Stony Brook University

Problems and Solutions for Backscatter-Based Tag-to-Tag Communication in the Internet of Things

A Dissertation presented

by

Zhe Shen

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

August 2015

Stony Brook University

The Graduate School

Zhe Shen

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Petar M. Djurić - Dissertation Advisor
Professor, Department of Electrical and Computer Engineering

Mónica F. Bugallo - Chairperson of Defense
Associate Professor, Department of Electrical and Computer Engineering

Sangjin Hong
Professor, Department of Electrical and Computer Engineering

Samir R.Das
Professor, Department of Computer Science

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Distributed Estimation in the Presence of Correlation

by

Zhe Shen

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2015

The Internet of Things (IoT) has been developing for decades and has been considered as the future of the Internet. The basic idea of the IoT is to enable a wide range of objects (things) around us to interact and cooperate with each other so that certain goals are reached. An important technology that will be part of the IoT is Radio Frequency IDentification (RFID). RFID is based on the concept of backscatter-based communication and the use of inexpensive RFID tags. A very exciting direction of work in the last few years has been the research on backscatter-based tag-to-tag (BBTT) communication systems, that is, systems that do not require the use of expensive RFID readers. In BBTT communication systems, two or more radio-less tags communicate with each other purely by backscattering an external signal. We study several problems of BBTT communication systems.

First, we investigate a unique phase cancellation problem that occurs in BBTT systems. The relative phase difference between the backscatter signal and the external excitation signal at the receiving tag often causes a complete cancellation of the baseband information contained in the envelope, which results in a loss of communication between the two tags. We theoretically analyze and experimentally demonstrate this problem. We then present a solution to the problem based on the design of a new backscatter modulator for tags that enables multi-phase backscattering.

Second, we address protocols for communication in BBTT systems. We note that the tags of BBTT systems have limitations including memory space and communication ranges. Considering these limitations and simple

applications, we choose and modify two existing anti-collision protocols that can be used in BBTT systems. We examine the performance of the proposed protocols through theoretical analysis of linear and complete networks and by computer simulations of general networks.

Third, we consider the problem of distributed Bayesian learning in BBTT systems. The BBTT system is composed of tags that can only communicate with their neighbors. These tags are tasked to learn by cooperation with the neighboring tags. More specifically, the objective of the tags is to obtain the global posterior distribution of an unknown parameter of a fictitious fusion center in a distributed way through the use of the Bayesian paradigm. The tags iteratively exchange information with their neighbors, and they update the summary of their information using the signals received from the neighbors. All the tags are assumed to know the topology of the network and keep all the new information in their memories. We propose a method based on a recent work and prove that the distribution of each tag can converge correctly using the proposed method. Furthermore, with the proposed method, convergence is achieved much faster than with the non-Bayesian and consensus-based algorithms. The proposed approach is general and applicable to other types of distributed systems.

Contents

List of Figures

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Petar M. Djurić for the long time support and guidance of my Ph.D research. He inspire me by his hardworking, knowledge and passionate attitude. Also he is a very nice person. I could not have imagined having a better advisor and mentor for my Ph.D study.

Secondly, I would like to show my gratitude to Prof. Mónica F. Bugallo for her assistance during my research and the opportunities she offered me to collaborate with other researchers, and to teach high school students.

Thirdly, I would like to thank Akshay Athalye. He first found the phase cancellation problem and gave a lot of help to solve this problem. I also would like to thank Professor Samir R.Das. He gave me many good suggestions for the anti-collision protocols selection. I am very thankful to Prof. Sangjin Hong and Prof. Samir R. Das for serving as committee members.

Fourthly, I thank my friends in the COSINE Laboratory, Inigo Urteaga, Cagla Tasdemir, Shishir Dash, Li Geng, Zhiyuan Weng, Yunlong Wang, Zheming Zhang, Zhongwen Ying, Kai Wang, Kezi Yu, Hechuan Wang and Lingqing Gan who have been supportive in every way.

Finally, special gratitude goes to my parents, Guowei Shen, Heli Zhang, and my wife, Sha Tong, for their warm love, continued patience, and endless support during the past years.

Chapter 1

Introduction

1.1 Overview

The Internet of Things (IoT) has been developing for decades with increasing pace. The basic idea of IoT is to endow physical objects (things) around us with the ability to interact and cooperate with each other in order to allow for reaching specific goals [1]. For example, obvious goals are to identify, track, and manage physical things in the real world through the network or the Internet. The application domains of the IoT include transportation and logistics, health care, and smart environments [2].

Radio Frequency Identification (RFID) devices will be indispensable components of the IoT. Modern RFID systems are composed of expensive readers and inexpensive tags. The readers communicate with the tags by using the principle of backscattering. In the IoT, ideally, one would want to avoid the use of RFID readers and to solely rely on the use of inexpensive tags. This requires that the tags directly communicate with each other. Very recently, progress has been made in the development

of backscatter-based tag-to-tag communication (BBTT) systems. In these systems, the tags can interconnect smart objects which is exactly one of the idea of the IoT [3]. In BBTT systems, an exciter sends continuous wave (CW) signals to provide power and let the tags communicate using backscattering. The exciters are far less expensive than the readers, which would permit scalability of the system. In brief, one exciter and several tags can implement a network with a low cost.

In a BBTT system, all the tags can backscatter the CW signal and receive backscattered signals from other tags. The signal that a tag receives is a superposition of the CW signal and the backscatter from the transmitting tag on the same frequency. The phase difference between them has a big impact on the amplitude of the received signal. They may cancel each other or decrease the amplitude in certain topological configurations. The tag is simple and low-cost, and therefore it cannot implement active IQ demodulation to extract the backscatter signal like an RFID reader. As a result, it suffers from the phase cancellation problem. In this dissertation, we address this problem in detail and provide solutions to it.

When a tag receives more than one backscatter signal from its neighbors, collision occurs and it cannot decode them successfully. In the commercial RFID system protocol ISO 18000-6C(Gen. 2), a reader detects tags and assigns talking time period to each tag [4]. The reader sends commands to change the states of tags or sends continuous wave (CW) signals to let the tags talk. A tag listens to the commands or modulates the CW signal using backscattering. However, in BBTT systems, there is no centralized reader to command the tags. Because of energy and memory limitations of the system, one approach is to avoid use of protocols that

require acknowledgment of signals and establishment of topologies. The existing anti-collision protocol cannot directly be used in a BBTT system. In this work, we consider protocols that are adapted to BBTT systems, and we study them from theoretical and practical points of view

The third topic of this dissertation is distributed learning. The topic is not only of direct interest for BBTT systems, but also for any other system composed of agents and without a central unit. In distributed learning, agents estimate the underlying state of nature using not only their local measurements, but also the information of other agents. If there is a fusion center with access to information from all the agents, it can obtain global optimal estimates of the desired quantities. However, in most cases of the IoT, each agent can only communicate with its neighbors. In other words, each agent only gets information from a few agents around it. We study a network of Bayesian cooperative agents whose objective is to obtain the posterior distribution of an unknown parameter in a distributed way. We cast the distributed learning as a consensus problem, where the agent can perform as well as a fictitious fusion center. There exist consensus-based methods, such as average consensus and gossip-based averaging algorithms. They are robust and do not require topology information, but they converge relatively slowly. They are suitable for use in dynamic networks with simple devices, such as temperature sensors. However, in some cases, as in BBTT systems, we want to allow for a much faster learning of the unknowns. We propose an approach that improves the convergence rate of learning over networks in comparison to standard approaches.

1.2 Contributions

In this thesis, we analyze and provide solutions to two problems in BBTT systems and develop a learning method over networks with improved learning rate.

Our first contribution is a solution to the phase cancellation problem in BBTT systems. This problem is particular for BBTT systems. We model the amplitudes and phases mathematically for both amplitude shift keying (ASK) and phase shift keying (PSK) modulations, and show how the problem arises in both the time domain and with phasor diagrams. We analyze the problem and point out the parameters that affect the values of cancellation phases. We also give several solutions and analyze their pros and cons. Theoretically, our proposed solution can avoid the problem perfectly except in one situation. The simulation and experiment results show that our analysis is correct and that the proposed solution can reduce phase cancellation significantly.

The second contribution is two anti-collision protocols that are modified for use in BBTT systems. We theoretically analyze the modified CSMA/CA protocol using Markov chains in complete networks and the framed slotted Aloha protocol in complete and linear networks. We investigate their performances in general networks using computer simulations.

The third contribution is a novel approach to distributed learning. We exploit the work reported in [5], where the authors present an efficient Bayesian learning method in a Gaussian setting. Namely, the authors assume that the agents in the network observe Gaussian signals with the same mean and with variance 1. The agents know the network topology and

use it to execute an efficient learning method of the unknown parameter. In the proposed solution, instead, we work with Bernoulli models in two scenarios. In the first one, the agents observe outcomes of a Bernoulli experiment, where the probability of success θ is unknown and where the observations are without errors. In the second scenario, these observations are with errors whose probabilities are known. In both cases, we show how the agents can reach a consensus in finite number of iterations and much faster than consensus-based methods.

1.3 Dissertation Organization

The dissertation is organized as follows. In Chapter 2, we formulate the phase cancellation problem in a BBTT system. We then propose solutions for this problem. The anti-collision protocols selection problem is addressed in Chapter 3 In Chapter 4, we consider the distributed Bayesian learning problem with Bernoulli models. Particularly, we consider the case that the observations are without and with errors. We conclude the dissertation with Chapter 5.

Chapter 2

Phase Cancellation Problem in BBTT System

2.1 Introduction

The science behind communication by means of reflected signals or backscatter has been studied for several decades. In recent years this technique has seen widespread practical application in the form of Radio Frequency Identification (RFID). Furthermore, it is expected that the RFID technology will play a prominent role in enabling the Internet of Things, where physical objects will be connected to other objects and to the cyberspace. Typical backscatter RFID systems consist of an active reader that transmits a query signal and passive tags that communicate with the reader by reflecting part of this signal. This communication technique allows for tags that are very inexpensive and consume very low power.

Traditional passive RFID has long been thought of as the 'last link' in the IoT, enabling visibility and connectivity for all items regardless of

size, cost and volume [4]. This is primarily due to the very low cost and low power consumption of passive RFID tags. However, use of traditional RFID technology requires a centralized model wherein one high-cost reader initiates and controls communication with a population of tags in its vicinity. The tags cannot autonomously communicate with each other nor can they initiate communication with their neighbors. This places a restriction on the capabilities and scalability of an IoT using standard RFID technology.

Backscatter based tag-to-tag communication (BBTT) is a paradigm that can overcome the above limitations and help realize the vision of the IoT. It can allow ubiquitously tagged objects to independently communicate with each other without being controlled by a single reader. In these systems, passive radio-less tags communicate by reflecting an external excitation signal. This passive tag-to-tag communication has recently started to gain research interest. Communication between two passive RFID tags at close range using backscatter communication has been demonstrated in [6] and the electromagnetic models that govern the communication in this system have been explored in [7]. In our previous work, we have developed a radio-less EPC Gen 2 compliant device (the sensatag) that can listen to backscatter signals from neighboring Gen 2 RFID tags and in turn communicate with a Gen 2 RFID reader using standard backscatter modulation. We employed this device to enable precise localization and tracking in passive RFID systems [8].

All these systems employ a standard UHF RFID reader as the source of the excitation signal to enable the backscatter communication. In [9], the authors have designed a system where tags communicate by backscattering

ambient TV signals. Similarly, in [10] a backscatter-based communication system that uses ambient WiFi signals as excitation source has been demonstrated. Although the frequency of the excitation signal and the sources of power for the tag differ in the above systems, the general design of the tags is similar. The tags consist of an antenna, a backscatter modulator, an envelope detection circuit, a comparator for digitizing the detected envelope, and a digital controller that implements the chosen communication protocol. The backscatter modulator consists of a switch that alters the reflection cross section (RCS) of the antenna between two states by varying the termination impedance. This allows for modulation of baseband data onto the reflected backscatter signal. Tags can employ ASK or PSK backscatters. In the former case, the real part, and in the latter, the imaginary part of the RCS are varied, respectively.

In this chapter, we explore phase cancellation issues that occur in such BBTT communication systems. Consider two tags in a BBTT scenario, both receiving an excitation signal, either from an active source or ambient radiation. The transmitting tag sends out baseband data using backscatter modulation. The receiving tag sees a signal that is a superposition of the excitation signal and the modulated backscatter from the transmitting tag. Since the tags are passive radio-less devices, the receiving tag cannot implement active IQ demodulation to extract the backscatter signal. Instead, it has to rely on passive envelope demodulation. In such a setup, the relative phase difference between the excitation signal and the modulated backscatter at the receiving tag has a big impact on the amplitude of the baseband envelope in the combined signal.

Based on the value of the phase difference between the excitation

and backscatter signals, the baseband envelope can be severely attenuated and completely cancelled out leading to a loss of communication between the two tags. Due to this phase problem, the amplitude of the received baseband signal does not decrease monotonically with increasing distance between the two tags. Instead, it varies between peaks and nulls, with the peaks getting successively smaller. This phenomenon significantly impairs the robustness of the tag-to-tag link.

In order to truly unlock the potential of BBTT communication, it is important that the tags can communicate over long distances with low bit error rates (BERs). Handling the challenge of phase cancellation is a critical step towards achieving that goal. We present a mathematical formulation of the phase problem and further illustrate it using phasor diagrams. We verify the models using simulations and lab experiments. In order to solve this problem, we propose a new backscatter modulation technique that uses phase diversity. We couple it with a combination scheme in the tag front end that can further exploit this diversity and increase the link range and robustness.

In summary, backscattering devices are traditionally intended to communicate with an active reader. Hence over the past decade a lot of design and optimization efforts have gone towards building devices to achieve this goal. The BBTT paradigm, on the other hand, opens up a new set of challenges which call for novel designs of backscatter modulators and tag front end circuits. Our solutions from this chapter to some of the challenges is one effort in that direction.

2.2 The phase cancellation problem

Passive tags achieve backscatter modulation by varying the complex impedance of the tag chip between two or more states. This alters the complex power reflection coefficient, s^2, and accordingly alters the amplitude and/or the phase of the reflected signal. The coefficient s^2 determines what fraction of the power incident on the tag is reflected. It can be expressed as [11][12]

$$|s|^2 = \left|\frac{Z_C - Z_A^*}{Z_C + Z_A^*}\right|^2, 0 \leq |s|^2 \leq 1, \tag{2.1}$$

where Z_C and Z_A are the impedances of the tag chip and tag antenna, respectively.

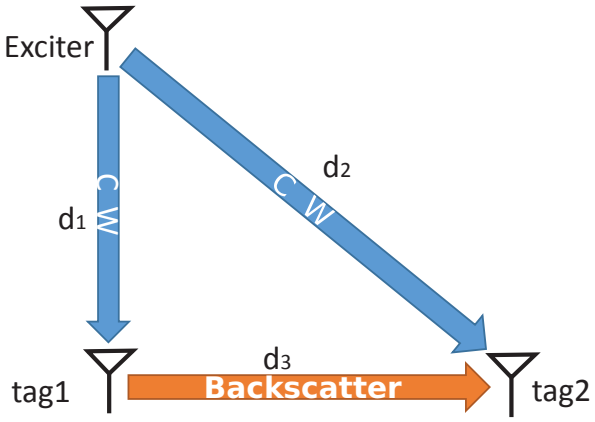

Figure 2.1: A two-tag BBTT system. The respective distances between the devices are d_1, d_2, and d_3.

In Fig. 2.1, we depict a BBTT system with one exciter and two tags. The exciter broadcasts the CW signal, and Tag 1 backscatters this signal and modulates this backscatter by varying its impedance between two states, $'0'$ and $'1'$. Tags can employ either ASK or PSK backscattering. In ASK backscattering, the real part of the chip impedance is varied between

the two states altering the amplitude of the reflected signal. By contrast, in PSK backscattering the imaginary part of the chip impedance is varied altering the phase of the reflected signal.

The signal power received by Tags 1 and 2 from the exciter is given by Friis' equations

$$P_{E\to T_1} = \frac{P_E G_E G_{T_1} \lambda^2}{(4\pi d_1)^2}, \tag{2.2}$$

$$P_{E\to T_2} = \frac{P_E G_E G_{T_2} \lambda^2}{(4\pi d_2)^2}, \tag{2.3}$$

where P_E is the output power of the exciter; G_E, G_{T_1} and G_{T_2} are the antenna gains of the exciter, Tag 1 and Tag 2, respectively; $\lambda = \frac{c}{f}$ is the wavelength of the signal; f is the frequency of the signal; c is the speed of light in free space; and d_1 and d_2 are distances as shown in Fig. 2.1.

When tag T_1 is backscattering, the power reflected by it in the two backscatter modulation states can be written as

$$P^0_{T_1} = k_0^2 P_{E\to T_1}, \tag{2.4}$$

$$P^1_{T_1} = k_1^2 P_{E\to T_1}, \tag{2.5}$$

where k_0^2 and k_1^2 are constants proportional to the reflection coefficient of the tag, $|s|^2$, in each of the two states. Again, using Friis' equation, the

power received at tag T_2 from tag T_1 in the two states can be written as

$$\begin{aligned} P^0_{T_1 \to T_2} &= \frac{P^0_{T_1} G_{T1} G_{T2} \lambda^2}{(4\pi d_3)^2} \\ &= \frac{k_0^2 P_E G_E G_{T1}^2 G_{T_2} \lambda^4}{(4\pi d_1)^2 (4\pi d_3)^2}, \qquad (2.6) \\ P^1_{T1 \to T2} &= \frac{P^1_{T_1} G_{T1} G_{T2} \lambda^2}{(4\pi d_3)^2} \\ &= \frac{k_1^2 P_E G_E G_{T1}^2 G_{T2} \lambda^4}{(4\pi d_1)^2 (4\pi d_3)^2}. \qquad (2.7) \end{aligned}$$

We note that in the case of PSK these two powers are the same.

Tag 2 "sees" the superposition of two signals viz. the CW from the exciter and the modulated backscatter from Tag 1. Since tag T_2 is a passive device, it employs an envelope detector to decode the backscatter signal received from tag T_1. Depending upon the relative phase difference between the received exciter and backscatter signals at tag T_2, the amplitude of the resultant signal may be the same in both states $'0'$ and $'1'$. When this occurs, the envelope detector is unable to detect the modulated backscatter despite tag T_1 being in the *range* of tag T_2 w.r.t. signal strength. This creates a "null spot" where the two tags cannot communicate. This phenomenon, which we refer to as phase cancellation, occurs in both ASK and PSK backscattering. We analyze the ASK and PSK modulations next.

2.2.1 ASK modulation

When the system uses ASK backscatter modulation, the tag alters the amplitude of the reflected signal in the states $'0'$ and $'1'$. The phase of the reflected signal in the two states remains the same. Let $S^0(t)$ and $S^1(t)$ represent the resultant superimposed signals received at tag T_2 when

backscattering tag T_1 is in state $'0'$ and $'1'$ respectively. Further, let $A^0_{T_1}$ and $A^1_{T_1}$ be the amplitudes of the received backscatter signal from tag T_1 in the two states, and let θ_E and θ_{T1} be the phases of the signals from the exciter and T_1 respectively at T_2 (the exciter is at phase 0). Note that since this is ASK backscattering, θ_{T1} will be the same in states $'0'$ and $'1'$.

From the geometry shown in Fig. 2.1, we can write the following expressions for the above mentioned phases:

$$\theta_E = \frac{2\pi f d_2}{c}, \tag{2.8}$$

$$\theta_{T_1} = \theta_{E\rightarrow T_1} + \theta_b + \frac{2\pi f d_3}{c}, \tag{2.9}$$

where $\theta_{E\rightarrow T_1}$ is the phase difference of the CW signal from the exciter at T_1, and θ_b is the phase difference introduced by the backscattering hardware. Using (2.8) and (2.9), the relative phase difference, θ_d, between the two superimposing signals at T_2 is given by:

$$\begin{aligned} \theta_d &= \theta_{T1} - \theta_E \\ &= \frac{2\pi f(d_1 + d_3 - d_2)}{c} + \theta_b. \end{aligned} \tag{2.10}$$

Then we can write

$$\begin{aligned} S^1(t) &= A_E \cos(\omega t + \theta_E) + A^1_{T_1} \cos(\omega t + \theta_{T_1}) \\ &= A_E \cos\omega t \cos\theta_E - A_E \sin\omega t \sin\theta_E \\ &\quad + A^1_{T_1} \cos\omega t \cos\theta_{T_1} - A^1_{T_1} \sin\omega t \sin\theta_{T_1} \\ &= (A_E \cos\theta_E + A^1_{T_1} \cos\theta_{T_1}) \cos\omega t \\ &\quad - (A_E \sin\theta_E + A^1_{T_1} \sin\theta_{T_1}) \sin(\omega t). \end{aligned} \tag{2.11}$$

The expression for $S^0(t)$ is analogous. Let A^0 and A^1 represent the amplitudes of the signal detected by the evelope detector of T_2 in the two states. Then we can write:

$$\begin{aligned} &A^1 = \\ &\sqrt{(A_E \cos\theta_E + A^1_{T_1} \cos\theta_{T_1})^2 + (A_E \sin\theta_E + A^1_{T_1} \sin\theta_{T_1})^2} \\ &= \sqrt{A_E^2 + 2A_E A^1_{T_1} \cos\theta_d + (A^1_{T_1})^2}, \end{aligned} \tag{2.12}$$

and

$$\begin{aligned} &A^0 = \\ &\sqrt{(A_E \cos\theta_E + A^0_{T_1} \cos\theta_{T_1})^2 + (A_E \sin\theta_E + A^0_{T_1} \sin\theta_{T_1})^2} \\ &= \sqrt{A_E^2 + 2A_E A^0_{T_1} \cos\theta_d + (A^0_{T_1})^2}. \end{aligned} \tag{2.13}$$

Phase cancellation arises when the above two amplitudes are equal, i.e., when $A^0 = A^1$. This entails that in that case tag T_2 cannot detect and demodulate the signal sent from tag T_1. From the above equations we can determine that phase cancellation occurs when

$$\begin{aligned} &\sqrt{A_E^2 + 2A_E A^0_{T_1} \cos\theta_d + (A^0_{T_1})^2} \\ &\qquad = \sqrt{A_E^2 + 2A_E A^1_{T_1} \cos\theta_d + (A^1_{T_1})^2}, \end{aligned}$$

or when

$$\theta_d = \theta_c = \cos^{-1}\left(-\frac{A^0_{T_1} + A^1_{T_1}}{2A_E}\right), \tag{2.14}$$

where θ_c represents the angle at which phase cancellation occurs.

Using the power - amplitude relationship $P = A^2/2R$ where R is the input resistance of the detector circuit, we get the following relations for

amplitudes of signals received at T_2

$$\begin{aligned} A_E &= A_{E\to T_2} = \sqrt{2P_{E\to T_2}R_{T_2}} \\ &= \frac{\lambda\sqrt{2P_E G_E G_{T_2} R_{T_2}}}{4\pi d_2}, \qquad (2.15) \\ A^0_{T_1} &= A^0_{T_1\to T_2} = \sqrt{2P^0_{T_1\to T_2}R_{T_2}} \\ &= \frac{k_0 G_{T_1}\lambda^2\sqrt{2P_E R_{T_2} G_E G_{T_2}}}{16\pi^2 d_1 d_3}, \qquad (2.16) \\ A^1_{T_1} &= A^1_{T_1\to T_2} = \sqrt{2P^1_{T_1\to T_2}R_{T_2}} \\ &= \frac{k_1 G_{T_1}\lambda^2\sqrt{2P_E R_{T_2} G_E G_{T_2}}}{16\pi^2 d_1 d_3}. \qquad (2.17) \end{aligned}$$

Substituting from (2.15) into (2.14), we conclude that cancellation occurs when $\theta_d = \theta_c$ where

$$\theta_c = \cos^{-1}\left(-\frac{(k_0+k_1)d_2\lambda G_{T1}}{8\pi d_1 d_3}\right). \qquad (2.18)$$

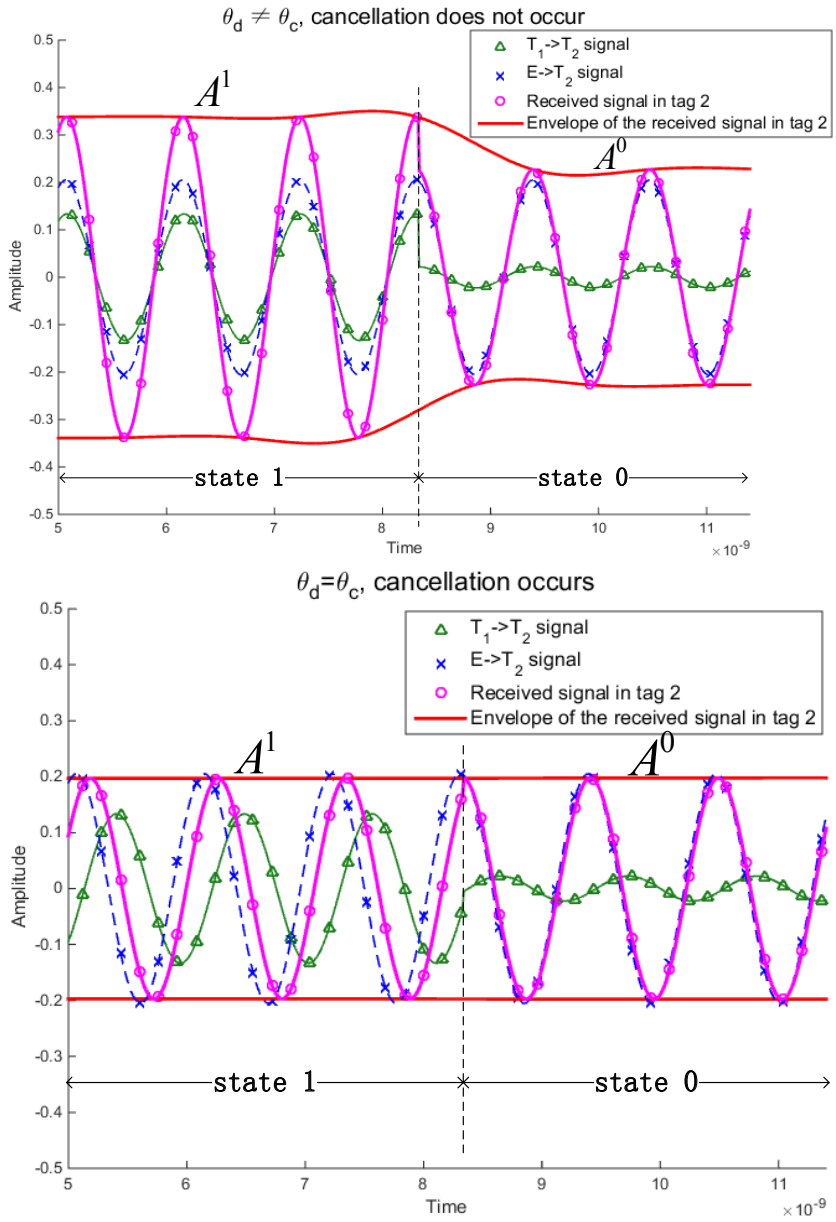

Figure 2.2: Top: Signal waveforms when there is no phase cancellation ($\theta_d \neq \theta_c$). Bottom: Signal waveforms when phase cancellation takes place ($\theta_d = \theta_c$).

