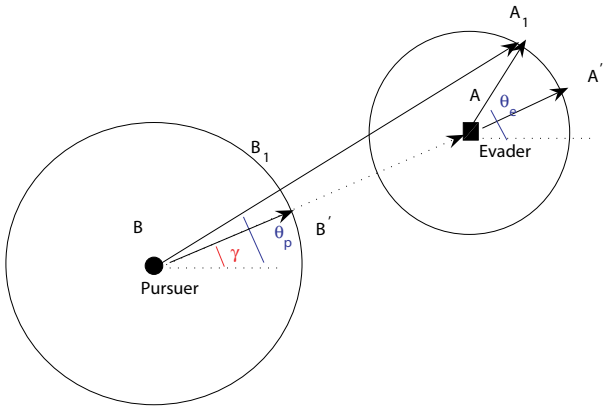

Figure 3.2: The pursuer and evader game for target capture game

evader can move are $v_p * dt$ and $v_e * dt$. All the possible locations are on the circle around current pursuer and evader locations. Point B' is the crosspoint of the circle around the pursuer and line BA. Point A' is crosspoint of the circle around the evader and the other side of line BA. We claim the point B' and A' are min-max optimal strategy pair for pursuer-evader movement during time dt. In other words, $\overline{B'A'}$ is the min-max distance after this movement.

Assume the evader moves to any other location $A_1, A_1 \neq A'$, the best strategy for pursuer is to move toward A_1, i.e., to point B_1 on pursuer circle. By Triangle Inequality,

$$\overline{BA} + \overline{A_1A} > \overline{BA_1} \Rightarrow \overline{BB'} + \overline{B'A} + \overline{A_1A} > \overline{BB_1} + \overline{B_1A_1}$$

because $\overline{BB_1} = v_p * dt = \overline{BB'}$ and $\overline{AA_1} = v_e * dt = \overline{AA'}$,

$$\Rightarrow \overline{B'A} + \overline{AA'} > \overline{B_1A_1} \Rightarrow \overline{B'A'} > \overline{B_1A_1}$$

So $\overline{B'A'}$ is the longest distance the evader can achieve. The best strategy for pursuer is to move towards the evader, while the evader tries to escape the pursuer.

$$\theta_e(x_0) = \gamma, \qquad \theta_p(x_0) = \gamma, \qquad \gamma = \tan^{-1}(\frac{y_r}{x_r})$$

On the other hand, if the evader moves to location A', the best strategy for the pursuer is to move directly toward A'.

Because the pursuer speed v_p is α times of the evader speed v_e, assume the final catch point is C, then $\overline{BC} = \alpha\overline{AC}$, and $\overline{BC} = \overline{BA} + \overline{AC}$, so we get:

$$\overline{AC} = \frac{\overline{BA}}{\alpha - 1} = \frac{\sqrt{x_r^2 + y_r^2}}{\alpha - 1} \Rightarrow V(x_0) = \frac{\overline{AC}}{v_e} = \frac{\sqrt{x_r^2 + y_r^2}}{(\alpha - 1) * v_e}$$

■

3.2.3 Optimal pursuit under communication constraints

Sampling rate requirements of the optimal pursuit strategy

In Section 3.2.2, we assumed that the global state is available to the pursuer at all times. This is an unrealistic assumption for a sensor network implementation where the information can be provided only at discrete time intervals. In this section, we derive the sampling rate requirements of the optimal strategy and show that it is inversely proportional to the relative distance between the pursuer and evader. The result is particularly important for sensor network implementations using resource constrained nodes, because it informs how the information data rate can be reduced based on the state of the game so as to conserve the energy and bandwidth resources of the network. Again, we use the min-max solution concept to formulate a robust pursuit strategy that will perform satisfactorily irrespective of evader motion. To design for the worst possible case of evader motion, we assume the pursuer has perfect

information about the location of the evader and the sampling period. The sampling period is then chosen such that the evader does not benefit from switching from the optimal direction given in Theorem. 7, although the evader's deviation will be detected by the pursuer after the sampling period interval.

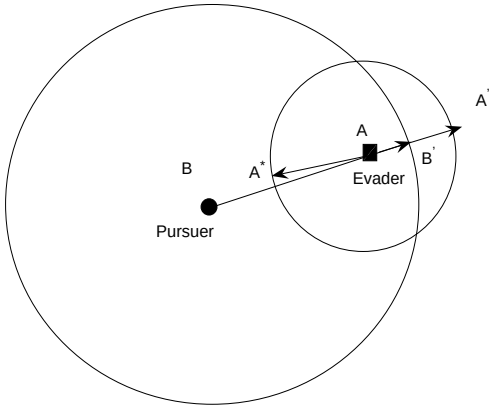

Figure 3.3: The pursuer and evader game for target capture

Theorem 8 *The evader does not deviate from its min-max equilibrium strategy if and only if sampling period T_{sample} with respect to the distance d_{pe} between the pursuer and evader satisfies:*

$$T_{samp}(d_{pe}) < \frac{d_{pe}}{v_p} \tag{3.3}$$

In other words, the sampling period should decrease proportionally with decreasing distance between evader and pursuer to guarantee that the evader does not have an incentive to deviate from its strategy.

Proof: Assume the pursuer moves first. For any time interval T_{sample}, ($v_p * T_{sample} < \sqrt{(x_r)^2 + (y_r)^2}$), as shown in Figure 3.2, the pursuer will move to B', where B' is the crosspoint of the circle around the pursuer and line BA. The evader can move to any location in the circle which is centered at A and has the radius $v_e * dt$. It will choose the location A_1 that has maximum $\overline{B'A_1}$. By Triangle Inequality, A' is the location that maximize the $\overline{B'A_1}$:

$$\overline{B'A} + \overline{AA_1} \geq \overline{B'A_1}$$

because $\overline{AA'} = v_e * T_{sample} \geq \overline{AA_1}$,

$$\Rightarrow \overline{B'A'} = \overline{B'A} + \overline{AA'} \geq \overline{B'A} + \overline{AA_1} \geq \overline{B'A_1} \Rightarrow \overline{B'A'} \geq \overline{B'A_1}$$

This means A' is the location that maximize $\overline{B'A_1}$.

However, if $v_p * T_{sample} \geq d_{pe}$, the evader can find a better location such that $\overline{B'A^*} > \overline{B'A'}$ as shown in Figure 3.3.

■

Effect of message losses

From the previous sampling rate analysis, to guarantee the optimum evader capture, the information must be updated before the pursuer reaches the previous evader location. For perfectly reliable communication links, this can be achieved by the pursuer issuing an evader location query shortly before reaching the critical point. However, in the presence of message losses, the pursuer needs to issue multiple queries within a sampling period and adjust the frequency of its queries according to the state of the game. As shown in the previous section, it suffices that the required sampling period decreases with decreasing distance between the pursuer and evader. We note that to minimize the frequency of the queries, it suffices that the network

communication protocol scale to provide higher reliability as the distance between the pursuer and evader decreases.

Theorem 9 *Let the relation between message loss probability and the distance between the pursuer and the evader be given by the function $p_M(d_{pe})$. For any initial state x, the sampling period condition for Nash Equilibrium given in Equation 8 will be satisfied with probability greater than $1-\epsilon$ if*

$$f_q(d_{pe}) > \frac{\log(\epsilon)v_p}{\log(p_M(d_{pe}))d_{pe}}$$

where $f_q(d_{pe})$ is the frequency of the evader location queries when its distance from the pursuer is d_{pe}.

Proof: Consider a global state update that occurs at state x. The pursuer can issue up to $f_q T_{samp}$ queries before it traverses the previous evader location. The number of queries has to be chosen such that the probability of getting at least one successful update at that period is greater than $1-\epsilon$:

$$1-(p_M(d_{pe}))^{f_q T_{samp}} \geq 1-\epsilon \Rightarrow f_q(d_{pe}) \geq \frac{\log(\epsilon)}{\log(p_M(d_{pe}))T_{samp}} > \frac{\log(\epsilon)v_p}{\log(p_M(d_{pe}))d_{pe}}$$

■

Effect of Packet Delay

The evader location information needs to be routed from the local fusion center to the pursuer through wireless multiple hop links. The multiple hop communication imposes non-negligible delays on the evader state information. We assume the network is time synchronized and the packets are time-stamped at the source so that the pursuer will be able to calculate the delay of the packets it received. To derive a robust pursuit strategy we design for the worst possible evader motion, by assuming

the evader will have perfect information about the pursuer location. Therefore at time interval t the evader has access to state information $[x_p(t), y_p(t), x_e(t), y_e(t)]$ and the pursuer has access to state information $[x_p(t), y_p(t), x_e(t-\Delta t), y_e(t-\Delta t)]$. Then consider the following strategies:

Evader Strategy $\tilde{u}_e$: The evader uses the current location information for the pursuer to calculate the optimal direction as given in Theorem 7.

Pursuer Strategy $\tilde{u}_p$: The pursuer estimates the worst case location $(\hat{x}_e(t), \hat{y}_e(t))$ of the evader by considering all the points that the evader can reach at Δt and choosing the one that yields the lowest game value $V(\hat{x}_p(t), \hat{y}_p(t), x_e(t), y_e(t))$.

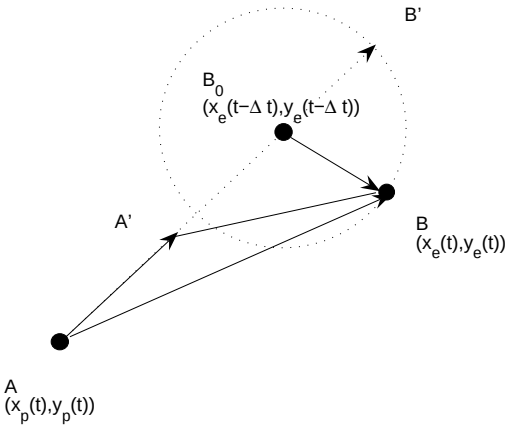

Figure 3.4: Effect of packet delay for target capture game

Theorem 10 *The strategies $\tilde{u}_p$ and $\tilde{u}_e$ are a Nash equilibrium of the pursuer-evader game with packet delays if the delay at each point is bounded by:*

$$\Delta t < \frac{d_{pe}(t - \Delta t)}{v_p}$$

where $d_{pe}(t - \Delta t)$ is the pursuer-evader distance at the time of packet transmission.

Proof: The pursuer moves to location A' at time t. At time $t - \Delta t$, the evader can move to anywhere on the circle (see Figure 3.4). To maximize its payoff, it must choose a location that maximizes $\overline{A'B}$. By the Triangle Inequality, B' is that location:

$$\overline{A'B_0} + \overline{B_0B} \geq \overline{A'B}$$

because $\overline{B_0B'} = \overline{B_0B}$,

$$\Rightarrow \overline{A'B'} = \overline{A'B_0} + \overline{B_0B'} \geq \overline{A'B} \Rightarrow \overline{A'B'} \geq \overline{A'B}$$

This will hold as long as

$$\Delta t * v_p = \overline{AA'} < d_{pe}(t - \Delta t) \Rightarrow \Delta t < \frac{d_{pe}(t - \Delta t)}{v_p}.$$

■

3.3 Pursuit-Evasion Game for asset protection

In this section, we continue our analysis of optimal pursuit control strategies in the presence of network effects for a more complex game – "Asset Protection" game.

3.3.1 Problem definition

As in the target capture case, we first consider a game between two players: a single pursuer and a single evader. The game state is given by the two dimensional

coordinates of the pursuer and evader $x = \{x_p, y_p, x_e, y_e\}$. Each player travels at constant speed v_p and v_e, and controls the direction of its motion, denoted by θ_p and θ_e.

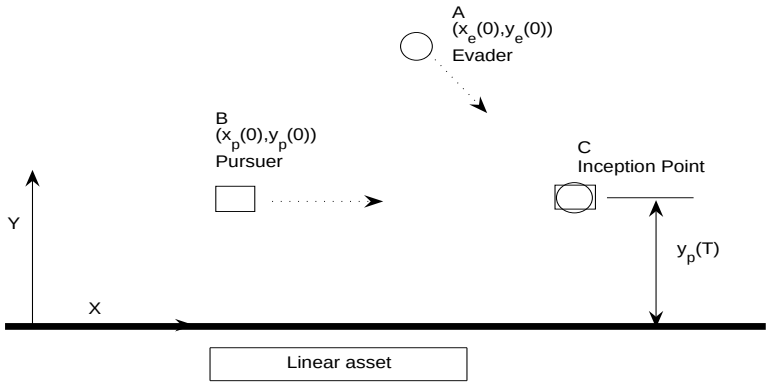

Figure 3.5: The pursuer and evader game

The linear asset is assumed to be infinitely long. With this assumption, the state space can be reduced to three dimensions by defining relative coordinates, $x_r = x_e - x_p$ and $y_r = y_e - y_p$. The state vector x evolves according to:

$$\dot{x} = \frac{\partial}{\partial t}\begin{bmatrix} x_r \\ y_r \\ y_p \end{bmatrix} = f(x, \theta_p, \theta_e) = \begin{bmatrix} v_e\cos(\theta_e) - v_p\cos(\theta_p) \\ v_e\sin(\theta_e) - v_p\sin(\theta_p) \\ v_p\sin(\theta_p) \end{bmatrix}$$

A catch is said to happen when $x_r^2 + y_r^2 < r^2$, where r is the catch radius. In the following, we consider the limiting case of $r \to 0$. The effect of finite catch radius is discussed in Section 3.6. Starting from the initial condition x_0, if the control strategies $u_p(x), u_e(x)$ satisfy the catch condition at time T then the payoff is given by $\mathcal{J}(u_p, u_e, x_0) = y_p(T)$. $y_p(T)$ is distance between the evader and asset at time T.

The game is zero-sum, so the pursuer's goal is to maximize $\mathcal{J}$ whereas the evader's goal is to minimize $\mathcal{J}$. Min-max optimal feedback strategies $u_p^*(x), u_e^*(x)$ are defined by the saddle condition:

$$\mathcal{J}_{u_p}(u_p, u_e^*, x_0) \leq \mathcal{J}(u_p^*, u_e^*, x_0) \leq \mathcal{J}_{u_e}(u_p^*, u_e, x_0) \tag{3.4}$$

We also note that the min-max optimal strategy pair $u_p^*(x), u_e^*(x)$ is also the Nash equilibrium [38] for this zero-sum game, where none of the players have an incentive to change its strategy unilaterally given the rival is maintaining its strategy choice.

For each initial condition x_0 the value of the game is defined as $V(x_0) = \mathcal{J}(u_p^*, u_e^*, x_0)$. The value function is uniquely defined irrespective of the number of min-max strategy pairs that satisfy the saddle point property in 3.4.

3.3.2 Optimal pursuit under perfect information

The value function and the associated optimal strategies for the game defined in Section 3.3.1 can also be derived using the Isaac conditions. Here we chose to present geometric solutions to provide intuition for the pursuit-evasion game under network effects.

Theorem 11 *If the ratio of the pursuer speed v_p to the evader speed v_e, α, is larger than 1, then the min-max optimal strategy for the evader and pursuers is given by:*

$$\theta_e(x) = \tan^{-1}(\tan\gamma + \alpha\sqrt{1 + (\tan\gamma)^2}) \tag{3.5}$$

$$\theta_p(x) = \tan^{-1}(\tan\gamma + \frac{\sqrt{1 + (\tan\gamma)^2}}{\alpha}) \tag{3.6}$$

where $\gamma = \tan^{-1}(\frac{y_r}{x_r})$ *and* $V(x) = y_p + \frac{\alpha^2 y_r + \alpha\sqrt{y_r^2 + x_r^2}}{\alpha^2 - 1}$.

Proof:

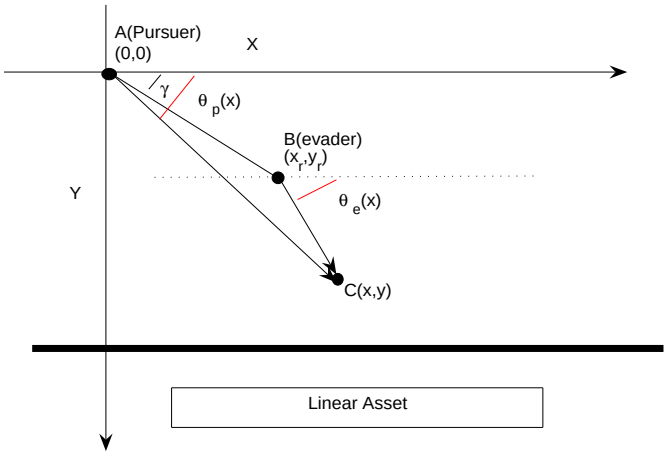

Figure 3.6: Linear asset protection with evader in between pursuer and asset

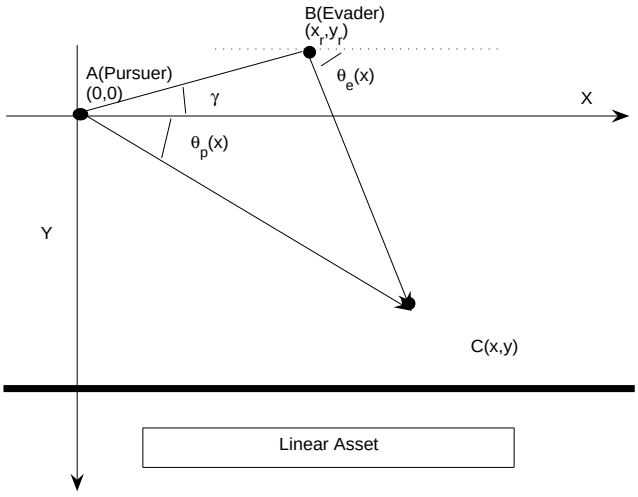

Figure 3.7: Linear asset protection with pursuer in between evader and asset

The min-max optimal strategies for the pursuer and evader is to directly to the boundary point of the circle that is closest to the target. In the following we characterize this critical boundary point:

We use a coordinate system to simplify the proof (cf. Figure 3.6 and Figure 3.7). Without loss of generality, we assume the pursuer location is $(0, 0)$; the evader location is (x_r, y_r). Let the location $C(x, y)$ be the location where the evader is caught by the pursuer. Because the pursuer speed v_p is α times of the evader speed v_e, then $\overline{AC} = \alpha\overline{BC}$ (Note: this straight line movement can be proved to be optimal). In coordinate form, we can rewrite this equation as:

$$\frac{\sqrt{x^2 + y^2}}{\sqrt{(x - x_r)^2 + (y - y_r)^2}} = \alpha$$
$$\Rightarrow x^2 - \alpha^2(x - x_r)^2 = \alpha^2(y - y_r)^2 - y^2$$

Maximizing y by differentiating the right side by x, we get:

$$\frac{d(x^2 - \alpha^2(x - x_r)^2)}{dx} = 0 \Rightarrow x = \frac{x_r\alpha^2}{\alpha^2 - 1}$$

Putting this equation into the previous equation, we get

$$(\alpha^2 - 1)y^2 - 2\alpha^2 y y_r + y_r^2\alpha^2 = \frac{x_r^2\alpha^2}{\alpha^2 - 1}$$
$$\Rightarrow y = \frac{\alpha^2 y_r + \alpha\sqrt{y_r^2 + x_r^2}}{\alpha^2 - 1}$$

Therefore,

$$\theta_e(x) = \tan^{-1}\frac{y - y_r}{x - x_r} = \tan^{-1}(\tan\gamma + \alpha\sqrt{1 + (\tan\gamma)^2})$$
$$\theta_p(x) = \tan^{-1}\frac{y}{x} = \tan^{-1}(\tan\gamma + \frac{\sqrt{1 + (\tan\gamma)^2}}{\alpha})$$
$$V(x) = y_p + \frac{\alpha^2 y_r + \alpha\sqrt{y_r^2 + x_r^2}}{\alpha^2 - 1}$$

■

At each time instant t, the pursuer will calculate the best location $(x', y'))$ that the evader can reach:

$$x' = \frac{x_r \alpha^2}{\alpha^2 - 1} + x_p, \qquad y' = \frac{\alpha^2 y_r + \alpha\sqrt{y_r^2 + x_r^2}}{\alpha^2 - 1} + y_p \tag{3.7}$$

then it will move toward that location.

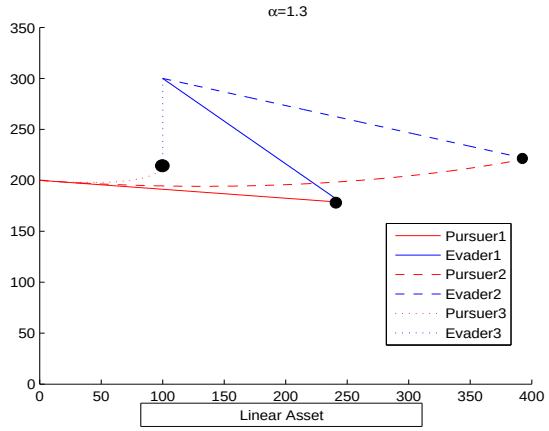

Figure 3.8: The Pursuer-Evader trajectory under perfect information

We illustrate the performance of the optimal strategy using a simulation. Results are given in Figure 3.8. The solid lines show the pursuer-evader trajectories when both employ min-max optimal strategies. The dashed lines show the case when evader uses non-optimal straight line strategies. We observe that min-max optimal pursuit strategy catches non-optimal evaders at a larger distance to the target.

3.3.3 Optimal pursuit under communication constraints

Sampling rate requirements of the optimal pursuit strategy

In this section, we derive the sampling rate requirements of the optimal strategy and show that it is inversely proportional to the relative distance between the pursuer

and evader. Again, we use the min-max solution concept to formulate a robust pursuit strategy that will perform satisfactorily irrespective of evader motion. To design for worst possible case of evader motion, we assume the pursuer has perfect information about the location of the evader and the sampling period. The sampling period is then chosen such that the evader does not benefit from switching from the optimal direction given in Theorem. 11, although the evader's deviation will be detected by the pursuer after the sampling period interval.

Theorem 12 *The evader does not deviate from its min-max equilibrium strategy if and only if the distance moved by the pursuer before getting the next sample of state information satisfies:*

$$v_p T_{sample} < \frac{\sqrt{\alpha^2(x_r)^2 + (\alpha(y_r) + \sqrt{(x_r)^2 + (y_r)^2})^2}}{\alpha} \tag{3.8}$$

Equivalently, the pursuer can move up to $\frac{(\alpha^2-1)}{\alpha^2}$ *of the total distance to the predicted evader location before sampling the global state without loss of optimality.*

Proof: Assume the pursuer moves first. It will move $\alpha * ds$ toward to the predicted optimal location (x, y), where ds is the maximum distance the evader can move during that time interval. Without loss of generality, we assume the initial location of pursuer is $(x_p, y_p) = (0, 0)$. So the next location based on the pursuer strategy is (x'_p, y'_p), which is decided by equation:

$$x'_p = \frac{\alpha^2 x_e ds}{\sqrt{\alpha^2 x_e^2 + (\alpha y_e + \sqrt{x_e^2 + y_e^2})^2}}, \qquad y'_p = \frac{(\alpha y_e + \sqrt{x_e^2 + y_e^2})\alpha ds}{\sqrt{\alpha^2 x_e^2 + (\alpha y_e + \sqrt{x_e^2 + y_e^2})^2}}$$

The evader can move to any location in the circle which is centered at (x_e, y_e) and has the radius ds. So, the next move for evader must satisfy:

$$(x'_e - x_e)^2 + (y'_e - y_e)^2 < ds^2$$

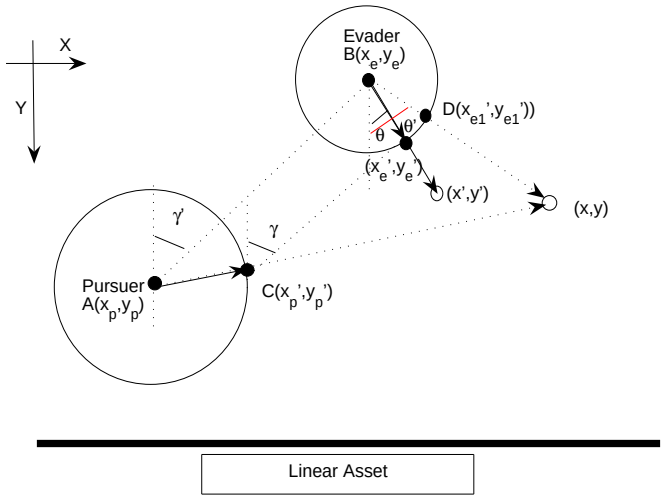

Figure 3.9: The sampling rate for tracking

and the next optimal location based on location (x'_e, y'_e) and (x'_p, y'_p) is:

$$y' = y'_p + \frac{\alpha^2(y'_e - y'_p) + \alpha\sqrt{(x'_e - x'_p)^2 + (y'_e - y'_p)^2}}{\alpha^2 - 1}$$

We want to find the maximum y' by changing (x'_e, y'_e). The constraints can be reformulated as: $x'_e = x_e + r\sin\theta, y'_e = y_e + r\cos\theta$. Let $F_x = x_e - x'_p$,$F_y = y_e - y'_p$, then we can get:

$$y' = y'_p + \frac{\alpha^2(F_y + r\cos\theta) + \alpha\sqrt{(F_x + r\sin\theta)^2 + (F_y + r\cos\theta)^2}}{\alpha^2 - 1}$$

To maximize y', the partial derivative with respect to θ is:

$$\frac{\partial y'}{\partial \theta} = \frac{-\alpha^2 r\sin\theta + \alpha\frac{(rF_x\cos\theta - rF_y\sin\theta)}{\sqrt{(F_x + r\sin\theta)^2 + (F_y + r\cos\theta)^2}}}{\alpha^2 - 1} = 0$$

To solve this equation, we let: $F'_x = F_x + r\sin\theta$, $F'_y = F_y + r\cos\theta$. Then the equation can be written as:

$$\alpha \sin\theta = \frac{(F'_x\cos\theta - F'_y\sin\theta)}{\sqrt{(F'_x)^2 + (F'_y)^2}}$$

The value of θ can be solved as:

$$\tan\theta = \frac{1}{\tan\gamma + \alpha\sqrt{1+(\tan\gamma)^2}}$$

where

$$\tan\gamma = \frac{F_y + r\cos\theta}{F_x + r\sin\theta} = \frac{y_e - y'_p + r\cos\theta}{x_e - x'_p + r\sin\theta} = \frac{y'_e - y'_p}{x'_e - x'_p}$$

Here, we claim the solution of the equation is $\theta = \theta'$. One important observation is that if $\tan\gamma = \tan\gamma'$ then $\tan\theta = \tan\theta'$. The other important observation is that when evader moves to (x'_{e_1}, y'_{e_1}) with distance r and pursuer moves to (x'_p, y'_p) with distance αr, the following equation holds:

$$\tan\gamma = \tan\gamma'$$

since line AB is parallel to line CD. To maximize y', the value of r should be $r = Max(r) = ds$ since the partial derivative of y' with respect to r is nonnegative when $\theta \in [0, \pi/2]$.

