EFFICIENT DATA COMMUNICATION PROTOCOLS FOR WIRELESS NETWORKS

by

Engin Zeydan

A DISSERTATION

Submitted to the Faculty of the Stevens Institute of Technology
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Engin Zeydan, Candidate

ADVISORY COMMITTEE

Cristina Comaniciu, Chairman Date

Didem Kivanc Tureli, Co-chairman Date

Yu-Dong Yao Date

Rajarathnam Chandramouli Date

Jose E. Ramirez-Marquez Date

STEVENS INSTITUTE OF TECHNOLOGY
Castle Point on Hudson
Hoboken, NJ 07030
2010

EFFICIENT DATA COMMUNICATION PROTOCOLS FOR WIRELESS NETWORKS

ABSTRACT

In this dissertation, efficient decentralized algorithms are investigated for cost minimization problems in wireless networks. For wireless sensor networks, we investigate both the reduction in the energy consumption and throughput maximization problems separately using multi-hop data aggregation for correlated data in wireless sensor networks. The proposed algorithms exploit data redundancy using a game theoretic framework. For energy minimization, routes are chosen to minimize the total energy expended by the network using best response dynamics to local data. The cost function used in routing takes into account distance, interference and in-network data aggregation. The proposed energy-efficient correlation-aware routing algorithm significantly reduces the energy consumption in the network and converges in a finite number of steps iteratively. For throughput maximization, we consider both the interference distribution across the network and correlation between forwarded data when establishing routes. Nodes along each route are chosen to minimize the interference impact in their neighborhood and to maximize the in-network data aggregation. The resulting network topology maximizes the global network throughput and the algorithm is guaranteed to converge with a finite number of steps using best response dynamics.

For multiple antenna wireless ad-hoc networks, we present distributed cooperative and regret-matching based learning schemes for joint transmit beamformer and power level selection problem for nodes operating in multi-user interference environment. Total network transmit power is minimized while ensuring a constant received signal-to-interference and noise ratio at each receiver. In cooperative and regret-matching based power minimization algorithms, transmit beamformers are selected from a predefined codebook to minimize the total power. By selecting transmit beamformers judiciously and performing power adaptation, the cooperative algorithm is shown to converge to pure strategy Nash equilibrium with high probability throughout the iterations in the interference impaired network. On the other hand, the regret-matching learning algorithm is noncooperative and requires minimum amount of overhead. The proposed cooperative and regret-matching based distributed algorithms are also compared with centralized solutions through simulation results.

Author: Engin Zeydan

Advisor: Cristina Comaniciu

Co-advisor: Didem Kivanc Tureli

Date: December 08, 2010

Department: Electrical and Computer Engineering

Degree: Doctor of Philosophy

Acknowledgements

Time is passing quickly and it has been more than four years since I have started my Ph.D work. Over the course of my graduate studies, I had the chance to continue my studies in two different universities, Stevens Institute of Technology and West Virginia University and two different cities, Hoboken, NJ and Montgomery, WV. The experiences and events that I have encountered in these cities will never fade away from my memory.

This thesis could not have been completed without support, encouragement and guidance of my family, friends and advisors. I was extremely fortunate to not have just one advisor during my graduate studies. First of all, I would like to thank my advisor Prof. Cristina Comaniciu for serving as a mentor during my graduate studies. She opened new doors for me with new research topics which have become the fundamental parts of this dissertation.

My co-advisors Dr. Didem-Kivanc and Prof. Ufuk Tureli are currently in West Virginia University Institute of Technology. Dr. Didem-Kivanc or “Didem” in short as she always wants everybody to call her as, is a very kind person. She always made time for me, had long discussions with me and gave me invaluable advice. Her help during my studies kept me motivated and added new and better dimensions to my research. I am grateful to her. Prof. Ufuk Tureli, on the other hand, always gave me a large amount of freedom during my graduate study and helped me a lot in difficult times with his generosity in financial support. Both Didem and Prof. Tureli helped me a lot to grow as a researcher and individual.

I am also grateful to my committee members Prof. Yu-Dong Yao, Prof. Rajarathnam Chandramouli and Prof. Jose E. Ramirez-Marquez for their valuable com-

ments and time.

Finally, my family has always given me strength throughout my entire life especially during my graduate studies in USA. Their love and support have guided me through the years. Mom and Dad, you have always supported me and been there for me. I owe you a lot for all the things I have achieved in life. I would also like to thank my sisters, Leyla and Ruken for their continuous support and encouragements. This dissertation is a tribute to you all.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As network architecture becomes more decentralized and nodes are more autonomous in decision making, efficient distributed algorithms become important for real time implementation of wireless networks. In designing distributed wireless networks, designers must deal with several big and important issues. Energy efficiency, interference management, throughput requirements, level of interactions or cooperations between users, data latency, reliability and quality of service are some of the significant challenges that should be addressed in a network operating in a multi-user environment. Game theory can be used to provide a powerful mathematical framework to model and analyze wireless networks. Interactions between users or nodes can be modeled as a game among different users and the desired outcome is the steady-state equilibrium point. In this dissertation, we address some of these important issues mentioned above in the context of wireless sensor networks (WSNs) and wireless ad-hoc networks. Furthermore, we use the principles of game theory to analyze the adaptations that can occur in such decentralized networks.