The phase cancellation phenomenon is shown in Fig. 2.2. The blue line (marker x) represents the signal received from the exciter at T_2 ($E \rightarrow T_2$), and the green line (marker $\triangle$) shows the signal received at T_2

from T_1 in states $'0'$ and $'1'$ ($T_1 \rightarrow T_2$). The superimposed signals S_0 and S_1 are plotted in purple (marker o), and the envelope, with amplitudes A^0 and A^1 is shown in red. When $\theta_d \neq \theta_c$, we can see that $A^0 \neq A^1$. Hence the envelop detector will be able to differentiate between the two levels and demodulate the backscatter signal allowing for communication between T_1 and T_2. However as seen in Figure 2.2, when $\theta_d = \theta_c$, then the amplitude of the received envelop in the two states is the same i.e. $A^0 = A^1$. In this case the envelope detector will be unable to demodulate the tag backscatter leading to a loss of communication between the two tags.

The phase cancellation phenomenon is further illustrated in Figures 2.3 and 2.4 with phasor diagrams showing the signals received at T_2 with and without phase cancellation, respectively. The resultant envelope amplitudes A^0 and A^1 received at T_2 in the two states are obtained by adding the vectors representing the signals received from the exciter and from the backscattering tag T_1 in the two states.

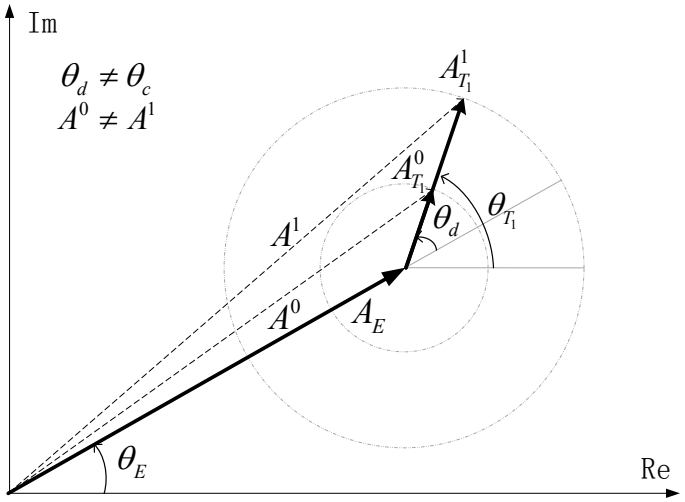

Figure 2.3: A phasor diagram of ASK modulation when phase cancellation does not occur. All the symbols are defined in the text.

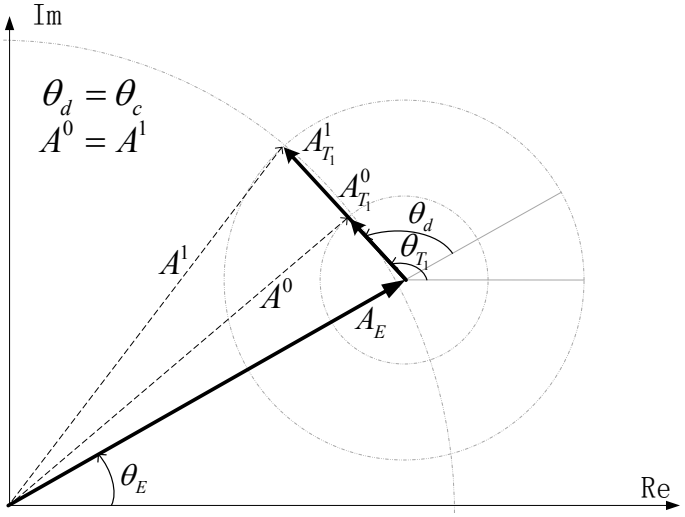

Figure 2.4: A phasor diagram of ASK modulation that represents settings with phase cancellation. All the symbols are defined in the text. Note that there are two angles θ_c of phase cancellation.

2.2.2 PSK modulation

When the system uses PSK backscatter modulation, the tag T_1 changes only the phase of the reflected signal in the states $'0'$ and $'1'$. The amplitude of the reflected signal in the two states is the same. At the receiving tag, this PSK backscatter signal superimposes with the CW signal from the exciter to produce two resultant signals, one for each state $'0'$ and $'1'$. As before, if the amplitudes of the resultant signals in the two states are equal, then the envelope detector will not be able to demodulate the backscatter signal. In this subsection, we analyze the phase cancellation phenomenon in a BBTT system utilizing PSK backscattering.

In PSK modulation, the amplitudes of the two states $'0'$ and $'1'$ are the same i.e. $A^0_{T_1} = A^1_{T_1} = A_{T_1}$. Using (2.15), we can write

$$A_E = A_{E\to T_2} = \frac{\lambda\sqrt{2P_E G_E G_{T_2} R_{T_2}}}{4\pi d_2}, \tag{2.19}$$

$$A_{T_1} = A_{T_1\to T_2} = \frac{kG_{T_1}\lambda^2\sqrt{2P_E R_{T_2} G_E G_{T_2}}}{16\pi^2 d_1 d_3}, \tag{2.20}$$

where k is a constant determined by the cross-section of the antenna of Tag 1.

The tag backscatters PSK signals by varying the imaginary (reactance) part of its power reflection coefficient between the two states. Let $\psi \in (0, 2\pi)$ be the phase difference of between the signals backscattered in states $'0'$ and $'1'$. For the phases of the states, we can write

$$\theta^0_{T_1} = \theta^0_{T_1\to T_2} = \frac{2\pi f(d_1+d_3)}{c} + \theta_b, \tag{2.21}$$

$$\begin{aligned}\theta^1_{T_1} &= \theta^1_{T_1\to T_2} = \theta^0_{T_1} + \psi \\ &= \frac{2\pi f(d_1+d_3)}{c} + \theta_b + \psi,\end{aligned} \tag{2.22}$$

where θ_b is the phase difference introduced by the backscattering mechanism in state $'0'$ and $\theta_b + \psi$ in state $'1'$.

Similar to the ASK modulation case, the resultant signals received at T_2 when T_1 utilizes PSK backscattering can be written as:

$$S^0_{PSK}(t) = A_E \cos(\omega t + \theta_E) + A_{T_1} \cos(\omega t + \theta^0_{T_1}), \tag{2.23}$$

$$S^1_{PSK}(t) = A_E \cos(\omega t + \theta_E) + A_{T_1} \cos(\omega t + \theta^0_{T_1} + \psi). \tag{2.24}$$

We can write the phase differences between the two superimposing signals, viz. the exciter signal and backscatter from T_1 at T_2 in the two states as:

$$\theta^0_d = \theta^0_{T_1} - \theta_E = \frac{2\pi f(d_1 + d_3 - d_2)}{c} + \theta_b, \tag{2.25}$$

$$\begin{aligned} \theta^1_d &= \theta^1_{T_1} - \theta_E = \theta^0_d + \psi \\ &= \frac{2\pi f(d_1 + d_3 - d_2)}{c} + \theta_b + \psi. \end{aligned} \tag{2.26}$$

Substituting from (2.25), (2.26) into (2.23), (2.24) and solving, we get the resultant envelope amplitudes in the two states as

$$A^0 = \sqrt{A_E^2 + 2A_E A_{T_1} \cos\theta^0_d + (A_{T_1})^2}, \tag{2.27}$$

$$\begin{aligned} A^1 &= \sqrt{A_E^2 + 2A_E A_{T_1} \cos\theta^1_d + (A_{T_1})^2} \\ &= \sqrt{A_E^2 + 2A_E A_{T_1} \cos(\theta^0_d + \psi) + (A_{T_1})^2}. \end{aligned} \tag{2.28}$$

As explained earlier, phase cancellation occurs when $A^0 = A^1$. Then, from (2.27) and (2.28) we deduce that this happens when $\cos\theta^0_d =$

$\cos(\theta_d^0 + \psi)$. Thus, the condition for phase cancellation is

$$\theta_d^0 = \theta_c = n\pi - \frac{\psi}{2}, n = 0, 1, 2, \ldots \tag{2.29}$$

Figures 2.5 and 2.6 show the phasor diagrams of signals received at T_2 when T_1 is backscattering using PSK modulation. The phasor diagrams depict the conditions where phase cancellation does not occur and where it occurs. We can see that the signals of the two states have equal amplitudes. However, the different phases of the states cause that the received signals have different amplitudes A^0 and A^1, unless phase cancellation occurs.

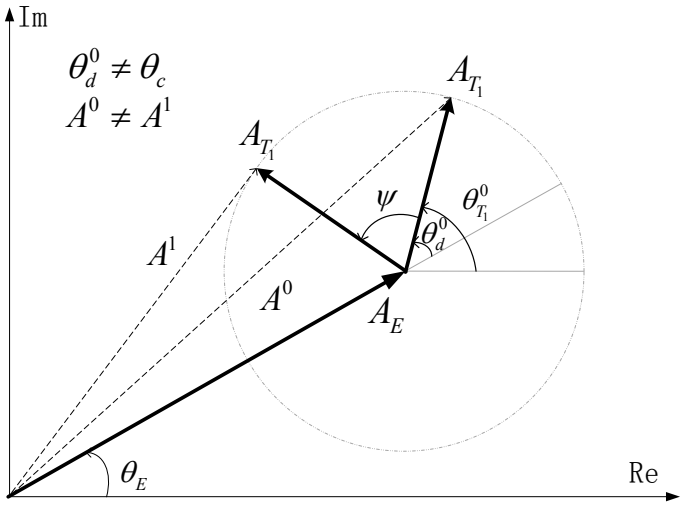

Figure 2.5: A phasor diagram of PSK modulation when phase cancellation does not occur. All the symbols are defined in the text.

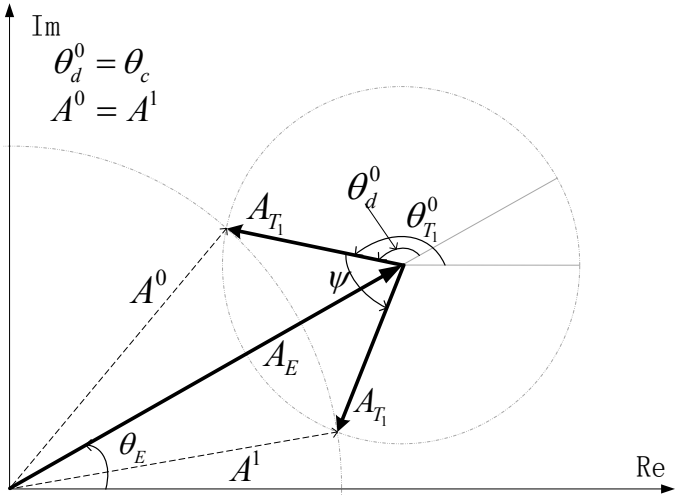

Figure 2.6: A phasor diagram of PSK modulation that represents settings with phase cancellation. All the symbols are defined in the text. Note that there are two angles θ_c of phase cancellation.

2.3 Possible Solutions

One solution for the phase cancellation problem is a tag design with multiple antennas. In [13], a design for ASK modulation with a dual-antenna is addressed. If the two received signals by the antennas have different phases θ_d, the phase cancellation can be avoided in some cases. For the PSK modulation, we can also use multiple antennas. The solution based on multiple antennas may be of limited value if the distance between the two antennas is small. In that case, the difference between A^0 and A^1 may be below the detecting threshold of the envelope detector.

Another solution can be based on sending the same message two or more times (time multiplex) or use frequency hopping by the exciters. According to the time multiplex, during one transmission, the tag sends a message in the usual way, and in other time slots, the tag resends the message but with different parameters that affect θ_b and ψ in (2.25) and (2.26). In that case, at least one transmission will satisfy $\theta_d \neq \theta_c$. We note that in our system, the positions of the exciter and the tags are fixed, and therefore we can only change θ_b, f, k_0 and k_1. The frequency can only be changed by the exciters and therefore, this solution is only possible when the exciters can be controlled. For example, one may have two or more synchronized exciters that emit CWs at different frequencies. Another alternative is to have the exciters employ frequency hopping. The effectiveness of this solution would depend on the bandwidth of the frequency hopping. For UHF RFID signals, this solutions would provide poor results.

In the next section we present our proposed solution based on changing θ_b.

2.4 Solutions based on phase-diverse backscatter modulation

Our proposed solution is based on the introduction of phase diversity in the backscattering via the use of an enhanced backscatter modulator. As compared to a multiple antenna solution, this solution keeps the size of the tag small. Moreover, the phase diversity introduced by a multi-antenna solution is non-deterministic and depends upon the separation between the antennas and the random geometries created by the environment. In order to achieve sufficient phase diversity, the separation between the antennas may need to be large which further increases tag size. By using an enhanced backscatter modulator, we can introduce a deterministic phase diversity into the backscatter link irrespective of the environment. This will allow systems to overcome the phase cancellation problem without requiring multiple antennas. Our proposed solution uses multi-phase backscattering employed by the transmitting tag. The tag will backscatter its information in two successive intervals with a deterministic phase difference between the backscattered signal in the two intervals. The use of phase diversity in backscattering implies that if there is a cancellation during one of the intervals, it is avoided in the other. In a straightforward implementation, this scheme will increase the robustness of the tag-to-tag link while reducing the throughput. However, in our future work, we will explore higher layer protocol mechanisms which introduce a handshaking mechanism to first determine the optimal phase to use in backscattering and then transmitting the information only once.

We now describe the proposed solution for the case of ASK

modulation and PSK modulation.

2.4.1 ASK modulation with multi-phase backscattering

Under this scheme, a backscattering tag will send out its data in two successive intervals. In each of these two intervals the tag uses different phases θ_b, which according to (2.10) creates two different phase differences between the excitation signal and backscatter signal at the receiving tag. A standard implementation of ASK modulation requires the use of two different impedances of the tag. By contrast, in the proposed solution, we use four impedances. Our design, shown in Fig. 2.7 (see also [14]), is similar to the QAM (Quadrature Amplitude Modulated) backscatter scheme from [15], [16] and [17]. The symbols Z_1 and Z_2 are two impedances with different real parts and the same imaginary part for the states $'0'$ or $'1'$, respectively. On the other hand, the impedances Z_3 and Z_4 are chosen such that the backscatter signals have the same amplitudes as when Z_1 or Z_2 are used, but their imaginary part is different from that of Z_1 and Z_2, respectively. As a result, the phase of the backscatter signals when they are used is changed.

Without loss of generality, we consider the phase of the backscatter signals to be 0 when Z_1 or Z_2 are selected and θ_n when Z_3 or Z_4 are selected

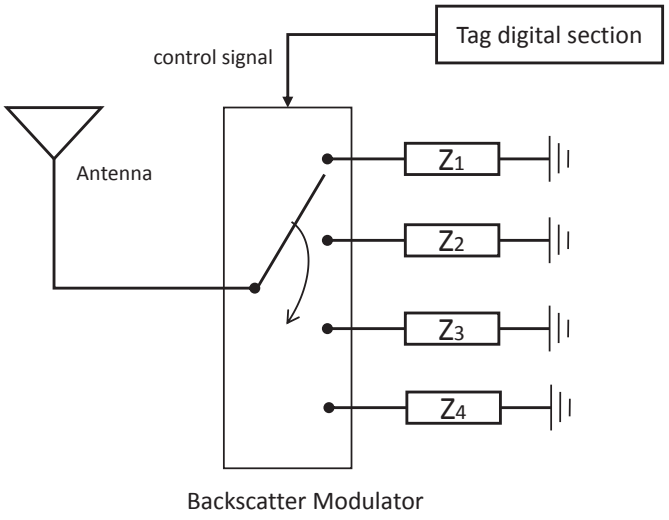

Figure 2.7: Block diagram of the new tag that uses four impedances for the ASK modulation.

As described earlier, in the first interval, the backscatter modulator switches the tag impedance between Z_1 and Z_2. In the next interval, this tag sends the same message again by switching between Z_3 and Z_4. In (2.18), the argument of arccos is $\frac{-(k_1+k_0)d_2\lambda G_{T1}}{8\pi d_1 d_3} < 0$, so the first θ_c is in the range of $(\frac{\pi}{2}, \pi]$, and for the second one we have $(2\pi - \theta_c) \in [\pi, \frac{3\pi}{2})$. The range of θ_d can be $[0, 2\pi)$, so phase cancellation can occur at two values of θ_d in one period.

The envelope detector has a threshold for discriminating the state $'0'$ from state $'1'$, and therefore our objective is to maximize $|A^1 - A^0|$. From Fig. 2.3, when $\theta_d = 0$ or π, we obtain the optimum result, which is $|A^1 - A^0| = |A^1_{T1} - A^0_{T1}|$. Therefore the optimum solution is to make $\theta_n = \pi - \theta_d$ or $2\pi - \theta_d$ ($\theta_n \geq 0$). But this solution requires a tag with increased hardware complexity, which goes against the reason why we use the backscatter-based system. So we use a fixed value of θ_n.

Because we only focus on θ_d, without loss of generality, we assume

$\theta_E = 0$. In Fig. 2.8, when $\theta_d = \theta_c$, the phase cancellation occurs during the first period. During the second period, θ_d is increased by θ_n, and then $A^0 \neq A^1$, which entails that phase cancellation does not occur. The phase θ_n is determined by Z_3 and Z_4. When considering the implementation of the proposed backscatter modulator, the values of the impedances to generate a specified phase difference in the backscattered signal can be calculated using methods described in [15] and [16].

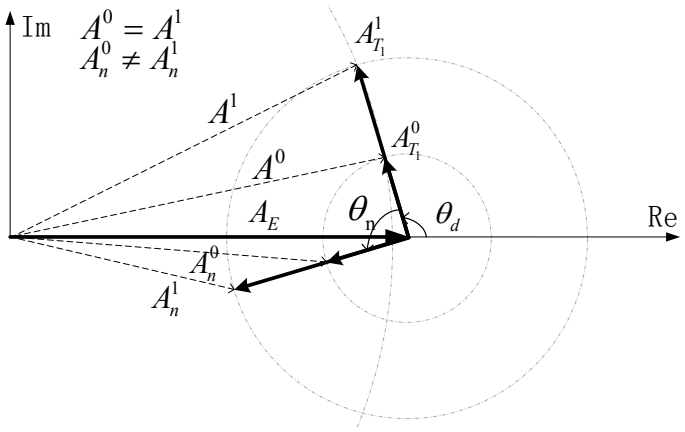

Figure 2.8: Phasor diagrams of multi-phase backscattering in ASK modulation.

2.4.2 PSK modulation with multi-phase backscattering

Our proposed method can also be used for PSK modulation. The implementation is similar in that it uses a backscatter modulator with four impedances and the tag sends the same message in two successive intervals. In the case of PSK modulation, the real parts of all impedances $Z_1 \cdots Z_4$ are the same and the difference between the imaginary parts of Z_1 and Z_2 is the same as that between those of Z_3 and Z_4. This means that the PSK

modulation index is the same in the two intervals, but there is deterministic phase difference between the signals received in the two intervals. We note that the performance for PSK modulation is better than that for ASK modulation. The reason is that the phase cancellation can theoretically be fully avoided when the phase difference between the two pair of impedances θ_n is in $(0, \frac{\pi}{2}]$. When phase cancellation occurs, the phase differences of the two states θ_d^0 and $\theta_d^1 = \theta_d^0 + \psi$ are symmetrical about the $n\pi$-axis. So when they both add $\theta_n \in (0, \frac{\pi}{2}]$, they will not be symmetrical anymore. A phasor diagram representing this is shown in Fig. 2.9. There is phase cancellation in the first period and so $\theta_d^0 = \theta_c$ and $A^0 = A^1$. Then in the second period, the phases of the two states are changed by θ_n and for the amplitudes we have $A'_0 \neq A'_1$.

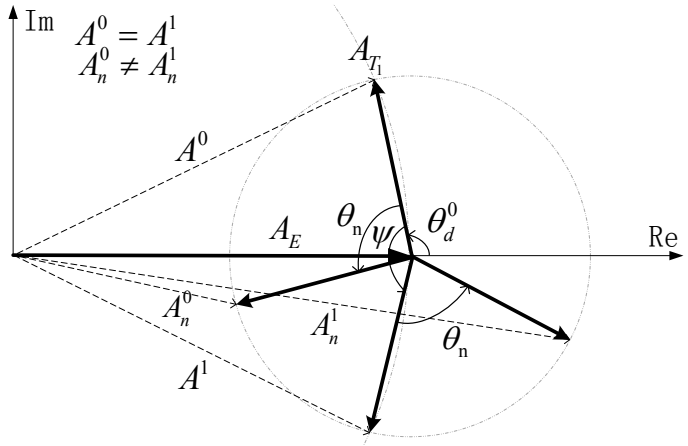

Figure 2.9: Phasor diagrams of multi-phase backscattering in PSK modulation.

2.4.3 Combination of signals

In a straightforward implementation of the above method, a transmitting tag backscatters the same signal in successive intervals and

the phase diversity ensures that one of these will be reliably detected by the receiving tag. This method also presents the opportunity to further improve performance by combining the backscattered signals received in the two intervals. Since the tag employs a purely analog circuit for envelop detection, one can develop a delay circuit to store the signal received in the first interval and then, combine it with the signal received in the second interval. With the envelope detector, one can form four amplitude differences: A_{d_1}, A_{d_2}, $A_{d_1} + A_{d_2}$ and $A_{d_1} - A_{d_2}$, where A_{d_1} and A_{d_2} are the amplitude differences in the first and second interval, respectively. The combination method can increase the differences of the amplitudes (almost double them), and therefore it can vastly improve the performance and communication ranges of the BBTT system. Implementation of this scheme would require a protocol wherein individual bits of symbols of a tag are repeated rather than the whole message. This is is because of the restriction on the amount of delay that can be built into the analog circuit. Because the demodulation is the same for ASK and PSK modulations, this method can be applied to both. In our future work, we will explore designs for tags that utilize phase-diverse backscatter modulation along with the above described signal combination.

2.4.4 Discussion

Theoretically, the proposed methods can avoid the phase cancellation problem almost entirely. Our solution requires four impedances, small changes in the digital section of the tag, and minor changes in the protocol. The approach can be extended to include multiple impedances allowing for more phase diversity. As mentioned earlier, in one of our future works, we

will be designing a protocol wherein two tags will first determine the best phase to use for communication during a handshaking period and then use that phase for a conversation. Having multiple possible impedances will be very beneficial in this approach.

Under a straightforward implementation (i.e. no signal combination, no special protocols) this proposed approach will increase the latency of the system. For most part, however, the applications of the BBTT system include backscattering of short messages. Therefore, the benefits of the multi-phase backscattering are much greater than the drawbacks.

We reiterate that the proposed solution cannot fully avoid the phase cancellation phenomenon using ASK modulation. Theoretically, when $\theta_d = \theta_c$ and $\theta_n = 2|\pi - \theta_d|$, θ_d changes from one cancellation phase $\theta_c \in (\frac{\pi}{2}, \pi]$ to another one, $2\pi - \theta_c$. Full cancellation can be avoided by using more impedances. Thus, we have a tradeoff between performance and latency.

2.5 Simulations and Experimental Results

2.5.1 Simulations

Here we present simulation results that demonstrate the performance of the proposed approach. We use a setting as shown in Fig. 2.1, where T_1 and the exciter are at fixed locations, and T_2 changes its location but stays in the plane defined by the locations of T_1, the exciter, and its initial location. More specifically, the position of T_1 is fixed at $(0, 0)$ and that of the exciter at $(0, 5)$ (thus, d_1 is equal to 5 m).

In the simulations, the power of the exciter was set at 13 dBm, and

G_{T_1}, G_{T_2} and G_R in (2.15) were 2 dB and 6 dB, respectively. Also we set $k_0 = 0.15$, $k_1 = 0.8$, R=50 Ω in (2.15) and θ_b due to Tag 1 for both states was set to π. Because we do not consider the amplifier circuit and other hardware resistance, the presented values of the amplitudes are relative. Also we simulate the proposed method ideally, which means that we do not consider other random effects, such as noise and multipath. We only focus on the curves before and after using our solutions. The simulation results of the PSK modulation are very similar to those of the ASK modulation, and therefore we do not show them.

2.5.1.1 Selection of θ_n

In (2.18), $k_0 + k_1$ is usually about 1 and $d_2 \approx d_1 >> d_3$. Since we use the UHF frequency, $\lambda \in [0.1, 1]$ m, and $\cos(\theta_c) = -\frac{(k_0+k_1)d_2\lambda G_{T1}}{8\pi d_1 d_3}$ is typically negative and very small. This means that θ_c is a little bit larger than $\frac{\pi}{2}$ or less than $\frac{3\pi}{2}$. From Figs. 2.4 and 2.8, when $\theta_d = \theta_c$, $\theta_d + \frac{\pi}{2}$ is close to π or 0, where we can obtain the largest amplitude differences between A^0 and A^1. So we fix θ_n to $\frac{\pi}{2}$.

2.5.1.2 Simulations in a two-dimensional space

First we show simulations in a two-dimensional space. In the experiment, Tag 2 only moves along the x axis starting at $(0, 0)$. We present the maximum amplitude difference between the two phases which reflect on the detection performance of the system. The results are shown in Fig. 2.10 for ASK modulation. The green line with downward-pointing triangle and the blue line with asterisk are the amplitude differences using one pair of impedances. When Tag 2 is in a position where $\theta_d = \theta_c$, the

value of the green line is 0. While in this position, the value of the blue line is not 0. This means that Tag 2 does not receive the message during the first time period, but receives it during the second period.

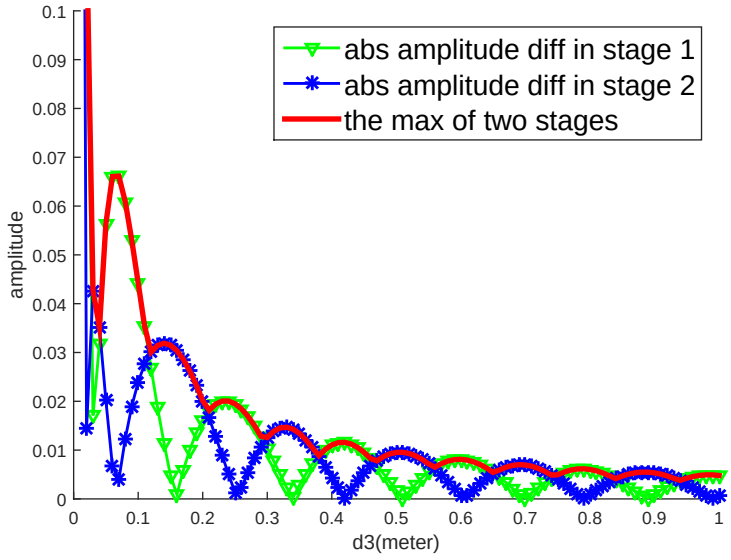

Figure 2.10: The amplitude differences using the proposed method for ASK modulation in a two-dimensional space.