To satisfy the condition of $\theta \in [0, \pi/2]$, we must guarantee:

$$x'_p \le x_e \quad \text{when} \quad 0 = x_p \le x_e$$

$$x'_p \ge x_e \quad \text{when} \quad 0 = x_p \ge x_e$$

Then we can get:

$$\alpha|ds| < \frac{\sqrt{\alpha^2 x_r^2 + (\alpha y_r + \sqrt{x_r^2 + y_r^2})^2}}{\alpha}$$

which is $\frac{(\alpha^2-1)}{\alpha^2}$ of total distance of current pursuer location to the predicted optimal location (this distance is defined as d_{pu}).

■

We extend the previous result to derive the following scaling property of the sampling period T_{sample} with respect to the distance d_{pe} between the pursuer and evader:

Theorem 13 *Optimal pursuit-evasion strategies of the perfect information game also yield Nash equilibrium of the game with discrete time updates if:*

$$T_{samp}(d_{pe}) \leq \frac{\alpha - 1}{\alpha v_p} d_{pe}$$

In other words, the sampling period should decrease proportionally with decreasing distance between evader and pursuer to guarantee that the evader does not have an incentive to deviate from its strategy to move directly to the predicted intercept point.

Proof: If we define u to be the location of the predicted intercept point then we have:

$$\begin{aligned}
\frac{(\alpha^2-1)}{\alpha^2} d_{pu} &= \frac{\sqrt{\alpha^2 x_r^2 + (\alpha y_r + \sqrt{x_r^2+y_r^2})^2}}{\alpha} = \frac{\sqrt{(\alpha^2+1)d_{pe}^2 + 2\alpha y_r d_{pe}}}{\alpha} \\
&\in [\frac{\sqrt{(\alpha^2+1)d_{pe}^2 - 2\alpha d_{pe}^2}}{\alpha}, \frac{\sqrt{(\alpha^2+1)d_{pe}^2 - 2\alpha d_{pe}^2}}{\alpha}] \\
&\Rightarrow \frac{(\alpha^2-1)}{\alpha^2} d_{pu} \in [\frac{\alpha-1}{\alpha} d_{pe}, \frac{\alpha+1}{\alpha} d_{pe}]
\end{aligned}$$

Then we have

$$v_p T_{samp} \leq \frac{\alpha-1}{\alpha} d_{pe} \Rightarrow T_{samp} \leq \frac{\alpha-1}{\alpha v_p} d_{pe}$$

■

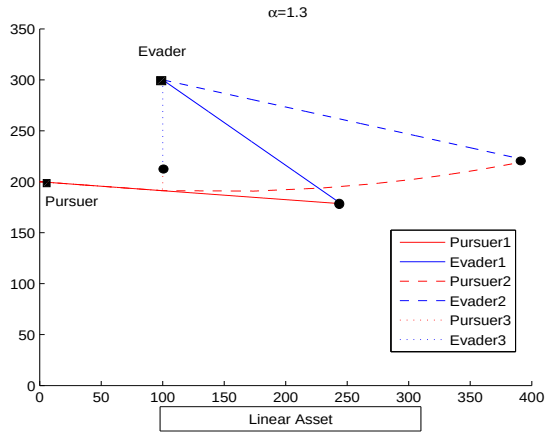

Figure 3.10: The Pursuer-Evader trajectory when using the T_{samp} update

We illustrate the performance of the reduced sample rate strategy using a simulation. The results are given in Figure 3.10. The solid lines show the pursuer-evader trajectories when both employ min-max optimal strategies, which is identical to the continuous update case. The dashed lines show the case when evader uses non-optimal straight line strategies. We observe that reduced sample rate pursuit strategy differs from its continuous information behavior for these cases but still catches these non-optimal evaders at a larger distance to the target.

Effect of message losses

The previous sampling rate analysis shows that the information must be updated before the pursuer reaches a critical point on the path to the predicted location defined in Theorem 12. In the presence of message losses, the pursuer needs to issue multiple queries within one sampling period and adjust the frequency of its queries according to the game state. To minimize the frequency of the queries, the network communication

protocol should scale to provide higher reliability as the distance between the pursuer and evader decreases.

Theorem 14 *Let the relation between message loss probability and the distance between the pursuer and the evader be given by the function $p_M(d_{pe})$. For any initial state x, the sampling period condition for Nash Equilibrium given in Equation 12 will be satisfied with probability greater than $1-\epsilon$ if*

$$f_q(d_{pe}) \geq \frac{\log(\epsilon)\alpha v_p}{\log(p_M(d_{pe}))(\alpha-1)d_{pe}}$$

where $f_q(d_{pe})$ is the frequency of the evader location queries when its distance from the pursuer is d_{pe}.

Proof: Consider a global state update that occurs at state x. The pursuer can issue up to $f_q T_{samp}$ queries before it traverses the critical distance $\frac{(\alpha^2-1)}{\alpha^2}d_{pu}$. The number of queries has to be chosen such that the probability of getting at least one successful update at that period is greater than $1-\epsilon$:

$$1-(p_M(d_{pe}))^{f_q T_{samp}} \geq 1-\epsilon \Rightarrow f_q(d_{pe}) \geq \frac{\log(\epsilon)}{\log(p_M(d_{pe}))T_{samp}} \geq \frac{\log(\epsilon)\alpha v_p}{\log(p_M(d_{pe}))(\alpha-1)d_{pe}}$$

■

Effect of Packet Delay

Similar to the target capture case, to derive a robust pursuit strategy we design for the worst possible evader motion, by assuming the evader will have perfect information about the pursuer location. Therefore at time increment t, evader has access to state information $[x_p(t), y_p(t), x_e(t), y_e(t)]$ and the pursuer has access to state information $[x_p(t), y_p(t), x_e(t-\Delta t), y_e(t-\Delta t)]$. Then consider the following strategies:

Evader Strategy $\tilde{u}_e$: The evader uses the current location information for the pursuer to calculate the optimal direction as given in Theorem 11.

Pursuer Strategy $\tilde{u}_p$: The pursuer estimates the worst case location $(\hat{x}_e(t), \hat{y}_e(t))$ of the evader by considering all the points that the evader can reach at Δt and choosing the one that yields the lowest game value $V(\hat{x}_p(t), \hat{y}_p(t), x_e(t), y_e(t))$

Theorem 15 *The strategies $\tilde{u}_p$ and $\tilde{u}_e$ are a Nash equilibrium of the pursuer-evader game with packet delays if the delay at each point is bounded by:*

$$\Delta t < \frac{\alpha - 1}{\alpha v_p} d_{pe}(t - \Delta t)$$

where $d_{pe}(t - \Delta t)$ is the pursuer-evader distance at the time of packet transmission.

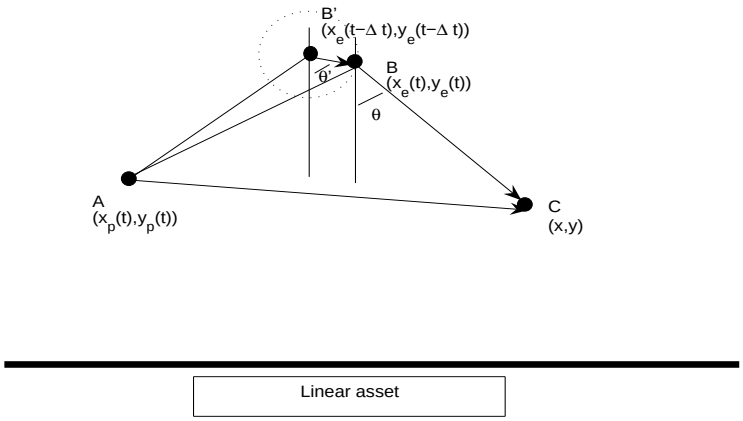

Figure 3.11: Effect of packet delay

Proof: Firstly, from the view point of pursuer, at time $t - \Delta t$, the evader can move to anywhere on the circle around B'. Without loss of generality, we assume the initial location of pursuer is $(x_p, y_p) = (0, 0)$. If the evader chooses the location B, the MaxMin y coordinate at time t is:

$$y = \frac{\alpha^2(y'_e + r\cos\theta') + \alpha\sqrt{(x'_e + r\sin\theta')^2 + (y'_e + r\cos\theta')^2}}{\alpha^2 - 1}$$

To maximize y, the partial derivative with respect to θ' is:

$$\frac{\partial y}{\partial \theta'} = \frac{-\alpha^2 r \sin\theta' + \alpha \frac{(r x'_e \cos\theta' - r y'_e \sin\theta')}{\sqrt{(x'_e + r\sin\theta')^2 + (y'_e + r\cos\theta')^2}}}{\alpha^2 - 1} = 0$$

It can be simplified as:

$$\alpha \sin\theta' = \frac{(x'_e \cos\theta' - y'_e \sin\theta')}{\sqrt{(x'_e + r\sin\theta')^2 + (y'_e + r\cos\theta')^2}}$$

Let: $x_e = x'_e + r\sin\theta'$, $y_e = y'_e + r\cos\theta'$. Then the equation can be written as:

$$\alpha \sin\theta' = \frac{(x_e \cos\theta' - y_e \sin\theta')}{\sqrt{(x_e)^2 + (y_e)^2}}$$

The value of θ' can be solved as:

$$\tan\theta' = \frac{1}{\tan\gamma + \alpha\sqrt{1 + (\tan\gamma)^2}}, \qquad \tan\gamma = \frac{y_e}{x_e} = \frac{y'_e + r\cos\theta'}{x'_e + r\sin\theta'}$$

Secondly, from the view point of evader, as shown in Theorem.11, the optimal value of θ based on state information $[x_p(t), y_p(t), x_e(t), y_e(t)]$ can be solved as:

$$\tan\theta = \frac{1}{\tan\gamma + \alpha\sqrt{1 + (\tan\gamma)^2}}$$

So, we have $\theta = \theta'$. Both pursuer and evader derive the same equilibrium C, so by the strategy of evader, we only need current location information to calculate equilibrium C. In fact, $B'BC$ should be a line.

Next, we need show the uniqueness of the equilibrium when both move to new location: In Figure 3.12, when evader moves from B_1 to B_2 with distance ds, the pursuer moves from A_1 to A_2 with $ds * \alpha$. We have:

$$\left.\begin{array}{l} |A_1C| = |B_1C| * \alpha \\ |A_1A_2| = |B_1B_2| * \alpha \end{array}\right\} \Rightarrow A_1B_1 // A_2B_2$$

Therefore, at the new location A_2,B_2, the evader decides the same equilibrium C as in location A_1,B_1, so does the pursuer.

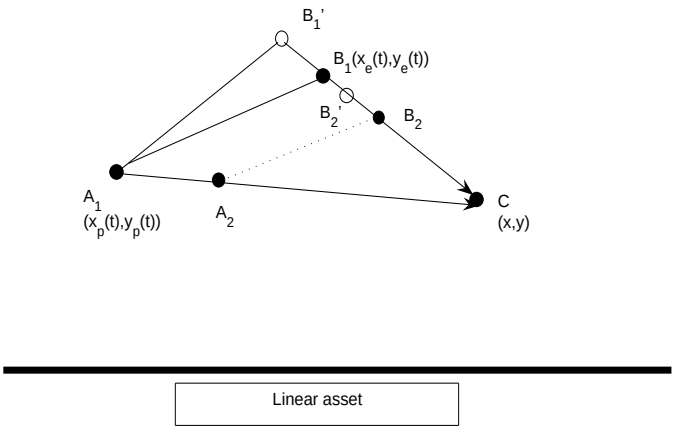

Figure 3.12: The uniqueness of equilibrium when packets are delayed

We observe that the predicted intercept point for the pursuer-evader game with packet delays at state $[x_p(t), y_p(t), x_e(t-\Delta t), y_e(t-\Delta t)]$ coincides with the predicted intercept location for the perfect information pursuit evader game at state $[x_p(t-\Delta t), y_p(t-\Delta t), x_e(t-\Delta t), y_e(t-\Delta t)]$. Therefore, we can use the results of Section 3.3.3 to bound the packet delay. Theorem 12 shows that if the packet is received before the pursuer travels distance of $\frac{\alpha-1}{\alpha} d_{pe}(t-\Delta t)$ the evader does not have an incentive to deviate from its equilibrium strategy. Therefore we should have:

$$v_p \Delta t < \frac{\alpha-1}{\alpha} d_{pe}(t-\Delta t) \Rightarrow \Delta t < \frac{\alpha-1}{\alpha v_p} d_{pe}(t-\Delta t)$$

■

3.4 Stabilization of the pursuer strategy

We have shown that Nash equilibrium can still hold despite communication constraints in both the target capture game and the asset protection game. In this

section, we discuss the stabilization properties of the pursuer strategy in the presence of state corruption as well as change in evader strategy.

3.4.1 State corruption

Wireless sensor nodes are deployed in harsh environments, not only is their communication unreliable, but the information about their state can also be corrupted. The optimal pursuer strategy is however based on the latest evader location information, and is thus independent of history information. Even if state information is corrupted, the pursuer should continue to query the latest evader location and move according to its optimal strategy. After it receives the correct evader location information, Nash equilibrium is reestablished.

3.4.2 Change in evader strategy

Every min-max equilibrium strategy enjoys the guarantee of a minimal payoff—regardless of what its opponent chooses to do. Our pursuer strategy thus has a sort of stabilization property, in the sense that irrespective of how the evader changes its strategy, the pursuer strategy is guaranteed to achieve a minimal payoff, i.e., catch distance. If the pursuer learns of a new strategy adopted by the evader and deviates from its own strategy to exploit the evader's strategy change, it opens itself up to the possibility that the evader reacts to the pursuer's change and the pursuer ends up with less payoff that it was guaranteed. In other words, following the min-max strategy is the evolutionary stable choice for the pursuer.

In addition, the intrinsic feature of Nash equilibrium is that if any player changes its strategy, it ends up with a worse payoff. Thus, in the target capture case, if evader

chooses a suboptimal strategy, it will be caught earlier than if it chooses the optimal strategy, as long as the pursuer maintains its strategy.

3.5 Experimental results

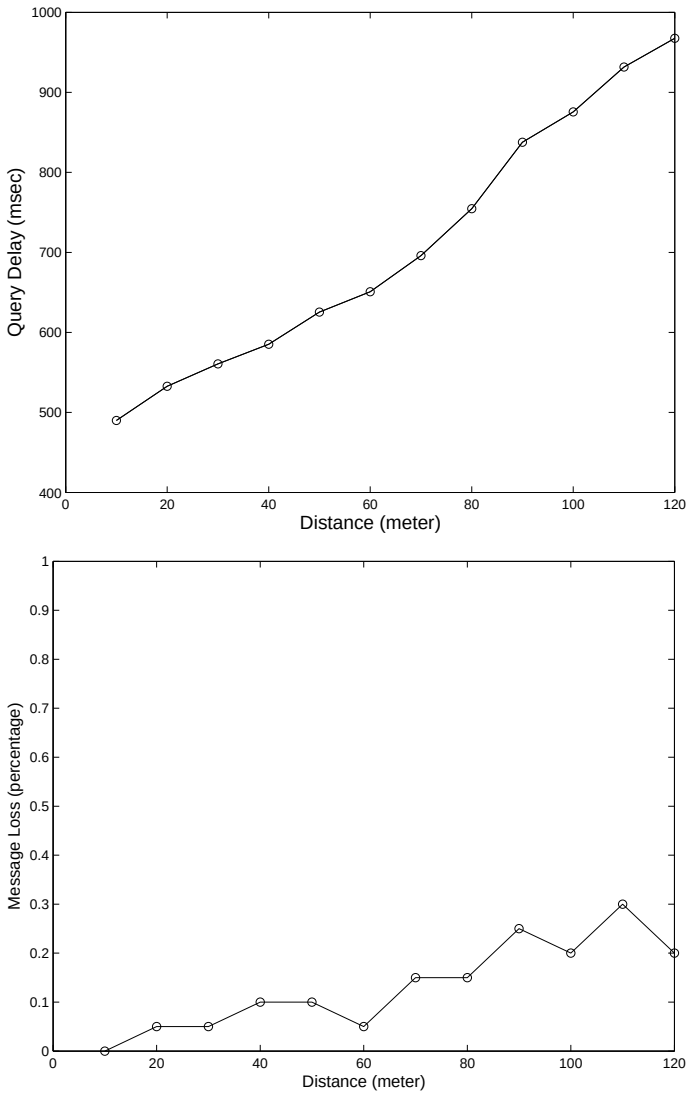

Figure 3.13: Experimental delay and message-loss rate using *Trail* networking service

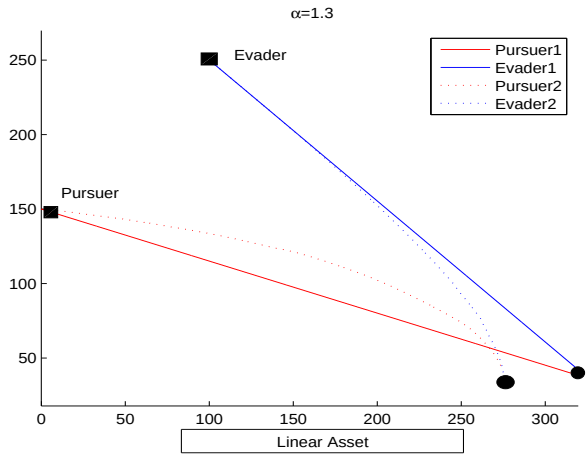

Figure 3.14: Pursuer-Evader trajectory in real experiment for asset protection

The results of Section 3.2 and Section 3.3 indicate the following requirements on the network protocol responsible for communicating evader track information to the pursuer agents: (i) Pursuer should determine the information refresh rate based on the requirements of the pursuit strategy, and (ii) Network delays should scale with the pursuer-evader distance. We have implemented a communication protocol called *Trail* that is compatible with these requirements. The overall system architecture for *Trail* is described in [31]. *Trail* offers the following pursuer controlled interface: *find evader* i, that returns the state of evader i to the pursuer agent issuing the query. The pursuer issuing the query itself could be mobile in which case the result is returned to the pursuer agent at its current location. In *Trail* object updates are local and it *Trail* provides a query time proportional to the distance from the object. Trail was implemented in a network of 105 XSM nodes in *Kansei* sensor network testbed at Ohio State University [3], where we used Garcia robots to serve as the mobile agents.

There are 2 objects in the system, one pursuer and one evader. The average find time and the variance of find times for an object at different distances, with 20 experiments at each distance, using *Trail* is shown in Figure 3.13. The object being found is mobile and the update messages due to this mobility can interfere with the find messages. When the reply to a find is not received before a threshold, it is considered to be lost. The fraction of lost messages with δ equal to 1.5 times the round trip network transmission time is also shown in Figure 3.13. These are used to build the loss and the reliability model for our pursuit-evasion game application.

We have used the experimental data to test the optimal pursuit strategy given in Section 3.3, where asset protection game is played. The results are given in Figure 3.14. There are two experiments. In both experiments the evader is assumed to know the current location of the pursuer and employ the optimal evading action. The solid lines are for the pursuit strategy that incorporates delays in to the pursuit strategy, the dashed lines are for the pursuit strategy that does not take delay into account and treats the location as if it is the current evader location. We observe that the delay tolerant algorithm can intercept even an evader that has information superiority at minimum possible distance, whereas an evader information superiority can achieve higher payoff facing an opponent which does not take delays into account.

3.6 Extensions

3.6.1 Non-zero catch radius

In practice, the catch condition should not be defined as $distance(P, E) = 0$, but as $distance(P, E) \leq r$ for some finite r. This can also relax the requirement to

increase sampling frequency near the catch. For this case, we give the following result for min-max strategies.

Target Capture Case

The non-zero catch radius only affects the optimal capture time, the optimal min-max strategy is still to go directly to previous evader location.

Asset Protection Case

In this case, the non-zero catch radius only affects the optimal intercept location $C(x, y)$. The optimal min-max strategy is still to go directly to $C(x, y)$, which can be calculated simply as: The point $C(x, y)$ (see Figure 3.15) can be calculated by

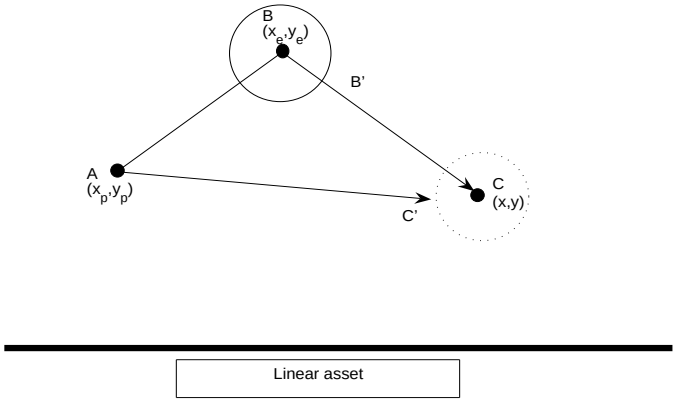

Figure 3.15: The effect of end game condition for asset protection

optimization:

$$\max_{\theta \in (0,2\pi)} y \qquad where \qquad |C'C| = r \text{ and } \frac{|AC'|}{|BC|} = \alpha$$

3.6.2 Multiple pursuer evader problems

Here, we consider n pursuer – m evader game with $n \geq m$, where each pursuer is restricted to catch only one evader. For instance, we can assume that the pursuer is immobilized at the time of a catch to detain the evader and more than one pursuer is not assigned to a given evader to reserve pursuer agents for future evader threats. The aim of the pursuer team is to minimize the catch time in the target capture case, or maximize a function of the distances to the asset in the asset protection case. $\mathcal{J}(u_p, u_e, x) = L(y_p^1(T_1), \ldots, y_p^n(T_n))$. The game is still zero-sum, so that the evader team tries to minimize(maximize) the same cost function. Common examples of cost functions are:

$$L(y_p^1(T_1), \ldots, y_p^n(T_n)) = \frac{1}{N} \sum_i y_p^i(T_i) \qquad or \qquad \min_i \{y_p^i(T_i)\}$$

We give the following result for this class of multiple pursuer-evader games. Let Σ be the set of all one-to-one assignment functions with the domain and range sets given as $\sigma : \{1, \ldots, m\} \to \{1, \ldots, n\}$. Then the value function $\mathcal{V}$ of the n pursuer – m evader game is given by :

$$\mathcal{V}(\{x_e^i\}_{i=1:m}, \{x_p^j\}_{j=1:n}) = \max_{\sigma \in \Sigma} L(V(x_e^1, x_p^{\sigma(1)}), \ldots, V(x_e^m, x_p^{\sigma(m)}))$$

In essence, the n pursuer – m evader game is reduced to first stage combinatorial optimization of the assignment problem followed by n two player pursuit games. We note that as long as both teams stick to min-max optimal strategies, no reassignment is required. In case the evaders deviate from their "assigned" pairs they will only achieve a lower score than their equilibrium strategy.

3.7 Conclusion

In this chapter, we studied differential games in networked environments with constrained communication resources leading to delays, losses and finite rates in information state updates. We focused on two typical differential games: pursuit-evasion game for target capture and an "asset protection" game, and formulated optimal strategies under communication constraints, established bounds on the information requirements of these strategies, and derived scaling laws for these bounds. In particular, we showed that the min-max optimal pursuer strategy of the full information game extends to networked games, and the stabilization properties of the pursuer strategy, provided that the sampling period and the delay in obtaining the evader state information updates scale linearly with the pursuer-evader distance.