2.5.1.3 Simulations in a three-dimensional space

In these simulations, we use the same settings as in the simulations in the two-dimensional space, except that Tag 2 can be in any position whose x and y coordinates are within $[-2, 2]$ m. In the presentation of our results, we use a threshold value for detecting the signal. When the signal is detected in both phases we use one symbol (and color), and similarly, different symbols (and colors) when the signal is detected only in the first or the second period, respectively.

The results are shown in In Fig. 2.11, the red ($'\cdot'$) and blue points ($'*'$) are the positions where the tag can receive the signal in the first

period, while the red and green ($'+'$) points are the positions where the tag successfully receives the signal in the second period. If we remove the green points, there are many positions where that tag cannot receive messages because of the phase cancellation problem. With the proposed method, the green points fill many new positions where the signal can be detected. However, there are still some locations where the signal is not detected, although the distances between these positions suggest that detection should take place. This can be explained by noting the shape of the red line in Fig. 2.10. This line still has local dips, and the positions that are not filled correspond to some of them.

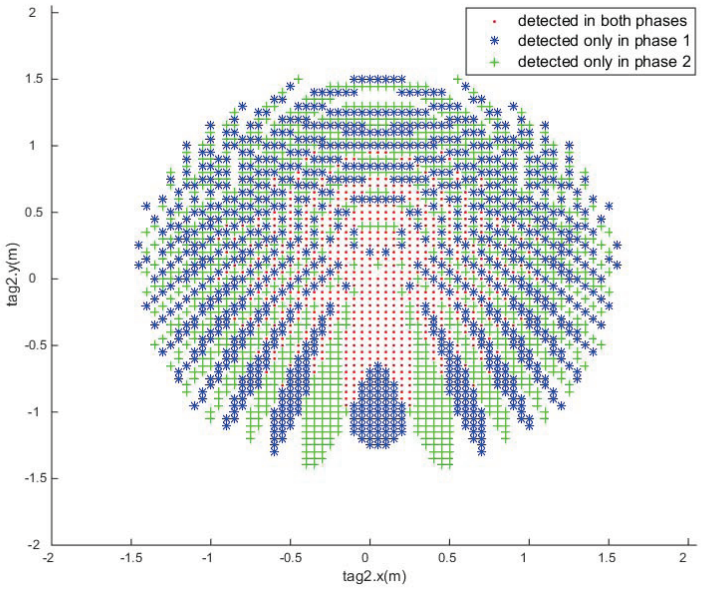

Figure 2.11: The amplitude differences using proposed method in 3d simulation for ASK modulation.

2.5.2 Experimental Results

Here we describe the lab experimental setup where we demonstrate the phase cancellation problem and the multi-phase backscattering approach that we propose to overcome it. Fig. 2.12 shows the tag prototype hat we used in the experiments. It consists of a printed dipole antenna connected to a backscatter modulator which consists of an Agilent ADG 902 single pole double throw (SPDT) RF switch. The envelop detector used for demodulating the backscatter is built in a separate board using a Shottky diode doubler [18]. The diode detector board is connected to one of the ports of the switch. This corresponds to state $'0'$ when the tag backscatters. A variable capacitance is connected to the other port of the switch and this corresponds to state $'1'$ of the backscattering tag. The switch control input of the backscattering tag is driven by a pulse signal from a function generator. At the receiving tag, the output of the envelope detector is connected to the oscilloscope. In Fig. 2.13, we show the experimental setup components of the system: one exciter, Tag 1 and Tag 2 (all circled). The prototype backscatter modulator is shown in Fig. 2.14. When the exciter is turned on, T_1 generates modulated backscatter

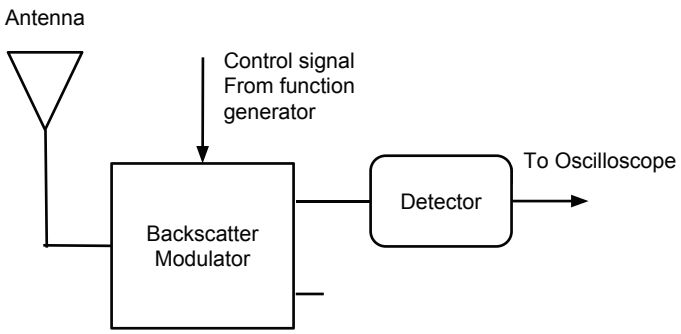

Figure 2.12: The tag prototype

and which is demodulated by T_2. The demodulated baseband signal is

Figure 2.13: The experimental setup composed of one exciter and two tags.

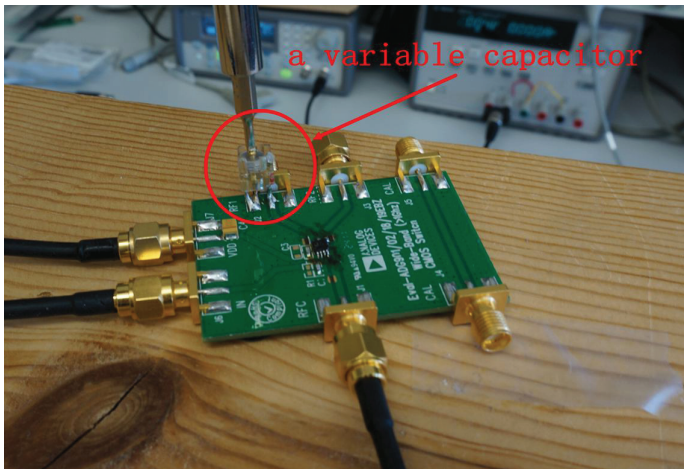

Figure 2.14: A prototype backscatter modulator with a variable capacitor.

observed on the oscilloscope. In the experiment, the two tags were placed close to each other so that the received signal power was large enough, and we could ignore all the noises and multi-path effects. We observed that when we moved T_2 in small increments away from T_1, the amplitude of the observed pulse signal on the oscilloscope varys continuously between peaks and nulls with gradually decreasing peaks as the tag moves further away. This observation is in line with the simulations shown in Fig. 2.10. We then placed T_2 close to T_1 but in a position where the oscilloscope showed almost a straight line i.e. where there was perfect phase cancellation. Then

keeping the positions of all devices the same, and manually varying the capacitance connected to the backscatter modulator of T_1, we observed the detected backscatter at T_2. The experimental results are shown in Figs. 2.15, 2.16, and 2.17. We can clearly see that when the variable capacitor changes the phase of the backscatter, the amplitude difference becomes much larger than in Fig. 2.15. This experiment demonstrates that the multi-phase backscattering approach can be successfully used to overcome the phase cancellation problem.

We would like to point out that the phase cancellation problem itself can also be demonstrated by using a standard RFID reader as an exciter, a standard RFID tag as T_1 and a similar (envelope detector/oscilloscope) setup for T_2. We have observed this during some of our prior work [19]. The above approach demonstrates both, the phase cancellation problem and our proposed approach to solve it for a generalized BBTT system.

2.6 Summary

In this chapter, we described the phase cancellation problem of backscatter-based tag-to-tag communication systems. In these systems, a tag receives the superposition of a CW and backscatter signals which have the same frequency but different phases. The combination of these two signals may cause that a tag cannot discriminate between the signals corresponding to states $'1'$ and $'0'$. We addressed and analyzed the problem for both ASK and PSK modulations. We presented several solutions and proposed a multi-phase backscattering method. According to the method, a tag sends a message twice using different pair of

impedances. Our simulations show that for ASK and PSK modulations phase cancellation can greatly be reduced. We have verified our simulations in lab experiments.

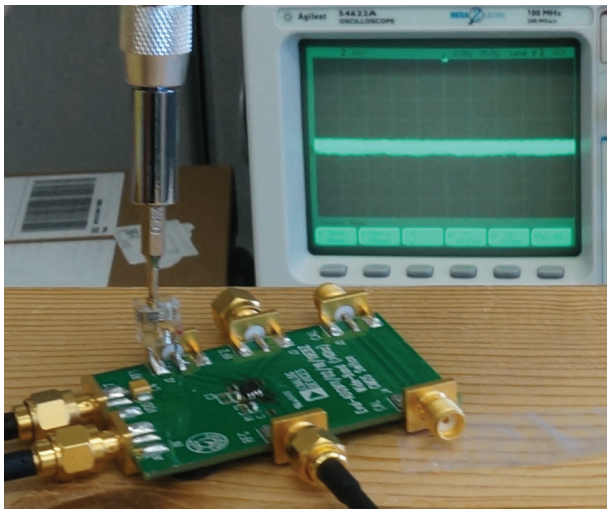

Figure 2.15: Phase cancellation. The amplitude difference between the states is about 0.

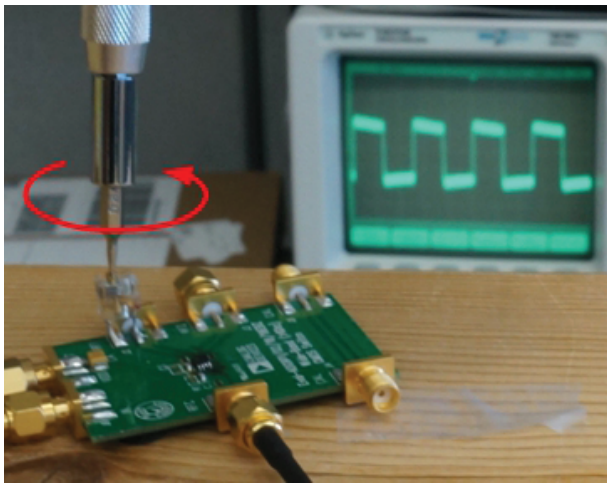

Figure 2.16: The amplitude difference between the two states increases when the value of the variable capacitor changes.

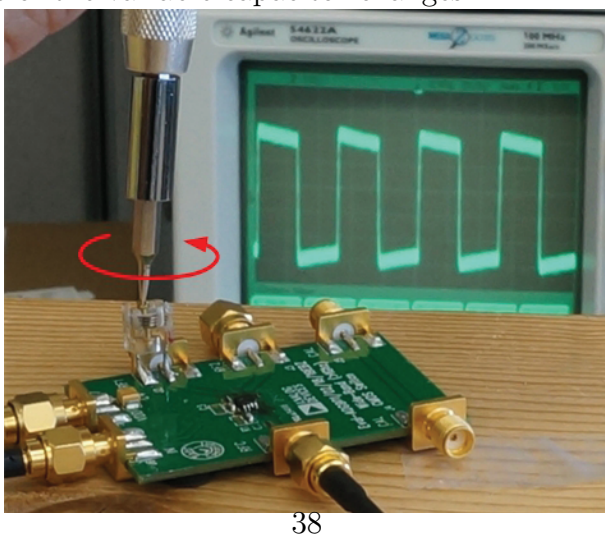

Figure 2.17: The amplitude difference between the two states when the phase due to the capacitor changes about $\pi/2$.

Chapter 3

Anti-Collision Protocol Selection

3.1 Introduction

In the previous chapter, we discussed the phase cancellation problem and presented a solution. In this chapter, we will discuss the collision problem in the same BBTT system. When more than one message is sent to a tag at the same time and this tag cannot receive any one of them successfully, a collision occurs. There are four different anti-collision methods illustrated in Fig 3.1: space division multiple access (SDMA), frequency domain multiple access (FDMA), time domain multiple access (TDMA), and code division multiple access (CDMA). Because of the limitation of RFID tags, we will focus only on TDMA.

In a regular RFID system, the reader performs as a centralized unit. Each tag applies for a time slot to transmit and the reader arranges and reserves a certain time slot for each tag. Therefore, a the collision can be

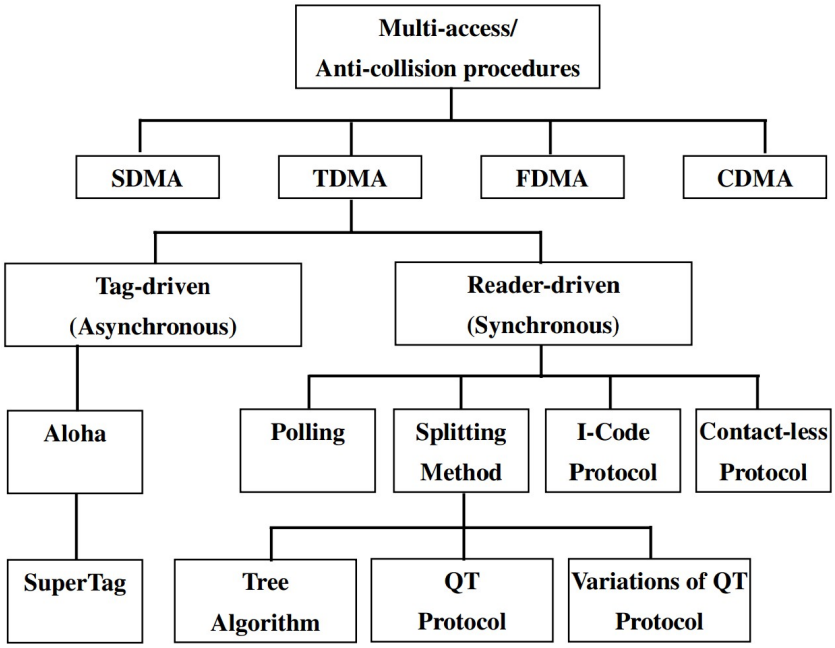

Figure 3.1: Existing protocols.

avoided. However, our BBTT system is a type of distributed backscatter system. There is no centralized unit to avoid or detect collisions. Each tag has the same hardware and runs the same program, so each tag performs the same except the tag connected to the sink. The sink is a computer that is connected to the continuous-wave(CW) generator and a regular tag. This computer can determine the transmitted message in the tag that is connected to the sink. The protocols discussed in this chapter need to be synchronized. The sink can turn the CW generator on and off to synchronize the tags.

Because, as of now, the memory in each tag can store only 4 to 5 IDs, we cannot use the acknowledge signals to detect collisions. For example, a tag receives 5 acknowledge signals, but it does not know the number of its neighbors. If it has 5 neighbors, no collision occurs. However, if it has 10 neighbors, 5 of them cannot receive the message. Further, the acknowledge

signals make the channel relatively busy.

Here, we consider a scenario to implement some simple applications. In our scenario, each tag keeps broadcasting a message in its memory and listening the rest of the time. When it receives a message that is different from the one in its own memory, it updates its memory by using a rule. This rule depends on the application. For example, in a library, the sink wants to check whether a book is in the network or not. The tag connected to the sink keeps sending the ID of the book. When other tags receive this ID, they check the received ID with their IDs. If the two are not the same, they store the new ID in their memory and keep sending it. If the two are the same, the tag keeps sending "true" which means that this book is in the network. Therefore, other tags sends "true" instead of the ID of the book, untill the sink receives the message "true". If the sink does not receive "true" for a certain time period, the sink will consider that the book is not in the network.

Therefore, after receiving a message, each tag keeps broadcasting a message. This is similar to the process carried out under the saturation condition. As mentioned earlier, we cannot detect or fully avoid the collision, so we want to find a good method to make the time between the query from the sink and the result from the other tags as short as possible in our scenario. CSMA/CA and framed slotted Aloha are two broadly used anti-collision protocols. In the following sections, we will modify these two protocols on the basis of our system and analyze them in complete networks. Most of the paper uses throughput as the performance metric. However, because of the requirements in our scenario, our performance metric is $E(T)$, which denotes the average transmission time from one

node to another. We also assume that the channel has ideal conditions. If two nodes can communicate with each other directly, there is an edge between these two nodes in the topology of the network.

3.2 Modified Slotted CSMA/CA Protocol

Carrier sense multiple access with collision avoidance(CSMA/CA) is a network multiple access method that uses carrier sensing to avoid collisions. In CSMA/CA, each node sends a message only when the channel is sensed to be idle. If the channel is busy, the node will wait a random backoff time and sense again.

The CSMA/CA methods in IEEE 802.15.4 and in IEEE 802.11 are slightly different. In IEEE 802.11, the nodes sense the channel for every time slot, and the backoff counter decreases only when the channel is sensed to be idle, as described in [20, 21]. However in the CSMA/CA of IEEE 802.15.4, a node decreases the backoff counter irrespective of the channel status, because carrier sensing consumes a certain amount of energy and IEEE 802.15.4 is designed for small wireless personal area networks (WPANs) with a low data rate [22].

In the slotted CSMA/CA algorithm [22], each node maintains three variables: NB, W, and CW. NB denotes the current backoff stage. W represents the backoff window that determines the number of backoff waiting slots before sensing a channel. CW indicates the contention window length, defining the number of backoff periods that need to be clear for carrier channel assessment (CCA) before transmission [23]. CW and NB are initialized to two and zero, respectively, before each transmission attempt.

W is initialized to W_0. CW will be reset to two each time the channel is assessed to be busy. A backoff counter is set to a random value in the range of $[0, W_0 - 1]$. In each time slot, the counter decreases by one. When it reaches zero, the node performs the first CCA. If the channel is idle, it performs the second CCA. If these two CCAs sense the channel to be idle, the node will transmit the message in the next time slot. After the message transmission, the backoff stage is reset to zero. Otherwise, if one of these two CCAs senses that the channel is busy, the backoff stage NB increases by one and W is doubled. If NB and W reach the maximal backoff stage m and window W_m, CSMA/CA must terminate with a channel-access-failure status. When the other nodes receive this message, they will send an acknowledge signal.

In our modified protocol, we use two variables: the backoff stage NB and the backoff window W. NB is in the range of $[0, m]$, and $W = W_0 2^{NB}$. Each node only performs the carrier sensing once before sending the message, because doing so reduces the energy consumption. Further, if the backoff stage NB reaches the maximal limit m, it maintains this stage instead of terminating.

The hidden node problem is that a node cannot sense nodes that are not its instant neighbors. CSMA/CA uses some mechanisms to solve this problem, such as Request to Send/Clear to Send. However, as of now, in our BBTT system, the memory of each tag can only store about 5 integers. The tag also needs the memory to maintain a status and perform calculations. Therefore, it is impossible to learn and establish a topology and let each tag remember its neighbors. We cannot use similar mechanisms to solve the hidden node problem. Further, the tags cannot use the acknowledge signals,

because they do not know who their neighbors are. Furthermore, they do not need them, because a tag will send the same information (assuming that it is not updated) again irrespective of whether its neighbors have receives it or not. We show an example of a 2-node transmission under a saturation traffic condition in Fig. 3.2. In this figure, (i, j) represents that this device is in the backoff stage i and the backoff counter is j now. The lengths of the CCA and a message are 1 and 6 time slots, respectively.

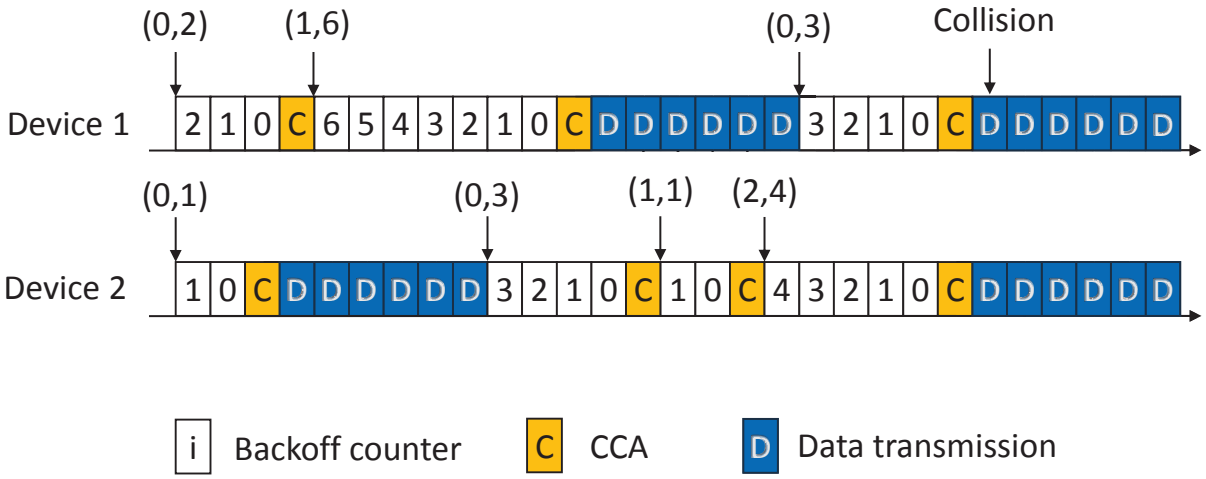

Figure 3.2: An example of a 2-node transmission.

3.2.1 Analysis in the Complete Network

We discus the similar methods of [20, 24, 25] in this section, but our protocol is different from their protocols. Let $s(t)$ be the stochastic process representing the stage,including the backoff stages and the transmission stages, and $b(t)$ be the stochastic process representing the backoff time counter for a given node at time t. We define two parameters: the stationary probability τ is that a node attempts its CCA within a time slot, and α is the probability that the node assesses the channel to be busy during the CCA. We assume that τ is a constant and is independent of all

stages and the other nodes. On the basis of this assumption, we can use the two-dimensional Markov chain $s(t), b(t)$ to derive these two parameters.

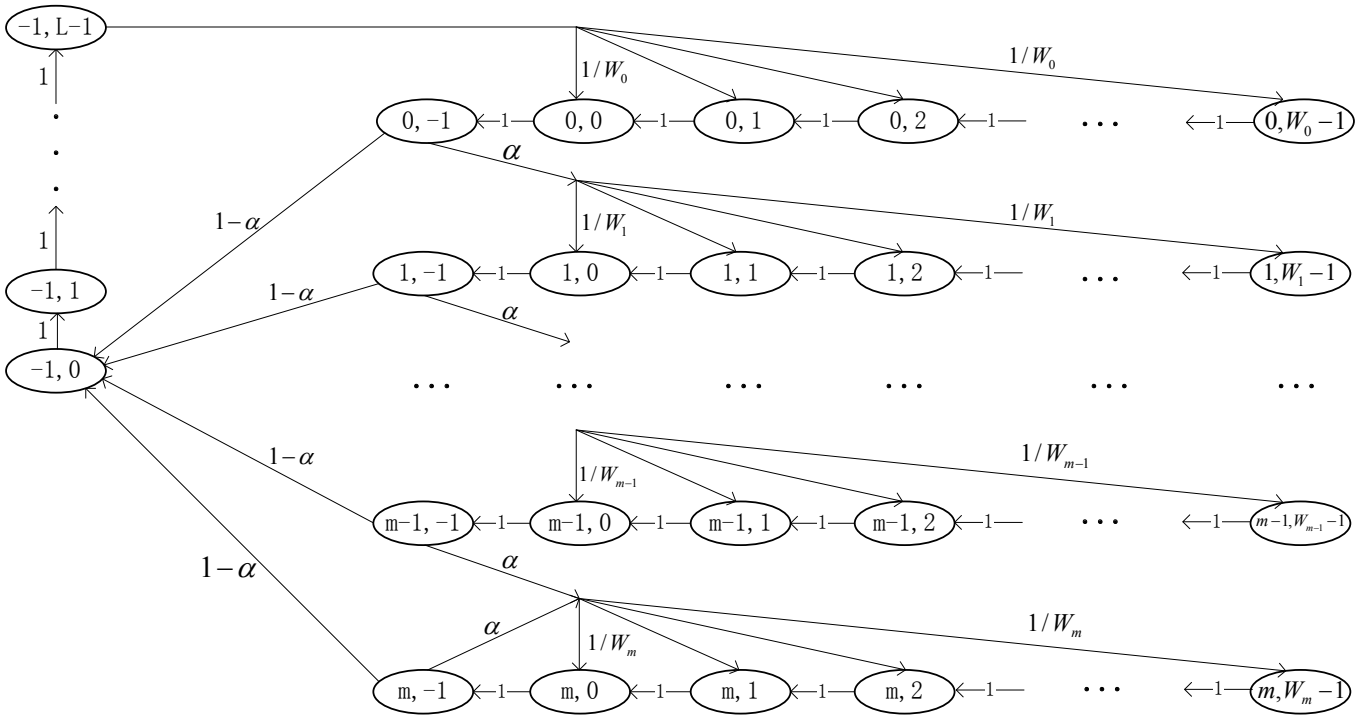

Figure 3.3: Markov model for our modified CSMA/CA protocol.

In each state of Fig. 3.3, the first number represents the stage $s(t) \in [-1, m]$. Each node is initially in stage 0 or after it sends a message. The largest stage is $s(t) = m$. Therefore, when a node is in stage m and senses that the channel is busy again, it will stay in stage m. The stage determines the contention window length $W_i = W_0 2^i$. At the beginning of each stage i, a node chooses a random number from $(0, W_i - 1)$, on the basis of the uniform distribution. In each stage, the backoff counter $b(t)$ decreases by 1 with the probability of 1 until it is equal to -1. The state $s(t), b(t) = -1$ means that this node attempts CCA. The state $s(t) = -1, b(t)$ means that the node is transmitting a message. L denotes the length of the messages.

We define the steady-state probability of the Markov chain as $b_{i,k} = P\{s(t) = i, b(t) = k\}, i \in (-1, m), k \in (-1, W_i - 1)$. Then from the Markov chain in Fig. 3.3, we have

$$\begin{aligned}
&P\{i, k|i, k+1\} = 1, && i \in (0, m), k \in (0, W_i - 1) \\
&P\{-1, k+1| - 1, k\} = 1, && k \in (0, L-2) \\
&P\{0, k|i, 0\} = \frac{1-\alpha}{W_0}, && i \in (0, m), k \in (0, W_0 - 1) \\
&P\{i+1, k|i, 0\} = \frac{\alpha}{W_{i+1}}, && i \in (0, m-1), k \in (0, W_{i+1} - 1) \\
&P\{m, k|m, 0\} = \frac{\alpha}{W_m}. && k \in (0, W_m - 1)
\end{aligned} \tag{3.1}$$

The third equation means that the probability that a node in stage i transmits a message and goes back to stage 0 is $\frac{1-\alpha}{W_0}$. The fourth and fifth equations represent that the node cannot send messages because the channel is busy. From Fig. 3.3, we can easily obtain

$$b_{i,0} = \alpha b_{i-1,0} = \alpha^i b_{0,0}, \qquad 0 \le i < m. \tag{3.2}$$

The stage m is different from the other stages:

$$\begin{aligned}
&b_{m,0} = \alpha(b_{m-1,0} + b_{m,0}) \\
&\Rightarrow b_{m,0} = \frac{\alpha}{1-\alpha} b_{m-1,0} = \frac{\alpha^m}{1-\alpha} b_{0,0}.
\end{aligned} \tag{3.3}$$

Using (3.2) and(3.3), we have

$$\begin{aligned}\sum_{i=0}^{m} b_{i,0} &= \sum_{i=0}^{m-1} b_{i,0} + b_{m,0} \\ &= \sum_{i=0}^{m-1} \alpha^i b_{0,0} + \frac{\alpha^m}{1-\alpha} b_{0,0} \\ &= \frac{1-\alpha^m}{1-\alpha} b_{0,0} + \frac{\alpha^m}{1-\alpha} b_{0,0} \\ &= \frac{b_{0,0}}{1-\alpha}.\end{aligned} \tag{3.4}$$

From the fourth equation, we obtain the following:

$$\begin{aligned}b_{i,k} &= \frac{W_i - k}{W_i} \alpha b_{i-1,0} \qquad 0 < i \leq m \\ &= \frac{W_i - k}{W_i} b_{i,0}.\end{aligned} \tag{3.5}$$

For stage 0, using (3.2) and (3.3), we obtain the following:

$$\begin{aligned}b_{0,k} &= \frac{W_0 - k}{W_0}(1-\alpha) \sum_{j=0}^{m} b_{j,0} \\ &= \frac{W_0 - k}{W_0}(1-\alpha)(\sum_{j=0}^{m-1} b_{j,0} + b_{m,0}) \\ &= \frac{W_0 - k}{W_0}(1-\alpha)(\frac{1-\alpha^m}{1-\alpha} b_{0,0} + \frac{\alpha^m}{1-\alpha} b_{0,0}) \\ &= \frac{W_0 - k}{W_0} b_{0,0}.\end{aligned} \tag{3.6}$$

Therefore,

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \qquad 0 \leq i \leq m \tag{3.7}$$

From (3.2) to (3.7), all states in the Markov chain can be expressed as functions by using $b_{0,0}$ and α. As mentioned earlier, each node is under

the saturation condition, which means that each node must be in one of the states shown in Fig. 3.3. Therefore, the summation of all states is equal to 1. Therefore, we can obtain the value of $b_{0,0}$ by

$$\begin{aligned}
1 &= \sum_{i=0}^{m}\sum_{k=-1}^{W_i-1} b_{i,k} + \sum_{i=0}^{L-1} b_{-1,i} \\
&= \sum_{i=0}^{m}\sum_{k=0}^{W_i-1} b_{i,k} + \sum_{i=0}^{m} b_{i,-1} + \sum_{i=0}^{L-1} b_{-1,i} \\
&= \sum_{i=0}^{m} b_{i,0} \sum_{k=-1}^{W_i-1} \frac{W_i - k}{W_i} + \sum_{i=0}^{m} b_{i,0} + L(1-\alpha)\sum_{i=0}^{m} b_{i,0} \\
&= \sum_{i=0}^{m} \frac{b_{i,0}}{W_i}(W_i^2 - \frac{(0+W_i-1)W_i}{2}) + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \sum_{i=0}^{m} b_{i,0}\frac{W_i+1}{2} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \sum_{i=0}^{m-1} b_{i,0}\frac{W_0 2^i + 1}{2} + b_{m,0}\frac{W_0 2^m + 1}{2} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \sum_{i=0}^{m-1} b_{0,0}\alpha^i \frac{W_0 2^i + 1}{2} + \frac{\alpha^m b_{0,0}(W_0 2^m + 1)}{2(1-\alpha)} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \frac{b_{0,0}}{2}\sum_{i=0}^{m-1}(W_0(2\alpha)^i + \alpha^i) + \frac{\alpha^m b_{0,0}(W_0 2^m + 1)}{2(1-\alpha)} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \frac{W_0 b_{0,0}(1-(2\alpha)^m)}{2(1-2\alpha)} + \frac{b_{0,0}(1-\alpha^m)}{2(1-\alpha)} + \frac{\alpha^m b_{0,0}(W_0 2^m + 1)}{2(1-\alpha)} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0} \\
&= \frac{W_0 b_{0,0}(1-(2\alpha)^m)}{2(1-2\alpha)} + \frac{b_{0,0}(1+W_0(2\alpha)^m)}{2(1-\alpha)} + \frac{b_{0,0}}{1-\alpha} + Lb_{0,0}
\end{aligned} \tag{3.8}$$

Then we get $b_{0,0}$ as a function of α

$$\begin{aligned}
b_{0,0} &= 1/(\frac{W_0(1-(2\alpha)^m)}{2(1-2\alpha)} + \frac{1+W_0(2\alpha)^m}{2(1-\alpha)} + \frac{1}{1-\alpha} + L) \\
&= \frac{2(1-\alpha)(1-2\alpha)}{W_0 - 6\alpha + 3 - \alpha W_0(1+(2\alpha)^m) + 2L(1-\alpha)(1-2\alpha)}
\end{aligned} \tag{3.9}$$

We also have that $\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-\alpha}$. We finally obtain the following first equation between τ and α

$$\tau = \frac{b_{0,0}}{1-\alpha} = \frac{2(1-2\alpha)}{W_0 - 6\alpha + 3 - \alpha W_0(1+(2\alpha)^m) + 2L(1-\alpha)(1-2\alpha)} \tag{3.10}$$

Now we derive the second equation. From the definition, we know that α denotes the probability of more than one transmission in the medium except in the given node in a given slot. Without any loss of generality, we assume that the given sensing node is i_0 and node i_1 transmits a message, and omit time t. Thus, we obtain the following:

$$\alpha = \sum_{j=1}^{n-1} \binom{n-1}{j} P(\bigcup_{k=1}^{j} s_{i_k} = -1 | b_{i_0} = 0) \tag{3.11}$$

$$= \sum_{j=1}^{n-1} \binom{n-1}{j} P(s_{i_1} = -1) P(\bigcup_{k=2}^{j} s_{i_k} = -1 | s_{i_1} = -1, b_{i_0} = 0) \tag{3.12}$$

$$= P(s_{i_1} = -1) \sum_{j=1}^{n-1} \binom{n-1}{j} P(\bigcup_{k=2}^{j} s_{i_k} = -1 | s_{i_1} = -1, b_{i_0} = 0), \tag{3.13}$$

$$\tag{3.14}$$

where $s_{i_k} = -1$ denotes the event that node i_k is transmitting a message and $b_{i_0} = 0$ means that the given node i_0 is sensing the channel.

The probability that node i_1 is transmitting a message is as follows:

$$\begin{aligned} P(s_{i_1} = -1) &= \sum_{j=0}^{L-1} P(s = -1, b = j) \\ &= LP(s = -1, b = 0) \\ &= L\tau(1-\alpha). \end{aligned} \tag{3.15}$$

Because each node attempts a CCA before transmitting a message, the only way that there is more than one transmission in a particular slot is that these nodes perform the CCA in the same slot. Thus the conditional probability is as follows:

$$P(\bigcup_{k=2}^{j} s_{i_k} = -1 | s_{i_1} = -1, b_{i_0} = 0) = \tau^j (1-\tau)^{N-2-j}. \tag{3.16}$$

The second part of (3.18) can be written as a summation of binomial distributions. Then

$$\begin{aligned} &\sum_{j=1}^{n-1} \binom{n-1}{j} P(\bigcup_{k=2}^{j} s_{i_k} = -1 | s_{i_1} = -1, b_{i_0} = 0) \\ =&\sum_{j=1}^{n-1} \binom{n-1}{j} \tau^{j-1} (1-\tau)^{n-1-j} \\ =&\frac{\sum_{j=1}^{n-1} \binom{n-1}{j} \tau^j (1-\tau)^{n-1-j}}{\tau} \\ =&\frac{1-(1-\tau)^{n-1}}{\tau}. \end{aligned} \tag{3.17}$$

This represents all possible situations in which more than one transmission occurs in nodes other than the transmitting node i_1.

Substituting (3.15) and (3.17) into (3.11), we have

$$\begin{aligned} \alpha &= L\tau(1-\alpha)\frac{1-(1-\tau)^{n-1}}{\tau}, \\ &= L(1-\alpha)(1-(1-\tau)^{n-1}) \\ \Rightarrow \alpha &= \frac{L(1-(1-\tau)^{n-1})}{1+L(1-(1-\tau)^{n-1})} \end{aligned} \tag{3.18}$$

Using (3.10) and (3.18), we obtain the values of τ and α numerically with the given parameters L, W_0 and m.

From the definitions of τ and α, we find that the probability of successfully transmitting a message for a given node in a time slot is $p = \tau(1-\tau)^{n-1}(1-\alpha)$. Therefore, the time at which a node sends a message to another node successfully follows a geometric distribution with parameter p. We can calculate the expectation time as follows:

$$E(T) = \frac{1}{p} = \frac{1}{\tau(1-\tau)^{n-1}(1-\alpha)}. \tag{3.19}$$

3.2.2 Model Validation and Simulation

In our BBTT system, we use 4-symbol Miller encoding [26]. The ID of a tag is a 32-bit integer. The head and the tail of a message are both 5 bits long. Therefore, each message is 42 bits long. After 4-symbol Miller encoding, a message requires $(5 + 32 + 5) * 4 = 168$ symbols. In the CCA, we use 3 symbols to sense whether a channel is busy or not. We use the 3 symbols as 1 unit time slot. Therefore sending a message requires 56 time slots.

In the simulation, we use different α values and calculate the new α', using (3.10) and (3.18). Intersections with the line $\alpha = \alpha'$ give the analytical values. We change the number of nodes from 2 to 50, and run the simulations 1000 times to calculate the average values of τ and α.

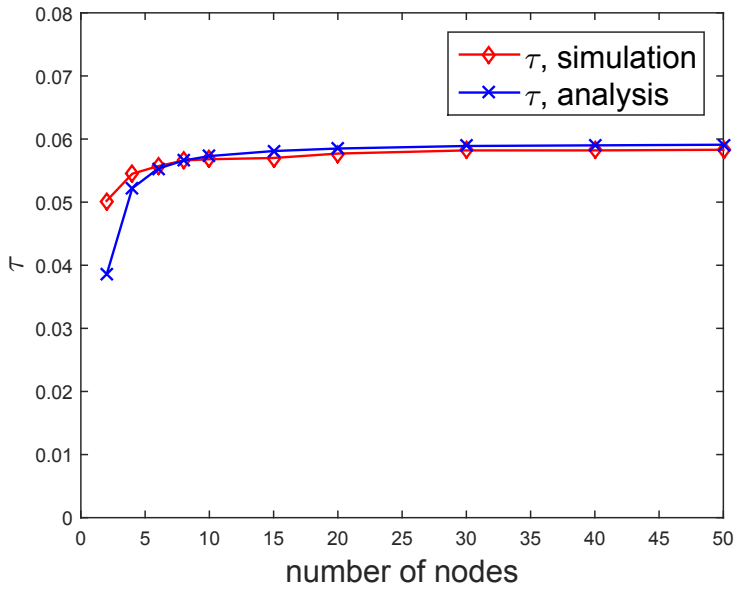

Figure 3.4: An example of a comparison between an analysis and a simulation of τ. $W_0 = 2^2$, $m = 3$ and $L = 56$.

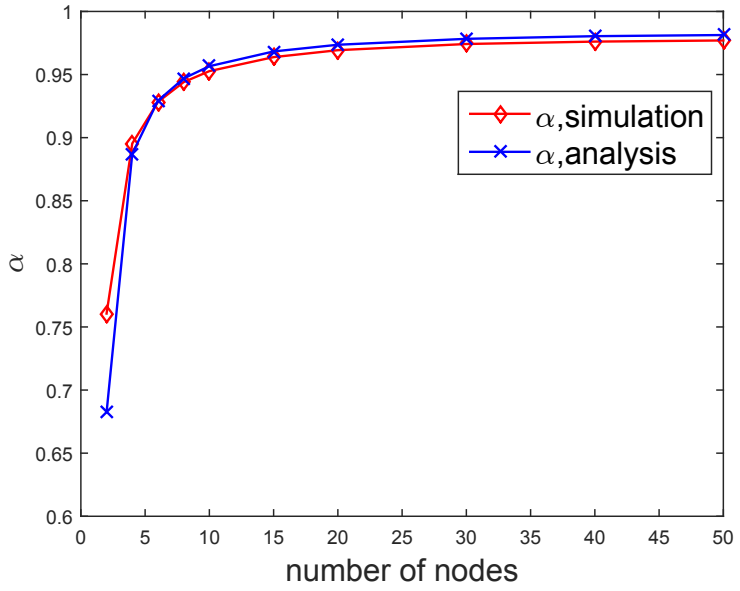

Figure 3.5: An example of a comparison between an analysis and a simulation of α. $W_0 = 2^2$, $m = 3$ and $L = 56$.

We compare the analytical results with the simulation results shown

in Figs. 3.4 and 3.5. Thus, we see that in most cases, our analytical results closely match the simulation results when the network has more than 2 nodes. The errors between them do not increase when the number of nodes increases. We consider that the independent assumption causes these errors. Because of the error accumulation and our assumptions, the expectation of a successfully transmission is not very accurate. The purpose of our analytical model is to obtain the optimum values of parameters W_0 and m and compare the performance with that of other protocols in complete networks.

3.3 Framed Slotted Aloha Protocol

The framed slotted aloha(FSA) anti-collision Protocol is broadly used in commercial RFID systems. It is usually considered simpler and has poorer performance than the CSMA/CA protocol. In [26], the researchers introduce several dynamic framed slotted Aloha algorithms to adjust the size of a frame. However, these algorithms increase the system cost because of the use of tag estimation functions and need a reader to command tags. Here we want to know the average transmission time under the setting of our BBTT system.

In the FSA protocol, each node sends one message within a time period. We define a parameter k as the number of time slots in one time period. The node generates a random number from 1 to k and chooses the time slot to send a message on the basis of this random number. Here, a time slot is defined as the time during which a node can send an entire message. Therefore, in our BBTT system, currently, one time slot consists

of 168 symbols.

Therefore, if two nodes choose the same time slot, a collision occurs, which means that nodes cannot receive or send messages successfully. Clearly, k and the degrees of tags determine the performance. We show an example of a 2-node transmission using this protocol in Fig. 3.6, where we set the number of time slots in one time period k as 10. We can see that in the second time period, two devices choose the fifth slot to transmit a message in, which causes a collision.

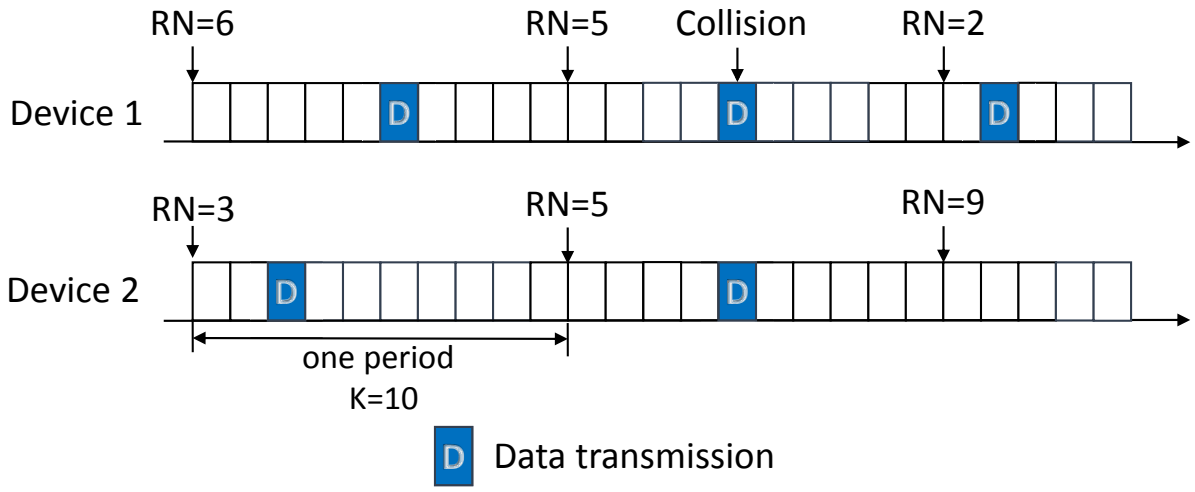

Figure 3.6: An example of a 2-node transmission. Here $k = 10$.

3.3.1 Collision Analysis of Complete Network

Let us assume that the given node i wants to send a message to the given node j. The degree of node j is d_j. Therefore, the probability that node i can transmit to node j successfully is as follows:

$$p_{i \to j} = \frac{\binom{k}{1}(k-1)^{d_j - 1}}{k^{d_j}} = \frac{(k-1)^{d_j - 1}}{k^{d_j - 1}}. \tag{3.20}$$

Because we choose one slot for node j, the probability that the other nodes choose of the rest $k-1$ slots is $\frac{(k-1)^{d_j-1}}{k^{d_j-1}}$. Let us use $T_{i\to j}$ to represent the number of periods in which node j can receive the message from node i. Clearly, $T_{i\to j}$ follows a geometric distribution. Therefore, the expected time can be calculated as follows:

$$\begin{aligned} E(T_{i\to j}) &= \sum_{T_{i\to j}=1}^{\infty} T_{i\to j}(1-p_{i\to j})^{T_{i\to j}-1}p_{i\to j} \\ &= \frac{1}{p_{i\to j}} = (\frac{k}{k-1})^{d_j-1}. \end{aligned} \tag{3.21}$$

For a complete network with n nodes, we can easily obtain the probability that one node transmits to another node successfully in one period, as follows:

$$p_{comp} = \frac{k(k-1)^{n-1}}{k^n} = (\frac{k-1}{k})^{n-1}. \tag{3.22}$$

The expected value of $T_{i\to j}$ in a complete network is as follows:

$$E(T_{comp}) = \frac{1}{p_{i\to j}} = (\frac{k}{k-1})^{n-1}. \tag{3.23}$$

Therefore, the average transmission time $E(T) = LkE(T_{comp})$, where L denotes the number of units (3 symbols) in one time slot. In our system, $L = 56$. We set the derivative to zero, then, we obtain the following:

$$\begin{aligned} \frac{\partial LkE(T_{comp})}{\partial k} &= \frac{\partial Lk(\frac{k}{k-1})^{n-1}}{\partial k} \\ &= (\frac{k}{k-1})^{n-1} - (n-1)\frac{k^{n-1}}{(k-1)^n} \\ &= (\frac{k}{k-1})^{n-1}(1-\frac{n-1}{k-1}). \end{aligned} \tag{3.24}$$

Therefore, the optimum value of k is just the number of nodes n.

3.3.2 Collision Analysis of Linear Network

In the FSA protocol, the random numbers of the sending node i's neighbor have no effect on the collision. However, the random numbers of the receiving node j's neighbors determine whether the collision occurs or not.

Figure 3.7: A 4-node linear network.

Considering the 4-node linear network shown in Fig. 3.7, we can obtain the expected number of transmission periods from node 1 to node 4

$$\begin{aligned} E(T_{1\to4}) &= E(T_{1\to2}) + E(T_{2\to3}) + E(T_{3\to4}) \\ &= \frac{1}{p_{1\to2}} + \frac{1}{p_{2\to3}} + \frac{1}{p_{3\to4}} \\ &= \frac{1}{\frac{k(k-1)^2}{k^3}} + \frac{1}{\frac{k(k-1)^2}{k^3}} + \frac{1}{\frac{k(k-1)}{k^2}} \\ &= 2\frac{k^2}{(k-1)^2} + \frac{k}{k-1}. \end{aligned} \tag{3.25}$$

For the linear network with m nodes, the expected transmission period from the start node to the end node is $E(T_{1\to m}) = \frac{(m-2)k^2}{(k-1)^2} + \frac{k}{k-1}$. The expected number of transmission time slots is $kE(T_{1\to m})$. Therefore, we find that there are only two cases in a linear network which are the nodes in the middle and the last node. Their degrees are 2 and 1, respectively. We define that the probability of a successful transmission from node i

to node j ($j \neq m$) in one period is $p_b = \frac{(k-1)^2}{k^2}$ and the probability of a successful transmission from node $m-1$ to m is $p_e = \frac{k-1}{k}$.

$$
\begin{aligned}
&p(T_{1\to m} = m-1) = p_b^{m-2} p_e \\
&p(T_{1\to m} = m) \\
=&\binom{m-2}{1}(1-p_b)p_b^{m-2}p_e + p_b^{m-2}(1-p_e)p_e \\
=&[(m-2)(1-p_b) + (1-p_e)]p_b^{m-2}p_e \\
&p(T_{1\to m} = m+1) \\
=&(\binom{m-2}{1} + \binom{m-2}{2})(1-p_b)^2 p_b^{m-2}p_e \\
&+ \binom{m-2}{1}(1-p_b)p_b^{m-2}(1-p_e)p_e + p_b^{m-2}(1-p_e)^2 p_e \\
=&[(\binom{m-2}{1} + \binom{m-2}{2})(1-p_b)^2 + \binom{m-2}{1}(1-p_b)(1-p_e) + (1-p_e)^2]p_b^{m-2}p_e.
\end{aligned}
\tag{3.26}
$$

Then we find that

$$
\begin{aligned}
&p(T_{1\to m} = m-1+n) \\
=&[\binom{m-3+n}{m-3}(1-p_b)^n + \binom{m-3+n-1}{m-3}(1-p_b)^{n-1}p_e \\
&+ \binom{m-3+n-2}{m-3}(1-p_b)^{n-2}p_e^2 + \cdots + p_e^n]p_b^{m-2}p_e.
\end{aligned}
\tag{3.27}
$$

3.3.3 Collision Analysis of 4-Node Square Network

Because there is no loop in the linear network, each transmission is independent of each other at a given time and in a given node. Here, we try to derive a simple 4-node square network with a loop.

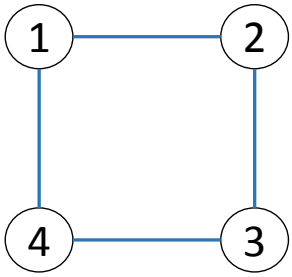

Figure 3.8: A 4-node square network.

For the network shown in Fig. 3.8, we assume that the start node is 1 and the destination node is 3. We define that

$$\begin{aligned} p = p_{1\to2} = p_{1\to3} = p_{2\to4} = p_{3\to4} = \frac{k(k-1)^2}{k^3}, \\ E(T_{1\to2}) = E(T_{1\to3}) = E(T_{2\to4}) = E(T_{3\to4}) = \frac{k^2}{(k-1)^2}. \end{aligned} \tag{3.28}$$

Then we derive the probabilities that $T_{1\to3} = 2$ and $T_{1\to3} = 3$ as follows:

$$\begin{aligned} & p(T_{1\to3} = 2) \\ =& p(T_{1\to2\to3} = 2) + p(T_{1\to4\to3} = 2) - p(T_{1\to2\to3} = 2, T_{1\to4\to3} = 2) \\ =& p^2 + p^2 - \frac{k(k-1)^3}{k^4}\frac{k(k-1)(k-2)}{k^3} \\ =& 2\frac{(k-1)^4}{k^4} - \frac{(k-1)^4(k-2)}{k^5} \end{aligned} \tag{3.29}$$

$$
\begin{aligned}
& p(T_{1\to 3}=3) \\
=& p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}\neq 2)+p(T_{1\to 4\to 3}=3, T_{1\to 2\to 3}\neq 2)-p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}=3) \\
=& p(T_{1\to 2\to 3}=3)-p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}=2)+p(T_{1\to 4\to 3}=3)-p(T_{1\to 4\to 3}=3, T_{1\to 2\to 3}=2) \\
& \qquad -p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}=3) \\
=& 2[C_2^1 p^2(1-p)-(\frac{k(k-1)^2}{k^4}p^3+\frac{k(k-1)^3}{k^4}\frac{k(k-1)}{k^3}p)]-p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}=3) \\
=& 4p^2(1-p)-2(\frac{(k-1)^8}{k^9}+\frac{(k-1)^6}{k^7})-p(T_{1\to 2\to 3}=3, T_{1\to 4\to 3}=3) \\
=& 4p^2(1-p)-2(\frac{(k-1)^8}{k^9}+\frac{(k-1)^6}{k^7}) \\
& \qquad -[\frac{kk^2+k(k-1)}{k^4}\frac{k(k-1)^3}{k^4}+2\frac{k(k-1)^2}{k^4}(1-p)p+\frac{k(k-1)^3}{k^4}\frac{k^2}{k^3}]\frac{k(k-1)(k-2)}{k^3} \\
=& 4p^2(1-p)-2(\frac{(k-1)^8}{k^9}+\frac{(k-1)^6}{k^7}) \\
& \qquad -[\frac{(k^2+k-1)(k-1)^3}{k^6}+2\frac{(2k-1)(k-1)^4}{k^7}+\frac{(k-1)^3}{k^4}]\frac{k(k-1)(k-2)}{k^3}
\end{aligned}
\tag{3.30}
$$

The exact equations of the probability of $T_{1\to 3}>3$ are very difficult to formulate, because we need to consider every possible situation. Therefore, for a general network, we need to use a computer simulation to obtain the expected number of transmission periods.

3.4 Comparison

In this section, we compare the performance of these two protocols.

3.4.1 Complete Networks

As mentioned earlier, we assume that the channel is under ideal conditions. If two nodes can communicate with each other directly, there is an edge between these two nodes in the topology of the network. We try different combinations of the parameters and choose the one with the best performance. The result is shown in Fig. 3.9. We can see that the modified CSMA/CA protocol performs better than FSA, because in the case of the FSA protocol, the use of a complete network increases the chances of a collision. The blue line is almost a straight line. This is because, using (3.25), when $k_{opt} = n$, we obtain the following:

$$\begin{aligned} \lim_{n\to\infty} E(T) &= \lim_{n\to\infty} Ln(\frac{n}{n-1})^{n-1} \\ &= \lim_{n\to\infty} Ln(1+\frac{1}{n-1})^{n-1} = Lne, \end{aligned} \tag{3.31}$$

where e denotes Euler's number.

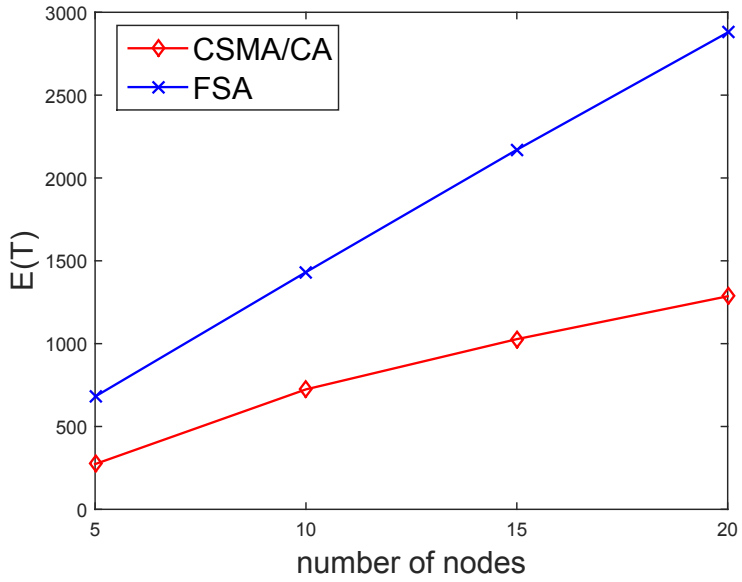

Figure 3.9: The simulation result in the complete networks with different number of nodes.