We proposed a novel min-max equilibrium concept for networked differential games by introducing an omniscient opponent which can maximally exploit the delays and the intersample periods in the information state updates. This equilibrium concept is applicable to a much larger class of differential games than the two games considered in this paper. In future work we will focus on formulating generic information rate bounds for these set of games. Finally, in this paper, we assumed that the quality of the evader state information received by the pursuer is perfect. A promising direction for future research is to study the effect of the uncertainty in the evader location estimate on the optimal pursuer strategies.

CHAPTER 4

RECEIVER CENTRIC POWER MANAGEMENT PROTOCOL

In this chapter, we introduce a new design concept for power management - Receiver Centric. Energy efficiency is widely understood to be one of the dominant considerations for Wireless Sensor Networks. Based on historical data and technology trends, the receiver energy consumption will dominate all energy, to the point that for the majority of applications, power management research must focus on receiver efficiency.

By modeling several popular MAC layer protocols, we derive bounds on performance for receiver efficiency. In particular, we analyze four abstract models, Synchronous Blinking (e.g. T-MAC, S-MAC), Long Preamble (e.g. B-MAC), Structured Time-Spreading (also called Asynchronous Wake-Up), and Random Time Spreading. These results strongly suggest that scheduling the receiver so as to minimize (or eliminate) the potential for interference (or collisions) could be from 10 fold to 100 fold more efficient than current practice.

We provide two new receiver scheduling methods, *Staggered On* and *Pseudorandom Staggered On*, both of which are designed to exploit the untapped opportunity

for greater receiver efficiency. Compared with the centralized deterministic scheduling in Staggering On, the decentralized scheduling in Pseudorandom Staggered On achieves only slightly lower energy efficiency.

Finally, we identify the challenges of implementing self-stabilizing receiver centric protocols. Our solution will be presented in Chapter 5.

4.1 Introduction

Energy is a fundamental bottleneck of wireless sensor networks. It is widely understood in the literature that radio communication is the dominant power consumption in all the components [19].

4.1.1 Receiver centricity

The following table shows the power specifications for the historical sequence of radios used by the Berkeley motes [42].

Vendor	RFM	Chipcon	Chipcon
Part No.	TR1000	CC1000	CC2420
Rx power (mW)	11.4	28.8	59.1
Tx power (mW)	36	49.5	52.2

Table 4.1: Historical progression of radios used in Berkeley motes.

For comparison, the power specifications of CPUs is also listed in the following table [42]. Although the amount of data is small, it suggests three trends:

1. The communication power consumption is increasing.

Type	ATmega163	ATmega128	MSP430
Active (mW)	15	8	3
Sleep (mW)	0.045	0.075	0.015

Table 4.2: Historical progression of processors used in Berkeley motes.

2. The receiving power consumption is growing much faster than the transmitting.

3. The CPU active power decreases steadily with time.

Those trends are, in fact, real and fundamental. The modest but steady increase in transmitter power is largely caused by an increase in the data rates. The more significant growth in receiver power is due to growth in receiver complexity. We expect the first trend to be restrained by system energy. However, it seems that that second trend may accelerate over the next 5 to 10 years because of sophisticated despreading and Forward-Error Correction (FEC), which will dramatically increase the relative power required by the receiver. In the future the receiver power may be 1 to 2 orders of magnitude higher than the transmitter power because of the cost of receiver computations and dramatic improvements in other sources of efficiency (as one example, nRF24Z1 by Samsung has 50% more power consumption in Rx than in Tx).

The dominance of receiver power consumption requires receiver centric power management design. This is different from the sender based design that current MAC layer protocols have assumed. In the sender centric design, the sender wakes up all the potential receivers during the transmission even if the message is unicast. In contrast, receiver centricity means the sender must follow the wake-up schedule of

receiver. In this case, it is common that only one receiver will wake up to receive its message in a region at one time.

4.1.2 Almost Always Off communication

In typical sensor network applications such as environment monitoring, the systems are required to survive for several years. This means most of nodes must be almost always off (**AAO**) to conserve energy. This almost always off communication paradigm is opposite to legacy software paradigms that assume the receiver is always on.

For a MAC protocol in a low duty cycle sensor network, energy is wasted due to the following sources of overhead [34]:

- **Idle listening**: Since a node does not know when it will be the receiver of its neighbors, it must keep its radio in receiving mode all the time.
- **Overhearing**: Since the radio channel is a shared medium, a node may receive packets that are not destined to it.
- **Collisions**: If two nodes transmit at the same time, packets may be corrupted. Hence, the energy used during transmission and reception is wasted.
- **Protocol overhead**: MAC headers and control packets are used for signaling (ACK/RTS/CTS). This source of overhead can be significant since many applications only send a few kilobytes of data per day.

In AAO networks, idle listening and overhearing are two major source of power consumption. The protocol overhead should also be minimized because the application traffic is low. However, the low duty cycle tends to alleviate collisions.

4.1.3 Our contributions

- In this paper, we identify the fact of receiver dominance in energy consumption and the design paradigm of receiver centricity, which is in contrast to the current sender based MAC layer design. We believe this new paradigm will dominate energy sensitive designs.

- We define an energy efficiency metric, using which we analyze the power management schemes embedded in current MAC layer protocols. Bounds on the performance suggest that sender based scheduling suffers inherently from overhearing and idle listening. These results show the limits of the sender based scheduling.

- We provide two receiver based scheduling techniques. One is centralized deterministic scheduling, the other is decentralized pseudo-random scheduling. Surprisingly, the decentralized pseudo-random scheduling achieves only slightly lower energy efficiency compared with the global scheduling. Both of the receiver based scheduling techniques show orders of magnitude improvement over current transmitter based scheduling protocols.

- We also discuss the challenges to implement a stabilizing receiver centric protocols.

4.1.4 Related work

About 20 power aware MAC layer protocols have been proposed in recent years. The power management methods embedded in those protocols fall into four categories: synchronous blinking (S-MAC[52], T-MAC[49]), asynchronous wake-up [55], [47], long

preamble (usually called low power listening in the WSN literature)(B-MAC[41]), and on-demand wakeup based on a second channel [45], [56]. In the synchronous blinking case, all the nodes wake up at the same time periodically; in the asynchronous wake-up case, every node wakes up using a complex pattern designed to ensure that any two neighbor nodes can communicate irrespective of the time shift between the patterns; in the long preamble case, the transmitter uses a long enough preamble so that all nodes are guaranteed to wake-up before it transmits and to remain awake until the transmission completes; in the on-demand approach, a second channel is used to wake up the main radio.

Because current MAC layer protocols assume that the underlying communication between sender and receiver is local broadcast, the energy wasted on overhearing is substantial. All the neighbors around the sender must wake up to receive the packet which may be a unicast packet. In contrast, TDMA based approaches (SS-TDMA[32], L-MAC[23]) can avoid overhearing, but their idle-listening overhead is non-negligible, unless the TDMA duty cycle exactly matches the application's data rate. Essentially, these protocols focus on providing higher throughput by collision avoidance and transmission scheduling, energy efficiency is only a secondary consideration. A new energy efficient MAC layer protocol is therefore needed for AAO communications.

In [48], a receiver based collision avoidance protocol is introduced. But its primary goal is collision avoidance, not for energy efficiency. Therefore, the protocol provided is not energy efficient. In this paper, we argue that the energy efficient MAC protocol should be designed based on receiver centricity.

The rest of this chapter is organized as follows. In Section 4.2, we formally define the system model and an energy efficiency metric to evaluate the performance

of the protocols. In Section 4.3, by generalizing common MAC protocols into several abstract models, we compare their energy efficiency and provide theoretically performance bounds for each abstract model. In Section 4.4, we explain the challenges of designing a stabilizing receiver centirc MAC protocol.

4.2 Definition and system model

4.2.1 Definitions

We generalize the frame format common in several protocols, such as IEEE 802.15.4, B-MAC, S-MAC, and T-MAC, into a common logical structure. The schematic view of this abstraction is described in the next figure:

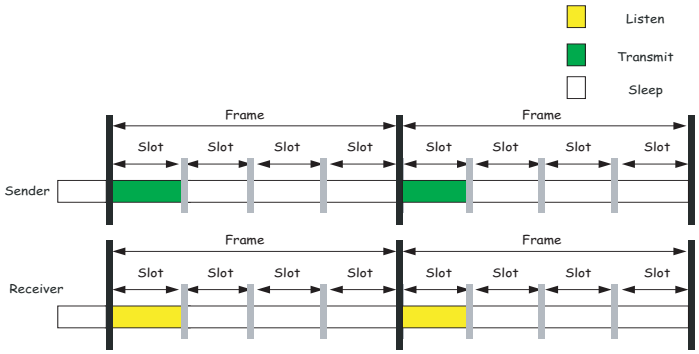

Figure 4.1: Structure of schedule

- Packet-length slot: A slot is a fixed time interval that is long enough to receive (send) a packet, and includes a "guard region" to allow for small scale time misalignments, which includes the radio startup time and packet delivery time. For CC2420 in Telos, the startup time is $0.57ms$ and packet delivery time is

about $1.5ms$ for a packet with 30 bytes payload. Different protocols may have different slot size. We use slot to discretized the continuous time.

- Preamble: A part of a slot that is used by the receiver to identify the start of a transmission. It appears before the header and is used internal to the radio for fine-scale time synchronization, carrier acquisition, etc. Usually, the preamble length is about 10% of the packet size. Most protocols can achieve better energy efficiency with shorter preambles. Used as the indication of sending request, preamble contains no information such as the destination or source address. On the receiver side, partial slot listening is used to detect the preamble. If there is no preamble detected, the receiver can sleep in the left slot time.

- Frame: A frame is the minimum interval over which a receiver is guaranteed to turn on at least once. Frame size is closely related to latency requirements.

We make the following assumptions:

- The cost of transmission and of reception are the same. This is true in the case of Chipcon CC2420. In fact, all the analysis we perform can be extended easily to other ratio models. We normalize the cost of sending in one slot to unity.

4.2.2 Problem statement

Traditional MAC layers are designed to achieve high throughput by collision avoidance. However, in the low duty cycle applications, the primary goal is to maximize goodput for a given energy budget, for which we propose be measured by **energy efficiency**, defined as:

$$E = \frac{\sum\sum M_i^j}{\sum\sum(S_i^j + R_i^j)} \tag{4.1}$$

where

$$S_i^j = \begin{cases} 1 & \text{When node } i \text{ transmits in slot } j \\ 0 & \text{When node } i \text{ sleeps in slot } j \end{cases}$$

$$M_i^j = \begin{cases} 2 & \text{Node } i \text{ succeeds in unicast at } j \\ 1 + N_r & \text{Node } i \text{ succeeds in broadcast at } j \\ 0 & \text{Otherwise} \end{cases}$$

$$R_i^j = \begin{cases} 1 & \text{When node } i \text{ listens in slot } j \\ 0 & \text{When node } i \text{ sleeps in slot } j \\ c_p & \text{Otherwise: partial slot listening} \end{cases}$$

(Note: N_r is the number of receivers in the broadcast).

The goal is to achieve maximum energy efficiency by scheduling transmission and reception.

Note:

- M_i^j is decided by the sender S_i^j, receiver R_i^j, and the possibility of collisions.
- If all the transmissions are well scheduled so that collisions are avoided, then for unicast communication:

$$\sum\sum M_i^j = 2\sum\sum S_i^j = 2\sum\sum R_i^j \Rightarrow E_{max} = 1$$

  Achieving such a schedule would require exact knowledge of the message generation pattern, which is almost never available.

4.2.3 Models

In our analysis, we consider the following models:

1. **Communication Model**: If a receiver receives more than two transmissions at the same time, none of them can succeed.
2. All communication is unicast.

3. **Traffic Model**: All the sensor nodes will send messages with the same probability p_t when they are active. If message is lost, it will be retransmitted with random delay.

4.2.4 Notations

Before analyzing the performance of different power management schemes, we define several variables:

- Let ϵ be the probability that on average a node needs to transmit in one slot. Typically, $\epsilon \in [10^{-6}, 1/500]$, and is determined only by the application and routing policy, not network reliability. It measures the message generating forwarding rate, without including the retransmissions.

- Let N_e be the average number of neighbors, this is determined by the communication range and the node density. Typically, $N_e \in [2, 6]$.

- Let η be the average number of nodes that would interfere with a particular transmission. Typically, $\eta \in [5, 50]$, because the interference range is significantly larger than the communication range.

- Let ψ be the overall duty cycle. Typically, $\psi \in [\frac{1}{32}, \frac{1}{256}]$. Mission lifetime dictates this. Here, the ψ is defined as number of active slots (sending and receiving) divided by total number of slots.

- Let ψ_r be the receiver duty cycle. ψ_r is defined as number of listening slots divided by total number of slots.

- Let T be the cycle time, or the duration of one frame. Typically, $T \in [0.1, 100]s$. The average single hop latency is half of this number.

- Let δ be the slot time, the time it takes to wakeup and power up the communications to send one packet. Typically, $\delta \in [5, 50]ms$.

- Let $c_p\delta$ the partial slot listening time. c_p is the percentage of the time that is spent on detecting channel activity.

Note: In a stable network where all communications are unicast, $2N_t\epsilon = \sum\sum M_i^j$ and $N_t\psi = \sum\sum(S_i^j + R_i^j)$, where N_t is total number of slots. Then the energy efficiency can be computed by:

$$E = \frac{\sum\sum M_i^j}{\sum\sum(S_i^j + R_i^j)} = \frac{2\epsilon}{\psi} \tag{4.2}$$

It measures the real goodput per energy consumption.

4.3 Energy efficiency analysis

In this section, we investigate the theoretical performance bounds of several abstract models that represent key features of widely used MAC protocols. The following assumptions are made in this section:

- The number of interfering nodes η is constant. In Section 4.3.7, we prove that our analysis is still valid in the case of varying η.

- To simplify our analysis, we do not consider CSMA effects in the analysis. We relax this assumption in Section 4.3.7.

- We assume a node will wake up for a full slot other than partial slot. In Section 4.3.7, we will analyze these protocols with partial slot listening enabled.

We begin with two lemmas.

Lemma 2 *Assume the probability of transmission for any node at slot t is p_t, then the conditional probability of collision p_c when a node wants to send at slot t is:*

$$p_c = 1 - (1 - p_t)^{\eta - 1} \tag{4.3}$$

Note: this equation is derived from the fact that for any receiver only one neighbor node can send out message. In addition, all the transmissions are independent. Clearly, the probability of collisions depends on the number of interfering nodes.

Lemma 3 *When one packet is sent, the expected number of transmissions is:*

$$E(Trans) = (1 - p_c)(1 + \sum_{k=1}^{\infty}(k+1) * p_c^k) = \frac{1}{1 - p_c} \tag{4.4}$$

where p_c is the probability of collision.

4.3.1 The Synchronous Blinking case

In this case, based on the global time, all the nodes wake up at the same time. During these short on-intervals any traditional protocol may be used. S-MAC and T-MAC belong to this category.

Theorem 16 *When $p_t^* = \frac{1}{\eta}$, the Synchronous Blinking Case attains its the maximal energy efficiency:*

$$E_{smax} = \max(\frac{2\epsilon}{\psi}) = \frac{2(1 - \frac{1}{\eta})^{\eta}}{\eta - 1} \approx \frac{2}{(\eta - 1)e} \tag{4.5}$$

Proof: Assuming the probability of transmission when a receiver is awake is p_t, then the percentage of time for transmission is defined as:

$$Tr = p_t * \psi \tag{4.6}$$

Tr can also be calculated by:

$$Tr = E(Trans) * \epsilon = \frac{\epsilon}{1 - p_c} = \frac{\epsilon}{1 - (1 - (1 - p_t)^{\eta - 1})} = \frac{\epsilon}{(1 - p_t)^{\eta - 1}} \tag{4.7}$$

By solving equation (4.6) and (4.7), we can get:

$$\frac{\epsilon}{\psi} = p_t(1 - p_t)^{\eta - 1}$$

By differentiating with respect to p_t, we get the maximal efficiency at $p_t^* = 1/\eta$:

$$E_{smax} = \max\left(\frac{2\epsilon}{\psi}\right) = \frac{2(1 - \frac{1}{\eta})^{\eta}}{\eta - 1} \approx \frac{2}{(\eta - 1)e}$$

Note: The approximation in the last step is the asymptote as $\eta \to \infty$, but it is already a fairly good approximation by the time $\eta = 5$.

■

Remark:

- If all the senders are well scheduled, they can send messages sequentially to avoid collisions. The energy efficiency under this assumption, E_{imax} is $2/\eta$. Thus,

  $$\frac{E_{smax}}{E_{imax}} = \frac{\frac{2(1 - \frac{1}{\eta})^{\eta}}{\eta - 1}}{\frac{2}{\eta}} = \frac{(1 - \frac{1}{\eta})^{\eta}\eta}{\eta - 1} \approx \frac{1}{e}$$

  Because of collisions, only $1/e$ of the messages are successfully transmitted.

- Since $\eta \gg e$, the maximal energy efficiency in this case is dictated by the number of interfering nodes.

4.3.2 The Long Preamble case

In this case, all the nodes wake up periodically. No time synchronization is required. If a node wants to send a message, it uses a long preamble. When a receiver

wakes up and if it detects an ongoing preamble, it stays awake for the message; otherwise it goes back to sleep. B-MAC [41] falls into this category. There are two cases as shown in Figure 4.2:

- In case one, a long preamble is used to wake up the receiver, all the nodes that hear the preamble will wake up. After the long preamble, the payload is transmitted.
- In case two, the same packet is sent repeatedly during the frame time and the receiver wakes up.

Our analysis focuses on case two since it is more power-efficient than the case one [41] and more practical for packet based radio such as CC2420.

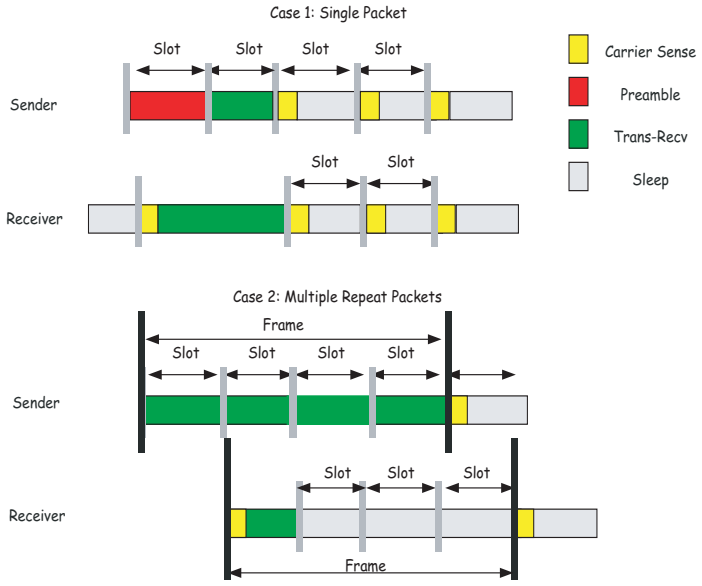

Figure 4.2: Structure of the Long Preamble case

Theorem 17 *When $p_t^* \approx \psi/2$, we get the highest energy efficiency in the Long Preamble case (case two):*

$$E_{lmax} = \max(\frac{2\epsilon}{\psi}) \approx \frac{\psi(1-\psi)^{\eta-1}}{2} \approx \frac{\psi}{2} \tag{4.8}$$

Proof: Assume the probability of transmission at every frame is p_t, so the probability that a receiver gets messages successfully can be calculated by:

$$p_s = p_t(1-2p_t)^{\eta-1}$$

Note: The factor of 2 is explained by the fact that the transmission in one slot can interfere with transmissions in two slots due to slot misalignment.

Let the receiver duty cycle be ψ_r during the long preamble transmission, the energy efficiency is:

$$\frac{\epsilon}{\psi} = \frac{p_s\psi_r}{p_t+\psi_r}$$
$$\psi = p_t + \psi_r$$

We can get:

$$\frac{\epsilon}{\psi} = (1-\frac{p_t}{\psi})p_t(1-2p_t)^{\eta-1} \tag{4.9}$$

Differentiating with p_t, we can get the maximum ϵ/ψ when

$$p_t^* = \frac{\psi}{\sqrt{\eta^2\psi^2+1-2\psi}+\eta\psi+1} \approx \frac{\psi}{2}$$

and $1 \gg \eta\psi$, we have the maximal efficiency:

$$E_{lmax} = \max(\frac{2\epsilon}{\psi}) \approx \frac{\psi(1-\psi)^{\eta-1}}{2} \approx \frac{\psi}{2}$$

■

Remark:

- To get the maximal energy efficiency, the receiver duty cycle must be approximately equal to the sender's duty cycle: $\psi_r = \psi - p_t^* \approx \frac{\psi}{2} \approx p_t^*$.

4.3.3 The Asynchronous Wake-up case

In this case, all the nodes wake up according to a schedule described in [55] and [47]. By using these schedules, it is possible to wake up in only k slots out of total k^2 slots and to guarantee that for any two nodes at least one slot exists during which both nodes are awake, no matter what shift exists between the two schedules. We regard these k^2 slots as one frame. In [47], a dynamic scheduler is developed to get different duty cycle by changing parameters. We define the frame length as n, i.e.:

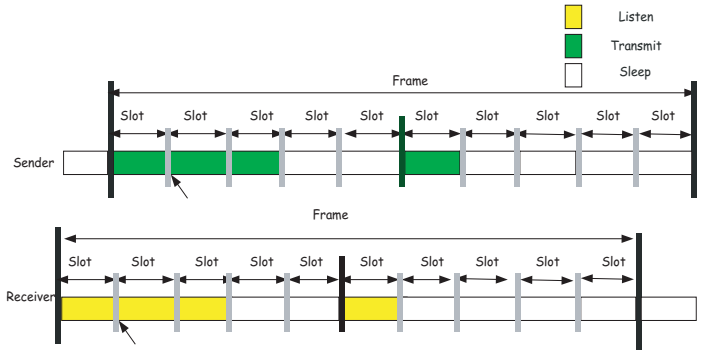

Figure 4.3: Structure view of the Asynchronous Wake-up case

$$n = \frac{T}{\delta} = k^2 \text{ (Note: } \psi \approx \frac{k}{k^2} = \frac{1}{k} = \frac{1}{\sqrt{n}} \text{)}$$

Theorem 18 *The maximal energy efficiency where ψ is small and $1 > 2\eta\psi$ is:*

$$E_{amax} = \max\left(\frac{2\epsilon}{\psi}\right) \approx \frac{2}{\sqrt{n}} = 2\psi \qquad (when \quad p_t^* = 1) \tag{4.10}$$

Proof: Assume the probability of transmission in one frame is p_t, then the conditional probability of collision given transmission is:

$$p_c = 1 - (1 - p_t * \frac{2}{\sqrt{n}})^{\eta-1}$$

Note: Similarly to the Long Preamble case, factor 2 is used to compensate for desynchronized slot. The percentage of transmission time Tr is

$$Tr = p_t * \psi \tag{4.11}$$

Tr can also be computed by equation:

$$Tr = E(Trans) * \epsilon * \sqrt{n} = \frac{\epsilon\sqrt{n}}{1 - p_c} = \frac{\epsilon\sqrt{n}}{1 - (1 - (1 - p_t * \frac{2}{\sqrt{n}})^{\eta-1})} = \frac{\epsilon\sqrt{n}}{(1 - p_t * \frac{2}{\sqrt{n}})^{\eta-1}} \tag{4.12}$$

by using similar steps as the Synchronous Blinking case, we can get:

$$\frac{\epsilon}{\psi} = \frac{1}{\sqrt{n}} p_t (1 - p_t * \frac{2}{\sqrt{n}})^{\eta-1} \tag{4.13}$$

By varying p_t, we can get the maximum energy efficiency. when $1 \leq 2\eta\psi$:

$$E_{amax} = \max(\frac{2\epsilon}{\psi}) = \frac{(1 - \frac{1}{\eta})^{\eta}}{\eta - 1} \approx \frac{1}{e(\eta - 1)} \tag{4.14}$$

$$p_t^* = \frac{\sqrt{n}}{2\eta} = \frac{1}{2\eta\psi} \tag{4.15}$$

when $1 > 2\eta\psi$, we have the maximal efficiency:

$$E_{amax} = \max(\frac{2\epsilon}{\psi}) = \frac{2}{\sqrt{n}}(1 - \frac{2}{\sqrt{n}})^{\eta-1} = 2\psi(1 - 2\psi)^{\eta-1} \tag{4.16}$$

$$p_t^* = 1 \tag{4.17}$$

In a low duty cycle sensor network, n is large enough, so the maximal efficiency is:

$$\max(\frac{2\epsilon}{\psi}) \approx \frac{2}{\sqrt{n}} = 2\psi$$

■

Remark:

- The energy efficiency of Asynchronous Wake-up method is proportional to total duty cycle, which is very low in a typical AAO network.
- Since no time synchronization is required, the method is robust to network uncertainty and mobility.

4.3.4 Random Time Spreading case

In this case, the wakeup schedule is totally random. Every time slot, the receiver will wake up with probability p_r. In addition, time synchronization is not required.

Theorem 19 *The maximal energy efficiency in low-duty-cycle random time spreading sensor network is:*

$$E_{rmax} = \max_{p_t \in [0,1]} (\frac{\epsilon}{\psi}) \approx \frac{2\psi}{\eta} \tag{4.18}$$

Proof: Assume the probability of sending a message in one time slot is p_t, then the probability of successfully receiving a message is:

$$p_{su} = N_e * \frac{p_t}{N_e} p_r (1-p_t)^{\eta-1} = p_t p_r (1-p_t)^{\eta-1}$$

$$\epsilon = p_{su}$$

$$\psi = p_r + p_t$$

The energy efficiency can be calculated by:

$$E_{rmax} = \max_{p_t \in [0,1]} (\frac{2\epsilon}{\psi}) = \max_{p_t \in [0,1]} (\frac{2p_t p_r (1-p_t)^{(\eta-1)}}{p_t + p_r}) \approx \frac{2\psi\eta}{(\eta+1)^2} (1 - \frac{\psi}{\eta+1})^{\eta-1} \approx \frac{2\psi}{\eta}$$

where

$$p_t^* = \frac{\sqrt{\eta^2 + 4\eta - 4} - \eta}{2(\eta-1)} p_r \approx \frac{p_r}{\eta}$$

■

Remark:

- This fully random wake-up case has the worst power efficiently because energy is wasted not only in time (duty cycle ψ), but also in space (η).
- Here, we ignore the effect of possibly unaligned slots. If we consider this effect, the energy efficiency is reduced by a factor of 2.

4.3.5 The Staggered On case

All the solutions we have described so far are sender based scheduling. They are intended as surrogates of the bulk of schemes in common use today. We provide one solution, which we call Staggered-On wake-up, in order to highlight the key difference between this case and the Synchronous Blinking case.

In this case, all the receivers are scheduled to wake up so that no receivers can interfere with each other; we call this **receiver collision** avoidance. Specifically, any transmitter that is within the communication range of one receiver is outside the interferences range of the other receiver as shown in Figure 4.4. The four circles in the figure mean the interference regions for four receivers. In this case, receivers that have overlapped interference region can not be active at the same time. Every node knows the wake-up schedule of their neighbor. If they want to send unicast message to a neighbor, they wait until the destination node is awake.

Theorem 20 *When $p^*_{tm} \approx 0.62/N_e$, we get the highest energy efficiency in the Staggered On case:*

$$E_{omax} = \max\left(\frac{2\epsilon}{\psi}\right) \approx 0.43 \tag{4.19}$$

where p^*_{tm} is the possibility of transmission when the neighboring receiver is on.

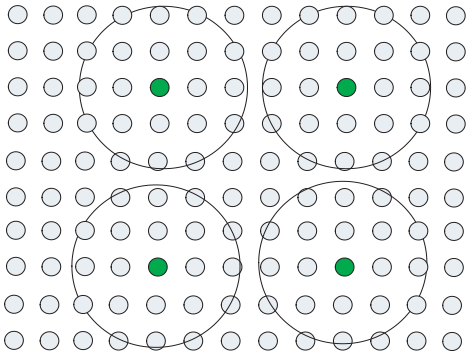

Figure 4.4: The spatial view of Staggered On case

Proof: Assume when any neighboring receiver is on, the probability of transmission is p_t, then the probability of transmitting to one particular receiver is $p_{tm} = p_t/N_e$, so the conditional probability of collision p_c can be calculated by:

$$p_c = 1 - (1 - p_{tm})^{N_e - 1}$$

Let the receiver duty cycle be ψ_r, then the sender duty cycle ψ_s is

$$\begin{aligned} \psi_s &= N_e * p_{tm} * \psi_r \\ \frac{\epsilon}{\psi_r} &= N_e p_{tm} (1 - p_{tm})^{N_e - 1} \\ \psi &= \psi_s + \psi_r = N_e * p_{tm} * \psi_r + \psi_r \end{aligned} \tag{4.20}$$

We have another equation for the sender duty cycle ψ_s:

$$\psi_s = E(Trans) * \epsilon = \frac{\epsilon}{1 - p_c} = \frac{\epsilon}{1 - (1 - (1 - p_{tm})^{N_e - 1})} = \frac{\epsilon}{(1 - p_{tm})^{N_e - 1}} \tag{4.21}$$

By solving equation (4.20) and (4.21), we can get:

$$\frac{\epsilon}{\psi} = \frac{N_e p_{tm} (1 - p_{tm})^{(N_e - 1)}}{N_e p_{tm} + 1} \tag{4.22}$$

By differentiating with respect to p_{tm}, we can get the maximal efficiency:

$$E_{omax} = \max(\frac{2\epsilon}{\psi}) \approx 0.43$$

when :

$$p^*_{tm} = \frac{\sqrt{5N_e^2 - 4N_e} - N_e}{2N_e(N_eN_e - 1)} \approx \frac{0.62}{N_e}$$

■

Remark: Because of sender collision, we can only successfully transmit 43% of the ideal capacity. However, compared with the Synchronous Blinking case, by scheduling the receiver, the energy efficiency increases by a degree of η, i.e. the number of interfering neighbors.

4.3.6 Pseudo-random Staggered On case

To overcome the difficulty of implementing and maintaining a global schedule in Staggered On case, we relax the constraints by letting every node wake up independently with probability ψ_r. There is no guarantee that **receiver collisions** are avoided. Every node has the wake-up schedule of their neighbors.

Theorem 21 *The maximal energy efficiency in the Pseudo-random Staggered On case is*

$$E_{domax} = \max(\frac{\epsilon}{\psi}) \approx E_{omax} * (1 - 0.62\psi_r)^{\eta - N_e} \approx 0.43(1 - 0.62\psi_r)^{\eta - N_e} \qquad (4.23)$$

Proof: The only difference between Pseudo-random Staggered On and centralized Staggered On is the expected number of interferers. Similar to the previous case,

$$\frac{\epsilon}{\psi_r} = N_e p_{tm}(1 - p_{tm} - N_e p_{tm}\psi_r/2)^{N_e - 1} * (1 - N_e * p_{tm}\psi_r)^{\eta - N_e}$$
$$\psi = \psi_s + \psi_r = N_e * p_{tm} * \psi_r + \psi_r$$

This equation is derived by the following observations:

- Assume a node named A wakes up as a receiver, since its neighbors know the schedule of A, they may act as senders. In addition, if some other neighboring nodes of A are receivers (wakeup), they may also transmit. Because of communication range, on average there is 50% chance that two neighbors of A are also neighboring. Therefore, the transmission probability is $p_{tm} + N_e * p_{tm}\psi_r/2$.

- For any other node that is not a neighbor, but in the interference range, the possibility of being active as a sender is $N_e * p_{tm}\psi_r$.

We have:

$$\frac{\epsilon}{\psi} = N_e p_{tm}(1 - p_{tm} - N_e p_{tm}\psi_r/2)^{N_e-1} * \frac{(1 - N_e * p_{tm}\psi_r)^{\eta - N_e}}{N_e p_{tm} + 1} \tag{4.24}$$

By differentiating with respect to p_{tm}, we get the maximal efficiency:

$$E_{domax} = \max(\frac{2\epsilon}{\psi}) \approx E_{omax} * (1 - N_e * p_{tm}\psi_r)^{\eta - N_e} \approx 0.43(1 - 0.62\psi_r)^{\eta - N_e} \tag{4.25}$$

when $p^*_{tm} \approx 0.62/N_e$

■

Remark: In this case, the energy efficiency decreases with the number of interferers. In addition, at very low duty cycle, the Pseudo-random scheduling can be as good as global Staggered On case.

4.3.7 Extensions to the analysis

Adaptation to interference range variation

The value of η is decided by interference range. In this section, we focus on its influence on the energy efficiency. We evaluate the influence of variation using two

standard distributions: uniform distribution and normal distribution to show that our analysis is still valid even under those variations. Here, we only show the Synchronous Blinking Case as an example.

a) Uniform distribution: Assume η is uniformly distributed in $[\eta_0 - \sigma, \eta_0 + \sigma]$. However, the wakeup schedule uses the average value η_0. Then the expected efficiency can be calculated by:

$$E(e_f) = \int_{\eta_0-\sigma}^{\eta_0+\sigma} \frac{1}{2\sigma} p(1-p)^{\eta-1} d\eta = \frac{p}{1-p} \frac{(1-p)^{\eta_0+\sigma} - (1-p)^{\eta_0-\sigma}}{2\sigma \log(1-p)}$$

Compared to the efficiency of the network with constant η_0,

$$E(e_{f_0}) = p(1-p)^{\eta_0-1}$$
$$\frac{E(e_f)}{E(e_{f_0})} = \frac{(1-p)^{\sigma} - (1-p)^{-\sigma}}{2\sigma \log(1-p)} \approx 1 \qquad (\text{ when } p = \frac{1}{\eta_0} \text{ is small })$$

b) Normal distribution: Assume η is normally distributed in (η_0, σ^2). However, the wakeup schedule uses the average value η_0. Then the expected efficiency can be calculated by:

$$E(e_f) = \int_{-\infty}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\eta-\eta_0)^2}{2\sigma^2}} p(1-p)^{\eta-1} d\eta = p(1-p)^{\eta_0-1} e^{\frac{\sigma^2 \log^2(1-p)}{2}}$$

Compared to the efficiency of the network with constant η_0,

$$E(e_{f_0}) = p(1-p)^{\eta_0-1}$$
$$\frac{E(e_f)}{E(e_{f_0})} = e^{\frac{\sigma^2 \log^2(1-p)}{2}} \approx 1 \qquad (\text{ when } p = \frac{1}{\eta_0} \text{ is small })$$

Carrier sensing and collision avoidance

To avoid collisions, carrier sensing can be applied in all the previous cases. However, this may increase the idle listening time. In addition, carrier sensing can not avoid the hidden terminal problem completely. So, the benefit for energy efficiency

by carrier sensing is limited. From previous analysis, we have seen the channel utilization can be up to 63% by simply letting every node access the channel randomly with probability $1/\eta$. In other words, the maximum improvement of energy efficiency by using carrier sensing is only 37%, which is less important than scheduling receivers properly.

Partial slot listening

Partial slot listening can reduce the idle listening because receiver can quickly go back to sleep if there is no traffic. The energy efficiency for the different cases can be computed by the following equations:

- Synchronous Blinking case:

$$E = \frac{2\epsilon}{\psi} = \frac{2p_t(1-p_t)^{\eta-1}}{c_p(1-p_t)^{\eta} + 1 - (1-p_t)^{\eta}} \tag{4.26}$$

- Long Preamble case:

$$E = \frac{2\epsilon}{\psi} = \frac{2p_t(1-p_t)^{\eta-1}(\psi - p_t)}{p_t + (\psi - p_t)((1-c_p)(1-p_t)^{\eta} + 1)} \tag{4.27}$$

- Asynchronous Wake-up case: no change since every node needs to wake up for a full slot to listen to any possible traffic.

- Random Time Spreading Case: no change since every node needs to wake up for a full slot to listen to any possible traffic.

- Staggered On case:

$$E = \frac{2N_e p_t(1-p_t)^{N_e-1}}{(1-c_p)N_e p_t + c_p + N_e p_t} \tag{4.28}$$

- Pseudo-random Staggered On case:

$$E_a = 2N_e p_t (1-p_t)^{N_e-1}(1-N_e p_t \psi_r)^{\eta-N_e}$$
$$E_b = (1-c_p)(1-(1-p_t)^{N_e}(1-N_e p_t \psi_r)^{\eta-N_e}) + c_p + N_e p_t$$
$$E = \frac{2\epsilon}{\psi} \approx \frac{E_a}{E_b} \tag{4.29}$$

Matching duty cycles

All of the efficiencies reported in this section have been computed for ideally chosen message rates. In each case, the derivation considers a range of communication load levels and selects the load level that maximizes the efficiency. Table 4.3 summarizes the relationship between message rate and duty cycle corresponding to optimal efficiency.

Case Name	Message Rate	Receiver duty cycle
Sync Blinking	$\frac{\psi}{\eta+1}$	$\frac{\psi\eta}{\eta+1}$
Long Preamble	$\psi^2/2$	$\psi/2$
ASync Wakeup	$\psi^2/2$	$\psi/2$
Fully random	$\frac{\psi}{\eta+1}$	$\frac{\psi\eta}{\eta+1}$
Staggered-On	$0.38\psi/N_e$	0.62ψ
Random-Staggered	$0.38\psi/N_e$	0.62ψ

Table 4.3: Receiver duty cycle and message generation rate at maximal energy efficiency.

Figure 4.5 shows how the efficiency varies with the ratio of message rate and receiver duty cycle. For some applications, the amount of traffic load might far exceed the bandwidth capacity of the network. In this case a policy of governing the message generation rate to match the optimal efficiency point of the network

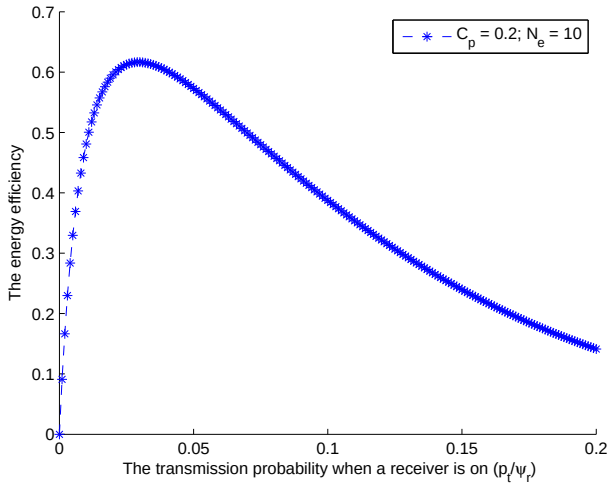

Figure 4.5: The loss of efficiency due to the mismatch between message generation rate and receiver duty cycle.

is required. On the other hand, if the network has bandwidth capacity that far exceeds the applications requirements, the network should pick a duty cycle that meets the needs of the application in the most efficient manner, i.e., according to the formula in the Table 4.3. Unless the message generation rate of application can be adjusted, the receiver should adjust in situ to the needs of the application. When the requirements of application are variable (or not know in advance), selecting a duty cycle corresponding to a worst case requirements, may reduce the efficiency in a couple of orders magnitude. Clearly in this case the receiver should adjust to the in situ needs of the application.

4.3.8 Summary of analysis results

Full slot listening

The energy efficiency comparison for all the methods is shown in Figure 4.6. For clarity, we translate these values into dB($20\log(E)$), show in the Figure 4.7. The figures are drawn under 1% total duty cycle and the number of neighbors is 6.

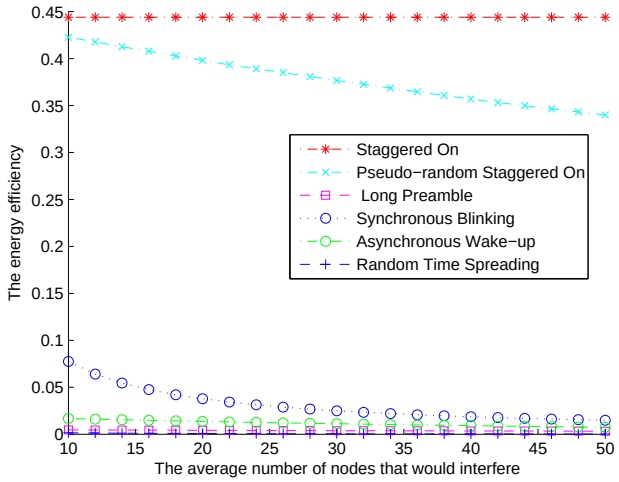

Figure 4.6: Energy efficiency comparison with Full Slot Listening

- Staggered On achieves the highest energy efficiency, followed by Pseudorandom Staggered On, Synchronous Blinking, Asynchronous Wake-up, Long Preamble, and Random Time Spreading.
- Pseudorandom Staggered On achieves slight worse energy efficiency than Staggered On, but still far better than other approaches.
- The energy efficiency of the Synchronous Blinking case, Random Time-Spreading and Pseudo-Random Staggered On case decrease with the number of interfering nodes.

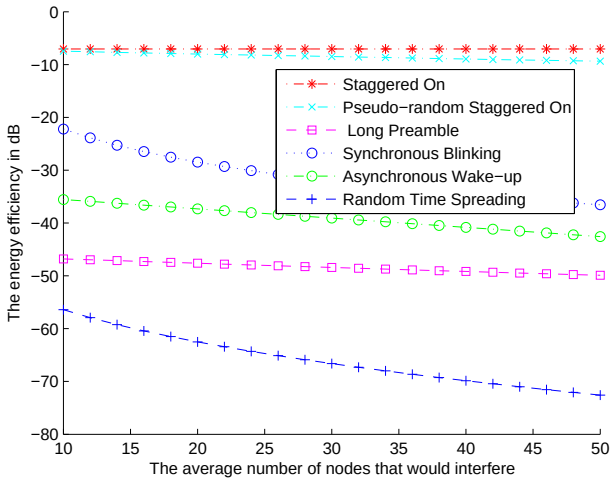

Figure 4.7: Energy efficiency comparison with Full Slot Listening (in dB)

- The more scheduling, the more energy efficiency we can get. By scheduling receiver as the Staggered On case, we can get highest energy efficiency.

Partial slot listening

The maximal energy efficiency comparison for these methods is shown in Figure 4.8 where $c_p = 0.1$. Those values are computed numerically by varying $p_t \in [0, 1]$.

Several observations:

- Staggered On and Synchronous Blinking case can benefit from partial slot listening. Their energy efficiency increases with a factor approaching 2 as $\delta_p \to 0$.

- Staggered On can achieve a 70% energy efficiency, then followed by Synchronous Blink, Asynchronous Wake-up, and Long Preamble case. In fact, they are in the same order as before.

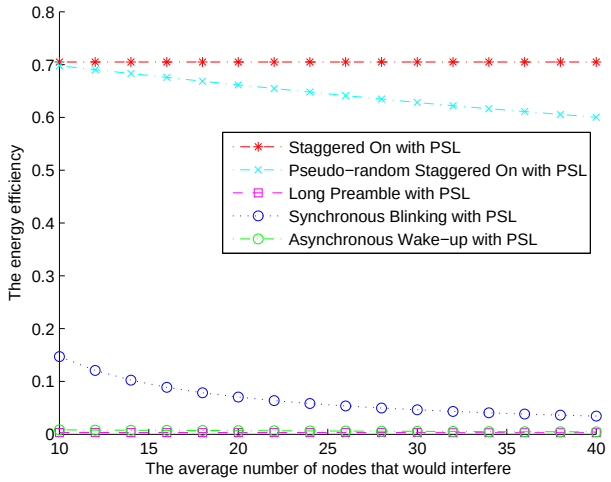

Figure 4.8: Energy efficiency comparison with Partial Slot Listening (PSL)

4.4 Challenges of implementing stabilizing receiver centric protocols

The analysis of the previous section indicates that it is possible to achieve one or two orders of better energy efficiency than current best practice. However, achieving the efficiencies associated with the two Staggered-On cases shown in Figure 4.7 requires solving engineering issues such as time-synchronization, management of neighbor tables. A stabilizing receiver centric protocol must guarantee that when nodes are out of sync, they can be rediscovered and resynchronized within reasonable time period.

Asynchronous neighbor discovery

For an always-on sensor network, neighborhood discovery is not an issue, since packet transmission can be overheard by neighboring nodes. However, in a low duty cycle mobile network, two neighboring nodes may never communicate because their

wakeup schedules never overlap. Even worse, for any wakeup schedule that is based on time synchronization, unsynchronized nodes may be lost forever because synchronized nodes and unsynchronized nodes are mutually unaware of each other.

In addition, link quality dynamics and clock variation also require continuous neighbor discovery. In [35], multiple schedules (time zones) have been shown to exist consistently in a network of 50 Mica2Dot motes using the S-MAC protocol. Our simulations also confirm that network partitions can break down time synchronization in low duty cycled network. We conclude that a synchronous protocol must coexist with energy efficient neighbor discovery in mobile, dynamic low duty cycle networks.

Time synchronization

Both Staggered On and Pesudo-random Staggered On require time synchronization. Any time sync protocol that is not inherently based on always on receivers can be applied in the receiver centric protocols. One such incompatible protocol is RBS [22], because RBS assumes all receivers are awake during synchronization. For typical physical clock, the period of time synchronization is 2 to 10 minutes. Considering the typical slot length is $5ms$, the duty cycle for time synchronization is less than 0.004%, which is negligible. Many time synchronization protocols such as [26][25][37] can be combined with receiver centric protocols. However, to guarantee stabilization, time synchronization protocol must be stabilizing.

In next chapter, we will provide a stabilizing receiver centric protocol called O-MAC.

4.5 Conclusion

In this chapter, we have argued that the receiver radio dominates the power consumption. By deriving the bounds on energy efficiency for various models, we have shown that receiver scheduling can increase the energy efficiency by orders of magnitude. In addition, we have provided two new receiver based scheduling methods: Staggered On and Pseudo-randomized Staggered On and designed one new MAC protocol that achieves the near optimal energy efficiency. The adaptivity in the protocol requires matching the duty cycle of the communication system to the needs of the application across variations in message generation rate. Finally, we have described challenges of implementing stabilizing receiver centric protocols.

CHAPTER 5

OMAC: A STABILIZING RECEIVER CENTRIC PROTOCOL FOR MOBILE NETWORKS

In this chapter, we present a stabilizing receiver centric MAC protocol, where robust asynchronous discovery occurs in parallel with synchronous (receiver centric unicast and broadcast) communication. Through asynchronous discovery, all unsynchronized nodes can rediscover the network within a bounded time, thus stabilizing synchronous communication. To minimize the energy consumption of asynchronous discovery, we design a wakeup schedule that achieves the optimal bound for 3 state (listen, beacon, and sleep) radios. And, to minimize the energy consumption of synchronous communication, we design a distributed stabilizing controller that adapts the duty cycle at each node to correspond to that node's actual communication traffic levels.

5.1 Introduction

Current practice in wireless sensor networks has exploited several wakeup-sleep schedules, which are broadly classified as synchronous or asynchronous. Synchronous approaches [52, 49, 53, 15, 28] have been shown to achieve higher energy efficiency than asynchronous ones [41, 54], both by analysis [53, 15] and by simulation [28].

5.1.1 Applying synchronous protocols in a mobile network

Bootstrapping a network requires initial discovery of nodes and initial node/time synchronization, and so cannot rely on a synchronous protocol. Bootstrapping has typically been handled, if at all, by running a less energy efficient asynchronous MAC in order to setup the network and then switching to a high efficiency protocol for routine operation. This two phase approach is unsuitable for mobile networks, which need to continuously add and subtract nodes and also to partition and recombine.

For an always-on sensor network, neighborhood discovery is not an issue, since packet transmission can be overheard by neighboring nodes. However, in a low duty cycle mobile network, two neighboring nodes may never communicate because their wakeup schedules never overlap. Even worse, for any wakeup schedule that is based on time synchronization, unsynchronized nodes may be lost forever because synchronized nodes and unsynchronized nodes are mutually unaware of each other.

In addition, link quality dynamics and clock variation also require continuous neighbor discovery. In [35], multiple schedules (time zones) have been shown to exist consistently in a network of 50 Mica2Dot motes using the S-MAC protocol. Our simulations also confirm that network partitions can break down time synchronization in low duty cycled network. We conclude that a synchronous protocol must coexist with energy efficient neighbor discovery in mobile, dynamic low duty cycle networks.

Both [47] and [55] have developed schedules that have the property that, for all possible time shifts for a schedule, there is a time when the active slots of nodes overlap. Their schedules are derived for 802.11 radios, where a node can both beacon and listen in one active slot (also called beacon interval). In WSN radios such as 802.15.4, however, there is a high cost and latency if nodes beacon and listen in

the same slot (as we explain in Section 5.3). This leads us to introduce a three-state (beacon, listen, and sleep) WSN radio model, for which we are unaware of any previous theoretical results about optimal bounds or schedules.

Another central aspect of bootstrap in a low duty cycle system is to select the duty cycle of the schedule so that the traffic in the network is communicated with high energy efficiency. Analytical results [15] indicate that different traffics require different duty cycles to achieve optimal energy efficiency. If the duty cycle is lower than required, higher collision or sender buffer overflow can happen; if the duty cycle is higher than required, energy is wasted on idle listening. In other words, it is important that nodes adapt their duty cycle according to traffic changes.

5.1.2 Summary of the results

In this work, we design an energy efficient protocol for mobile WSNs that provides robust asynchronous and receiver centric synchronous communication. Specifically, it exports three services in parallel: asynchronous discovery, synchronous unicast, and simultaneous broadcast. Each service has a different wakeup time interval that either yields no overlap with the others or or overlaps with low probability; this is achieved via pseudo-random slot selection per service.