3.4.2 General Networks

To evaluate the performances in general cases, we randomly generate 20-node connected networks. The proposed protocols are sensitive to the degrees, so we select four networks with different ranges of degrees as examples to simulate. The ranges of degrees in these four networks are $degree \in [3, 6]$(low degree, the degree distribution is shown in Fig. 3.12), $degree \in [8, 14]$(medium degree, as shown in Fig. 3.13), $degree \in [12, 15]$(medium degree, as shown in Fig. **??**) and $degree \in [3, 6]$(high degree, as shown in Fig. **??**).

For these four given networks, we run our programs using the two proposed protocols. The start node and the destination node are always 1 and 20, respectively. The setting of modified CSMA/CA is similar to that used earlier. We fix $L = 56$, and use different combinations of parameters

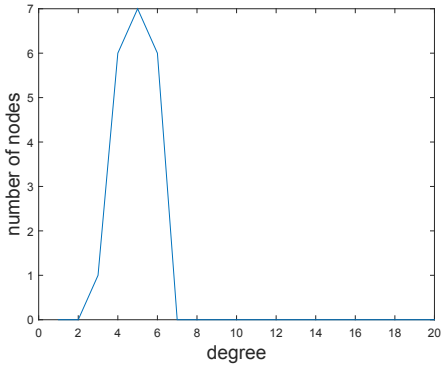

Figure 3.10: The degree distribution of a low degree example ($degree \in [3, 6]$).

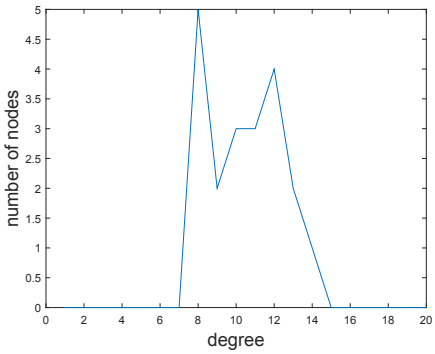

Figure 3.11: The degree distribution of a medium degree example ($degree \in [8, 14]$).

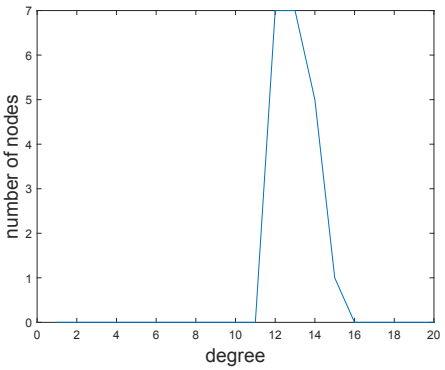

Figure 3.12: The degree distribution of a medium degree example ($degree \in [12, 15]$).

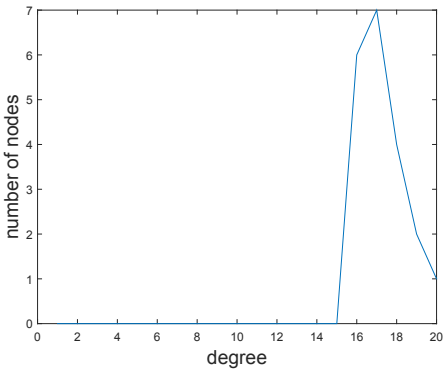

Figure 3.13: The degree distribution of a high degree example ($degree \in [16, 20]$).

W_0 and m. For each of these combinations, we run the simulation 500 times and calculate the average. We choose the smallest average transmission time as one of the results of all the combinations. For the second proposed protocol, there is only one parameter k. We use different values of k and run the simulation 10000 times for each k to get the average. The smallest average transmission time is used as one of the result.

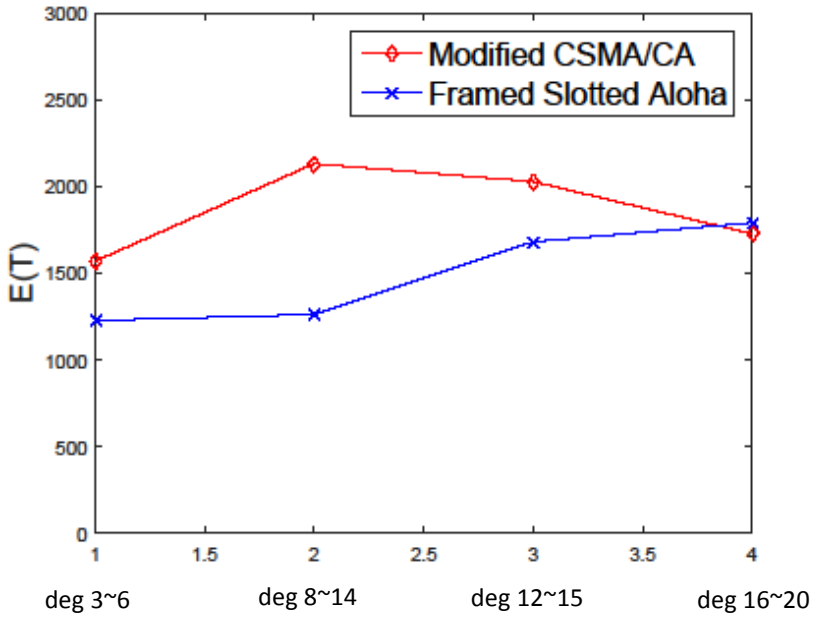

Figure 3.14: The simulation result of the four generated networks.

The results are shown in Fig. 3.14. We can see that the performance of the FSA protocol is better than that of the modified CSMA/CA protocol when the degrees are not high. Because of the randomness, the differences in the average transmission time between these two protocols are different.

From simulation results shown in Fig. 3.9 and Fig. 3.14, we consider that if the network is more crowed, the modified CSMA/CA should perform better. However, if it is not, the FSA protocol is a better choice.

3.4.3 Discussion

CSMA/CA is usually considered to be better than FSA, but our simulations show the opposite results. We consider that this is because the modified CSMA/CA suffers a lot by hidden and exposed terminal problems. In the modified CSMA/CA, the sending node needs to listen to the unrelated neighbors and does nothing with the neighbors of the receiving nodes. Therefore, each node does a lot of useless backoff waiting.

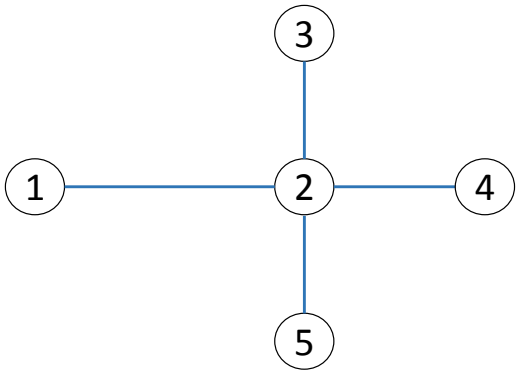

Figure 3.15: A 5-node network.

Consider the 5-node network example shown in Fig. 3.15, where the sink is connected to node 1, and node 2 is the destination. Here, node 1 wants to send a message to node 2. An example of the channel states is shown in Fig. 3.16, where $W_0 = 4$ and $m = 4$. The y-axis denotes the time slots and the x-axis represents the channel states. When the channel state is -6, this node receives more than one signal. It also means that a collision occurs in this node. If the channel state is a positive integer, the node whose ID is this integer is transmitting a message. We can find that node 1 and node 3 only have one neighbor node, node 2. If node 2 does not send a message at the beginning, these two nodes will keep sending messages.

Thus, node 2 rarely has a chance to send or receive messages. The channel in node 2 is blocked by nodes 3, 4, and 5, which are the hidden nodes of node 1. If node 2 is the sink and node 1 is the destination, the result is similar. Nodes 1, 3, 4, and 5 may keep sending messages and node 2 senses them, so the probability that node 2 can transmit a message successfully is very small. These problems are called hidden and exposed node problems. If we do not change the tag mechanism and scenario, we can increase the initial waiting time W_0 to reduce the effect of these problems, as shown in Fig. 3.17. However, this will increase the average transmission time.

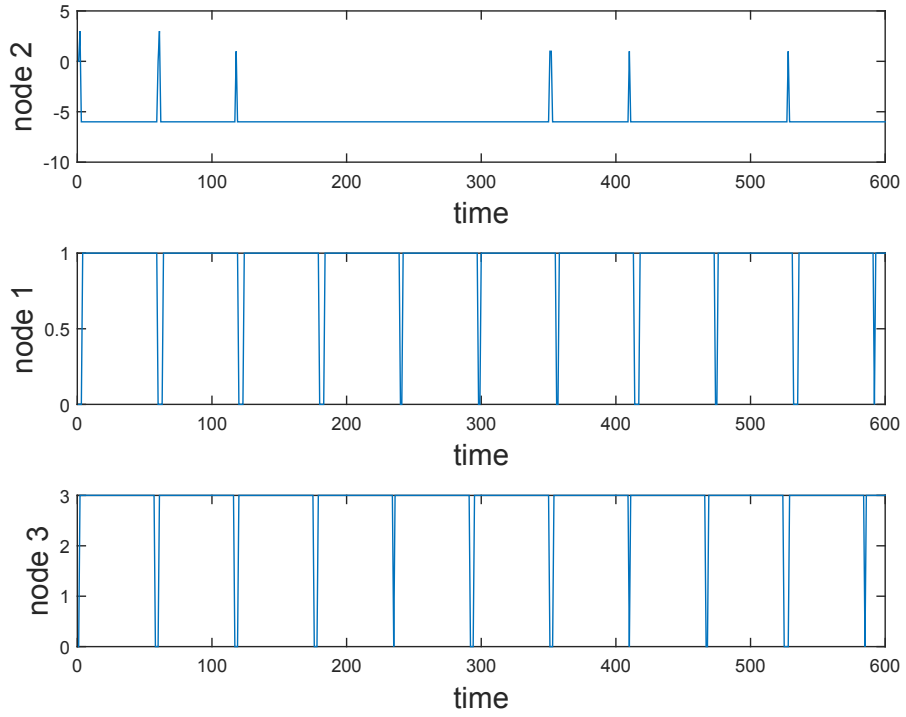

Figure 3.16: The channel states of nodes 2, 1, and 3 in the example where $W_0 = 2^2$ and $m = 3$.

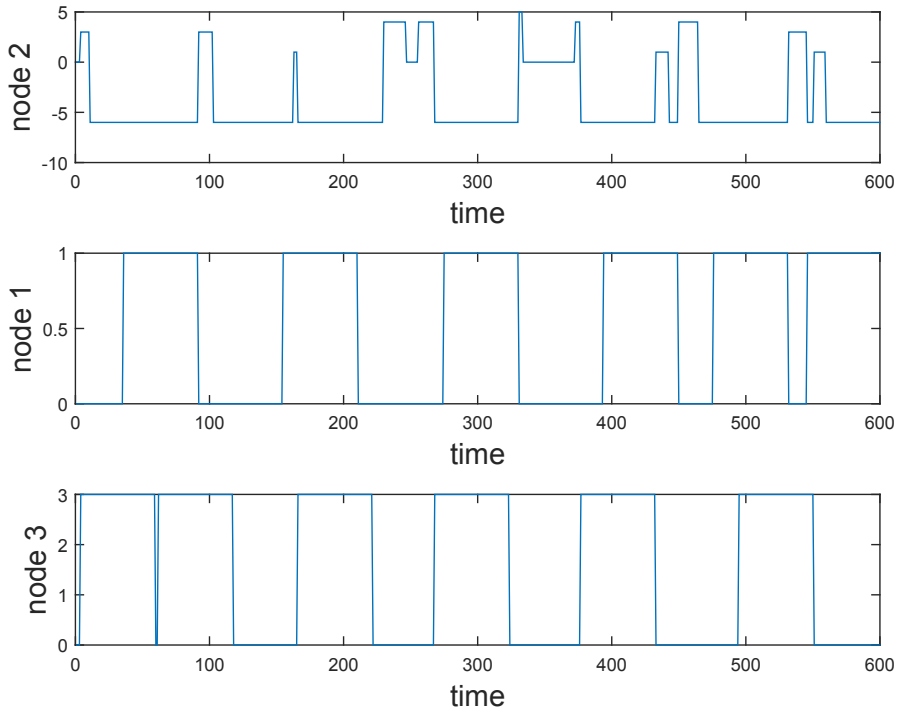

Figure 3.17: The channel states of nodes 2, 1, and 3 in the example where $W_0 = 2^6$ and $m = 3$.

3.5 Summary

In this chapter, we proposed two protocols to reduce the effects of collision in our BBTT system. We theoretically analyzed them in complete networks, and compared their performances in general networks by using computer simulations. We measured the performances by using the average transmission time from one node to another node. The results showed that if each node could communicate with most of the other nodes, such as in complete networks or in networks where each nodes had a high degree, CSMA/CA could transmit messages quickly. However, in other networks, the framed slotted Aloha protocol performed better. Furtherm in practice,

it was difficult to adjust the two parameters in the modified CSMA/CA protocol, compared with the one parameter in FSA, which had a significant influence on the performance.

Chapter 4

Distributed Bayesian Learning with Bernoulli Models

4.1 Introduction

The previous chapters talked about two problems in the BBTT system. In this chapter, we will present the distributed Bayesian learning problem which is a very popular topic. It is a problem that affects not only the BBTT system, but also the more general IoT systems. We study a network of Bayesian cooperative agents that observe Bernoulli signals with unknown parameters. Their objective is to obtain the global posterior distribution of these parameters without the presence of a fusion center in as short a time as possible. The network of the agents is connected and can be represented by an undirected graph. Every agent has its own initial private signals which are independent from the signals of the remaining agents. From these signals the agents obtain information, which they share with their neighbors. Thus, the agents learn not only from their private

signals, but also from the signals of their neighbors. In each iteration, each agent broadcasts its signal once, and receives its neighbors' signals. When the agents receive the signals from their neighbors, they store the signals if these signals contain new information, and then these agents generate new signals for their neighbors. After a finite number of iterations, each agent is guaranteed to have the complete information needed to obtain the global posterior of an unknown parameter. In other words, all the agents achieve a consensus on the posterior of a fictitious fusion center.

In the past decade or so, within the signal processing community, the problem of distributed learning has attracted considerable interest because of its applications. A more complete survey on applications is provided in [27]. Previous work on Bayesian learning in networks can be found in [28, 29, 30, 31, 32, 33]. In [33], a similar Bayesian learning method is studied, but in this model, each agent uses only the signals received in the current iteration. Other types of learning approaches are based on the consensus [27, 34, 35, 36] and the gossip methods [37, 38, 39, 40].

In [5], an efficient Bayesian learning method with Gaussian distributions was addressed. In this paper, we exploit the underlying idea from [5] to solve problems in more general settings. We use the results of our derivation on a Bernoulli model in two scenarios. In both cases, we show by simulations how the agents can reach a consensus in a finite number of iterations and much faster than consensus-based methods. In practice, this translates to less communication and energy consumption.

Consensus-based and gossip methods usually do not need to know the structure of a network. However, if the agents know the topology, for example, as in static man-made networks that are designed for long-term or

permanent use, one can apply methods for much quicker learning. In this paper, as in [5], we assume that each agent has the knowledge of the network topology. If in case the agents do not know it, they can still implement our learning method and achieve the same result as if they knew the topology. The learning, however, will require additional communication. But once the learning is completed, the coefficients can be readily used in subsequent estimation tasks.

4.2 Problem Statement

We address two problems. In the first one, the agents observe Bernoulli outcomes without errors and in the second, with errors. In the latter case, each agent knows only its own probability of error. The agents form a connected network described by an undirected graph $G = (\mathcal{N}_A, E)$, where $\mathcal{N}_A = \{1, 2, \cdots, n\}$ is the set of agents and E is the set of edges in the network. In other words, if agents A_i and A_j are connected, then the edge (i, j) represents the connection that satisfies $(i, j) \in E$. The topology of the network is time invariant and it is known to every agent. Agents only communicate with their neighbors. The communications occur in stages. In each stage, the agents broadcast their information and receive information from their neighbors.

4.2.1 Bernoulli Model without Errors

We assume that there is a state of the world $\theta \in [0, 1]$ which means $p(x = 1) = \theta$ and $p(x = 0) = 1 - \theta$, where x is the real observation. So x follows the Bernoulli distribution with the parameter θ. Suppose there

is a connected network of agents which observe $x_j \in \mathbb{R}^{k_j \times 1}$ without errors, where x_j is a vector of k_j zeros and ones. The subscript j refers to the jth agent and k_j to the length of the observed vector. We refer to these observations as private signals, and we denote them as $y_j \in \mathbb{R}^{k_j \times 1}$. Since the signal vector x_j is observed without errors, we have $y_j = x_j$.

The agents assume a prior for the probability θ, and it is a Beta density with parameters $\alpha_{j,0}$ and $\beta_{j,0}$, which is represented as $Beta(\alpha_{j,0}, \beta_{j,0})$. They use their private signals, to update this prior to the posterior $p(\theta|y_j, \alpha_{j,0}, \beta_{j,0})$, $j = 1, 2, \cdots, n$. Thus, at this stage, every agent has its own posterior. Now the agents start repeatedly exchanging information with their neighbors. After every exchange, they modify their posteriors of θ. Their objective is achieving the same posterior as that of a fictitious fusion center, i.e., the posterior $p(\theta|y_{1:n}, \alpha_{1:n,0}, \beta_{1:n,0})$. Furthermore, the objective is to reach the desired posterior in a finite number of exchanges.

4.2.2 Bernoulli Model with Errors

In the second model everything is the same as in the first one except that now the agents observe x with errors. More specifically, we model the observation y by

$$p(y = 1|x = 0) \;=\; p(y = 0|x = 1) \;=\; \epsilon \tag{4.1}$$

where the probability of error $\epsilon \in [0, 0.5)$. The model is depicted in Fig. 4.1. As pointed out earlier, each agent has in general a different probability of error, and each agent knows its own error. The agent, however, does not

know the probabilities of errors of the other agents.

The objectives are the same as stated in the previous subsection. Again, the agents have their own priors about θ, $Beta(\alpha_{j,0}, \beta_{j,0})$ and they aim at estimating $p(\theta|y_{1:n}, \alpha_{1:n,0}, \beta_{1:n,0}, \epsilon_{1:n})$.

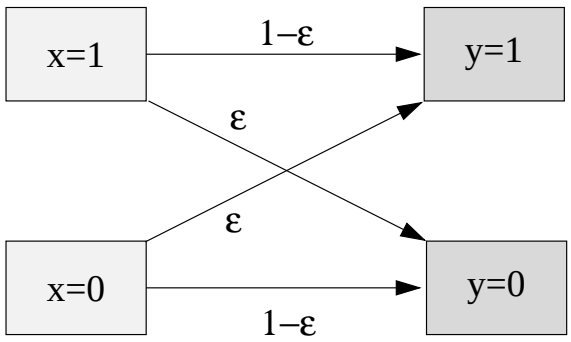

Figure 4.1: The model with observation errors.

4.3 Methods for Distributed Learning

In this chapter, we introduce three methods for distributed learning. In these methods, each agent observes its private signal and begins to exchange information with its neighbors. The first method is the average consensus method, and the second is the gosssip-based averaging method which can be viewed as a special case of the original average consensus method. The third method, referred to as sequential learning, is a Bayesian learning method where agents exchange information sequentially. With this method, sometimes a phenomenon called herding emerges, which means that each agent ignores its observation and always sends the same information.

4.3.1 Average Consensus Algorithm

In the field of distributed computing and estimation, consensus problems are important and have a long history. Consensus means all agents in the network reach an agreement regarding a certain quantity of interest that depends on the state of all agents [27]. If the sum of the state of all nodes is an invariant quantity, we call this method as average-consensus algorithm [41].

Consider an undirected graph $G = (V, E)$ with n agents, where the goal of each agent is to reach a consensus via exchanging information with its neighbors. By reaching a consensus, we mean asymptotically converging to a one-dimensional agreement space. Let $A = [a_{ij}]$ be the adjacency matrix of this graph and N_j be the set of neighbors of the agent j. From [27], an iterative form of this consensus algorithm can be written as follows:

$$x_j(k+1) = x_j(k) + \pi \sum_{i \in N_j} a_{ij}(x_i(k) - x_j(k)), \tag{4.2}$$

where $x_j(k)$ is the state of agent j at iteration k and $0 < \pi < 0.5$ is the step-size of the method. It also can be written in a vector form:

$$x(k+1) = Wx(k), \tag{4.3}$$

with $W = I - \epsilon L$, where I is the identity matrix and L is the Laplacian matrix of graph G. The following theorem provides important results about the method.

Theorem 4.1. *Consider a network of agents with topology G applying the distributed consensus algorithm* (4.2) *with parameter $0 < \epsilon < 1/\Delta$ where*

Δ is the maximum degree of the network. Let G be a strongly connected digraph. Then

i) The consensus is asymptotically reached for all initial states;

ii) The group decision value is $\alpha = \sum_i w_i x_i(0)$ with $\sum_i w_i = 1$;

iii) If the digraph is balanced (or W is doubly-stochastic), an average-consensus is asymptotically reached and $\alpha = (\sum_i x_i(0))/n$.

For the proof, see [27].

The speed of convergence of the consensus algorithm is quantified by the second smallest eigenvalue of the graph Laplacians called algebraic connectivity [42].

From (4.3), we see that each agent receives the neighbors' states and calculates the weighted average in each iteration. The weights are given by W, which is a constant matrix. There are other choices of weights, and they can be time-varying. In [43], the authors study two types of time varying weights. The first one is called Maximum-degree weights. It sets the constant weight $1/n$ to all the neighbors, and chooses the self-weights so that the sum of weights at each node is 1:

$$W_{ij}(t) = \begin{cases} \frac{1}{n} & \text{if agent } i \text{ and } j \text{ are neighbors,} \\ 1 - \frac{d_i(t)}{n} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

where $d_i(t)$ is the degree of agent i at time t. Note that, here it is assumed that the topology of the network may change.

The second type of time-varying weights is named Metropolis weights,

and it is defined as

$$W_{ij}(t) = \begin{cases} \frac{1}{1+max\{d_i(t),d_j(t)\}} & \text{if agent } i \text{ and } j \text{ are neighbors,} \\ 1 - \sum_{k \in N_i} W_{ik}(t) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

From [43], we also have the following theorem:

Theorem 4.2. *If the collection of communication graphs that occur infinitely often are jointly connected, then the iteration process* (4.3) *converges with either the maximum-degree weights or the Metropolis weights, and*

$$\lim_{t \to \infty} x(t) = (\frac{1}{n} 1^\top x(0))1,$$

for all $x(0) \in \mathbb{R}^n$.

Because the average consensus method is easy to apply, it has been broadly used in peer-to-peer and ad hoc networks where the sensor topologies may be time-varying. More result on consensus-based methods can be found in [35, 41, 44, 45, 46].

4.3.2 Gossip-based Averaging Algorithm

Gossip-based algorithms such as the push-sum protocol [37, 47] are important alternatives to the Laplacian-based average consensus algorithms in previous section. It is similar to the average consensus algorithm, but can be used in many challenging situations. In this scheme, each agent only communicates with a randomly chosen neighbor. To model this scheme, we introduce a matrix P which is described as follows: In the

kth time slot, agent i is selected, and contacts some neighboring agent j with probability P_{ij} [48]. At this time, both agents set their values equal to the average of their current values.

Here we still use the model of the previous section. Our goal is still to compute the average of all initial values in a distributed manner. The iteration form is

$$x(k) = W(k)x(k-1), \tag{4.6}$$

where from the n agents, one selects agent i and j for gossiping with probability $\frac{1}{n}P_{ij}$ (the probability that agent i is selected is $1/n$, and the probability that it contacts agent j is P_{ij}), and the random matrix $W(k)$is

$$W_{ij} = I - \frac{(e_i - e_j)(e_i - e_j)^\top}{2}, \tag{4.7}$$

where $e_i = [0 \cdots 0 \ 1 \ 0 \cdots 0]^\top$ is an $n \times 1$ vector with the ith component equal to 1.

One result about this algorithm is given in [49]:

Theorem 4.3. *For a complete graph, there exists a gossip algorithm such that the $1/n$-averaging time of the algorithm is $O(logn)$.*

In [48], it is proved that the convergence of the method is guaranteed. Also, upper and lower bounds of the convergence rate are obtained. More specifically, for the upper bound for the ϵ-averaging time, the following lemma is proved:

Lemma 4.1. *For any initial vector $x(0)$, for $k \geq K^*(\epsilon)$*

$$Pr\left(\frac{\parallel x(k) - \bar{x}1 \parallel}{\parallel x(0) \parallel} \geq \epsilon\right) \leq \epsilon,$$

where

$$K^*(\epsilon) \triangleq \frac{3\log\epsilon^{-1}}{\log\lambda_2(W)^{-1}}.$$

The lower bound for the ϵ-averaging time is provided by another lemma:

Lemma 4.2. *For the above gossip algorithm, there exists and initial vector* $x(0)$*, such that for* $k < K_*(\epsilon)$

$$Pr\left(\frac{\parallel x(k) - \bar{x}\mathbf{1} \parallel}{\parallel x(0) \parallel} \geq \epsilon\right) \geq \epsilon,$$

where

$$K^*(\epsilon) \triangleq \frac{0.5\log\epsilon^{-1}}{\log\lambda_2(W)^{-1}}.$$

4.3.3 Sequential Learning and Herding

In this section, we consider a group of agents that are rational and fully Bayesian. Each agent, based on its private observation of a noisy underlying state process, selfishly optimizes its local utility and broadcasts its action sequentially. This protocol leads to an interesting phenomenon which is called herding. Herding means that the agents eventually choose the same action irrespective of their observations.

One of the models in [31, 50, 51] is as follows:

There is a network with a finite number of agents. They take actions in a pre-determined order. They use $k = 1, 2, \cdots$ to represent the order in which agents act and also view k as the discrete time instant when agent k acts.

At time 0, we randomly chose the state of Nature $x \in \mathbb{X} = \{1, \cdots, X\}$ with probability $\pi_0 = (\pi_0(i), i \in \mathbb{X})$,$\pi_0(i) = P(x = i)$. Let $y_k \in \mathbb{Y} =$

$\{1, 2, \cdots, Y\}$ denote the private observation of agent k with the probability $P(y|x)$ and let $a_k \in \mathbb{A} = \{1, 2, \cdots, A\}$ denote the local decision that agent k takes. Define the sigma algebras:

$$\begin{aligned} \mathcal{H}_k &\quad \sigma - \text{algebras generated by } (a_1, \cdots, a_{k-1}, y_k), \\ \mathcal{G}_k &\quad \sigma - \text{algebras generated by } (a_1, \cdots, a_{k-1}, a_k). \end{aligned} \tag{4.8}$$

Here we define two posterior distributions. The first one is the public belief π_k which is the posterior of the state x given all the actions taken up to time k,

$$\pi_k = (\pi_k(i), i \in \mathbb{X}) \quad \pi_k(i) = P(x = i|a_1, \cdots, a_k). \tag{4.9}$$

The second distribution is η_j which is the private belief of agent j. This is the posterior of the state x given all the actions taken up to time k, along with the current private observation y_k at time k,

$$\eta_k = (\eta_k(i), i \in \mathbb{X}) \quad \eta_k(i) = P(x = i|a_1, \cdots, a_{k-1}, y_k). \tag{4.10}$$

We also define two diagonal matrices of conditional probabilities by

$$\begin{aligned} &B_y = diag(P(y_k = y|x = 1), \cdots, P(y_k = y|x = X)), \quad y \in \mathbb{Y}, \\ &B_y^\pi = diag(P(a_k = a|x = 1, \pi_{k-1} = \pi), \cdots, P(a_k = a|x = X, \pi_{k-1} = \pi)), \quad a \in \mathbb{A}. \end{aligned} \tag{4.11}$$

where B_y is a model parameter assumed known, and π is a X-dimensional probability vector.