We formulate the problem of optimal neighbor discovery for a duty cycled network, and provide a class of optimal wakeup schedules that achieves neighbor discovery with minimum energy. Our 3-state schedules consume $1/\sqrt{2}$ of energy required by other approaches to achieve neighbor discovery in a discovery frame.

Next, to maintain an optimal duty cycle in the presence of dynamic traffic, we provide a duty cycle adaptation mechanism that is based on feedback from channel

utilization and collisions. We identify a metric, the **Activity Ratio**, using which we transform the duty cycle optimization problem into a fixed point control problem.

Finally, experimental results on the TelosB low power radio platform, as well as simulation results, show that our protocol provides stabilization via asynchronous discovery, as well as higher energy efficiency by using synchronous communication combined with duty cycle adaptation.

5.1.3 Related work

In [19], radio communication is revealed as the dominant power consumer among all components. There are a large number of protocols to schedule radio wakeup:

Synchronous protocols:

The simplest synchronous protocol wakes up all the nodes at the same time. S-MAC [52] and T-MAC [49] are notable variants of this approach. At a time, only one receiver can receive a unicast message while the other receivers idle listen.

The concept of receiver-centric power management protocols was first introduced in [15]. In receiver-centric protocols, receivers are scheduled to wake up in different slots. Ideally, only one receiver wakes up at a slot and potential sender only need to wakeup to transmit in that slot; idle listening is avoided. The key feature of the protocol is that it avoids the vast majority of collisions by staggering or scheduling receiver on times rather than staggering or scheduling transmission times. In [28], one receiver-centric MAC protocol called Crankshaft is presented, wherein node ID is used to decide wakeup schedule. The duty cycle in Crankshaft MAC is fixed and not easily changed.

Asynchronous protocols:

Asynchronous approaches have been extensively studied and adopted in MAC layer protocols, in part because they assume less about node coordination and time synchronization and this reduces system complexity. A well-known approach uses Low Power Listening (LPL) whereby nodes wake up periodically and independently to check channel activity, and when a node wishes to send a message it sends out a long preamble first to wake up the receiver [41, 54].

Theoretical results on efficient, deterministic wakeup-sleep schedules for 802.11 networks are presented in [47] and [55]. To minimize energy consumption, all nodes wake up according to a schedule with only $\sqrt{N}$ wakeup slots out of total N slots. The schedule guarantees that, for any two nodes, there is a slot during which both nodes are awake, no matter what time shift exists between the two schedules. In [47], a dynamic scheduler is developed that uses only $\sqrt{2N}$ wakeup slots to get different duty cycles by changing parameters, based on a torus quorum system.

5.2 Energy efficient protocol design for mobile networks

Our solution to integrating the unicast and broadcast aspects of communication, which are synchronous for reasons of energy efficiency, with the robust discovery and duty cycle adaptivity aspects of control, which are asynchronous, is to create virtual MAC services that operate in parallel. In particular, unicasts, broadcasts, and discover each have different wakeup time intervals that either do not overlap with the others or that overlap with low probability; this is realized via pseudo-random slot selection.

5.2.1 Major components in protocol

Our protocol thus has four components, as shown in Figure 5.1.

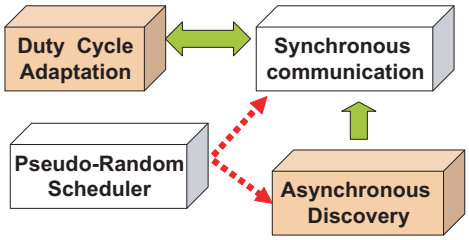

Figure 5.1: The major components in the protocol

a) Synchronous communication

Once nodes have discovered each other (including their ID), they share sufficient information to send and receive synchronous communications with respect to neighbors. Recall that our protocol supports synchronous unicasts and simultaneous broadcast. Specifically, the unicast protocol performs the following tasks at each node: it maintains at each node a pseudo-random unicast-receiver schedules for every neighbor; it buffers messages; it schedules transmission to correspond to the time when the intended neighboring recipient's receiver is on; it explicitly ACKs each message, if the ACK mode is enabled; and it deals with collisions by employing a low complexity back-off scheme. (We note that unicast protocol allows a synchronous broadcast to be simulated by sending one unicast to each neighbor. Such a broadcast is invoked by identifying the receiver ID as U_BCAST.)

The simultaneous broadcast protocol is used when a broadcast must serve as a synchronization barrier. (If a broadcast is performed merely to get data to all of a nodes neighbors, we advocate simulating the broadcast with a series of unicasts as described above.) In the simultaneous broadcast case, instead of requiring the

transmitter to accommodate the receiver's schedule, the receiver is required to accommodate the transmitters schedule. That is, each node maintain a pseudo-random broadcast schedule for each of its neighbors; during its broadcast slot, all of its neighbors are required to listen. Such a broadcast is invoked by identifying the receiver ID as S_BCAST.

b) Asynchronous discovery

Asynchronous discovery acts as the base line of communication and stabilization. It uses a wakeup schedule that we described in Section 5.3. The discovery beacons enable each node to know all of its neighboring nodes within a single discovery frame length with high probability, and thus make the network robust against partitioning that may be induced by mobility and link dynamics.

The asynchronous discovery protocol also provides an interface for asynchronous broadcast communication for services such as time synchronization which must work even when the network is not synchronized. In this broadcast is invoked by identifying the receiver ID as A_BCAST, the content of the broadcast is implemented as an overlay atop the discovery beacon messages of the protocol.

c) Duty cycle adaptation

This component deals with cases where the traffic is not well characterized at compile time. It uses the mechanism described in Section 5.4, to adjust the duty cycle at each node according to traffic changes experienced locally.

d) Pseudo-random scheduler

The purpose of the pseudo-random selection of slots is to avoid systematic conflicts between (a) the schedules of neighbors for each of the communication services and (b) the schedules of the communication services of each node.

Specifically, a pseudo-random number generator is used in synchronous communication to select the slot during which a receiver node will listen within each frame. It is necessary for any neighboring node that wishes to send a message to the node to be able to figure out this slot. This requires that the sending node be able to reproduce the same pseudo-random slot selection that was used by the receiver node. This can be achieved by transferring 1) the last slot assignment, 2) the current frame length, and 3) the seed of the receiver to the sender via the discovery beacons. A pseudo-random number generator is also used in asynchronous discovery for randomizing the discovery wakeup schedule. If all nodes wakeup at exactly the same time their wakeup and beacon slots will be identical and no discovery will occur. Randomizing the schedule avoids this "zero-shift" issue that is inherent to any deterministic schedule.

Figure 5.2 shows the communications between neighbors.

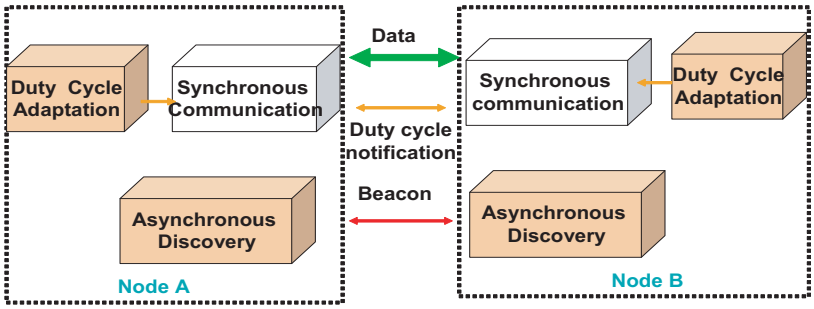

Figure 5.2: The communications between neighbors

5.2.2 Key protocol parameters

Synchronous unicast frame length

The duty cycle of the synchronous communication is determined by the application. If the frame length is N, the duty cycle is $1/N$. Typically, N is $\in [50, 500]$ for a low duty cycle application.

Broadcast frame length

A key difference between unicast frame length and broadcast frame length is that, because broadcast events are high energy events compared to unicasts (i.e., all of the nodes neighbors must be awake for the event), the broadcast frame length will typically be significantly longer than the frame length used in synchronous unicasts, e.g. 500 to $10,000$ slots.

Discovery frame length

The duty cycle of the discovery protocol is determined by the latency requirements for discovery. Typically these requirements can be fixed at compile time, but if they were to change in situ or to be different from region to region, it would be possible to change the frame length dynamically. The appropriate discovery frame length is much larger than normal data frame length. Typically, it is above $10,000$ slots for a low duty cycle application.

Duty cycle adaptation parameters

The details of these parameters are described in Section 5.4.

5.2.3 Relationship with other components

The relationship between our MAC protocol and other network protocols is shown in Figure 5.3.

A key decision is whether to subsume time synchronization within the MAC. On the one hand, time synchronization requires neighbors to exchange time information

without assuming synchronous communication, and on the other hand, synchronous MAC protocols needs time synchronization. This chicken-and-egg dilemma suggests that subsuming time synchronization is simple; however, this is undesirable as it greatly limits the portability of the MAC across network platforms. We avoid the assumption simply by letting time synchronization messages be exchanged via asynchronous broadcasts. In addition, global time synchronization must maintain a global state — global time. This global state must be stabilizing despite of node joining or leaving, network merging and partition. However, designing such protocol is not an easy task. In our OMAC protocol, we use neighbor time tracking, where each node only records the time differences to its neighbors. This avoids maintaining global time stabilization, but providing sufficient time sync support for OMAC, which only depends on neighboring time differences.

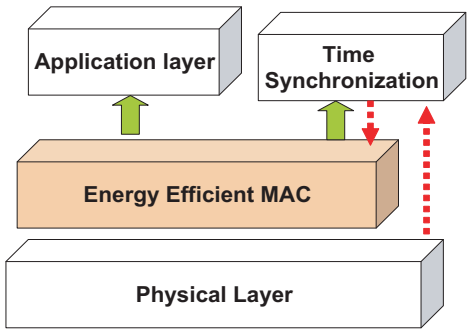

Figure 5.3: The Relationship between MAC and other componets

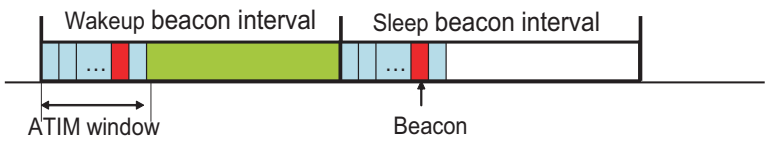

Figure 5.4: Structure of the 802.11 beacon interval

5.3 Continuous asynchronous discovery

5.3.1 Why a 3-state radio model?

In the power saving mode of 802.11, time is divided into beacon intervals in each of which up to one data packet can be communicated. As shown in Figure 5.4, a beacon interval begins with an ATIM (ad hoc traffic indication map) window, which consists of multiple slots that a node can choose from to send its beacon. Each node is awake during each ATIM window. Previous wakeup schedules [47, 55] associate a wakeup or sleep state with each beacon interval. It follows that if the wakeup beacon intervals of two neighboring nodes overlap, they will exchange beacons with high probability.

The 2-state 802.11 model is however not easily extended to WSN radios:

1. The cost of sending beacons in WSN radios is comparable to that of sending data packets, because of small data packet sizes (typically several tens of bytes), while in 802.11 it is much smaller, as the data packet sizes (typically > 512 bytes) are much larger. Thus using ATIM window to avoid collision is costly for WSN radios.

2. Using the 802.11 beacon interval structure in low data-rate WSN radios (say $250kbps$) would yield interval lengths of over $200ms$. This would imply a large latency (e.g., $80s$ for 5% duty cycle) that can be unsuitable for sensor networks.

Considering the cost and latency, we eschew design of fine-grain wakeup schedules for WSN radios. In keeping with the trend set by the 802.15.4 standard, where the unit of sending and receiving is a packet, as opposed to a bit or a byte, we select packet-sized slots as the unit of our schedules. In every slot, a node can be in only one of three states: beacon, listen, or sleep.

5.3.2 System model

The network consists of a number of mobile nodes. The operation of each radio can be viewed as a sequence of frames, each consisting of a constant number of constant time length slots as shown in Figure 5.5. We assume initially that the slot boundaries across different nodes are aligned, but then relax this assumption later. Each node u follows a schedule S_u that dictates its state per slot. S_u is represented as:

$$S_u(j) = \begin{cases} 0 & \text{if node } u \text{ sleeps in slot } j \\ 1 & \text{if node } u \text{ beacons in slot } j \\ 2 & \text{if node } u \text{ listens in slot } j \end{cases}$$

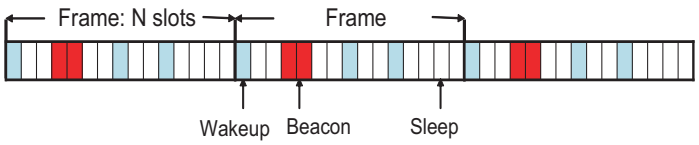

Figure 5.5: A schedule in the 3-state radio model

- N: number of slots in a frame.
- Let u and v range over the network nodes. We denote the number of sleeping, beaconing, and listening slots per frame as n_u^s, n_u^b, n_u^l and n_v^s, n_v^b, n_v^l respectively.

5.3.3 Problem statement

Our goal is that each node wakes up as little as possible, while being able to discover new nodes as quickly as possible. Before defining the problem formally, we distinguish two types of neighbor discovery: unidirectional discovery and mutual discovery.

Definition 5 (unidirectional discovery) *We say u and v achieve unidirectional discovery iff for any integer shift $T \in [0, N-1]$, $\exists i, j$ such that $(S_u(i+T) = 1 \wedge S_v(i) = 2) \vee (S_u(j+T) = 2 \wedge S_v(j) = 1)$.*

Unidirectional discovery implies that at least one node can discover the other node within a frame, but there is no guarantee that both nodes can find each other. However, mutual discovery guarantees discovery in both directions.

Definition 6 (mutual discovery) *We say u and v achieve mutual discovery iff for any integer shift $T \in [0, N-1]$, $\exists i, j$ such that $(S_u(i+T) = 1 \wedge S_v(i) = 2) \wedge ((S_u(j+T) = 2) \wedge S_v(j) = 1)$.*

We now define the optimal SBL (Sleep-Beacon-Listen) problem for unidirectional discovery and mutual discovery.

Definition 7 (optimal SBL for unidirectional discovery) *Given a fixed N, design S_u and S_v so as to achieve (1) $\min\{n_u^b + n_u^l + n_v^b + n_v^l\}$ and (2) unidirectional discovery.*

The SBL problem is thus to minimize the number of active slots (beacon slots and listen slots) while guaranteeing unidirectional discovery. The SBL problem for mutual discovery is defined likewise as:

Definition 8 (optimal SBL for mutual discovery) *Given a fixed N, design S_u and S_v so as to achieve (1)* $\min\{n_u^b + n_u^l + n_v^b + n_v^l\}$ *and (2) mutual discovery.*

5.3.4 Optimal bound for deterministic schedule

In this section, we characterize the minimum number of active slots required to achieve unidirectional discovery and mutual discovery.

Theorem 22 *To achieve unidirectional discovery, the following condition must hold:*

$$n_u^b \cdot n_v^l + n_u^l \cdot n_v^b \geq N$$

Proof: Since both u and v may be out of sync with any shift, without loss of generality, we fix the schedule of node u and only shift the schedule of v. In other words, we let $S_v(i+T)$ be $S_u(i)$, where T is the shift and $0 \leq i < N$. For any beacon slot $j \in [0, N-1]$ in u, the total number of listening slots that j overlaps during the shift is n_v^l. For all beacon slots in u, the total number of beacon-listen overlapping pairs is $n_u^b \cdot n_v^l$. Similarly, the total number of listen-beacon overlapping pairs is $n_u^l \cdot n_v^b$. None of these pairs repeat during the shift $T \in [0, N-1]$. Then the total number of "discovery" slots, N_d, is $n_u^b \cdot n_v^l + n_u^l \cdot n_v^b$. On the other hand, by unidirectional discovery, at least one beacon-listen or listen-beacon pair is guaranteed for every shift. So the total number of "discovery" slots, N_d, is at least N. So: $n_u^b \cdot n_v^l + n_u^l \cdot n_v^b = N_d \geq N$

■

Theorem 23 *To achieve mutual discovery, the following condition must hold:*

$$n_u^b \cdot n_v^l + n_u^l \cdot n_v^b \geq 2N$$

Proof: The proof is similar to Theorem 22, the only difference being that mutual discovery achieves at least one beacon-listen and one listen-beacon pair per every shift. So the total number of "discovery" slots, N_d, is at least $2N$. Thus we have: $n_u^b \cdot n_v^l + n_u^l \cdot n_v^b = N_d \geq 2N$

■

For ease of deploying WSNs, it is convenient to use the same wakeup schedule for all nodes. We call such schedule design symmetric. We get the following corollary by using Theorem 22 and Theorem 23.

Corollary 1 (Bound for symmetric SBL solution) *Given N, $n_u^b = n_v^b = n_b$, and $n_u^l = n_v^l = n_l$, any SBL solution must satisfy:*

$$2n_b \cdot n_l \geq N \textit{(for unidirectional discovery)}$$

$$n_b \cdot n_l \geq N \textit{(for mutual discovery)}$$

For an optimal symmetric design, the minimum number of total active slots are determined by the following corollary 2.

Corollary 2 (Bound for optimal symmetric SBL solution) *Given N, $n_u^b = n_v^b = n_b$, and $n_u^l = n_v^l = n_l$, the optimal SBL solution must satisfy :*

$$\min\{n_u^b + n_u^l + n_v^b + n_v^l\} \geq \begin{cases} 2\sqrt{2N} & \textit{(for unidirectional discovery)} \\ 4\sqrt{N} & \textit{(for mutual discovery)} \end{cases}$$

To achieve this bound, n_b and n_l must satisfy:

$$\begin{cases} n_b = n_l = \sqrt{2N}/2 & \textit{(for unidirectional discovery)} \\ n_b = n_l = \sqrt{N} & \textit{(for mutual discovery)} \end{cases}$$

(Note: For ease of exposition of theorems and proofs, we round $\sqrt{N}$ to the nearest larger integer when it is not an integer.)

Proof: $\min\{n_u^b + n_u^l + n_v^b + n_v^l\} = 2 \cdot \min\{n_b + n_l\} \geq 4\sqrt{n_b \cdot n_l}$. By Corollary 1, $\sqrt{n_b \cdot n_l} \geq \sqrt{2N}/2$ holds in the unidirectional discovery case, and $\sqrt{n_b \cdot n_l} \geq \sqrt{N}$ holds in the mutual discovery case. So, $\min\{n_u^b + n_u^l + n_v^b + n_v^l\} \geq 4\sqrt{n_b \cdot n_l} \geq 2\sqrt{2N}$ (for unidirectional discovery) or $4\sqrt{N}$ (for mutual discovery). To make the first "$\geq$" equal, n_b and n_l must satisfy $n_b = n_l$, so we can get the value of n_b and n_l in Corollary 2.

■

5.3.5 Optimal deterministic schedule

In this section, we describe schedules that achieve the optimal bound. To satisfy discovery (unidirectional or mutual), the active slots of a node and its neighbor must have at least one overlap for all possible time shifts. The quorum based schedules in [47] satisfy this condition, but they are 2-state schedules. We further extend them to 3-state schedules and achieve significantly better performance (and that too deterministically).

First, we introduce the concept of a block. We divide a frame into blocks, each with X slots such that there is at least one active (beacon or listen) slot per block.

- X: block size
- Y: number of blocks in a frame
- N: frame size, $N = X \cdot Y$

Optimal unidirectional discovery schedule

Although unidirectional discovery only guarantees discovery in one direction, it consumes less energy than mutual discovery. In addition, if one node discovers the other one, it can notify the other one about its schedule by using a synchronous ACK.

Theorem 24 *The schedule based on the following rules achieves unidirectional discovery:*

1. $S(i \cdot X) = 1$ *for all integer* $i \in [0, Y-1)$.
2. $S(X \cdot (Y-1) + j) = 2$ *for all integer* $j \in [0, X/2]$. *(Note: by this rule,* $S(X \cdot (Y-1)) = 2$*)*
3. $S(X \cdot (Y-1) + (X/2+1)) = 1$.

One instance of these schedules is shown in Figure 5.6. The numbers in the table are slot indexes. Generally speaking, this schedule requires node beaconing over the

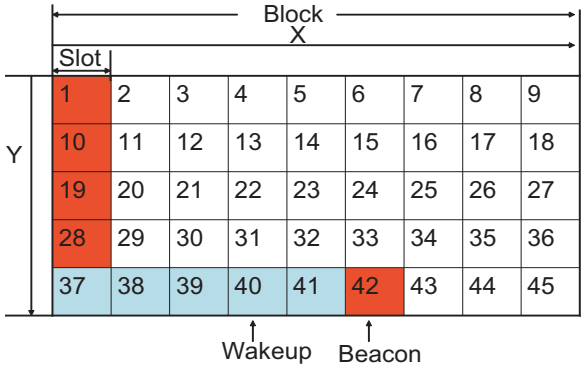

Figure 5.6: Optimal wakeup schedule for unidirectional discovery

slots in one column (except for one), and listening in more than half of the slots in the excepted row. The proof validates that unidirectional discovery holds for every possible shift.

Proof: For any two nodes u and v, without loss of generality, let the time v be shifted wrt u by Δ, where Δ is an integer expressed as: $\Delta = a \cdot (X \cdot Y) + b \cdot (X) + c$ and $(0 \leq b < Y, 0 \leq c < X)$. So their schedules are:

$$S_u(j) = S(j); \quad S_v(j) = S(j + \Delta) = S(j + b \cdot X + c)$$

Case 1 $(c = 0)$: $S_u(j) = S(j)$ and $S_v(j) = S(j + b \cdot X)$. Apparently, if $b = 0$, then $S_u(j) = S(j) = S_v(j)$, both nodes use the same wakeup schedule and are in sync. In the following, we assume $b \neq 0$. When $j = (Y - 1 - b)X$, $S_u(j) = S((Y-1-b)X), S_v(j) = S((Y-1)X)$. According to rule 1, and $(Y-1-b) \in [0, Y-1)$, so $S_u(j) = S((Y - 1 - b)X) = 1$. According to rule 2, $S_v(j) = S((Y - 1)X) = 2$. So, node v can listen the beacon of node u at slot $j = (Y - 1 - b)X$.

Case 2 $(c \neq 0, b = 0)$: $S_u(j) = S(j)$ and $S_v(j) = S(j + c)$.

If $c \leq X/2+1$, let $j = X \cdot (Y-1) + (X/2+1-c)$. Since $0 \leq (X/2+1-c) \leq X/2$, $S_u(j) = S(X \cdot (Y - 1) + (X/2 + 1 - c)) = 2$, based on rule 2. In addition, $S_v(j) = S(j + c) = S(X \cdot (Y - 1) + (X/2 + 1)) = 1$, according to rule 3. So, u can listen to the beacon of v in slot $j = X \cdot (Y - 1) + (X/2 + 1 - c)$.

If $c > X/2 + 1$, let $j = X \cdot (Y - 1) + (X - c)$. Since $0 \leq (X - c) \leq X/2$, $S_u(j) = S(X \cdot (Y-1) + (X-c)) = 2$, based on rule 2. In addition, $S_v(j) = S(j+c) = S(X \cdot (Y - 1) + X) = S(0) = 1$, according to rule 1. So, u can listen to the beacon of v in slot $j = X \cdot (Y - 1) + (X - c)$.

Case 3 $(c \neq 0, b \neq 0)$: $S_u(j) = S(j)$ and $S_v(j) = S(j + b \cdot X + c)$.

If $c \leq X/2+1$, let $j = X \cdot (Y-1) - b \cdot X$. $S_u(j) = S(X \cdot (Y-1-b)) = 1$, based on rule 1. In addition, since $c \leq X/2+1$, $S_v(j) = S(j+b \cdot X+c) = S(X \cdot (Y-1)+c) = 2$, according to rule 2. So, v can listen to the beacon of u in slot $j = X \cdot (Y-1) - b \cdot X$.

If $c > X/2+1$, let $j = X \cdot (Y-1) + (X-c)$. Since $0 \leq (X-c) \leq X/2$, $S_u(j) = S(X \cdot (Y-1) + (X-c)) = 2$, based on rule 2. In addition, If $b \neq Y-1$, $S_v(j) = S(j + b \cdot X + c) = S(X \cdot (Y-1) + X + b \cdot X) = S(b \cdot X) = 1$, according to rule 1. So, u can listen to the beacon of v at slot $j = X \cdot (Y-1) + (X-c)$. If $b = Y-1$, let $j = X \cdot (Y-1) + (X/2+1)$. So, $S_u(j) = 1$, based on rule 3. In addition, $S_v(j) = S(j + b \cdot X + c) = S(X \cdot (Y-1) + (Y-1) \cdot X + 1 + X/2 + c) = S((Y-1)X + (c+1-X/2)) = 2$, according to rule 2. So, v can listen to the beacon of u at slot $j = X \cdot (Y-1) + (X/2+1)$.

■

Based on the schedule described in Theorem 24, we achieve the optimal bound, as is shown in the following corollary.

Corollary 3 *If $X = 2 \cdot Y$, the schedule in Theorem 24 is optimal.*

Proof: When $X = 2 \cdot Y$, $X = \sqrt{2N}$, so the total number of active slots is $2 \cdot X = 2\sqrt{2N}$, which is the optimal bound for unidirectional discovery.

■

Optimal mutual discovery schedule

Mutual discovery is required for many applications, especially in mobile networks. In addition, it also has inherent robustness against unaligned slot boundaries, as shown in Theorem 27. The following theorem defines an optimal mutual discovery schedule.

Theorem 25 *The schedule based on the following rules achieves optimal mutual discovery when $X = Y$:*

1. *$S(i \cdot X) = 2$ for all integers $i \in [0, Y-3]$.*

2. $S(X \cdot (Y-1)+j) = 1$ *for all integers* $j \in [1, X-1]$.

3. $S((Y-2) \cdot X) = 1$

4. $S(X \cdot (Y-2)+1) = 2$; $S(X \cdot (Y-4)+1) = 2$.

This schedule requires node beaconing in the slots of one column and listening in the slots of one row, with only a few exceptions as mentioned in the schedule. Figure 5.7 shows one instance of an optimal schedule for mutual discovery.

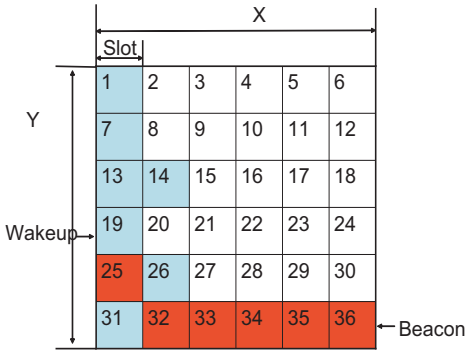

Figure 5.7: Optimal wakeup schedule for mutual discovery

Proof: Similar to that Theorem 24, it involves checking that mutual discovery holds under every possible shift.

■

5.3.6 Optimal schedule without slot alignment

In this section, we relax the assumption that the slots of different nodes are aligned. A successful communication happens when a receiver hears the full preamble and detects the start of frame delimiter (SFD), as shown in Figure 5.8. After SFD is detected, the receiver can stay active during the slot to receive the data packet. Since

the preamble length is small compared with the slot length, we say discovery occurs when a receiver wakes up for one slot and a preamble is received during that slot.

Unidirectional discovery schedule

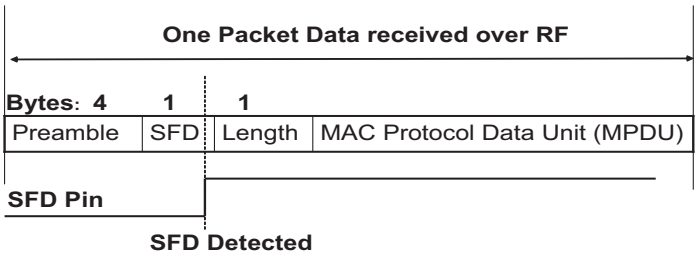

Figure 5.8: Receiving a packet in CC2420, an 802.15.4 radio

Theorem 26 *The schedule defined by Theorem 24 achieves unidirectional discovery even when the slots of different nodes are unaligned.*

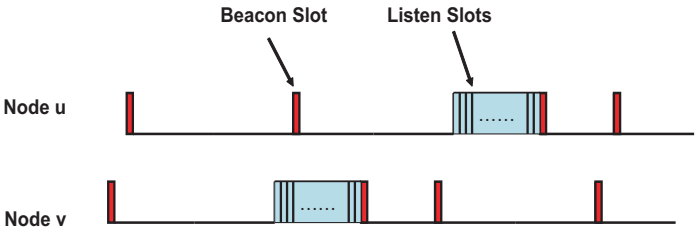

Figure 5.9: Unidirectional discovery without slot alignment

The schedule is shown in Figure 5.9. The block that nodes listen is called half listen block, where nodes listen more than half of the block. Except this block, node beacons at the start slot of every other block. If this half listen block overlaps with

any beacon slot, neighbor is discovered. The proof is to check this can happen in all possible shifts.

Proof: The proof is similar to Theorem 24, except we validate that discovery occurs for all possible real number shifts. We omit the details here.

■

We note that not all schedules that achieve unidirectional discovery schedule in aligned slots case suffice unaligned slots case. A counter example is shown in Figure 5.10. This schedule is very similar to the schedule in Theorem 24, except that its beacon and listen slots are switched. It is easily checked that this schedule achieves unidirectional discovery in the aligned slots case but not in unassigned slots case.

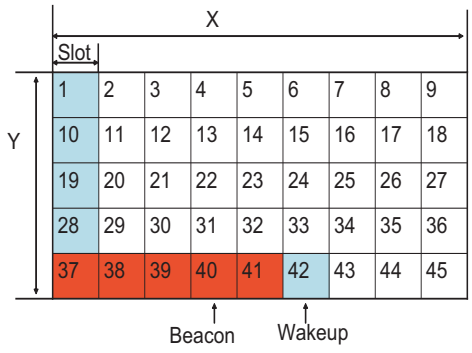

Figure 5.10: Unidirectional discovery schedule for aligned (but not unaligned) slots

In contrast, mutual discovery schedules have a robust feature that guarantees discovery even when slots are unaligned.

Mutual discovery schedule

Theorem 27 *Any algorithm that achieves mutual discovery when the slots of different nodes are aligned also achieves mutual discovery when those slots are unaligned.*

Proof: For any two nodes u and v, without loss of generality, let the time of v be shifted w.r.t u by Δ, where Δ is a real number, which can be expressed as: $\Delta = a \cdot (X * Y) + b * (X) + c + \delta$, where $(0 \leq b \leq Y-1,\ 0 \leq c \leq X-1,\ 0 < \delta < 1)$. When $\delta = 0$, the slots are aligned. According to mutual discovery, there exists a slot i in u that hears a beacon from v. When $\delta = 1$, there exists a slot j in u that hears a beacon from v. In other words, $S_u(j) = 2, S_v(j) = 1$. When $\delta \in (0,1)$, because u continues listening at slot j, $S_u(j+\delta) = 2$; also v beacons at the shifted time $j+\delta$, i.e., $S_v(j+\delta) = 1$. So, u hears a beacon at time $j+\delta$. Similarly, a v also always hears a beacon from u, which means mutual discovery is guaranteed even when slots are not aligned.

■

This theorem implies the following corollary:

Corollary 4 *The schedule in Theorem 25 achieves optimal mutual discovery even when the slots of different nodes are unaligned.*

5.3.7 Comparison with previous work

To use the optimal 2-state schedules proposed by [47] and [55], a node has to randomly select to beacon or listen during wakeup. To achieve mutual discovery, a 2-state schedule requires on average 8 discovery frames. This is because:

- In every frame, one node can discover the other one with probability 0.5.
- To have mutual discovery, the probability for every two discoveries is 0.5.

For a frame length N, to guarantee mutual discovery in 8 frames, our scheme requires $2\sqrt{8N} = 4\sqrt{2N}$, while an optimal 2-state schedule needs $8\sqrt{N}$ on average. So our

optimal 3-state schedule uses only $1/\sqrt{2}$ of energy that the 2-state schedule uses. Figure 5.11 shows the time taken to achieve mutual discovery in different duty cycle for neighbor discovery. In addition, our 3-state schedule guarantees mutual discovery, while the 2-state schedule only achieves this only with probability 0.5.

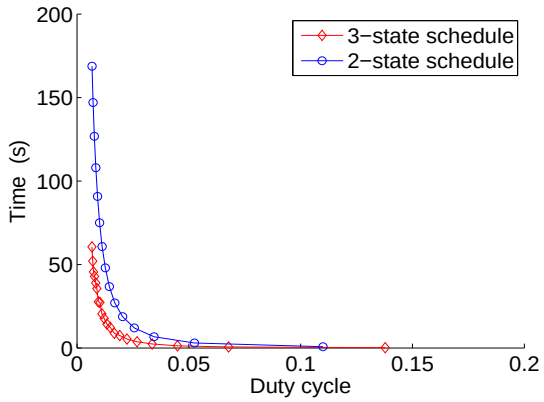

Figure 5.11: Comparison between 3-state schedule and 2-state schedule

5.4 Duty cycle adaptation

For any particular traffic level, there is an optimal duty cycle that maximizes the energy efficiency, i.e., the goodput, for a fixed power consumption. In this section, we focus on locally adapting the node duty cycle to that optimal point.

5.4.1 Receiver based collision detection

It is important to detect collisions at the receiver to estimate the incoming traffic. CC2420 provides CCA (Clear Channel Arbitration) status, by which we can detect collisions. Traditionally, CCA is used for implementing CSMA at the sender. However, it also allows the receiver to be aware of unreadable preambles caused

by overlapping transmissions. Consistent collisions indicate an under-provisioned communication system, while rare collisions indicate over-provisioning. The rate of receiver collision provides sufficient information for receiver to adjust its duty cycle.

5.4.2 Activity ratio as the control metric

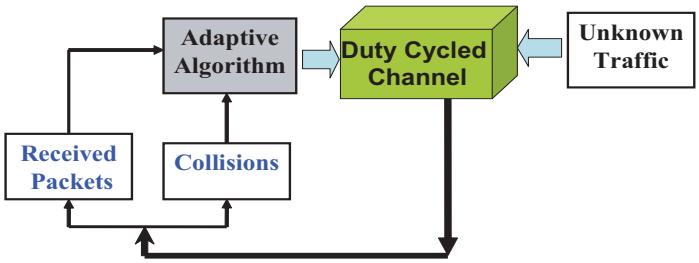

Figure 5.12: The structure of adaptive duty cycle control system.

Short term variations in traffic may be absorbed by a buffering mechanism implemented in the MAC protocol. But slightly longer term traffic variations require adaptation of the duty cycle. We have opted to use the control structure shown in Figure 5.12. There are two interesting points to note: 1) there is no attempt to directly send state information from the sender, and 2) control is based on knowledge of the number of collisions and the number of received packets. This scheme is purely local. In addition the control algorithm attempts to deal with interfering traffic from an unknown origin.

Let n_i be the total number of idle listening slots, n_c be the total number of collision slots, and n_r be the total number of successfully received slots. Then the **activity ratio**, defined as

$$r_a = \frac{n_r + n_c}{n_r + n_c + n_i},$$

captures the performance of the link over a wide range of environments. In particular, the optimal duty cycle depends strongly on the number of neighbors that send data to it and the amount of the other traffic in its vicinity. In practice, these metrics are hard to estimate and prone to large error. However, the activity ratio corresponding to the efficiency only, and only weakly depends on these hard to estimate aspects of the environment.

As a partial justification of this assertion, consider a uniform traffic pattern.

Theorem 28 *In a network with uniform traffic, the optimal receiver energy efficiency is achieved when the activity ratio is in* $[0.64, 0.75]$*, no matter how many neighbors are transmitting.*

Proof: Let η define the number of nodes sending to the receiver of interest. We denote the sender's duty cycle by d_s and the receiver's duty cycle by d_r. The probability p that a sender transmits when the receiver is on is $p = d_s/d_r$.
The receiver **energy efficiency** is defined as the ratio of energy spent on successful transmission to the total energy spent by the receiver. In this situation, the energy efficiency is:

$$E = \eta \cdot p(1-p)^{\eta-1}$$

It follows that the maximal energy efficiency occurs when,

$$p = p_0 = 1/\eta \qquad \Rightarrow \qquad d_r = \eta * d_s$$

As already mentioned, both η and d_s are unknown to the receiver and difficult to estimate accurately. However, the activity ratio in this case is:

$$r_a = 1 - \frac{n_i}{n_r + n_c + n_i} = 1 - P_{idle} = 1 - (1-p)^{\eta}.$$

The receiver energy efficiency is maximized when

$$r_{a\_opt} = 1 - (1 - p_0)^{\eta} = 1 - (1 - \frac{1}{\eta})^{\eta}$$

The value of r_{a_opt} is shown in Figure 5.13. Note that the value depends only weakly on η and is in the range $[0.64, 0.75]$.

■

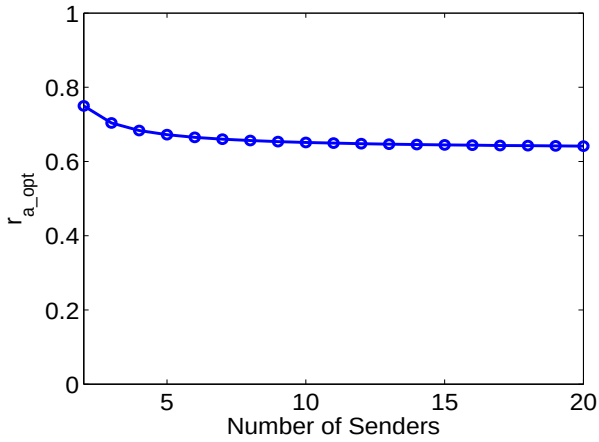

Figure 5.13: Activity ratio at optimal duty cycle for different number of senders

In the case of nonuniform traffic, there is a lower bound of activity ratio during which optimal receiver energy efficiency is achieved as described by following theory:

Theorem 29 *In a network with random traffic, the optimal receiver energy efficiency is achieved only when the activity ratio is in* $[0.64, 1]$*, no matter how many neighbors are transmitting.*

Proof: Let η define the number of nodes sending to the receiver of interest. We denote the sender i's duty cycle by d_i and the receiver's duty cycle by d_i. The

probability p_i that a sender transmits when the receiver is on is $p_i = d_i/d_r$.

The receiver **energy efficiency** is:

$$E = \sum_{i=1}^{i=\eta}(p_i \prod_{j=1,j\neq i}^{j=\eta}(1-p_j))$$

Firstly, when maximum E is obtained, the receiver duty cycle d_r should be smaller than $\sum_{i=1}^{i=\eta} d_i$, i.e. $d_{r0} \leq \sum_{i=1}^{i=\eta} d_i$.

The activity ratio in this case is:

$$\begin{aligned} r_a = 1 - P_{idle} &= 1 - \prod_{j=1}^{j=\eta}(1 - \frac{d_i}{d_{r0}}) \\ &\geq 1 - (\frac{\sum_{j=1}^{j=\eta}(1-\frac{d_i}{d_{r0}})}{\eta})^\eta = 1 - (1 - \frac{\sum_{j=1}^{j=\eta} d_i}{d_{r0}\cdot\eta})^\eta \\ &\geq 1 - (1-\frac{1}{\eta})^\eta \geq 1 - e^{-1} = 0.64 \end{aligned}$$

■

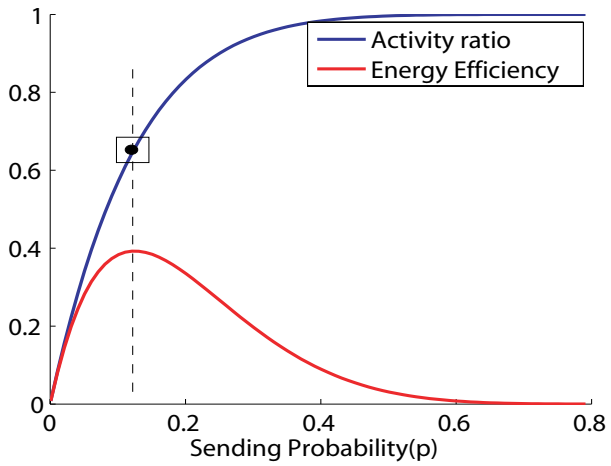

Figure 5.14: Activity ratio and energy efficiency as a function of traffic level

Figure 5.14 shows performance as a function of receiver duty cycle for a network with 8-nodes sending to a single receiver. Note that the activity ratio is monotonic

with significant slope around the optimal point. This implies that this parameter can be usefully controlled in the vicinity of the optimum. The box around the figure shows the operating region resulting from controlling $r_a \in [0.64, 0.75]$.

By using the activity ratio, a duty cycle optimization problem has been transformed into a fixed point control problem. Defining a feedback mechanism that controls the value of r_a to keep it in the range $[0.64, 0.75]$ works well.

5.4.3 A generic control algorithm

Figure 5.15 presents a simple generic control algorithm. Let:

- d_r be the receiver duty cycle,
- A_{max} be the maximum activity ratio,
- A_{min} be the minimum activity ratio,
- α be duty cycle increasing rate, and
- β be duty cycle decreasing rate;

Input: $r_a, d_r(k)$
Output: $d_r(k+1)$
Parameter: $A_{min}, A_{max}, \alpha, \beta$
if $(r_a > A_{max})$
  $d_r(k+1) = d_r(k) + d_r(k) * \alpha;$
else if $(r_a < A_{min})$
  $d_r(k+1) = d_r(k) - d_r(k) * \beta;$
end

Figure 5.15: Basic adaptive duty cycle algorithm

This algorithm converges quickly in most cases. It will settle on a good, but sub-optimal, rather than spend energy on incremental improvements.

Our previous derivation assumes uniform traffic from all the neighbors. However, in real deployments, traffic is frequently unbalanced. Ironically, unbalanced sending traffic increases energy efficiency by avoiding collisions. In the extreme case, where only one sender transmits at a time, the optimal energy efficiency is obtained when the activity ratio is 1. However, as the control point for the activity ratio approaches 1, the feedback becomes unstable. In order to maintain some duty cycle gain margin, our simulations indicated that the control point should not be above about 0.85, i.e. $A_{max} = 0.85$. The worst case occurs when uniform traffic from a large number of neighbors. In the limit as the number of neighbors goes to infinity, the activity ratio for uniform traffic is 0.64. Setting the control point below this value only introduces more idle listening, without any corresponding benefit, therefore $A_{min} = 0.64$.

The parameters α and β control the speed of convergence. Larger values improve the MAC efficiency by more quickly reaching the optimal dusty cycle, but too large a value will lead to instability. In order to estimate a reasonable maximum value of α, we consider an ad-hoc criterion that starting inside the desired operating box, shown in Figure 5.14, we should not overshoot landing outside the desired operating box. This suggests the following relationships.

5.4.4 Stabilization of feedback control algorithm

Although the previous generic algorithm achieves fast convergence, stability is an issue for this method. If α and β are not chosen carefully, it is possible that active ratio r_a may oscillate. To redress this problem, we have refined our algorithm so

that it guarantees stabilization (as proven via the Lyapunov method.) For reasons of space, we relegate the refined algorithm and its proof to Chapter 2 or a technical report [12].

5.4.5 Comparison with previous work

Extant work in duty cycle adaptation [54] first estimates incoming traffic and then calculates an optimal duty cycle. This approach has two issues: One, estimation of dynamic traffic can be error prone and yield an incorrect duty cycle. And two, collisions are ignored in measuring the incoming traffic. However, collisions resulting from a lower-than-desired duty cycle are an important indicator. To illustrate this, we designed a simulation where 5 senders transmit to one receiver with the same probability. The traffic estimation algorithm runs on the receiver, unaware of the number of possible senders. Figure 5.16 shows that traffic estimation that considers collisions performs better than one that excludes collisions as in XMAC [54]. As traffic per sender increases, the performance benefit of considering collisions also increases.

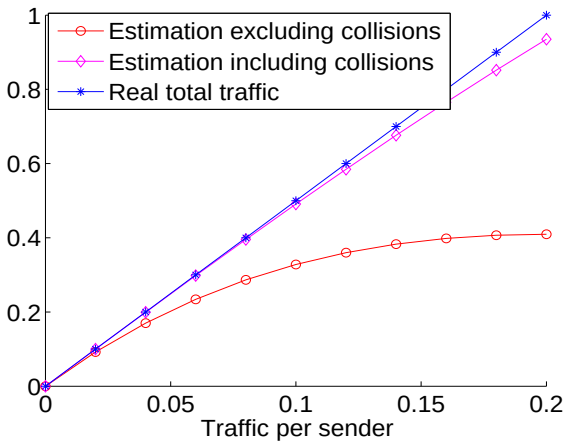

Figure 5.16: Comparison of estimation including vs. excluding collisions

5.5 Protocol evaluation

We have implemented the asynchronous discovery protocol and synchronous communication with duty cycle adaptation on the TelosB platform, using TinyOs 1.x and 2.x. In this section, we show their performance in experiments in a realistic mobile environment. In addition, we also evaluate their scalability in simulations using MATLAB. The experiments are at a scale of 10 nodes with neighborhood sizes up to 10 nodes, whereas the simulations consider neighborhood sizes of up to 40 nodes.

5.5.1 Asynchronous discovery protocol

To evaluate the performance of asynchronous discovery, we conducted an experiment and also designed a simulation to verify its scalability. Our experiment is based on the mutual discovery schedule described in Theorem 25. Since our asynchronous discovery protocol is purely local, the performance depends only on the degree of node, i.e., the number of neighbors. We believe a degree of 8 represents a relative dense network, so we used 9 TelosB motes. One attached to a laptop represented a node in an established network, while the other 8 were mobile nodes simultaneously attempting to join the node. The experiment was conducted in an engineering building, which is a noisy RF environment, including at least 3 802.11 access points in the vicinity of the nodes. All nodes were randomly placed and moved about on a flat table, within communication range of the attached node. To guarantee random shift of our clock, all nodes were reset after every experiment. The parameters of the experiment are:

- Slot length: 10 ms

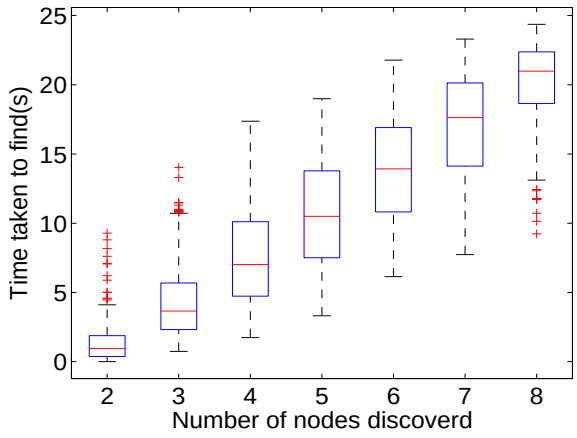

Figure 5.17: Time taken to discover neighbor

- Frame length: 2500 slots
- Packet length: 20 bytes

This implies that the frame length is about $25s$ and the duty cycle is $101/2500 \approx 4\%$. Figure 5.17 shows the time required to discover all the nodes. The figure reveals that time taken to discover all the 8 nodes does not exceed the frame length in 240 trials.

It should be pointed out that this experiment represents the worst case in the sense that typically most of the 8 nodes would be within range of more than one node in the existing network. Significantly improving the chance of early discovery by one of the in-network nodes.

Our simulation used the same parameters as the experiment. Its goal is to evaluate the discovery ratio, which is defined as the percentage of discovered nodes among all neighbors within one frame. Figure 5.18 shows that when the number of neighbors increases, the discovery ratio decreases. This is due to increased beacon collisions induced by increasing the number of neighbors. However, even with a very high

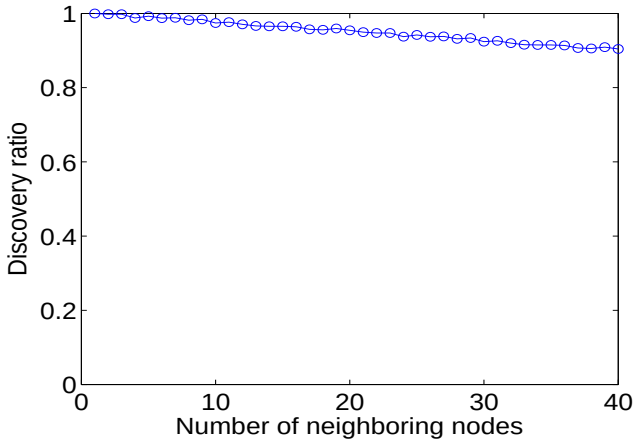

Figure 5.18: Scalability of asynchronous discovery

number of neighbors, the asynchronous discovery protocol still achieves above 90% discovery ratio.

5.5.2 Synchronous protocol with duty cycle adaptation

The experiments described in this subsection were conducted in the same environment as the asynchronous discovery experiment. Our duty cycle adaptation scheme is also local, and its performance depends only on the degree of node and not the network size. We use a degree of 10 to represent a relative dense network. We use a larger slot length of $20ms$. Initially, nodes (receivers) operate with a frame length of 50 slots, i.e., the duty cycle is $1/50 = 0.02$. A time synchronization protocol also runs on those nodes. We conduct two experiments with the parameters:

- Slot length: 20 ms
- Initial frame length: 50 slots
- Packet length N: 30 bytes

- $A_{max} = 0.85$, $A_{min} = 0.64$
- $\alpha = 5 \cdot (r_a - A_{max})$, $\beta = 1 \cdot (A_{min} - r_a)$

Experiment 1: (varying traffic) In this experiment, we use 10 senders and 1 receiver, deployed randomly and within communication range of each other. The duty cycle for each sender is 1.2%, i.e., for each frame the probability of a transmitter attempting to transmit when the receiver is on is $0.012 \cdot N$. The receiver duty cycle, decided by the receiver frame length N, is updated every 32 frames, which we'll call a round. The initial aggregate transmission rate is $0.012 \cdot 10 = 12\%$. After 100 rounds, the number of transmitters drops to 5 (i.e., about 6%), and after 140 round it drops to 2 nodes, (i.e., about 2.4%). Figure 5.19 shows the throughput and the receiver duty cycle. Initially, receiver duty cycle is far below incoming traffic, so it increases exponentially to reach the steady state. When incoming traffic decreases, receiver duty cycle matches incoming traffic fairly well. Collision rate and activity ratio are shown in Figure 5.20. The algorithm maintains a steady activity ratio and low collision rate.