Our goal is for each agent to choose the optimal action to optimize

a social welfare function (network utility). One of the sequential learning protocols in [31] is:

For $k = 1, 2, \cdots$

1) Private Belief Update: Agent k obtains a multivariate measurement $y_k \sim p(\cdot|x)$ and updates its Bayesian private belief as

$$\eta_k = T(\pi_{k-1}, y_k), \quad \text{where} \quad T(\pi, y) = \frac{B_y \pi}{\sigma(\pi, y)}, \quad \sigma(\pi, y) = 1^\top B_y \pi. \quad (4.12)$$

2) Selfish Optimization: Agent k then chooses action $a_k = \arg\min_{a \in \mathbb{A}} \{c_a^\top \eta_k\}$ to minimize its expected cost given its available information. Therefore,

$$a_k(\pi_{k-1}, y_k) = \arg\min_{a \in \mathbb{A}} \mathbb{E}\{c_a^\top \eta_k\} = \arg\min_{a \in \mathbb{A}} \{c_a^\top T(\pi_{k-1}, y_k)\}. \quad (4.13)$$

3) Recommendation Forwarding: Agent k broadcasts its action (recommendation) a_k to the other agents.

4) Social learning: Based on the action a_k in step 2, every agent (except k) updates its Bayesian social belief by

$$\pi_k = T(\pi_{k-1}, a_k), \quad \text{where} \quad T(\pi, a) = \frac{B_a^\pi \pi}{\sigma(\pi, a)}, \quad \sigma(\pi, a) = 1^\top B_a^\pi \pi. \quad (4.14)$$

From the above protocol, herding and cascades behaviors can be defined as [31, 32],

Definition 1. *1) An individual agent k herds on the public belief π_{k-1} when its action is independent of its private observation y_k. That is*

$$a_k(\pi_{k-1}, y_k) = \arg\min_{a \in \mathbb{A}} \{c_a^\top T(\pi_{k-1}, y_k)\} \quad \textit{is independent of} \quad y \in \mathbb{Y}. \quad (4.15)$$

2) A herd of agents takes place at time $\bar{k}$, if the actions of all agents after time $\bar{k}$ are identical, i.e., $a_k = a_{\bar{k}}$ for all $k > \bar{k}$

3) An information cascade takes place at time $\bar{k}$, if the social beliefs of all agents after time $\bar{k}$ are identical, i.e., $\pi_k = \pi_{\bar{k}}$ for all $k > \bar{k}$. So social learning ceases.

Based on the above protocol and definition, [31] presents a theorem that shows that information cascade occurs with probability 1. More specifically,

Theorem. *The above social learning protocol leads to an information cascade in finite time with probability 1. That is: there exists a finite time $\bar{k}$ at which social learning ceases and all individual agents herd.*

As a consequence of this theorem, agents eventually will ignore their observations and always send the same action. This is a very interesting behavior. For example, in a social network, it is as if a person always chooses to believe a gossip that may not be truthful instead of truths. More materials about herding can be found in [32, 52, 53].

4.4 Distributed Bayesian Learning with Bernoulli Models

From the previous chapter, we can see the consensus-based method is powerful and easy to apply in wireless sensor networks. In many situations, the topology of the networks changes continuously as new nodes join and old nodes leave the network. Algorithms for such networks need to be robust against changes in topology. Additionally, these agents

operate under limited computational, communication, and energy resources [48]. Because the consensus-based methods can work without the need of knowing the topology of the network and each agent applies very simply update rules, these methods work very well in such networks. In cases when the agents know the topology, for example, when some networks are designed for a long term or permanent use, these methods cannot use the topology to make the convergence quicker. Here we address this scenario by studying Bayesian learning with Bernoulli models.

4.4.1 Solution to the Problem without Errors

In chapter 2, we stated two problems. The first one was learning the probability of an event in a distributed manner where the agents observe Bernoulli outcomes. In the first problem, we assume that there is no observation error, and each agent j has k_j observations. Therefore, the posterior distribution of θ of each agent j can be written as

$$\begin{aligned} p(\theta|y_j, \alpha_{j,0}, \beta_{j,0}) &\propto p(y_j|\theta)p(\theta|\alpha_{j,0}, \beta_{j,0}) \\ &\propto \theta^{s_j} (1-\theta)^{k_j - s_j} \theta^{\alpha_{j,0}-1} (1-\theta)^{\beta_{j,0}-1} \\ &= \theta^{s_j+\alpha_{j,0}-1} (1-\theta)^{k_j - s_j + \beta_{j,0}-1}, \end{aligned} \tag{4.16}$$

where s_j and the prior are defined by $s_j = \sum_{i=1}^{k_j} y_{j,i}$ and $p(\theta|\alpha_{j,0}, \beta_{j,0}) \propto \theta^{\alpha_{j,0}-1} (1-\theta)^{\beta_{j,0}-1}$, respectively. Before communicating with others, the agents process their own data and obtain their own posteriors, which are

Beta densities with parameters

$$\alpha_j(0) = s_j + \alpha_{j,0} \tag{4.17}$$

$$\beta_j(0) = k_j - s_j + \beta_{j,0}, \tag{4.18}$$

where $j = 1, 2, \cdots, n$. In our method, $\alpha_j(0)$ and $\beta_j(0)$ are the private signals which are sent to others at the first iteration.

From (4.16), we can readily show that the posterior distribution of θ of a fictitious fusion center is given by

$$\begin{aligned} p(\theta|y_{1:n}, \alpha_{1:n,0}, \beta_{1:n,0}) &\propto \prod_{j=1}^{n} p(y_j|\theta) p(\theta|\alpha_{j,0}, \beta_{j,0}) \\ &\propto \theta^{\sum_{j=1}^{n}(s_j+\alpha_{j,0}-1)} (1-\theta)^{\sum_{j=1}^{n}(k_j-s_j+\beta_{j,0}-1)} \\ &= \theta^{\alpha_{fc}-1} (1-\theta)^{\beta_{fc}-1}, \end{aligned} \tag{4.19}$$

where

$$\alpha_{fc} = \sum_{j=1}^{n} (s_j + \alpha_{j,0}) - n + 1 \tag{4.20}$$

$$\beta_{fc} = \sum_{j=1}^{n} (k_j - s_j + \beta_{j,0}) - n + 1. \tag{4.21}$$

Therefore, the posterior of the fusion center is a Beta density with parameters given by (4.20) and (4.21). Since from (4.17) and (4.18)

$$\alpha_{fc} = \sum_{j=1}^{n} \alpha_j(0) - n + 1 \tag{4.22}$$

$$\beta_{fc} = \sum_{j=1}^{n} \beta_j(0) - n + 1, \tag{4.23}$$

it is clear that if the agents know the sums $\sum_{j=1}^{n} \alpha_j(0)$ and $\sum_{j=1}^{n} \beta_j(0)$, the

agents will be able to construct the same posterior as the fictitious fusion center.

4.4.2 The Diffusion Method

Once the agents form their initial posteriors, they begin communicating with their neighbors. Now, the goal of the agents (obtaining the two sums) is equivalent to finding the average values of all the $\alpha_j(0)$s and $\beta_j(0)$s, respectively. With the method described in the sequel, the agents exchange information in stages where at each stage t, the agent A_j broadcasts to its neighbors $\bar{\alpha}_j(t)$ and $\bar{\beta}_j(t)$, i.e, its estimates of $\sum_{j=1}^{n} \alpha_j(0)/n$ and $\sum_{j=1}^{n} \beta_j(0)/n$, respectively. This information is possibly used by the neighbors to update their estimates which they broadcast at the next stage.

In the sequel, we describe the updating process of $\bar{\alpha}_j(t)$. The process of $\bar{\beta}_j(t)$ is analogous. Let the estimate $\bar{\alpha}_j(t)$ be a linear combination of all the $\alpha_i(0)$s, that is

$$\bar{\alpha}_j(t) = \sum_{i=1}^{n} h_{j,i}(t)\alpha_i(0), \tag{4.24}$$

where $h_{j,i}(t) \geq 0, j \in \mathcal{N}_A$ and $\sum_{i=1}^{n} h_{j,i}(t) = 1$. How the agents choose the coefficients $h_{j,i}(t)$ is explained below.

We enforce that the newly generated $\bar{\alpha}_j(t+1)$ is in the space spanned by all the received signals by A_j up to t and its own $\alpha_j(0)$. So $\bar{\alpha}_j(t+1)$ is a linear combination of all the received signals and $\alpha_j(0)$. At every iteration, each agent keeps a received signal from a neighbor only if it is not in the space spanned by the set of signals present in its memory. The agent has

this information because the agents knows the topology of the network. If the signal is linearly independent from the signals in its memory, the agent adds the new signal to the memory. Thus, the memory of the agent keeps growing with time. The memory is described by a $l(t) \times n$ matrix H, whose rows are vectors of the form $h_j(t) = [h_{j,1}(t), h_{j,2}(t), \cdots, h_{j,n}(t)]$.

In the rest of this section, we focus on A_j, and therefore we will omit in the notation the subscript j. Also, for simplicity, in the rest of this section, we assume $k_j = k$, i.e., that the number of all the agents are all equal. Let $r(t) \in \mathbb{R}^{l(t)\times 1}$ be a vector of the received signals, which are linearly independent. Note that $r(1)$ is the private signal of the agent and $l(t)$ is the number of linearly independent signals available to the agent by time t.

The agent's memory can be represented by

$$r(t) = H(t)\alpha \tag{4.25}$$

where

$$\alpha \;=\; [\alpha_1(0), \cdots, \alpha_n(0)]^\top . \tag{4.26}$$

For example, at time instant $t = 1$, A_j has a signal $r(1) = \alpha_j(0)$ and a matrix $H(1) = [0, 0, \cdots, 0, 1, 0, \cdots, 0]$, where the entry one appears at the jth location.

At time t, the agent has signals in its memory. The new estimate of each agent is a linear combination of the signals from its memory, that is,

$$\bar{\alpha}(t+1) = \phi^\top(t) r(t), \tag{4.27}$$

where $\phi(t) = (\phi_1(t), \cdots, \phi_{l(t)}(t))^\top$.

The variance of $\bar{\alpha}(t+1)$ is then

$$\begin{aligned} var[\bar{\alpha}(t+1)] &= \mathbb{E}[(\phi^\top(t)r(t) - \phi^\top(t)\mathbb{E}[r(t)])^2] \\ &= \phi^\top(t)\mathbb{E}\left[(r(t) - \mathbb{E}[r(t)])(r(t) - \mathbb{E}[r(t)])^\top\right]\phi(t) \\ &= \phi^\top(t)C_r(t)\phi(t), \end{aligned} \tag{4.28}$$

where $\mathbb{E}[\cdot]$ represents the expectation operator and $var[\cdot]$ represents the variance of the random variable inside the brackets.

We obtain the covariance matrix of the memory $r(t)$ from

$$\begin{aligned} C_r(t) &= \mathbb{E}[(r(t) - \mathbb{E}[r(t)])(r(t) - \mathbb{E}[r(t)])^\top] \\ &= H(t)\mathbb{E}\left[(\alpha - \mathbb{E}(\alpha))(\alpha - \mathbb{E}(\alpha))^\top\right]H^\top(t) \\ &= H(t)C_\alpha H^\top(t) \\ &= \theta(1-\theta)kH(t)H^\top(t). \end{aligned} \tag{4.29}$$

Now, using (4.28) and (4.29), we can write

$$var[\bar{\alpha}(t+1)] = \theta(1-\theta)k\phi^\top(t)H(t)H^\top(t)\phi(t). \tag{4.30}$$

We want to choose $\phi(t)$ so that we minimize the variance $var[\bar{\alpha}(t+1)]$. Before the minimization, we set the constraint $\phi^\top(t)1_{l(t)} = 1$ so that the estimate $\bar{\alpha}(t+1)$ is unbiased. The notation $1_{l(t)} \in \mathbb{R}^{l(t)\times 1}$ signifies a vector whose entries all equal to one.

We can readily show that $\phi(t)$ is obtained from

$$\phi^\top(t) = \frac{1_{l(t)}^\top\big(H(t)H^\top(t)\big)^{-1}}{1_{l(t)}^\top\big(H(t)H^\top(t)\big)^{-1}1_{l(t)}}. \tag{4.31}$$

Once $\phi^\top(t)$ is found from (4.31), the agent computes $\bar{\alpha}(t+1)$ as in (4.27) and broadcasts the obtained value to its neighbors.

The reason for the need that each agent knows the structure of the network is that the structure determines the vector $h(t+1)$ for constructing the signal $\bar{\alpha}(t+1)$. With the knowledge of the structure, the agents calculate the $h(t+1)$s of all the other agents. Namely, from

$$\begin{aligned} \bar{\alpha}(t+1) &= \phi^\top(t) r(t) \\ &= \phi^\top(t) H(t) \alpha, \end{aligned}$$

they find that

$$h(t+1) = \phi^\top(t) H(t). \tag{4.32}$$

Therefore, when an agent receives a signal from a neighbor, it knows if there is new information in it. If there is new information, it stores it in its memory; otherwise it throws it away. At each stage, the agent computes the matrices $H(t)$ of all the agents.

When the agent does not receive new information, it does not broadcast. When all the information diffuses throughout the network, the agents have the posterior that is identical to that of the fusion center. Namely, at that stage all the agents have the same estimates $\bar{\alpha}$ and $\bar{\beta}$, which are then used in (4.22) and (4.23) to obtain the same statistics as those of the fictitious fusion center. These statistics are subsequently employed to obtain the posterior estimate of θ.

In the case when the agents do not know the topology of the network, they can still obtain the optimal posterior distribution of θ. Then, they

do not compute the matrices $H(t)$, but instead when they broadcast the estimates $\hat{\alpha}(t)$, they also broadcast the associated vectors $h(t)$.

4.4.3 The Posterior

At each iteration t, each agent j can generate a new signal $\alpha_j(t)$ which is an estimate of the average of all private signals. The agent j can obtain a new updated posterior which is a Beta density with parameters $n\alpha_j(t)$ and $n(k - \alpha_j(t))$. In other words, in each iteration, each agent estimates the parameters of the posterior of a fusion center. So if the communication is interrupted, each agent j still can obtain an updated belief.

In the above, our role is to minimize the variance of each signal. Here we use another derivation and show how the agents update their beliefs. We define the private signals as $\alpha = [\alpha_1(0), \cdots, \alpha_n(0)]$, $\beta = [\beta_1(0), \cdots, \beta_n(0)]$. Because we assume the number of observations of each agent k is the same for all agents, we have that each $\beta_j(0) = k - \alpha_j(0) + 2$.

From our first model, each private signal $\alpha_j(0)$ has a Binomial distribution. However, all the signals are independent, so we have

$$\begin{aligned}
\alpha, \beta &\sim \prod_{i=1}^{n} \binom{k}{\alpha_i(0) - 1} \theta^{\alpha_i(0)-1} (1-\theta)^{\beta_i(0)-1} \\
f_\alpha(\alpha) &\propto \theta^{\sum_{i=1}^{n} \alpha_i(0) - n} (1-\theta)^{\sum_{i=1}^{n} \beta_i(0) - n} \\
&= \theta^{\sum_{i=1}^{n} \alpha_i(0) - n} (1-\theta)^{\sum_{i=1}^{n} (k - \alpha_i(0) + 2) - n} \\
&= \theta^{\mathbb{1}_n^\top \alpha - n} (1-\theta)^{kn - \mathbb{1}_n^\top \alpha + n} \\
&= \theta^{n(\frac{k\theta \mathbb{1}_n^\top \alpha}{\mathbb{1}_n^\top E[\alpha]} - 1)} (1-\theta)^{n(k - \frac{k\theta \mathbb{1}_n^\top \alpha}{\mathbb{1}_n^\top E[\alpha]} + 1)},
\end{aligned} \tag{4.33}$$

where $\mathbb{1}_n \in \mathbb{R}^{n \times 1}$ is an all ones vector, and $f_\alpha(\alpha)$ is the probability density

function (pdf) of the vector α which is a Beta distribution with parameters $(\mathbb{1}_n^\top\alpha-n+1)$ and $(kn-\mathbb{1}_n^\top\alpha+n+1)$. We write the pdf as in (4.33), because we want each signal is unbiased and has the same expectation $k\theta$. From our method, we know $r = H\alpha$, where $H \in \mathbb{R}^{l\times n}$, $l \leq n$ and $rank(H) = l$. Because H is a full row rank matrix, the general solution of α from $r = H\alpha$ is

$$\alpha = H^\dagger r + (I - H^\dagger H)Y, \forall Y \in \mathbb{R}^n, \tag{4.34}$$

where Y can be any n dimensional vectors with real numbers, and I is the identity matrix and $H^\dagger$ is the Moore Penrose pseudoinverse matrix of H which is defined as $H^\dagger = H^\top(HH^\top)^{-1} \in \mathbb{R}^{n\times l}$. The least-norm solutions of this undetermined equations is $\alpha = H^\dagger r$. It also is the only inverse function. So from the formula of the transformation of random vectors, we have

$$\begin{aligned}
f_r(r) &\propto f_\alpha(H^\dagger r) \\
&= \theta^{n(\frac{k\theta\mathbb{1}_n^\top H^\dagger r}{\mathbb{1}_n^\top E[H^\dagger r]}-1)}(1-\theta)^{n(k-\frac{k\theta\mathbb{1}_n^\top H^\dagger r}{\mathbb{1}_n^\top E[H^\dagger r]}+1)} \\
&\propto \theta^{n(\frac{\mathbb{1}_n^\top H^\dagger r}{\mathbb{1}_n^\top H^\dagger \mathbb{1}_l}-1)}(1-\theta)^{n(k-\frac{\mathbb{1}_n^\top H^\dagger r}{\mathbb{1}_n^\top H^\dagger \mathbb{1}_l}+1)} \\
&= \theta^{n(\frac{\mathbb{1}_n^\top H^\top(HH^\top)^{-1} r}{\mathbb{1}_n^\top H^\top(HH^\top)^{-1}\mathbb{1}_l}-1)}(1-\theta)^{n(k-\frac{\mathbb{1}_n^\top H^\top(HH^\top)^{-1} r}{\mathbb{1}_n^\top H^\top(HH^\top)^{-1}\mathbb{1}_l}+1)} \\
&= \theta^{n(\frac{\mathbb{1}_l^\top (HH^\top)^{-1} r}{\mathbb{1}_l^\top (HH^\top)^{-1}\mathbb{1}_l}-1)}(1-\theta)^{n(k-\frac{\mathbb{1}_l^\top (HH^\top)^{-1} r}{\mathbb{1}_l^\top (HH^\top)^{-1}\mathbb{1}_l}+1)},
\end{aligned} \tag{4.35}$$

where $\mathbb{1}_n^\top H^\top = \mathbb{1}_l^\top$. We can see that the pdf of r is a Beta distribution.

Now let us assume that r_0 is the signal already in the memory, and

r_1 is the new received linearly independent signal. Then we can derive the new posterior as follows:

$$\begin{aligned}
&p(\theta|r_1, r_0) \\
&\propto p(r_1|\theta, r_0)p(\theta|r_0) \\
&= \frac{p(r_1, r_0|\theta)}{p(r_0|\theta)}p(\theta|r_0) \\
&\propto p(r_0, r_1|\theta) = p(r_2|\theta) = f_{r_2}(r_2|\theta) \\
&\propto \theta^{n(\frac{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} r_2}{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} \mathbb{1}_{l'}} - 1)} (1-\theta)^{n(k - \frac{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} r_2}{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} \mathbb{1}_{l'}} + 1)},
\end{aligned} \tag{4.36}$$

where $H_2 = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix} \in \mathbb{R}^{l' \times n}$, $r_2 = \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \in \mathbb{R}^{l' \times 1}$ and the prior is uniform distribution. From our proposed method, the new signal is the estimate of the average of all private signals, which is $\bar{\alpha}_j(t) = \frac{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} r_2}{\mathbb{1}_{l'}^\top (H_2 H_2^\top)^{-1} \mathbb{1}_{l'}}$. From (4.22) and (4.23), we see that the posterior is a Beta density with two parameters $(n\bar{\alpha}_j(t) - n + 1)$ and $(nk - n\bar{\alpha}_j(t) + n + 1)$. Clearly, this posterior is the same as (4.36) which is updated by the Bayesian theorem.

In summary, the proposed algorithm can be written as follows:

4.4.4 An Example

We use a simple example to explain how this diffusion method works. The topology of this network is shown in Figure 4.2. We assume that there is no observation error and that each agent j has a different k_j.

Algorithm 1 the process of our method in each agent j

for $j = [1 : n]$ **do**
  observe and generate a private signal $\alpha_j(0)$;
end for
while $t = [2 :$time limit$]$ or all $\alpha(t)$s does change **do**
  for $j = [1 : n]$ **do**
    receive neighbors' signals;
    add linearly independent signals to memory and obtain new memory $r_{j,\alpha}(t)$, $r_{j,\beta}(t)$;
    add corresponding coefficients vectors of added signals and obtain new coefficients matrix $H_j(t)$;
    bring $H_j(t)$ to (4.31) to get $d_j(t)$;
    obtain a new signal $\hat{\alpha}_j(t+1)$ using (4.27);
    do step 7 to 10 except the parts of $r_{j,\alpha}(t)$ and $r_{j,\beta}(t)$ for all other agents, and obtain all $H_j(t)$s and $d_j(t)$s in the network;
  end for
end while

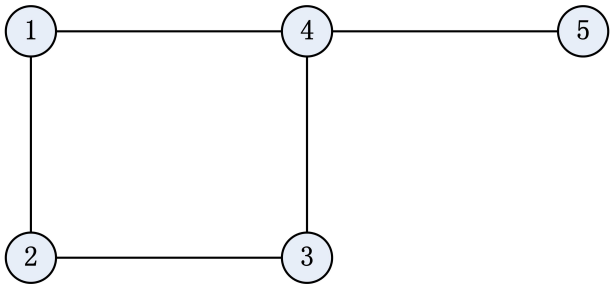

Figure 4.2: Network topology

Time Instant 0

Before the exchanges start, each agent j, $j \in [1, 5]$, has k_j private observations. So the posterior of agent j is

$$\begin{aligned} p(\theta|y_{j,1}, \cdots, y_{j,k_j}) &= \frac{\theta^{\sum_{i=1}^{k_j} y_{j,i}}(1-\theta)^{k_j - \sum_{i=1}^{k_j} y_{j,i}} p(\theta)}{\int_0^1 \theta^{\sum_{i=1}^{k_j} y_{j,i}}(1-\theta)^{k_j - \sum_{i=1}^{k_j} y_{j,i}} p(\theta) d\theta} \\ &= \frac{\theta^{\alpha_j(0)-1}(1-\theta)^{\beta_j(0)-1} p(\theta)}{\int_0^1 \theta^{\alpha_j(0)-1}(1-\theta)^{\beta_j(0)-1} p(\theta) d\theta} \\ &= Beta(\alpha_j(0), \beta_j(0)), \end{aligned} \tag{4.37}$$

agent	1	2	3	4	5
number of 0	5	3	12	24	6
number of 1	8	6	20	36	5
$\alpha_j(0)$	6	4	13	25	7
$\beta_j(0)$	9	7	21	37	6

Table 4.1: The initial states at time instant 0

where $Beta(\cdot)$ is the beta pdf, and $\beta_j(0) = k_j - \alpha_j(0) + 2$.

Here we choose $\alpha_j(0)$ and $\beta_j(0)$ as private signals (also we can choose $\alpha_j(0)$ and $k_j(0)$). The private signals $\alpha_j(0)$ and $\beta_j(0)$ are the estimates of the average values $\bar{\alpha}$ and $\bar{\beta}$. Also, they are the sufficient statistics. In this example, the initial conditions and private signals are shown in Table 4.1.

Time Instant 1

At $t = 1$, all the agents begin to exchange signals. Each agent j sends $\alpha_j(0)$ and $\beta_j(0)$ to its neighbors. Because the diffusion process of β is the same as α, we only use the process of α to explain the diffusion. Because all the $\alpha_j(0)$s are independent, each memory keeps all the received signals. The received signals and the memories are shown in Table 4.2, where $H(1)$ is from (4.25). From their own memories, the agents can use (4.29) and (4.31) to calculate the coefficients for each signal in the memories. Because all the received signals are independent, the covariances of the memories are all diagonal matrices. So we can have the new estimates of all the

agent	received signals	H(1)
1	$\alpha_1(0)$,$\alpha_2(0)$,$\alpha_4(0)$	$\begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&1&0\end{matrix}$
2	$\alpha_1(0)$,$\alpha_2(0)$,$\alpha_3(0)$	$\begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\end{matrix}$
3	$\alpha_2(0)$,$\alpha_3(0)$,$\alpha_4(0)$	$\begin{matrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{matrix}$
4	$\alpha_1(0)$,$\alpha_3(0)$,$\alpha_4(0)$,$\alpha_5(0)$	$\begin{matrix}1&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{matrix}$
5	$\alpha_4(0)$,$\alpha_5(0)$	$\begin{matrix}0&0&0&1&0\\0&0&0&0&1\end{matrix}$

Table 4.2: the memory of each agent at time instant1

agents, as follows:

$$\begin{pmatrix}\alpha_1(1)\\ \alpha_2(1)\\ \alpha_3(1)\\ \alpha_4(1)\\ \alpha_5(1)\end{pmatrix} = \begin{pmatrix}1/3 & 1/3 & 0 & 1/3 & 0\\ 1/3 & 1/3 & 1/3 & 0 & 0\\ 0 & 1/3 & 1/3 & 1/3 & 0\\ 1/4 & 0 & 1/4 & 1/4 & 1/4\\ 0 & 0 & 0 & 1/2 & 1/2\end{pmatrix} \begin{pmatrix}\alpha_1(0)\\ \alpha_2(0)\\ \alpha_3(0)\\ \alpha_4(0)\\ \alpha_5(0)\end{pmatrix}. \tag{4.38}$$

Time Instant 2

At $t = 2$, each agent still keeps the signal in its memory and sends the new estimate (4.38) to its neighbors. The new memory is presented in Table 4.3. We can see the agent 2 does not add $\alpha_3(1)$ to its memory, because it is a linear combination of the signals in agent's 2 memory. In other words, it is not a new information.

Now we focus on agent 2. From (4.29), the covariance is

agent	memories	H(2)
1	$\alpha_{1,2,4}(0)$,$\alpha_{2,4}(1)$	1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1/3 1/3 1/3 0 0 1/4 0 1/4 1/4 1/4
2	$\alpha_{1,2,3}(0)$,$\alpha_{1}(1)$	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1/3 1/3 0 1/3 0
3	$\alpha_{2,3,4}(0)$,$\alpha_{2,4}(1)$	0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1/3 1/3 1/3 0 0 1/4 0 1/4 1/4 1/4
4	$\alpha_{1,3,4,5}(0)$,$\alpha_{1}(1)$	1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1/3 1/3 0 1/3 0
5	$\alpha_{4,5}(0)$,$\alpha_{4}(1)$	0 0 0 1 0 0 0 0 0 1 1/4 0 1/4 1/4 1/4

Table 4.3: the memory of each agent at time 2

$$C_{r_2}(2) = H_2(2)C_\alpha H_2^\top(2) = \theta(1-\theta)kH_2(2)H_2^\top(2)$$
$$= \theta(1-\theta)k \begin{pmatrix} 1 & 0 & 0 & 1/3 \\ 0 & 1 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \end{pmatrix}. \tag{4.39}$$

Then we use (4.31) to calculate the coefficients for each signal in the memory.

$$\phi_2^\top(2) = \frac{\mathbb{1}^\top C_r^{-1}(2)}{\mathbb{1}^\top C_r^{-1}(2)\mathbb{1}} = \begin{pmatrix} 0 & 0 & 1/4 & 3/4 \end{pmatrix}. \tag{4.40}$$

So the new signal of agent 2 at $t = 2$ is

$$\alpha_2(2) = \begin{pmatrix} 0 & 0 & 1/4 & 3/4 \end{pmatrix} \begin{pmatrix} \alpha_1(0) \\ \alpha_2(0) \\ \alpha_3(0) \\ \alpha_1(1) \end{pmatrix}$$
$$= \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1(0) \\ \alpha_2(0) \\ \alpha_3(0) \\ \alpha_4(0) \\ \alpha_5(0) \end{pmatrix}. \tag{4.41}$$

The processes of all the other agents are the same, and the results are shown in (4.42). We can see that the signals of agents 1, 3 and 4 have

reached the MMSE (minimum mean square error) estimates which is the average value of all the private signals, but agent 2 and 5 have not. This makes sense, because the distance between them is the longest.

$$\begin{pmatrix} \alpha_1(2) \\ \alpha_2(2) \\ \alpha_3(2) \\ \alpha_4(2) \\ \alpha_5(2) \end{pmatrix} = \begin{pmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/4 & 0 & 1/4 & 1/4 & 1/4 \end{pmatrix} \begin{pmatrix} \alpha_1(0) \\ \alpha_2(0) \\ \alpha_3(0) \\ \alpha_4(0) \\ \alpha_5(0) \end{pmatrix}. \tag{4.42}$$

Time Instant 3

At $t = 3$, agents 1, 3 and 4 send their new signals which are the MMSE estimates. So agent 2 can directly use the received signal from agent 1 or 3 and agent 5 can use the signal form agent 4. We use the same method for β. Then all the agents reach the MMSE estimates of $\bar{\alpha}$ and $\bar{\beta}$. The posteriors of all the agents are the same as that of a fusion center.

Also we show how the posterior of θ of agent 2 changes in each time step in Figure 4.3.

4.4.5 Convergence Proofs

In this section, we prove that the signal of each agent converges to the average of all private signals. Here we assume that the network is a connected and undirected graph, and still use the same notations and conditions as above.

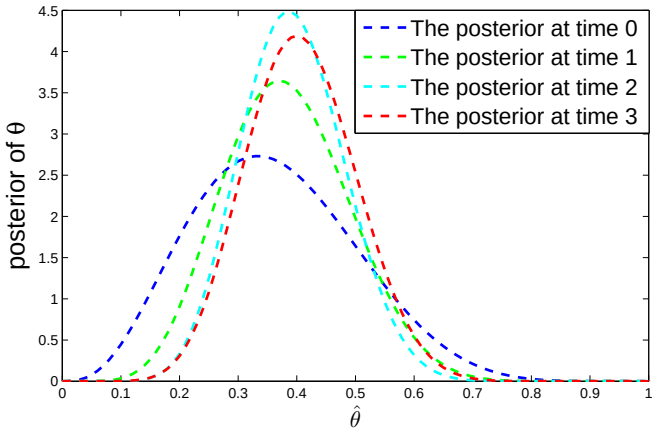

Figure 4.3: The posterior of θ of agent 2 at time 0

4.4.5.1 Second Moment Convergence

Lemma 4.3. *Using our proposed diffusion method and definitions, for each agent j, the variances of the signals of this agent will not increase with t, i.e., $var(\alpha_j(t+1)) \leq var(\alpha_j(t))$.*

Proof. From the previous section, we can obtain $\alpha_j(t) = \phi_j^\top(t-1)r_j(t-1) = \sum_{i=1}^{l_j(t-1)} \phi_{j,i}(t-1)r_{j,i}(t-1)$ and $\alpha_j(t+1) = \phi_j^\top(t)r_j(t) = \sum_{i=1}^{l_j(t)} \phi_{j,i}(t)r_{j,i}(t)$, where $l_j(t)$ is the number of signals at time t and $l_j(t) \geq l_j(t-1)$. Clearly, the first $l_j(t-1)$ elements of $r_j(t)$ are exactly the same as $r_j(t-1)$. So if we set the first $l_j(t-1)$ elements of $\phi_j(t)$ to be equal to $\phi_j(t-1)$ and the rest to be 0s, we have $\alpha_j(t+1) = \alpha_j(t)$. Because $\alpha_j(t+1)$ has the minimum variance than all other linear combinations of $r_{j,i}, i \in [1, l_j(t)]$ which include the combination to generate $\alpha_j(t)$. So $Var(\alpha_j(t+1)) \leq Var(\alpha_j(t))$. □

Lemma 4.4. *Using the proposed method, the variances of the signals of all the agents converge to γ.*

Proof. From Lemma 4.3, we know that the variance of each agent j's estimator is monotonously decreasing. And because the variances should be

greater or equal to 0, each variance must converge to a positive real number γ_j. For agent i which is the neighbor of agent j, we assume the variance will converge to γ_i. Without loss of generality, we assume $\gamma_j < \gamma_i$. So there must exits an iteration time t so that $\gamma_j < var(\alpha_j(t)) < \gamma_i < var(\alpha_i(t))$.

At time t, agent i can receive a signal $\alpha_j(t)$ from agent j. If it is not in the space spanned by agent i's memory $r_i(t)$, agent i adds it to its memory; otherwise agent i just ignores it. No matter in which situation, $\alpha_j(t)$ becomes a signal in the space spanned by agent i's memory. Namely, $\alpha_j(t)$ does not contain new information for agent i, and agent i can obtain $\alpha_j(t)$ by setting appropriate coefficients at time t. From our proposed method, we know that $var(\alpha_i(t+1))$ is the lowest variance all over this space which include all possible combinations of $r_i(t)$, so we can obtain $var(\alpha_i(t+1)) \leq var(\alpha_j(t)) < \gamma_i$. It's in contradiction with previous assumption. Therefore, γ_j must be equal to γ_i. And because the topology of this network is a connected graph, all agents will converge to a same variance γ. □

4.4.5.2 First Moment Convergence

Lemma 4.5. *Given the conditions from 4.4, the covariance of the signals of the connected agents converges to γ [33].*

Proof. From Lemma 2, we assume that all the variances of actions $\alpha_i(t)$s will converge to γ. So for any two connected agents i and j, there exists an iteration t where $var(\alpha_i(t)) < \gamma + \epsilon$ and $var(\alpha_j(t)) < \gamma + \epsilon$, and where ϵ can be any real number.

Because $\frac{1}{2}(\alpha_i(t)+\alpha_j(t))$ is in the space spanned by agent i's memory at iteration t, $Var(\alpha_i(t+1)) \leq Var(\frac{1}{2}(\alpha_i(t)+\alpha_j(t)))$. Also,

$$\begin{aligned} Var(\frac{1}{2}(\alpha_i(t)+\alpha_j(t))) &= \frac{1}{4}Var(\alpha_i(t)) + \frac{1}{4}Var(\alpha_j(t)) + \frac{1}{4}2Cov(\alpha_i(t),\alpha_j(t)) \\ &< \frac{1}{2}(\gamma+\epsilon+Cov(\alpha_i(t),\alpha_j(t))) \end{aligned} \tag{4.43}$$

So we have $\gamma < \frac{1}{2}(\gamma+\epsilon+Cov(\alpha_i(t),\alpha_j(t)))$ which can be represented to $Cov(\alpha_i(t),\alpha_j(t)) > \gamma-\epsilon$. Otherwise, from Cauchy−Schwarz inequality, we have

$$Cov(\alpha_i(t),\alpha_j(t)) \leq \sqrt{Var(\alpha_i(t))Var(\alpha_j(t))} < \gamma+\epsilon. \tag{4.44}$$

Since $\gamma-\epsilon < Cov(\alpha_i(t),\alpha_j(t)) < \gamma+\epsilon$ and ϵ can be any real number, we have $Cov(\alpha_i(t),\alpha_j(t)) = \gamma$. Therefore, if agents i and j are neighbors, the covariance of the two actions converges to γ. □

Theorem 4.4. *All signals will converge to the same value.*

Proof. From Lemma 4.5, we can see correlation coefficient will be 1. It means signals will be linear dependent. Because all signals $\alpha(t) = \sum_{i=1}^{n} h_i(t)\alpha_i(0), \sum_{i=1}^{n} h_i(t) = 1$ for any agents and at any iteration, all signals will converge to the same signal. □

4.4.5.3 Convergence to the Average Value

Given the conditions from above, we know that all $\alpha_i(0)$s are independent and have the same variance. Each signal is a linear combination: $\alpha(t) = \sum_{i=1}^{n} h_i(t)\alpha_i(0)$ with a constrain $\sum_{i=1}^{n} h_i(t) = 1$.

Using the method of Lagrange multipliers, we can know that the average of all $\alpha_i(0)$s has the lowest variance, which is $\frac{var[\alpha_i(0)]}{n}$, where n is the number of agents.

Lemma 4.6. *For each agent j, if $h_j(t) < \frac{1}{n}$ which is the coefficient of agent j's private signal $\alpha_j(0)$ in a signal $\alpha_j(t)$, then agent j can adjust $h_j(t) = \frac{1}{n}$ to obtain a new signal which has a lower variance(where n is the number of agents, and we assume $n \geq 2$).*

Proof. For simplifying the notation, we omit the time argument t and subscript j, and assume the variances of all the private signals to be equal to 1. For each agent j, $\alpha = h_j\alpha_j(0) + \sum_{i=1\&i\neq j}^{n} h_i\alpha_i(0)$. Because the agent j has its private signal $\alpha_j(0)$, it can modify α to $\alpha' = \frac{1}{n}\alpha_j(0) + \frac{n-1}{n}\frac{\sum_{i=1\&i\neq j}^{n} h_i\alpha_i(0)}{\sum_{i=1\&i\neq j}^{n} h_i}$. We have $var(\alpha) = \sum_{i=1}^{n} h_i^2$ and $var(\alpha') = \frac{1}{n^2} + \frac{(n-1)^2}{n^2(\sum_{i=1\&i\neq j}^{n} h_i)^2}\sum_{i=1\&i\neq j}^{n} h_i^2$. Because $\sum_{i=1}^{n} h_i = 1$, we have $\sum_{i=1\&i\neq j}^{n} h_i = 1 - h_j$. Then we have

$$
\begin{aligned}
var[\alpha'] &= \frac{1}{n^2} + \frac{(n-1)^2}{n^2(1-h_j)^2}\sum_{i=1\&i\neq j}^{n} h_i^2 \\
&= \frac{1}{n^2(1-h_j)^2}[(1-h_j)^2 + (n-1)^2\sum_{i=1\&i\neq j}^{n} h_i^2] \\
&= \frac{1}{n^2(1-h_j)^2}[(n-1)^2\sum_{i=1}^{n} h_i^2 + (1-h_j)^2 - (n-1)^2h_j^2] \\
&= \frac{1}{n^2(1-h_j)^2}[(n-1)^2\sum_{i=1}^{n} h_i^2 + (nh_j - 2h_j + 1)(1-nh_j)]
\end{aligned}
\tag{4.45}
$$

We want to prove $var[\alpha'] < var[\alpha]$ which can be represented as

$$\begin{aligned}
&var[\alpha'] < var[\alpha] \\
&\frac{1}{n^2(1-h_j)^2}[(n-1)^2\sum_{i=1}^{n} h_i^2 + (nh_j - 2h_j + 1)(1-nh_j)] < var[\alpha] \\
&(n-1)^2 var[\alpha] + (nh_j - 2h_j + 1)(1-nh_j) < n^2(1-h_j)^2 var[\alpha] \\
&(nh_j - 2h_j + 1)(1-nh_j) < [n^2(1-h_j)^2 - (n-1)^2] var[\alpha] \\
&(nh_j - 2h_j + 1)(1-nh_j) < (1-nh_j)(2n - nh_j - 1) var[\alpha]
\end{aligned} \tag{4.46}$$

Because we know $h_j < \frac{1}{n}$ and $n \geq 2$, we can obtain $1 - nh_j > 0$, $(n-2)h_j + 1 > 0$ and $(n(2-h_j) - 1) > 0$. Continuing (4.46), we want to prove

$$\begin{aligned}
&(nh_j - 2h_j + 1) < (2n - nh_j - 1) var[\alpha] \\
&var[\alpha] > \frac{(n-2)h_j + 1}{n(2-h_j) - 1}
\end{aligned} \tag{4.47}$$

We assume that the variances of all the private signals are equal to 1 and that $var[\alpha] > var[\bar{\alpha}] = \frac{1}{n}$, and thus we just need to prove

$$\begin{aligned}
&\frac{1}{n} > \frac{(n-2)h_j + 1}{n(2-h_j) - 1} \\
&2n - nh_j - 1 > n^2 h_j - 2nh_j + n \\
&n - 1 > n^2 h_j - nh_j \\
&h_j < \frac{n-1}{n(n-1)} = \frac{1}{n}
\end{aligned} \tag{4.48}$$

Because $h_j < \frac{1}{n}$ is our assumption, we can adjust $h_j = \frac{1}{n}$ to obtain a new signal which has a lower variance. □

Theorem 4.5. *With the proposed method, the signals of all the agents will converge to the average* $\bar{\alpha} = \sum_{i=1}^{n} \frac{1}{n}\alpha_i(0)$*, if each* $\alpha_j(0)$ *is independent and has the same variance, where the number of agents* $n > 2$*.*

Proof. According to Lemma 4.6 and Theorem 4.4, we assume that all signals will converge to $\alpha(t) = \sum_{i=1}^{n} h_j\alpha_j(0)$ $(\alpha(t) \neq \bar{\alpha})$ with the smallest variance γ. So $var[\bar{\alpha}] < var[\alpha(t)]$. Because $\sum_{i=1}^{n} h_i = 1$, there exists a coefficient $h_i < \frac{1}{n}$. From Lemma 4, agent i can modify h_i to $\frac{1}{n}$ to obtain a new signal with a smaller variance. This is in contradiction with the former assumption. So when the signals of all the agents converge, all the coefficients of this converged signal should not be less than $\frac{1}{n}$. Also $\sum_{i=1}^{n} h_i = 1$, so the signals of all the agents will converge to the average $\bar{\alpha} = \sum_{i=1}^{n} \frac{1}{n}\alpha_j(0)$. □

4.4.5.4 The Upper and Lower Bounds

Two theorems about the upper bound on the convergence are given on [5].

Theorem 4.6. *If some agents estimator has not changed after* $2d$ *iterations, then the process has converged.*

Theorem 4.7. *The process stops after* $2n \cdot d$ *iterations.*

where n is the number of agents and d is the diameter of the network.

For an agent which is in the center of the network, the process can converge to the average within $2n\lceil\frac{d}{2}\rceil$ steps, where $\lceil\frac{d}{2}\rceil = \frac{d}{2}$ when d is a even number or $\lceil\frac{d}{2}\rceil = \frac{d+1}{2}$ when d is an odd number. It is because the memory determines the signal. If the memory does not change, then the signal will not change. Therefore every time an agents signal changes (in most $2\lceil\frac{d}{2}\rceil$

iterations), the dimension of the space the agents memory spans increases by at least one. Clearly this cannot happen more than n times.

After the center agent reaches the optimum signal, this signal takes $\lceil \frac{d}{2} \rceil$ steps to be transmitted to all others. So all the agents can converge to the average in at most $(2n+1)\lceil \frac{d}{2} \rceil$ iterations.

For the lower bound, because it takes d steps to propagate information through the network, the convergence cannot happen faster than d.

4.4.6 Solution to the Problem with Errors

The Bernoulli model with errors(4.1) is more difficult for processing than the model from the previous section. Here, we first use the moment matching method to approximate the local posterior, and then use a similar diffusion method as in Section 4.4.1. The difference is that the variances of all the private signals are not the same, and so each agent needs two signals in each iteration. For the posterior of the fusion center we can write

$$p(\theta|y_{1:n}, \alpha_{1:n,0}, \beta_{1:n,0}, \epsilon_{1:n}) = \prod_{j=1}^{n} p(y_j|\theta, \epsilon_j) p(\theta|\alpha_{j,0}, \beta_{j,0}) \tag{4.49}$$

where $p(y_j|\theta, \epsilon_j)$ is the likelihood of the jth agent and $p_j(\theta)$ is its prior. For the local likelihood of each agent j we write

$$\begin{aligned} p(y_j|\theta, \epsilon_j) &= \prod_{i=1}^{k_j} p(y_{j,i}|x_{j,i}=1, \epsilon_j) p(x_{j,i}=1|\theta) + p(y_{j,i}|x_{j,i}=0, \epsilon_j) p(x_{j,i}=0|\theta) \\ &= \prod_{i=1}^{k_j} \epsilon_j^{1-y_{j,i}} (1-\epsilon_j)^{y_{j,i}} \theta + \epsilon_j^{y_{j,i}} (1-\epsilon_j)^{1-y_{j,i}} (1-\theta). \end{aligned} \tag{4.50}$$

Then we can write the local posterior of each agent j as

$$\begin{aligned}
&p(\theta|y_j, \alpha_{j,0}, \beta_{j,0}, \epsilon_j) \\
&\propto \prod_{i=1}^{k_j} [\epsilon_j^{1-y_{j,i}}(1-\epsilon_j)^{y_{j,i}}\theta + \epsilon_j^{y_{j,i}}(1-\epsilon_j)^{1-y_{j,i}}(1-\theta)]\theta^{\alpha_{j,0}-1}(1-\theta)^{\beta_{j,0}-1} \\
&= C_1\theta^{k_j+\alpha_{j,0}-1}(1-\theta)^{\beta_{j,0}-1} + ... + C_{k_j+1}\theta^{\alpha_{j,0}-1}(1-\theta)^{k_j+\beta_{j,0}-1},
\end{aligned} \tag{4.51}$$

where $C_i, i \in [1, k_j + 1]$ are coefficients which are determined by the observations y_j and the error ϵ_j. Clearly, the local posterior is a mixture of Beta densities with $k_j + 1$ components. We can assume that the prior of each agent is the same uniform distribution: $p(\theta) = 1, \theta \in [0, 1]$. Then the local likelihood is the same as the local posterior. Therefore, (4.49) and (4.51) imply that the posterior can be written as a complicated mixture of Beta densities with many components. Because the posterior of the fusion center is very complicated, we use a method to approximate the local posterior by one Beta density.

4.4.6.1 Moment Matching

We use the same mechanism for learning as in the previous section. This can only be accomplished with approximations. We propose that the agents approximate the mixtures with a single Beta density by the moment matching method.

We assume that the Beta density approximates local belief, i.e. $q(\theta) = \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$. The expectation and variance of a Beta random variable are $E_q[\theta] = \frac{\alpha}{\alpha+\beta}$ and $var_q[\theta] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$, respectively.

From (4.50), the expectation of the local belief is

$$\begin{aligned} E_p[\theta] &= \int_0^1 \theta p(\theta|y_1, ..., y_k, \epsilon) d\theta \\ &= \sum_{i=1}^{k+1} C_i \int_0^1 \theta \cdot \theta^{\alpha_0 + k_j - i} (1-\theta)^{\beta_0 + i - 2} d\theta \\ &= \sum_{i=1}^{k+1} C_i Be(\alpha_0 + k - i + 2, \beta_0 + i - 1). \end{aligned} \tag{4.52}$$

For the variance of the local belief, we have

$$\begin{aligned} var_p[\theta] &= E_p[\theta^2] - E_p^2[\theta] \\ &= \int_0^1 \theta^2 p(\theta|y_{j,1}, ..., y_{j,k_j}, \epsilon_j) d\theta - E_p^2[\theta] \\ &= \sum_{i=1}^{k+1} C_i Be(\alpha_0 + k - i + 3, \beta_0 + i - 1) - E_p^2[\theta]. \end{aligned} \tag{4.53}$$

According to the moment matching method, we want the expectations of the local belief and $q(\theta)$ be the same, i.e., $E_q[\theta] = E_p[\theta]$ and $var_q[\theta] = var_p[\theta]$. So we obtain

$$\begin{cases} \alpha_j(0) = \frac{E_p^2[\theta](1 - E_p[\theta])}{var_p[\theta]} - E_p[\theta] \\ \beta_j(0) = \frac{\alpha}{E_p[\theta]} - \alpha, \end{cases} \tag{4.54}$$

where $E_p[\theta]$ and $var_p[\theta]$ are expectation and variance of the real local belief (4.50). Thus, we can use one Beta density with parameter $\alpha_j(0)$ and $\beta_j(0)$ to represent the local posterior.

4.4.6.2 The Diffusion Method

The difference with the previous by proposed method is that the $k_j(0)$s of all the agents are different. Also, $k_j(0)$ is not the number of observation of agent j anymore, and instead, it is just equal to $\alpha_j(0) + \beta_j(0) - 2$. So we assume k_j is a random valuable, where all the k_js have the same expectation and variance. Therefore the expectations and the variances of all $\alpha_j(0)$s are also the same.

We use two entities, $\alpha_j(0)$ and $k_j(0)$, to play the roles of $\alpha_j(0)$ from the previous section simultaneously, and in each iteration, the agents exchange the two estimates of the averages $\bar{\alpha} = \sum_{j=1}^{n} \alpha_j(0)$ and $\bar{k_j} = \sum_{j=1}^{n} k_j(0)$. Each agent receives and sends two signals and maintains two memories $r_\alpha(t)$ and $r_k(t)$. However, the matrix $H(t)$ is the same for these two signals, because it is only determined by the topology of the network. Each two signals are unbiased estimates of the averages of the private signals. From the proposed method, each agent can finally obtain $\bar{\alpha}$ and $\bar{k}$, which are defined by

$$\bar{\alpha} = \frac{1}{n} \sum_{j=1}^{n} \alpha_j(0), \qquad \bar{k} = \frac{1}{n} \sum_{j=1}^{n} k_j(0). \tag{4.55}$$

With them, each agent can have the same posterior as that of a fusion center which is a Beta density with parameters $(n\bar{\alpha} - n + 1)$ and $(n\bar{k} - n\bar{\alpha} + n + 1)$.

We point that this diffusion method can be used in any problem whose goal is to obtain the average value of private signals. We just need to assume that all the private signals have the same variance.

4.4.7 Discussion

First we comment on the use of one Beta density to approximate the mixture Beta density. From (4.51), we can see that the sum of two parameters of each component of the mixture Beta density is the same and equal to $k_j + \alpha_{j,0} + \beta_{j,0} - 2$. The modes of the components ($\frac{\alpha-1}{\alpha+\beta-2}$) have the same interval 1. So for given k_j, $\alpha_{j,0}$ and $\beta_{j,0}$, if we ignore the coefficients, all the components are close to each other. Only the coefficients determine the shape of the local belief. From Figure 4.4, we can see that the shape is like a "bell curve". We note that the mode is determined by several largest coefficients. Therefore we want to use one Beta density to approximate the real local belief which is a mixture Beta density.

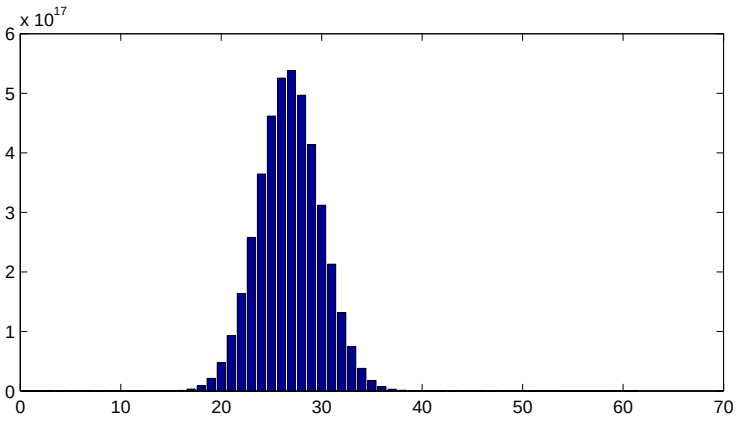

Figure 4.4: The coefficients of the real local belief with 60 observations.

In Figure 4.6, we present two examples that show the approximation. We assume $\theta = 0.8, \epsilon = 0.2$ and the two parameters of prior are set to $\alpha_{j,0} = \beta_{j,0} = 2$.

We can also use the variance inference method to approximate the local posteriors. One method is to minimize the Kullback-Leibler(KL) divergence [54] to obtain two private signals $\alpha_{j,0}$ and $\beta_{j,0}$. The Kullback-

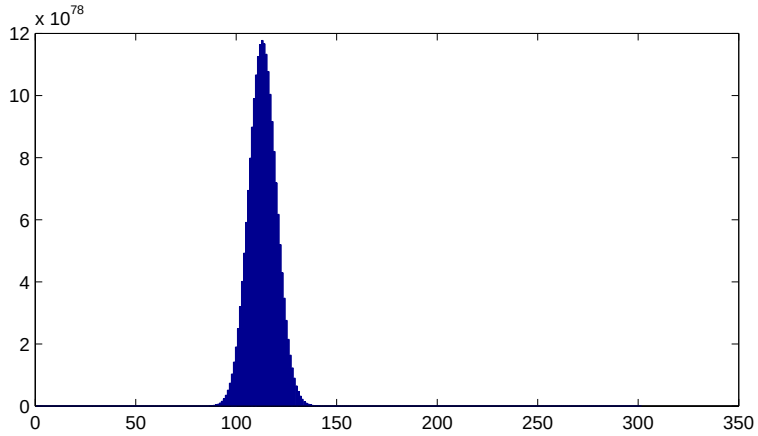

Figure 4.5: The coefficients of the real local belief with 300 observations.