Experiment 2: (communication interference) The previous experiment used only one receiver. This experiment is designed to show the effect of our protocol with several receivers in the region. Here we only have 5 nodes sending to the instrumented receiver, but the other 5 nodes send to a large number of other receivers. All transmitters potentially interfere with each other.

The 5 nodes that are transmitting to the instrumented node use a duty cycle of 1.2%, as before. However, the 5 nodes that are transmitting to the other nodes use a higher duty cycle of 2%. As Figure 5.21 shows, the instrumented receiver's duty cycle converges to the rate of the incoming traffic. The instrumented receiver

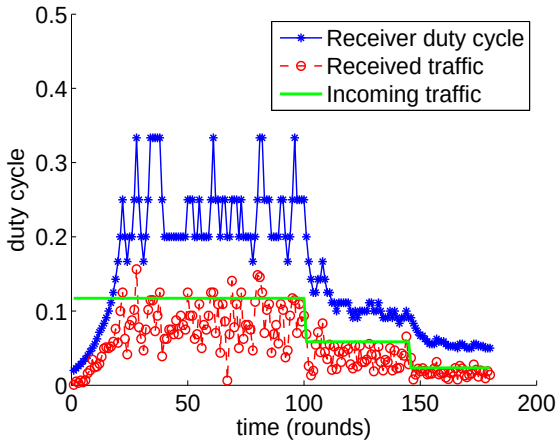

Figure 5.19: Experiment 1: Duty cycle adaptation under varying traffic

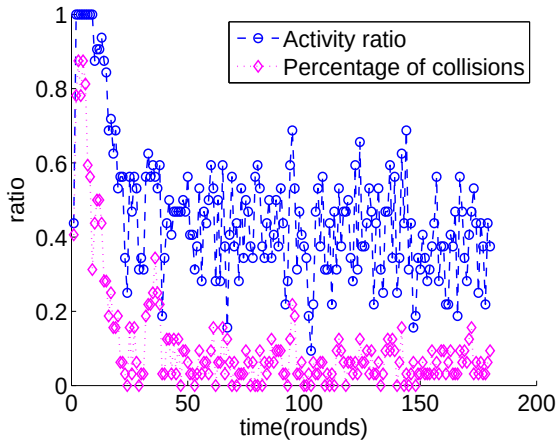

Figure 5.20: Experiment 1: Activity ratio and collisions

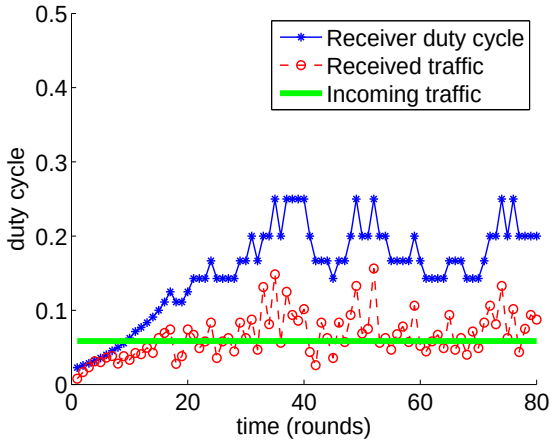

Figure 5.21: Experiment 2: Duty cycle adaptation under interfering traffic

also occasionally overhears messages intended for other receivers and fails to hear its own messages due to collisions with messages destined for the other receivers. However, Figure 5.22 shows fewer collisions occur than in the previous experiments; this is because the other receiver's slot rarely coincides with the reception slot of the instrumented receiver.

Simulation: (SeeSaw traffic) To further verify the algorithm, we have run several simulations [12]. Here we show just two results. In one simulation, there are 10 senders and 1 receiver within communication range. We use the same parameters as previous experiment. Every sender generates traffic randomly. The average rate starts at 0.001 packets per frame and linearly increases to 0.05 packets per frame and then decreases to back to the initial rate before repeating the cycle.

- Sender i: $p : 0.001 \rightarrow 0.05 \rightarrow 0.001 \cdots$

As Figure 5.23 shows, even for a 10 node, highly varying traffic network, our adaptive algorithm works well. As we have stated before, activity ratio is independent

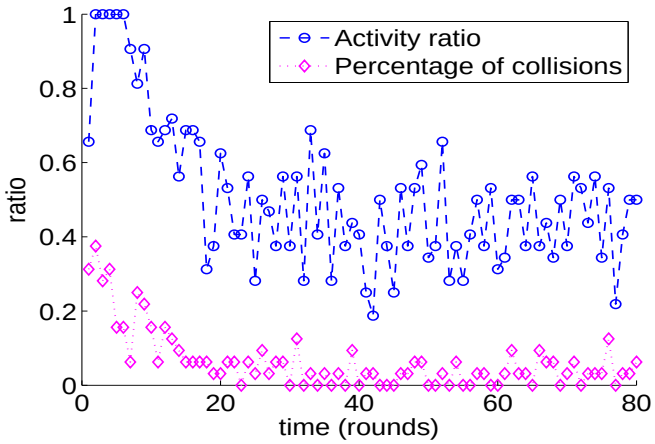

Figure 5.22: Experiment 2: Activity ratio and collisions

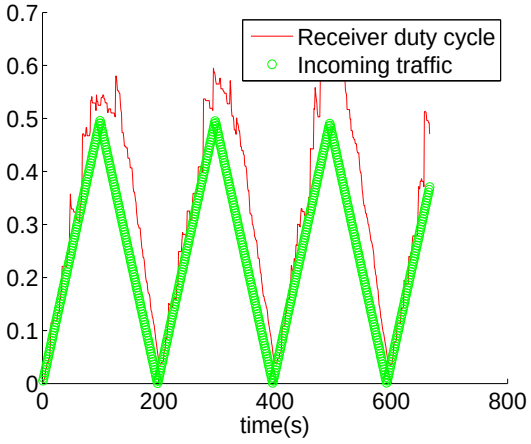

Figure 5.23: Simulation of duty cycle adaptation in SeeSaw traffic

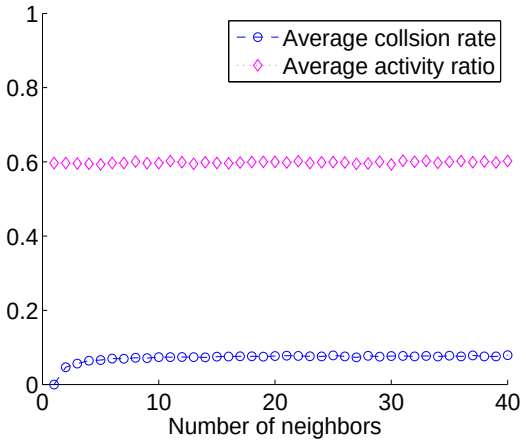

Figure 5.24: Scalability of duty cycle adaptation

of number of senders, this guarantees similar performance even number of senders increases.

To evaluate the scalability of duty cycle adaptation, we change the number of senders, but with the same total traffic. Other settings are the same as before. As Figure 5.24 shows, duty cycle adaptation performs well with different number of neighbors.

5.5.3 Comparison to LPL

We use the LPL (low power listening) implementation in Tinyos 2.0.2 as baseline to compare with OMAC. We performed two experiments.

Experiment 1: Duty cycle under no contention In this experiment, Five senders transmits message to a receiver. To conduct a fair comparison, we set the same idle listening duty cycle for nodes with either OMAC or BMAC (13%). Figure 5.25 shows the duty cycle of receiver with OMAC and BMAC. OMAC maintains low duty cycle at 13%, but BMAC increases quickly when traffic increases.

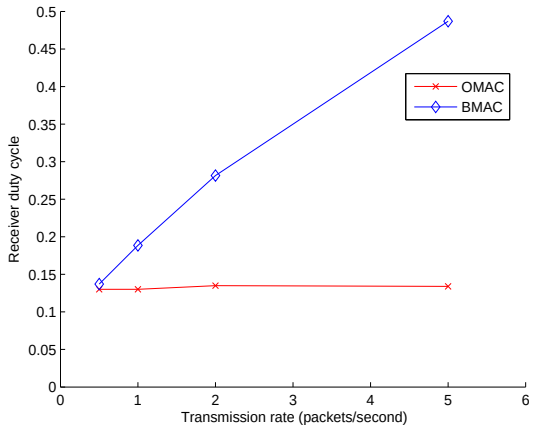

Figure 5.25: Duty cycles of receiver with increasing traffic

Experiment 2: Duty cycle under contention In this experiment, several senders transmit messages to a receiver. All the nodes are within communication of each other. When we increase the number of of senders, the receiver duty cycle increase for BMAC, but OMAC maintain low duty cycle as shown in Figure 5.26 and Figure 5.27. In addition, OMAC achieves better reception success rate (see Figure 5.28)because of the staggered on wakeup schedule.

5.6 Conclusion

System level analysis often suggests that WSNs should strive for duty cycles on the order of 1% while actually communicating at a duty cycle above 0.1%. In practice there are very few examples of deployments that achieve this level of performance. The key reason is the difficulty of getting the receiver off most of the time. Synchronous MACs can achieve this level of performance, but have not been widely adopted largely

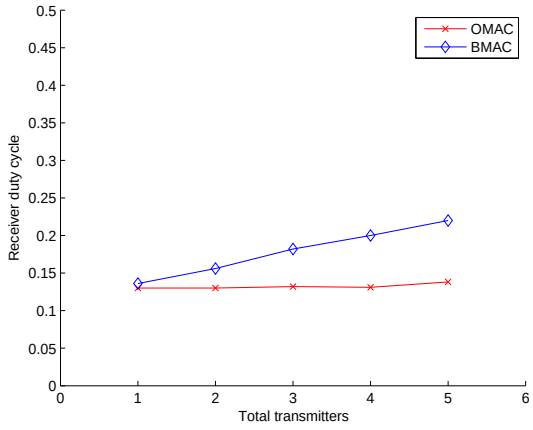

Figure 5.26: Duty cycles of receiver with contending senders, 1 packet per 3 seconds

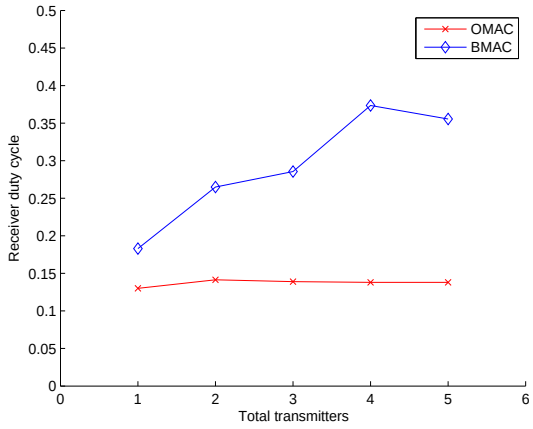

Figure 5.27: Duty cycles of receiver with contending senders, 1 packet per second

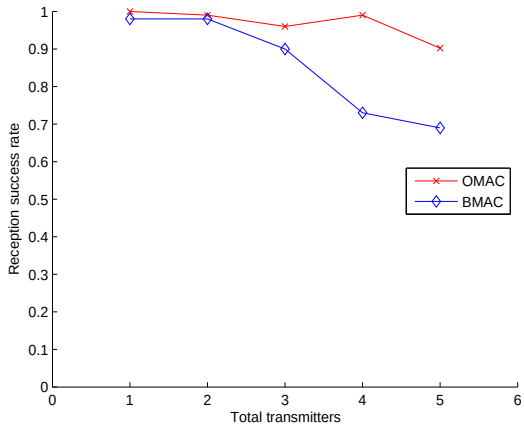

Figure 5.28: Reception success rate, 1 packet per second

because of the system stabiliztion implications for discovery and bootstrapping. These problems have seemed insurmountable for mobile networks.

This paper shows an approach for getting a high efficiency asynchronous discovery protocol to co-exist with an even higher efficiency synchronous communication protocol. The result should allow 1% to 5% duty cycles to be achieved, even for mobile networks. This in turn should allow for a whole new class of applications based on long life, battery powered, mobile nodes. The result could be as substantial as increasing the battery life from a few days to a year or from a couple of weeks to several years.

CHAPTER 6

FEATURE CALIBRATION IN SENSOR NETWORKS

Despite recent theory development, methods of calibration that accurately recover signals from biased sensor readings remain limited in their applicability. Acoustic sensors, for instance, which have been popular in low power wireless sensor networks, are difficult to calibrate in this manner, given their significant hardware variability, large dynamic range, sensitivity to battery power level, and complex spatial/temporal environmental variations.

In this chapter, we submit that the applicability of calibration is broadened by lifting the calibration problem from the level of sensors to that of sensing applications. We show feasibility of easy, accurate calibration at the level of application-specific features, via an example of recovering the feature of acoustic signal-to-noise ratio (SNR) that is useful in event-detection applications. By easy, we mean there is an efficient, purely local, and stimulus-free procedure for recovering SNR (that compares measured variances for multiple randomly chosen sensitivities, effected via acoustic sensor hardware support); unlike extant calibration methods, the procedure does not need to rely on any synchronization among nodes, long-term correlation between their respective environments, or assumptions about training events. And by accurate, we mean the procedure yields low error in SNR estimation. We provide experimental

validation of the difficulty of directly calibrating acoustic signals and the accuracy of our SNR calibration procedure.

6.1 Introduction

Calibration remains a fundamental yet rather unsolved problem for sensor networks. The trend towards using low-cost components has exacerbated sensor variability. Many deployment environments are uncontrolled, unpredictable, and even harsh, they thus impact sensing in a complex, dynamic way. Manual or factory calibration is typically impractical.

Existing methods for automatic in-the-field calibration have focused on accurately recovering signals from biased sensor readings. They been validated in applications that monitor the temperature, humidity, light, and radio in environments where there are known stimuli or correlations among the readings of multiple sensors. In our experience, however, there are common cases where these assumptions do not hold, and calibrating measurement sensors is difficult or inaccurate. One such case is acoustic sensor networks, which are attractive given their relatively low cost, low power, and long detection range for various target types [6, 24, 44]. Being wide band, acoustic sensing is sensitive to complex environmental factors such as wind or building noise, and this makes correlations between sensor nodes dynamic and hard to compute. In addition, the large dynamic range of acoustic response and the limited bandwidth of A-D channels introduce significant quantization errors. Similar arguments hold for low power radar, seismic, magnetic and other sensors, making traditional signal calibration methods unsuitable.

We propose eschewing the difficulty of recovering signals from direct measurements, by focusing instead on recovering features that are core to the application at hand. In an application for intruder detection using acoustic sensors, for instance, we would focus on the calibrating the feature (or features) that detection relies upon. The feature would depend upon whether a Neyman-Pearson criterion detector, Bayes risk criterion detector, sequential hypothesis test detector, or some other type of detector that was chosen. In turn, calibration so as to recover the feature, as oppose to to signal itself, will more directly improve the performance of the detector, i.e., detector accuracy in terms of false alarms or intruder misses.

For ease of illustrating feature calibration, we consider in this chapter the relatively simple feature of signal-to-noise-ratio (SNR). SNR is broadly useful in the class of binary hypothesis detectors. We give an efficient, purely local procedure for nodes to recover their SNR and experimentally validate the accuracy of this procedure in both indoor and outdoor settings. Our illustration also reveals other differences between SNR calibration and traditional signal calibration: The former focuses on relative value calibration, i.e., estimating the relative difference between the presence of intruders and their absence, whereas the latter considers absolute value calibration; the former thus does not require calibration of offset, since this offset is canceled in considering the difference between intruders and backgrounds. Also, the former explicitly considers noise, e.g., in the environment and in the hardware, whereas the latter typically avoids this consideration.

6.2 The Problem of SNR calibration

The goal of detection applications is to determine whether an event has occurred in the field of sensing. Detection is based on a binary hypothesis test: is the signal an event or just the environmental background. More specifically, a binary hypothesis test is:

$$H_0 : \quad x \sim f_0(x)$$

$$H_1 : \quad x \sim f_1(x)$$

where functions f_0, f_1 are probability density functions of signal x in the hypothesis H_0 and H_1 respectively. H_0 means that there is no event, and H_1 means that there is an event.

A rich class of detectors, including Neyman-Pearson criterion detector, Bayes risk criterion detector, and sequential hypothesis testing, are designed based on the likelihood ratio test:

$$\Lambda(x) = \frac{f_0(x)}{f_1(x)} \tag{6.1}$$

$$H_0 : \quad \Lambda(x) < \eta$$

$$H_1 : \quad \Lambda(x) \geq \eta$$

6.2.1 Feature model

The SNR feature for the k^{th} sample inside time cell (or time window) t is defined as:

$$SNR(t) = \frac{|x(t,k) - \mu(t)|}{\sigma(t)} \tag{6.2}$$

where $\mu(t)$ and $\sigma(t)$ are respectively the mean and the standard variance of the t^{th} cell reading.

We choose this feature for detectors that are based on likelihood ratio test, to calculate $\Lambda(x)$. In fact, one type of Neyman-Pearson detector, the CA-CFAR (Cell Average - Constant False Alarm Rate) detector, relies entirely on SNR, by designing its likelihood ratio test to be:

$$H_0 : \quad SNR(t) < \eta$$

$$H_1 : \quad SNR(t) \geq \eta$$

The problem of calibrating detectors to deal with uncertainty in the signal is thus lifted to calibrating the SNR feature. Note also that the larger the SNR, the better the performance of its detector.

6.2.2 Sensor measurement model

Each sensor i measures $y_i(t,k)$ instead of the signal $x_i(t,k)$, and this measurement depends on unknown gain α_i and unknown offset β_i parameters. For simplicity, we assume that the relationship between $y_i(t,k)$ and $x_i(t,k)$ follows the linear model:

$$x_i(t,k) = \alpha_i \cdot y_i(t,k) + \beta_i + \epsilon_i(t,k) \tag{6.3}$$

The model includes a random error ($\epsilon_i(t,k)$) that is introduced by sensor nonlinearity, quantization, and other sensor hardware variations, such as thermal drift of electronic components, electrical noise inside circuits, changing resistance of sensor wires, etc. We henceforth refer to this random error as hardware noise.

The parameters α_i and β_i can change over time, but we assume that during the short periods of time in which calibration is performed, these parameters remain constant.

We denote the variance of the measurement y_i as $\overline{\sigma_i}^2$, and the signal variance of x_t as σ_i^2. Since the hardware noise, $\epsilon_i(t,k)$, is independent of the measurement, the relationship between signal variance, measurement variance, and hardware noise variation is:

$$\sigma_i{}^2(t) = \alpha_i^2 \overline{\sigma_i}^2(t) + Var(\epsilon_i) \tag{6.4}$$

This equation indicates that the relationship between the measurement variance and signal variance is linear, since our measurement model is linear. The gain of signal variance is the square of the gain (i.e., α_i^2). We assume that hardware noise, $\epsilon_i(t,k)$, is stationary, and hence $Var(\epsilon_i)$ is constant.

6.2.3 SNR calibration problem

Given our feature and sensor measurement models, the measured $\overline{SNR}$ is calculated by:

$$\overline{SNR} = \frac{|y(t,k) - \overline{\mu}(t)|}{\overline{\sigma}(t)} \tag{6.5}$$

The feature calibration in this case is to recover the correct value of SNR at each sensor in the network.

6.3 Related work

There are several notable calibration methods for sensor networks. Hightower [29] and Whitehouse [51] show how to calibrate radio transceivers to accurately obtain radio signal strength information (RSSI) by controlling some radios to provide stimuli. In LaSLAT, Taylor et al [46] use a moving target as a stimulus to calibrate sensors encountered by the target.

Another approach assumes correlation between readings of sensors [11, 7]. Bychkovskiy et al [11] present a two-phase calibration process for dense sensor deployments. The first phase is pair-wise calibration, wherein temporal correlation between a pair of neighboring sensors is evaluated. The second step achieves consistency at a group level. Their underlying assumption is that neighboring sensors observe the same phenomena. Balzano et al [7] relax the density assumption, by showing that if sensors oversample their environment, the unknown sensor gain can be perfectly recovered provided the measurements are random and some incoherence conditions hold.

In contrast, acoustic SNR calibration cannot typically assume availability of stimuli or existence of correlation between sensors.

6.4 Feature calibration

To recover SNR, as shown in Equation (6.2) and (6.3), we need to recover the correct value of gain, α, and the hardware noise variation, $Var(\epsilon)$. However, if we know that $Var(\epsilon)$ is sufficiently small compared to the environment noise variance, $\sigma^2(t)$, feature calibration can be avoided. This is stated in the following theorem :

Theorem 30 *If $Var(\epsilon) \ll \sigma^2(t)$, SNR is approximated by $\overline{SNR}$ without knowing the exact value of gain α and $Var(\epsilon)$.*

Proof: $\overline{SNR}$ is calculated by:

$$\overline{SNR} = \frac{|y(t,k) - \overline{\mu}(t)|}{\overline{\sigma}(t)} = \frac{|((x(t,k) - \beta) - (\mu(t) - \beta))/\alpha|}{\overline{\sigma}(t)}$$
$$= \frac{|x(t,k) - \mu(t)|}{\sqrt{\sigma^2(t) - Var(\epsilon)}} \approx \frac{|x(t,k) - \mu(t)|}{\sigma(t)} = SNR$$

■

A typical illustration for acoustic sensors is that in loud environments, the effect of hardware variation can be ignored because it is dominated by the environment noise.

That said, in acoustically quiet or other environments, hardware variations cannot be ignored, and we must perform feature calibration to avoid negatively impacting detection performance.

6.4.1 Hardware assisted SNR calibration procedure

To calibrate SNR, we assume that the acoustic sensor node has sensitivity control, which controls the amplification of the sensor signal, but without affecting the hardware noise. In every time cell t, each node computes the measurement variance $\overline{\sigma}^2(t)$ during cell t. Our calibration procedure is the following:

Procedure

1. At time cell t, compute the measurement variance $\overline{\sigma}^2(t)$ at sensitivity $\gamma(t)$.

2. At time cell t+1, change sensitivity to some different value $\gamma(t+1)$, and compute $\overline{\sigma}^2(t+1)$.

3. For a measurement y in time cell t, recover SNR via Equation (6.6). Let $\gamma = \frac{\gamma(t+1)}{\gamma(t)}$),

$$SNR = \frac{|y(t,k) - \overline{\mu}(t)|}{\sqrt{\overline{\sigma}^2(t) + \frac{\overline{\sigma}^2(t+1) - \gamma^2\overline{\sigma}^2(t)}{\gamma^2 - 1}}} \tag{6.6}$$

End of procedure

Our procedure is optimal when the gain and the hardware noise variance are maximum likelihood estimates,

$$\alpha = \sqrt{\frac{(\gamma^2 - 1)\sigma^2(t)}{\overline{\sigma}^2(t+1) - \overline{\sigma}^2(t)}} \tag{6.7}$$

$$Var(\epsilon) = \frac{\overline{\sigma}^2(t+1) - \gamma^2\overline{\sigma}^2(t)}{\overline{\sigma}^2(t+1) - \overline{\sigma}^2(t)}\sigma^2(t) \tag{6.8}$$

as shown in the following theorem.

Theorem 31 *The estimates for signal and hardware noise variance given in Equation (6.7) and (6.8) are the maximum likelihood estimates under Gaussian assumption for signal and noise.*

Proof: Consider normal distributed signal embedded in normal distributed noise model for $y(t)$. For notational convenience, we assume a zero mean signal and define $z(t) = x(t)/\alpha$ without loss of generality.

$$y(t,k) = \gamma(t)z(t,k) + \frac{e(t,k)}{\alpha}$$

where z and e/α are zero mean normal distributed variables with unknown variance σ_z^2 and σ_e^2 (note that $\sigma_z^2 = \sigma^2(t)/\alpha^2$ and $\sigma_e^2 = Var(\epsilon)/\alpha^2$). Let $\gamma_1 = \gamma(t), \gamma_2 = \gamma(t+1)$, the log-likelihood of two sets of samples at time t and $t+1$ are given up to a constant by:

$$\begin{aligned} L(y;\sigma_z,\sigma_e) = &-\frac{n}{2}\log(\gamma_1^2\sigma_z^2 + \sigma_e^2) - \frac{\sum_k y(t,k)^2}{2(\gamma_1^2\sigma_z^2 + \sigma_e^2)} \\ &- \frac{n}{2}\log(\gamma_2^2\sigma_z^2 + \sigma_e^2) - \frac{\sum_k y(t+1,k)^2}{2(\gamma_2^2\sigma_z^2 + \sigma_e^2)} \end{aligned}$$

Then taking derivatives with respect to σ_z and σ_e to find the stationary point of the likelihood function yields the pair of equations:

$$\begin{aligned}
-\frac{n\gamma_1^2\sigma_z}{\gamma_1^2\sigma_z^2+\sigma_e^2}+\frac{\gamma_1^2\sigma_z\sum_k y(t,k)^2}{(\gamma_1^2\sigma_z^2+\sigma_e^2)^2}-\frac{n\gamma_2^2\sigma_z}{\gamma_2^2\sigma_z^2+\sigma_e^2} \\
+\frac{\gamma_2^2\sigma_z\sum_k y(t+1,k)^2}{(\gamma_2^2\sigma_z^2+\sigma_e^2)^2}=0 \\
-\frac{n\sigma_e}{\gamma_1^2\sigma_z^2+\sigma_e^2}+\frac{\sigma_e\sum_k y(t,k)^2}{(\gamma_1^2\sigma_z^2+\sigma_e^2)^2}-\frac{n\sigma_e}{\gamma_2^2\sigma_z^2+\sigma_e^2} \\
+\frac{\sigma_e\sum_k y(t+1,k)^2}{(\gamma_2^2\sigma_z^2+\sigma_e^2)^2}=0
\end{aligned}$$

Solving for σ_z and σ_e we get Equation (6.7) and (6.8).