Leibler divergence [54] of the real posterior and approximated posterior is given by

$$
\begin{aligned}
&D_{KL}(q||p) \\
&= \int_0^1 q(\theta) ln \frac{q(\theta)}{p(\theta)} d\theta \\
&= \int_0^1 \frac{1}{B(\alpha,\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} ln \frac{\frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}}{C_1\theta^{k_j+\alpha_0-1}(1-\theta)^{\beta_0-1} + ... + C_{k_j+1}\theta^{\alpha_0-1}(1-\theta)^{k_j+\beta_0-1}} d\theta \\
&= \frac{1}{B(\alpha,\beta)} \Bigg(\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} ln \frac{1}{B(\alpha,\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta \\
&- \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} ln[C_1\theta^{k_j+\alpha_0-1}(1-\theta)^{\beta_0-1} + ... + C_{k_j+1}\theta^{\alpha_0-1}(1-\theta)^{k_j+\beta_0-1}] d\theta \Bigg) \\
&= \frac{1}{B(\alpha,\beta)} \Bigg(\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} [(\alpha-1)ln\theta + (\beta-1)ln(1-\theta) - lnB(\alpha,\beta)] d\theta \\
&- \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} ln[C_1\theta^{k_j+\alpha_0-1}(1-\theta)^{\beta_0-1} + ... + C_{k_j+1}\theta^{\alpha_0-1}(1-\theta)^{k_j+\beta_0-1}] d\theta \Bigg).
\end{aligned}
\tag{4.56}
$$

Since

$$
\int_0^1 \frac{1}{B(\alpha,\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} ln\theta d\theta = E[ln\theta] = \psi(\alpha) - \psi(\alpha+\beta), \tag{4.57}
$$

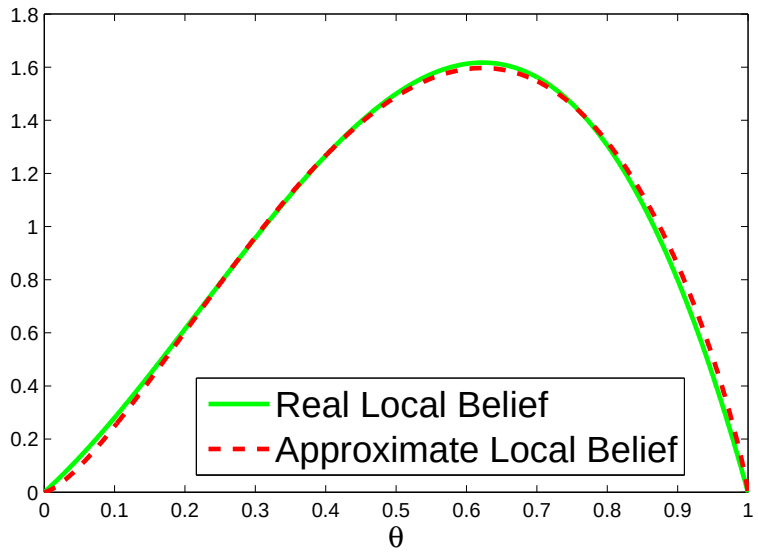

Figure 4.6: Comparison of the real local belief with the approximated belief based on 1 observation. It shows the performance is well.

where $\psi(x) = \frac{d}{dx}\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function, and assuming $t = 1 - \theta$. We obtain

$$\begin{aligned}
&\int_0^1 \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}ln(1-\theta)d\theta \\
&= -\int_1^0 \frac{1}{B(\alpha,\beta)}(1-t)^{\alpha-1}t^{\beta-1}lntdt = \int_0^1 \frac{1}{B(\alpha,\beta)}t^{\beta-1}(1-t)^{\alpha-1}lntdt \\
&= \psi(\beta) - \psi(\alpha+\beta).
\end{aligned} \tag{4.58}$$

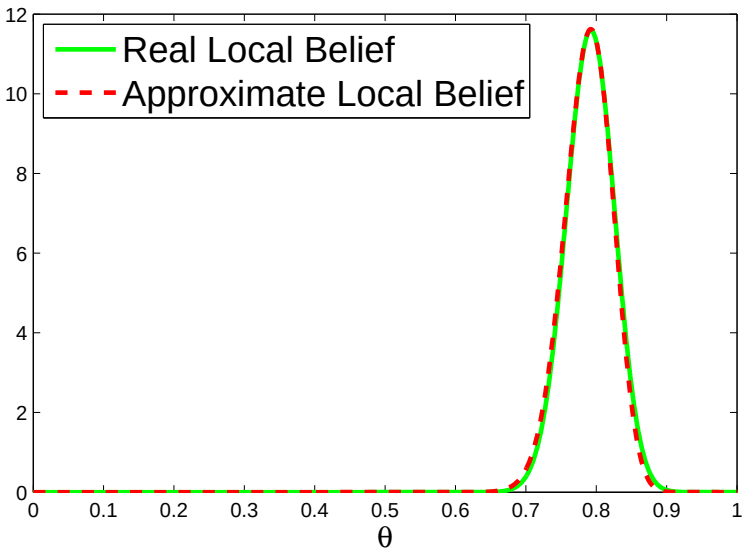

Figure 4.7: Comparison of the real local belief with the approximated belief based on 500 observations. It shows the errors do not accumulate.

According to (4.56), (4.57) and (4.58) , we find that:

$$
\begin{aligned}
&D_{KL}(q||p) \\
&= \underbrace{(\alpha-1)[\psi(\alpha)-\psi(\alpha+\beta)]+(\beta-1)[\psi(\beta)-\psi(\alpha+\beta)]-lnB(\alpha,\beta)}_{f(\alpha,\beta)} \\
&\underbrace{-\frac{1}{B(\alpha,\beta)}\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} ln[C_1\theta^{k_j+\alpha_0-1}(1-\theta)^{\beta_0-1}+...+C_{k_j+1}\theta^{\alpha_0-1}(1-\theta)^{k_j+\beta_0-1}]d\theta}_{g(\alpha,\beta)}\,.
\end{aligned}
\tag{4.59}
$$

We can minimize the KL divergence to obtain α and β using a numerical method to solve (4.59). The calculation, however, is computationally very intensive. For example, we randomly generate one observation with $\epsilon = 0.2$. The parameters α and β that minimize KL divergence and moment matching are 1.9049, 2.4147 and 1.8966, 2.4138, respectively. The KL divergences of these two methods are 5.817×10^{-4}

and 5.9511×10^{-4}, respectively. If we randomly generate 500 observations, the parameters α and β using moment matching are 114.207, 47.437. The KL divergences of these two methods are very close, which are all about 7.9205×10^{-4}. We can that see the moment matching method can obtain similar results as the minimization of the KL divergence. When the number of observations increases, the KL divergence of the moment matching method does not increase significantly.

We note that, because our estimation is the mode of the mixture Beta distribution, we can use the mode and the variance as matching parameters. The simulations show that this method is better than the moment matching. It, however, requires calculation of the mode of the mixture Beta distribution.

4.4.8 Simulations

Because the results for the model without errors is very similar, here we only present simulation results that demonstrate the performance of the proposed method for the model with errors. We note that the convergence for the two models, given the same priors of θ and the same observations of the agents (but with $\epsilon = 0$ and $\epsilon \neq 0$) is exactly the same.

We conducted an experiment with a network of 11 agents. The topology of the network was taken from [55] and is shown in Fig. 4.8. We set the unknown probability to $\theta = 0.76$. The errors and number of observations of all the agents were randomly chosen from the uniform distributions on $[0.01, 0.49]$ and $[1, 100]$, respectively. For comparison purposes, we also implemented the average consensus method from [27]. The value of the step-size of the consensus algorithm was set to 0.23.

In Figure 4.10, we show one realization of the evolution of the agents' MAP estimates with iterations for the proposed and the consensus-based methods. We can see that the estimates of each agent of the proposed method converge to estimate obtained from the exact average of the approximated sufficient statistics of all the agents. They do it at the fourth iteration. For the same network and the same signals and parameters, the agents need about 30 iterations to reach a consensus. Thus, with the proposed method, convergence is achieved much faster. Furthermore, for this network, the method always converges at the fourth iteration, whereas the consensus-based method has a variable convergence time. We observe, again, that this gain is achieved by knowledge of the network topology and more intensive computations of the agents that implement our method as opposed the ones that rely on the consensus-based method.

In Figure 4.11, we present a histogram of the error in the MAP estimates of the agents. We ran 500 trials with different sets of observations and parameters and simulated as above, and in each of them we recorded the error of the agent's estimate defined by $e_i = \widehat{\theta}_i - \widetilde{\theta}_i$, where $\widehat{\theta}_i$ and $\widetilde{\theta}_i$ are the agents' and fusion center estimates in the ith trial, respectively. The results show that these errors are very small.

Also we are curious about whether the errors are going to accumulate for a large number of agents. From the proposed method, we can find the signal of each agent will converge to the average value, so we omit the diffusion process and only compare the MAP estimates based on the average of all approximated private signals and in the fusion center without any approximations. Figure 4.13 and 4.14 show that the errors are also very small. The reason why the errors are unbiased may be that this

approximation method has a tendency to be close to the center which is 0.5.

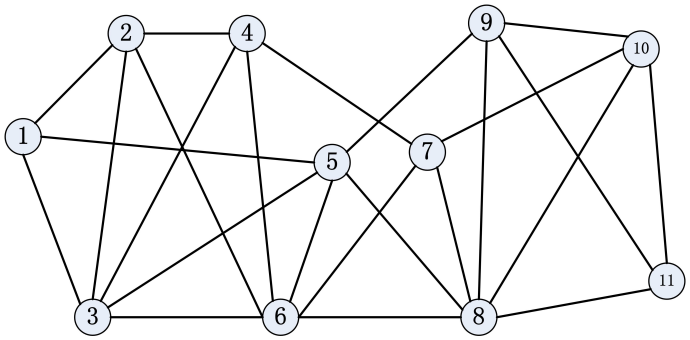

Figure 4.8: The network of agents in the experiments.

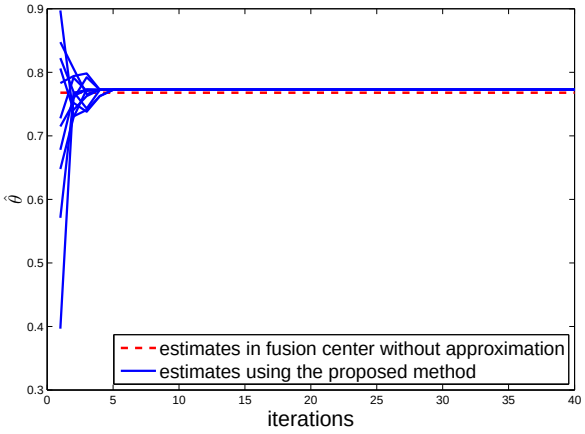

Figure 4.9: The MAP estimates of all the agents as a function of iteration number obtained by the proposed method (solid lines). The MAP of the fusion center (the dashed line).

4.5 Summary

In this chapter, we presented a Bayesian learning method in a network of agents, where the agents aim at estimating the posterior distribution of a probability in a Bernoulli model. Each agent knows the structure of the network and stores the received signals from their neighbors only if they

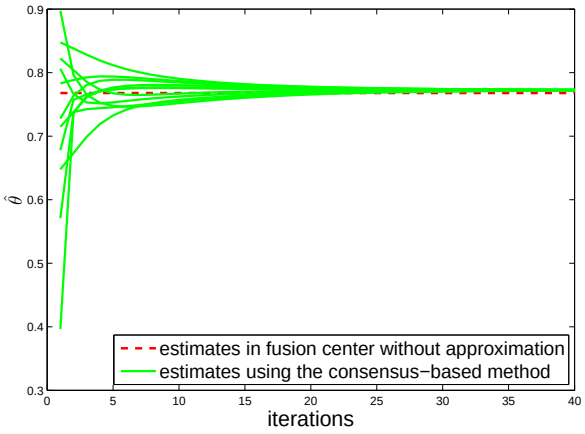

Figure 4.10: The MAP estimates of all the agents as a function of iteration number obtained by the consensus-based method (solid lines). The MAP of the fusion center (the dashed line).

are linearly independent from the previously received signals. The agents use the newly received linearly independent signals to modify their signals and subsequently they broadcast them to their neighbors. We demonstrate the performance of the method with computer simulations and compare it with the performance of a consensus-based method.

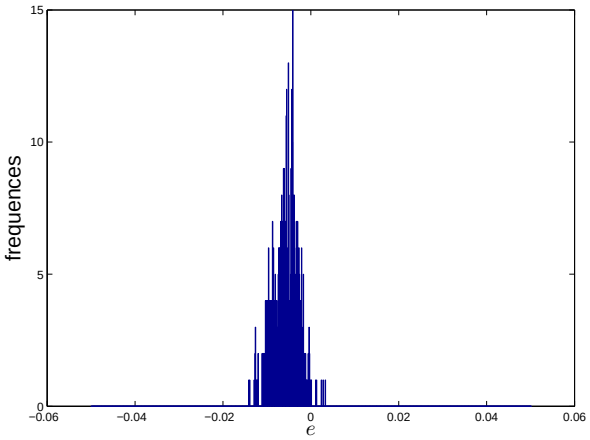

Figure 4.11: A histogram of errors of the proposed method from 500 trials.

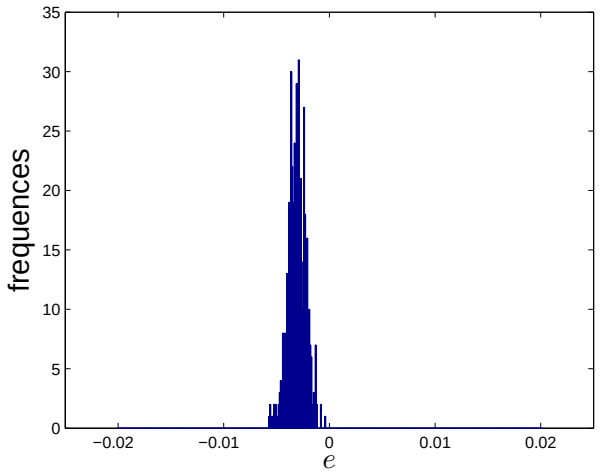

Figure 4.12: A histogram of errors of 100 agents from 500 trials ($\theta = 0.76$).

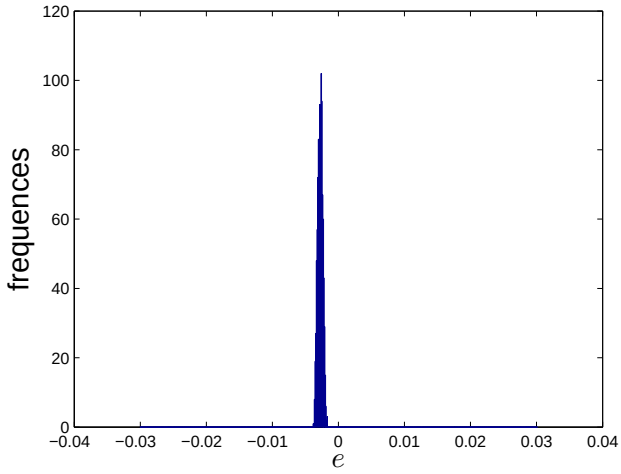

Figure 4.13: A histogram of errors of 1000 agents from 1000 trials ($\theta = 0.76$).

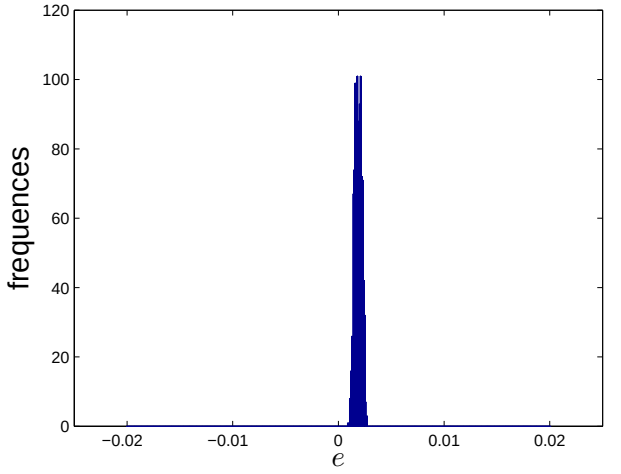

Figure 4.14: A histogram of errors of 1000 agents from 1000 trials ($\theta = 0.32$).

Chapter 5

Conclusions and Future Works

In this dissertation, we addressed three problems of backscatter-based tag-to-tag (BBTT) communication systems. First, we investigated a unique phase cancellation problem that occurs in BBTT systems. These are systems where two or more radio-less devices (tags) communicate with each other purely by reflecting (backscattering) an external signal (whether ambient or intentionally generated). A transmitting tag modulates baseband information onto the reflected signal using backscatter modulation. At the receiving tag, the backscattered signal is superimposed to the external excitation and the resulting signal is demodulated using envelope detection techniques. The relative phase difference between the backscatter signal and the external excitation signal at the receiving tag has a large impact on the envelope of the resulting signal. This often causes a complete cancellation of the baseband information contained in the envelope, and it results in a loss of communication between the two tags. This problem is ubiquitous in all BBTT systems and greatly impacts the reliability, robustness and communication range of such systems. We

theoretically analyzed and experimentally demonstrated this problem for devices that use both ASK and PSK backscattering. We then presented a solution to the problem based on the design of a new backscatter modulator for tags that enables multi-phase backscattering. We also proposed a new combination method that can further enhance the detection performance for BBTT systems. We examined the performance of the proposed techniques through theoretical analysis, computer simulations, and lab experiments.

For BBTT systems, the transmission collision between tags cannot be fully avoided under its current limitations. We proposed two anti-collision protocols based on existing protocols. We modified the CSMA/CA protocol and modeled it by using Markov chains in a complete network and framed the slotted Aloha protocol in both complete and linear networks. These models were used to obtain the best parameters for the protocols. We then compared the performances between these two protocols in complete networks. We measured the performances by the average transmission time from a sink node to a destination node. For the general networks, we tried different combinations of parameters to get the best set for each protocol and compared them using simulations. The results showed that the modified CSMA/CA performs better than the FSA protocol when the network is very crowded, but the FSA is better in more general cases.

We also addressed problems of distributed learning in networks of cooperative agents. First, we discussed two consensus-based methods: the average consensus, the gossip-based averaging methods, and a sequential learning method. Then, we presented a Bayesian learning method in a network of agents, where the agents aim at estimating the posterior

distribution of a probability in a Bernoulli model. Each agent knows the structure of the network and stores the received signals from their neighbors only if they are linearly independent from previously received signals. The agents use the newly received linearly independent signals to modify their signals and subsequently they broadcast them to their neighbors. We also prove that each agent can eventually reach the same information, and can obtain the global posterior of a fusion center using this information. We applied this method to Bernoulli models in two scenarios. In the first one, we omitted the observation errors and assumed that each agent has the same number of observations and the same prior. In the second one, we considered a model with observation errors. In this setting, because the local and global posteriors are complicated, we approximated the local posteriors by the moment matching method. For the addressed problem, this method is much easier and quicker than methods based on variational inference that use the Kullback-Leibler divergence. Finally, we demonstrated the performance of the method with computer simulations and compared it with the performance of the consensus-based method.

There are many directions for future work. For the phase cancellation problem in Chapter 2, one area of work is to design a protocol that will allow the tags to choose the best phase during a handshaking period. One can also use delay circuits to store the first signal and combine in various ways with signals received subsequently. With this approach we expect to increase the communication ranges between the tags. To that end, we did some simulations and experiments using the ambient signals instead of CW signals. It is important to point out that systems that exploit ambient signals also suffer from phase cancellation problem.

For protocols in BBTT systems, one may use more complicated schemes to reduce the unnecessary transmissions, such as limiting the transmission times of each tag. These schemes are much more difficult to analyze, but they may bring better performance. They are certainly worthy of further investigation.

In distributed Bayesian learning, it would be interesting to study different models using the proposed method. They include a model where the agents receive observations in each iteration and the unknown variable of interest, θ, is time varying. We believe that the upper bound of convergence can be made stricter. In [5], the authors conjecture that this bound is $O(n)$ iterations. We believe that the topology is more important than the number of agents in the network, i.e., that it is a function of a function of d, the diameter of a network.

Bibliography

[1] D. Giusto, A. Lera, G. Morabito, and L. Atzori, *The Internet of Things*. Springer, 2010.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[4] D. M. Dobkin, *The RF in RFID: UHF RFID in Practice*. Newnes, 2012.

[5] E. Mossel and O. Tamuz, "Efficient Bayesian learning in social networks with Gaussian estimators," *arXiv preprint arXiv:1002.0747*, 2010.

[6] P. V. Nikitin, S. Ramamurthy, R. Martinez, and K. Rao, "Passive tag-to-tag communication," in *RFID (RFID), 2012 IEEE International Conference on*, pp. 177–184, IEEE, 2012.

[7] G. Marrocco and S. Caizzone, "Electromagnetic models for passive tag-to-tag communications," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 11, pp. 5381–5389, 2012.

[8] A. Athalye, V. Savic, M. Bolic, and P. M. Djuric, "Novel semi-passive rfid system for indoor localization," *Sensors Journal, IEEE*, vol. 13, no. 2, pp. 528–537, 2013.

[9] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: wireless communication out of thin air," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 39–50, ACM, 2013.

[10] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-fi backscatter: Internet connectivity for rf-powered devices," in *Proceedings of the 2014 ACM conference on SIGCOMM*, pp. 607–618, ACM, 2014.

[11] K. Kurokawa, "Power waves and the scattering matrix," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 13, no. 2, pp. 194–202, 1965.

[12] P. V. Nikitin, K. S. Rao, S. F. Lam, V. Pillai, R. Martinez, and H. Heinrich, "Power reflection coefficient analysis for complex impedances in rfid tag design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 9, pp. 2721–2725, 2005.

[13] J. Wang and M. Bolic, "Exploiting dual-antenna diversity for phase cancellation in augmented rfid system," in *Smart Communications in*

Network Technologies (SaCoNeT), 2014 International Conference on, pp. 1–6, IEEE, 2014.

[14] A. Athalve and P. M. Djuric, "Rfid system and method for localizing and tracking a moving object with an rfid tag," Oct. 12 2010. US Patent 7,812,719.

[15] S. J. Thomas, E. Wheeler, J. Teizer, and M. S. Reynolds, "Quadrature amplitude modulated backscatter in passive and semipassive uhf rfid systems," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 60, no. 4, pp. 1175–1182, 2012.

[16] C. Boyer and S. Roy, "Coded qam backscatter modulation for rfid," *Communications, IEEE Transactions on*, vol. 60, no. 7, pp. 1925–1934, 2012.

[17] S. Thomas and M. S. Reynolds, "Qam backscatter for passive uhf rfid tags," in *RFID, 2010 IEEE International Conference on*, pp. 210–214, IEEE, 2010.

[18] "Designing detectors for rf/id tags, application note 1089." `http://www.spelektroniikka.fi/kuvat/schot14.pdf`.

[19] A. Athalye, V. Savic, M. Bolic, and P. Djuric, "Novel semi-passive rfid system for indoor localization," *Sensors Journal, IEEE*, vol. 13, pp. 528–537, Feb 2013.

[20] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[21] E. Ziouva and T. Antonakopoulos, "Csma/ca performance under high traffic conditions: throughput and delay analysis," *Computer communications*, vol. 25, no. 3, pp. 313–321, 2002.

[22] L. S. Committee *et al.*, "Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans)," *IEEE Computer Society*, 2003.

[23] J. He, Z. Tang, H.-H. Chen, and S. Wang, "An accurate markov model for slotted csma/ca algorithm in ieee 802.15. 4 networks," *Communications Letters, IEEE*, vol. 12, no. 6, pp. 420–422, 2008.

[24] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance analysis of slotted carrier sense ieee 802.15. 4 medium access layer," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 9, pp. 3359–3371, 2008.

[25] F. Shu and T. Sakurai, "A new analytical model for the ieee 802.15. 4 csma-ca protocol," *Computer networks*, vol. 55, no. 11, pp. 2576–2591, 2011.

[26] M. Bolic, D. Simplot-Ryl, and I. Stojmenovic, *RFID systems: research trends and challenges*. John Wiley & Sons, 2010.

[27] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[28] D. Gale and S. Kariv, "Bayesian learning in social networks," *Games and Economic Behavior*, vol. 45, no. 2, pp. 329–346, 2003.

[29] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar, "Bayesian learning in social networks," *The Review of Economic Studies*, vol. 78, no. 4, pp. 1201–1236, 2011.

[30] P. M. Djurić and Y. Wang, "Distributed Bayesian learning in multiagent systems: Improving our understanding of its capabilities and limitations," *Signal Processing Magazine, IEEE*, vol. 29, no. 2, pp. 65–76, 2012.

[31] V. Krishnamurthy, "Quickest time herding and detection for optimal social learning," *arXiv preprint arXiv:1003.4972*, 2010.

[32] C. P. Chamley, *Rational Herds: Economic Models of Social Learning*. Cambridge University Press, 2003.

[33] E. Mossel and O. Tamuz, "Iterative maximum likelihood on networks," *Advances in Applied Mathematics*, vol. 45, no. 1, pp. 36–49, 2010.

[34] R. M. Murray, "Consensus protocols for networks of dynamic agents," 2003.

[35] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[36] D. Li, Q. Liu, X. Wang, and . Lin, "Consensus seeking over directed networks with limited information communication," *Automatica*, vol. 49, no. 2, pp. 610–618, 2013.

[37] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[38] Y. Wang and P. M. Djurić, "A gossip method for optimal consensus on a binary state from binary actions," 2013.

[39] H. Wang, X. Liao, and T. Huang, "Average consensus in sensor networks via broadcast multi-gossip algorithms," *Neurocomputing*, vol. 117, pp. 150–160, 2013.

[40] S. Wu and M. G. Rabbat, "Broadcast gossip algorithms for consensus on strongly connected digraphs," *Signal Processing, IEEE Transactions on*, vol. 61, no. 16, pp. 3959–3971, 2013.

[41] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, pp. 951–956, IEEE, 2003.

[42] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[43] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 63–70, IEEE, 2005.

[44] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[45] I. D. Schizas and G. B. Giannakis, "Consensus-based distributed estimation of random signals with wireless sensor networks," in *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*, pp. 530–534, IEEE, 2006.

[46] U. A. Khan, S. Kar, and J. M. Moura, "Distributed average consensus: Beyond the realm of linearity," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pp. 1337–1342, IEEE, 2009.

[47] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pp. 482–491, IEEE, 2003.

[48] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[49] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pp. 565–574, IEEE, 2000.

[50] V. Krishnamurthy and A. Aryan, "Quickest detection of market shocks in agent based models of the order book," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 1480–1485, IEEE, 2012.

[51] V. Krishnamurthy, "Quickest detection pomdps with social learning: Interaction of local and global decision makers," *Information Theory, IEEE Transactions on*, vol. 58, no. 8, pp. 5563–5587, 2012.

[52] A. V. Banerjee, "A simple model of herd behavior," *The Quarterly Journal of Economics*, vol. 107, no. 3, pp. 797–817, 1992.

[53] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," *Journal of political Economy*, pp. 992–1026, 1992.

[54] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[55] A. J. Bean and A. C. Singer, "Cooperative estimation in heterogeneous populations," in *Proceedings of the Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 696–699, 2011.