■

Remark When multiple calibration readings are taken at times $t{+}1,\ldots,t{+}m$ with different sensitivities $\gamma_{t+1},\ldots,\gamma_{t+m}$, maximum likelihood interpretation gives us , via a generalization of Theorem 31, a way to estimate signal and hardware noise parameters using the calibration measurements jointly. Specifically, the maximum likelihood estimates for σ_z and σ_e are obtained by solving the following pair of equations:

$$\begin{aligned}
\sum_m -\frac{\gamma_m^2\sigma_z}{\gamma_1^2\sigma_z^2+\sigma_e^2}+\frac{\gamma_m^2\sigma_z\overline{\sigma}^2(t+m)}{(\gamma_m^2\sigma_z^2+\sigma_e^2)^2}=0 \\
\sum_m -\frac{\sigma_e}{\gamma_m^2\sigma_z^2+\sigma_e^2}+\frac{\sigma_e\overline{\sigma}^2(t+m)}{(\gamma_m^2\sigma_z^2+\sigma_e^2)^2}=0
\end{aligned}$$

It follows that SNR is recovered by:

$$SNR=\frac{|x(t,k)-\mu(t)|}{\sigma(t)}=\frac{|y(t,k)-\overline{\mu}(t)|}{\sqrt{\overline{\sigma}^2(t)+\sigma_e^2}} \tag{6.9}$$

Equation 6.9 indicates that our SNR calibration procedure produces smaller SNR value (i.e. $SNR<\overline{SNR}$) by considering hardware noise on each node. Therefore, false alarms are reduced as a result of our SNR calibration.

6.4.2 Comparison with other approaches

For the acoustic phenomena being monitored, our hardware assisted procedure does not assume dense deployment, as in [11], or correlations between nodes in either smoothness space or frequency space, as in [7]. In addition, both [11] and [7] requires synchronized sampling of the signal, which introduces a need for time synchronization. In contrast, our procedure is purely local, and therefore not subject to those constraints.

6.5 Evaluation

Our experiments are based on the wireless Extreme Scale Motes (XSMs, which are derivatives of the Mica2 mote family) which include a microphone and on-board circuitry that consists of an amplifier stage, a high pass filter, a low pass filter, and a sensitivity control unit.

6.5.1 Impact of hardware

Since XSMs are battery powered, we begin by evaluating the influence of battery level. In addition, hardware variations between nodes are also important.

Battery Level

We conducted an experiment to explore whether battery level affects acoustic sensitivity. In this experiment, an XSM powered by two AA batteries logged measured sound level and its variance on a laptop computer. The environment in this case is an office room, with modest acoustic activity during the days and relative quiet during the nights. Our data collection lasted over 3 days until this node ran out of power.

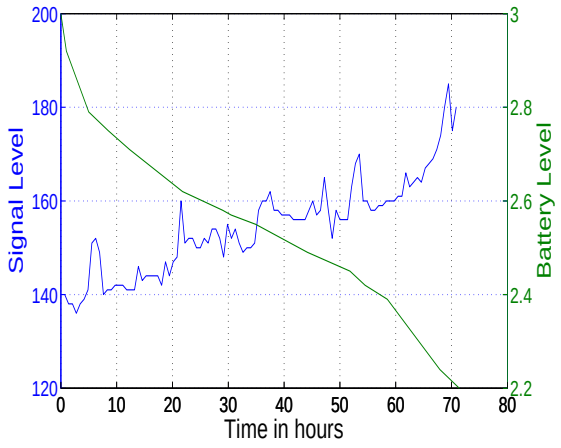

Figure 6.1: Sound & battery level vs. time

In Figure 6.1, the moving average of sound level is displayed along with the battery level over time. Surprisingly, sound level increases when the battery level decreases. This is because battery level determines the reference voltage of the amplifier, which in turn has a negative impact on sensitivity. Therefore, decreasing battery level increases the gain α. In order to discover the time varying gain, a calibration procedure should be run periodically.

Remark. In 2005, as part of DARPA NEST-FE experiments in Richmond Field Station, we performed acoustic based target detection using the Trio sensor nodes, which were based on the XSM mote sensor board and also had a solar harvester. When the power level was low, we observed that false alarms occurred frequently, most likely as a result of the lack of calibration with respect to the power level. *End of remark.*

Node differences

In this experiment, we co-located two XSMs side-by-side on the ground, and observed the effect of a car passing by the two. As Figure 6.2 shows, there is a significant difference in their estimation in this case, where hardware should be the primary difference.

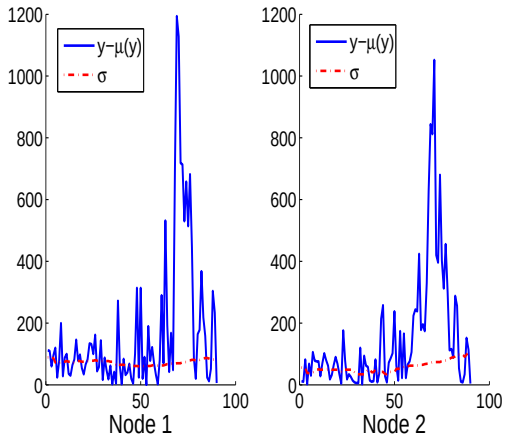

Figure 6.2: Detection of a car at two co-located outdoor sensors

To better evaluate hardware variation, we conducted an experiment with 24 closed located XSMs chosen from a larger indoor XSM array in our Kansei wireless sensor network testbed (www.cse.ohio-state.edu/kansei). The experiment was conducted at a time when the environment was particularly quiet and, therefore, likely to be similar at the chosen nodes. Figure 6.3 shows statistics of the moving average sound level and the sound level standard variation. We observe that the sound level does not have a normal distribution but fits a χ^2 distribution, which follows since the sound level is

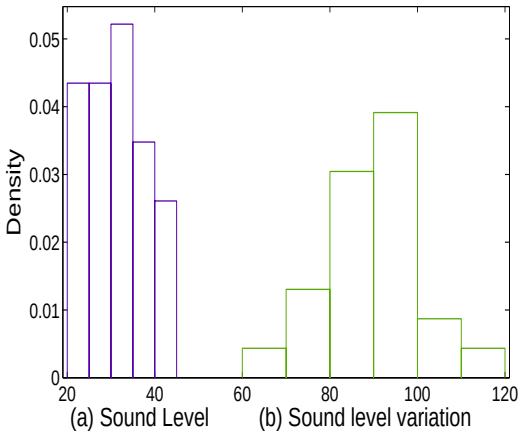

Figure 6.3: Density of sound level & sound level variation indoors

an energy measurement, i.e., related to a power of the signal (in this case, Gaussian noise). In contrast, sound level standard variation does have a normal distribution.

6.5.2 Impact of environment

Environmental noise sources

In both indoor and outdoor acoustic environments, complex influence of environment noise must be considered. In our outdoor ExScal experiments, for instance, wind noise was a serious consideration: Our measurements showed that the noise in strong winds ($> 4m/s$) is $5 - 10dB$ higher than light winds ($< 2m/s$). In addition, there was large and unpredictable spatial/temporal variation in wind noise. The Kansei location where the XSM array is housed is directly exposed to several manufacturing workshops and complex acoustics abound.

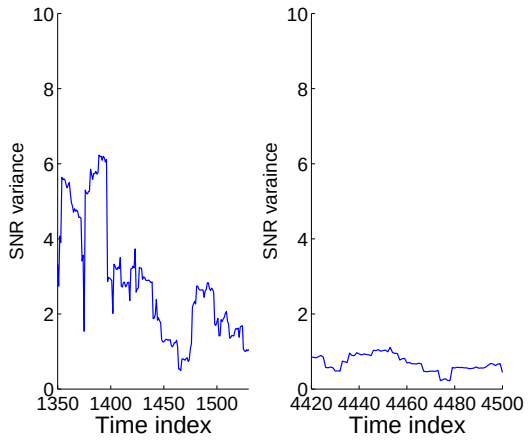

Figure 6.4: SNR variance in (a) low environment noise (b) high environment noise

Influence on SNR

When environment noise varies, SNR variance also changes. Figure 6.4 shows SNR variance in low environment noise and high environment noise. We can see that SNR has large variance in low environment noise, but small variance in high environment noise. This is likely because hardware noise dominates in Figure 6.4(a). As Theorem 30 states, SNR calibration is necessary when environment noise is low.

6.5.3 Acoustic feature calibration

To validate the accuracy of our SNR calibration procedure, we conducted an experiment on 4 XSMs selected within a small 2×2 region in Kansei. In such a small area, environment noise is similar for those XSMs. As Figure 6.5 shows, their readings are highly correlated. A single preprogrammed acoustic event was played between time 10 and 25 at a significant distance from the region; the testbed environment was quiet at other time. Since XSMs have a programmable amplifier,

they support our hardware assisted calibration procedure, enabling us to collect SNR data in both the uncalibrated and calibrated modes. As Figure 6.6 shows, when nodes were uncalibrated, their recorded SNR was significantly dispersed in the presence of the acoustic event. But when nodes were calibrated, all exhibited a much more similar response to the acoustic event.

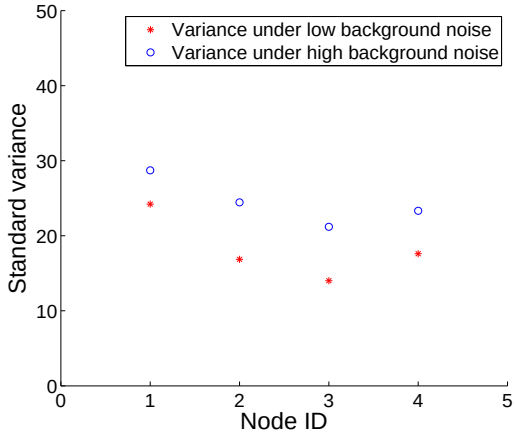

Figure 6.5: Variance correlation between two neighboring nodes

6.6 Discussion, conclusions, and future work

The concept of feature calibration accommodates a broad spectrum of application features, such as influence field in target classification or (bandlimited) frequency response in target detection/classification. On one hand, a feature calibration procedure is applicable only to applications that rely on that feature. On the other hand, the feature calibration approach enables deployment in richer environments, more efficient calibration procedures, and potentially higher accuracy by focusing on the accuracy that is most relevant to the application.

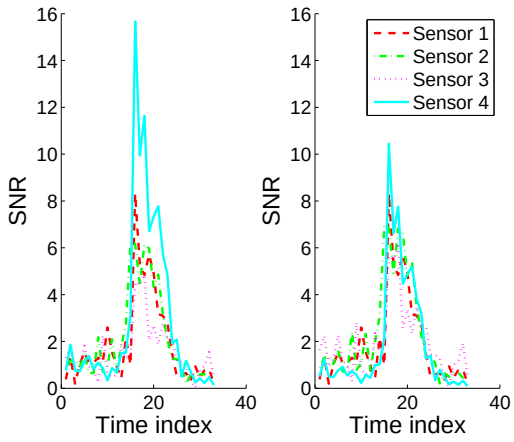

Figure 6.6: SNR of (a) uncalibrated sensor nodes (b) calibrated sensor nodes

In this chapter, we have limited ourselves to SNR, which is a common feature in many binary hypothesis detectors. It is not our intent however to suggest that acoustic SNR-based detectors are optimal. Similarly, we have intentionally limited ourselves to a simple linear sensor measurement model. In practice, a nonlinear model may suit better; in this regard, we note that our SNR calibration procedure is readily generalized for nonlinear models where inversion is feasible.

We note that one source of inaccuracy in our experimental validation of our calibration procedure is the error in the measurement of sensitivity parameter, γ. In other work, we have considered how the sensitivity itself is calibrated on line, and incorporating that technique would further improves our SNR calibration.

In sum, we have taken only a first step in investigating the feature calibration approach, and much remains to be further explored in its theoretical and practical aspects.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this dissertation, we have studied stabilization issues in the context of sensor networks. We have provided three methods that achieve stabilization for dynamic systems with varying equilibrium, exemplified by two applications in sensor networks. One is pursuit-evasion game, the other is duty cycle adaption. To address the challenges of designing robust strategies for surveillance systems with communication constrains, we provide stabilizing pursuer strategies for two pursuit-evasion games: target capture game and asset protection game. To address the challenges of designing robust energy efficient power management protocols for almost alway off mobile networks, we provided a stabilizing receiver centric protocol—OMAC.

We have studied differential games in networked environments with constrained communication resources leading to delays, losses and finite rates in information state updates. We showed that the min-max optimal pursuer strategy of the full information game extends to networked games, and the stabilization properties of the pursuer strategy, provided that the sampling period and the delay in obtaining the evader state information updates scale linearly with the pursuer-evader distance.

To achieve maximal energy efficiency, we have identified one design paradigm - receiver centricity, through theoretical comparison with a broad spectrum of current

approaches that are based on sender centricity. In addition, we have designed optimal asynchronous discovery protocol that can be applied with any synchronous protocol, to rediscover nodes that are out of sync because of mobility, link dynamics, or clock variations. The synchronous receiver centric protocol can achieve optimal energy efficiency along with our stabilizing duty cycle adaptation scheme.

We also have introduced the concept of feature calibration which accommodates a broad spectrum of application features. We designed hardware assisting calibration method for acoustic detector by recovering SNR feature.

7.1 Future work

Stabilization despite unreliable sensing

Sensor networks are usually deployed in harsh environments, which have great impact on sensing capability. In addition, even in the same environment, different sensor nodes creates different readings because of hardware differences. To achieve reliable system, one method is to design applications that can tolerate unreliable measurements. In other words, our goal is to design pursuit strategies despite uncertainty measurements.

To study this, we can model the uncertainty of measurements proportional to distance linearly or polynomially, denoted as:

$$R = f(d) = \gamma \cdot d$$

where R is the upper bound of errors and d is distance between pursuer and evader. This is a realistic model because target locations are provided by underlying network snapshot services, in which information accuracy is linearly proportional to distance.

With the distance between pursuer and evader decreases, pursuer receives more accurate measurement of evader location, which may help pursuer catch evader eventually. However, if accuracy increases too slow with distance increasing, pursuer may fail to catch evader. Our goal is to derive conditions for γ under which pursuit strategies can converge to a successful catch.

O-MAC improvement

In term of O-MAC, there are several possible improvements we need to investigate:

- **Traffic estimation:** In the future, I would like to investigate the improvement by using traffic model. If traffic pattern can be estimated effectively, the duty cycle can be switched to optimal operating point directly. For instance, if receivers correctly estimate the number of neighbors and their traffic, receivers can efficiently allocate bandwidth for incoming traffic. However, this approach may introduce instability or vulnerability because of inaccurate traffic estimation.

- **Sender-receiver coordination:** Also, it is interesting to exploit the relationship of senders and receiver to improve efficiency, fairness and convergence. If different senders have different bandwidth requests for a receiver and those requests can be communicated to receiver, involving senders into receiver duty cycle adaptation can be more effective. Senders can coordinate their transmissions to share active slots of receiver, even avoid collisions.

- **Multiple channels:** Another approach to avoid receiver collisions is using multiple channels, called FHSS (frequency hopping spread spectrum). In other words, different nodes listen at different channels. In 802.15.4, there are 16 distinct frequency channels in 2.4 GHz. However, adding this level of flexibility

also introduces complexity for stabilization. In duty cycled network, neighboring discovery becomes two dimensions: time and frequency. New energy efficient wakeup schedules incorporated with frequency hopping are required.

Gain calibration

Gain control is important in WSN because high dynamic range linear sensors, high bit-width Analog to Digital (A2D) converters, and high bit-width computations drive the price and energy cost dramatically. Gain control can facilitate sensing inherently high dynamic range phenomena such as acoustics on practical hardware, which requires to compare measurements made at different gain settings. Unfortunately, for low cost sensors this requires a limited form of calibration.

Let M denote the measured signal value. Let R denote the physical value. Assume, by way of example, that the gain is well modeled by

$$M = (Ga_1 + a_0)P + \epsilon$$

where a_1 and a_0 are unknown constants that vary from device to device, ϵ is the affect of noise, and G is the gain setting which can be controlled by the software. The ability to control G, in this model (or some similar model), is the defining characteristic of gain control. In pratical, a_1 and a_0 may not be truly constant, but they change slowly compared to the rate of changes in G , often on a time scale comparable the life of the device, but at least no faster than the temperature change associated with daily temperature fluctuations. The auto calibration problem is then to estimate the values of a_1 and a_0 , in situ from a sequence of (M_i, G_i) values without precise knowledge of the corresponding P_i values.

BIBLIOGRAPHY

[1] E. Anceaume, X. Défago, M. Gradinariu, and M. Roy. Towards a theory of self-organization. In *Proceedings of the 9th International Conference on Principles of Distributed Systems (OPODIS)*, 2005.

[2] A. Arora, P. Dutta, S. Bapat, and V. Kulathumani et al. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks, Special Issue on Military Communications Systems and Technologies*, 46(5):605–634, July 2004.

[3] A. Arora, E. Ertin, M. Nesterenko R. Ramnath, and W. Leal. Kansei: A high-fidelity sensing testbed. *IEEE Internet Computing, Special issue on Large-Scale Sensor Networks*, pages 18–31, March 2006.

[4] A. Arora and M. Gouda. Closure and convergence: A foundation of fault-tolerant computing. *IEEE Transaction of Software Engineering*, 19(10):1015–1027, 1993.

[5] A. Arora and M. Nesterenko. Unifying stabilization and termination in message-passing systems. *Distributed Computing*, 17(3):279–290, 2005.

[6] A. Arora, R. Ramnath, E. Ertin, S. Bapat, V. Naik, and H. Cao *et al.*(30 authors). Exscal: Elements of an extreme scale wireless sensor network. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 102 – 108, 2005.

[7] L. Balzano and R. Nowak. Blind calibration of sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, April 2007.

[8] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1999.

[9] J. Beauquier, C. Genolini, and S. Kutten. Optimal reactive k-stabilization: the case of mutual exclusion. In *Proceedings of the 18th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 209–218, 1999.

[10] J. Burman, S. Kutten, T. Herman, and B. Patt-Shamir. Asynchronous and fully self-stabilizing time-adaptive majority consensus. In *Proceedings of the 9th International Conference on Principles of Distributed Systems (OPODIS)*, 2005.

[11] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak. A collaborative approach to in-place sensor calibration. In *Lecture Notes in Computer Science*, volume 2634, page 301316, 2003.

[12] H. Cao, A. Arora, K. W. Parker, and T .H. Lai. Continuous asynchronous discovery with efficient synchronous communication for mobile networks. Technical Report OSU-CISRC-4/07-TR34, The Ohio State University, April 2007.

[13] H. Cao, E. Ertin, and A. Arora. Minimax equilibrium of networked differential games. Technical Report OSU-CISRC-4/07-TR35, The Ohio State University, April 2007.

[14] H. Cao, E. Ertin, V. Krishnan, M. Sridharan, and A. Arora. Differential games in large scale sensor actuator networks. In *Proceedings of the 5th Symposium on Information Processing in Sensor Networks (IPSN)*, pages 77–84, 2006.

[15] H. Cao, K. W. Parker, and A. Arora. O-MAC: A receiver centric power management protocol. In *Proceedings of the 14th IEEE International Conference on Network Protocols, ICNP*, 2006.

[16] P. Chen and S. Sastry. Pursuit controller performance guarantees for a lifeline pursuit-evasion game over a wireless sensor network. In *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec. 2006.

[17] A. Dasgupta, S. Ghosh, and S. Tixeuil. Selfish stabilization. In *Proceedings of the 8th International Symposium on Stabilization, Safety, and Security on Distributed Systems (SSS), Lecture Notes in Computer Science*, volume 4280, pages 231–243, 2006.

[18] A. Dhama, J. Oehlerking, and O. Theel. Verification of orbitally self-stabilizing distributed algorithms using lyapunov functions and poincaré maps. In *Proceedings of the 12th International Conference on Parallel and Distributed Systems*, 2006.

[19] L. Doherty, B. A. Warneke, B. E. Boser, and K.S.J. Pister. Energy and performance considerations for smart dust. *International Journal of Parallel Distributed Systems and Networks*, 4:121–133, 2001.

[20] S. Dolev and N. Tzachar. Empire of colonies: Self-stabilizing and self-organizing distributed algorithms. In *Proceedings of the 10th International Conference on Principles of Distributed Systems (OPODIS)*, 2006.

[21] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proceedings of the 4th Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[22] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, December 2002.

[23] L. Feeney and M. Nilsso. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the 20st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, page 15481557, 2001.

[24] Brian P. Flanagan and Kenneth W. Parker. Crobust distributed detection using low power acoustic. Technical report, MITRE Corporation, Mclean, VA, March 2005.

[25] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava. Estimating clock uncertainty for efficient duty-cycling in sensor networks. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2005.

[26] S. Ganeriwal, R. Kumar, and M.B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, California, USA, October 2003.

[27] M. G. Gouda and T. Herman. Adaptive programming. *IEEE Transaction of Software Engineering*, 17(9):911–921, 1991.

[28] G. Halkes and K. Langendoen. Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks. In *European conference on Wireless Sensor Networks (EWSN)*, 2007.

[29] J. Hightower, C. Vakili, G. Borriello, and R. Want. Design and calibration of the spoton ad-hoc location sensing system. In *Unpublished*, August 2001.

[30] R. Isaacs. *Differential Games*. Kruger Publishing Company, Huntington, NY, 1975.

[31] V. Kulathumani, A. Arora, M. Demirbas, and M. Sridharan. Trail: A distance sensitive network protocol for distributed object tracking. In *Proceedings of European Conference on Wireless Sensor Networks (EWSN)*, 2007.

[32] S. Kulkarni and M. Arumugam. Tdma service for sensor networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS), ADSN workshop*, page 604609, Tokyo, Japan, March 2004.

[33] S. Kutten and B. Patt-Shamir. Adaptive stabilization of reactive protocols. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science, 24th International Conference (FSTTCS)*, pages 396–407, 2004.

[34] K. Langendoen and G. Halkes. Energy-efficient medium access control. *Embedded Systems Handbook, CRC press*, August 2005.

[35] Y. Li, W. Ye, and J. Heidemann. Energy and latency control in low duty cycle mac protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, Mar 2005.

[36] Z. Manna and A. Pnueli. Models for reactivity. *Informatica*, pages 609–678, 1993.

[37] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.

[38] J. Nash. Noncooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[39] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Proceedings of the IEEE International Conference on Decision and Control*, Paradise Island, Bahamas, Dec 2004.

[40] O.Theel. Exploitation of lyapunov theory for verifying self-stabilizing algorithms. In *Proceedings of the 14th Symposium on Distributed Computing (DISC'00)*, volume 1914, Toledo, Spain, 2000.

[41] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2st international conference on Embedded networked sensor systems (SenSys)*, 2004.

[42] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th Symposium on Information Processing in Sensor Networks (IPSN)*, pages 364–369, 2005.

[43] L. Schenato, S. Oh, and S. Sastry. Swarm coordination for pursuit evasion games using sensor networks. In *Proceedings of the International Conference on Robotics and Automation*, Barcelona, Spain, Apr 2005.

[44] X. Sheng and Y.-H. Hu. Energy based acoustic source localization. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, pages 285–300, Palo Alto, CA, April 2003.

[45] S. Singh and C. S. Raghavendra. Pamas: power aware multi-access protocol with signaling for ad-hoc networks. *ACM Computer Communication Review*, 28(3):526, 1998.

[46] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, page 2733, 2006.

[47] Y. C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for ieee 802.11-based multihop ad hoc networks. In *Proceedings of the 21th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2002.

[48] A. Tzamaloukas and J. J. Garcia-Luna-Aceves. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *Proceedings of the 20st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001.

[49] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, 2003.

[50] S. Verdu and H. Poor. On minimax robustness: A general approach and applications. *IEEE Transactions on Information Theory*, IT-30:328–340, 1984.

[51] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, page 5967, 2002.

[52] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[53] W. Ye, F. Silva, and John Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[54] G. Yee, E. Anderson, and R. Han. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[55] R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, pages 35–45, 2003.

[56] R. Zheng and R. Kravets. On-demand power management for ad hoc networks. In *Proceedings of the 22st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.