WSNs consists of tiny sensor devices that are spatially distributed in large numbers and in high density in a certain region. Each sensor has a unique ID and a radio interface which is used to communicate with some of the sensors around it. They are often deployed in order to cooperatively sense, process and deliver information of the targeted physical environment to the desired person of interest which is usually called the sink node. Data collected at a sensor node is routed through intermediate nodes until it reaches the sink nodes. Due to their vast potential application areas in environmental and habitat monitoring, surveillance, early disaster warnings, military and security, transport and health care to name a few, WSNs have attracted plethora of research efforts [1] [2]. Some of the properties and requirements that should exist in protocols designed for WSNs are energy-efficiency, continuous or query and event driven processing capabilities, distributed solutions using as much local information as possible and self-organizing processing abilities [1]. Depending on application requirements, different communication protocols with different objective functions can be designed using WSNs. Energy-efficiency[3], network lifetime [4], throughput [5], latency [6], data accuracy [7], capacity [8], end-to-end delay [9] and security [10] are some of the main objective requirements that are commonly optimized in the context of WSNs.

Note that the amount of sensory data in WSNs can be large. Therefore, exploiting the spatial correlation between adjacent nodes can significantly reduce the amount of data that needs to be transmitted using multi-hop transmission. Problems such as excessive energy usage, buffer overflow of nodes near sink nodes and loss of data can be reduced by filtering out the data redundancy. In WSNs, designing *data aggregation trees* is a key strategy to reduce the redundancy and network load in the network [11]. During data aggregation, sensed data gathered at the intermediate node is combined with data from different sensors and transmitted to the next node for further processing. One of the most important challenges in designing an efficient data aggregation scheme is the limited energy supply of each sensor node. Therefore, it is critical to design energy efficient routing algorithms for data aggregation in WSNs. The data routing scheme along with the data aggregation model decides when and where data flows will meet and how data aggregation will be performed. Our goal in Chapter 3 is to propose an energy-efficient routing scheme for WSNs that takes advantage of spatial correlation between neighboring nodes and performs data aggregation accordingly. On the other hand, fast and efficient delivery of data may be another important requirement for WSNs when large amounts of data need to be delivered in a timely or short period of time. In this case, the network should be able to provide high throughput and timely delivery of the data. In Chapter 4, we introduce a new data routing algorithm to maximize throughput for correlated data.

In wireless ad-hoc networks operating in a “multi-user” environment (i.e. when the number of users is larger than one), maintaining a certain quality-of-service for each receiver is a challenging problem. Fluctuations in the received signal strength due to channel conditions, interference, other users’ transmission parameters or any other external obstacle may reduce the efficiency and reliability of wireless networks. Simply increasing the transmit power for each user is not an applicable solution, because it also increases the interference in the environment and is also not suitable for battery power limitations. The design of smart multiple access control algorithms for power (or energy) efficient communication in wireless ad-hoc networks while guaranteeing certain reliability constraints is another focus of this dissertation.

Employing multiple-input multiple- output (MIMO) beamforming in wireless systems maximizes the spectral efficiency and reliability of wireless communication. MIMO systems have been widely adopted in many future wireless communication standards (e.g., WiMAX, 3GPP LTE, etc.). Previous works on using MIMO beamforming techniques for MIMO communications have shown that there is great promise in using limited feedback scheme for single user systems. However, in multiuser MIMO systems, there are significant problems that need to be addressed before multi-user MIMO algorithms become widely adopted. Some of these problems are interference mitigation and management, resource allocation issues, amount of feedback information needed for the distributed algorithms, energy and power consumption issues and coordination between users’ strategies in the network. Therefore, addressing all these problems by designing “efficient” distributed multi-user MIMO algorithms is still an open research problem. For this reason, in Chapter 5, we provide efficient power minimization algorithms for the solution of the joint transmit beamformer and power adaptation problem in wireless ad-hoc networks while guaranteeing a certain quality of service to each receiver or user. In our MIMO multi-user algorithms, we address the recent concerns such as interference management, coordination between users and resource allocation (transmit power) issues in ad-hoc networks.

1.1 Main Contributions of this Dissertation

To summarize, the main contributions of this dissertation are:

1. In Chapter 3, we introduce an energy-efficient routing scheme for correlated data for WSNs and establish the convergence to pure Nash equilibrium using the congestion and potential game formulation.

2. In Chapter 4, we introduce a throughput maximizing correlated data routing for WSNs, and study the convergence properties using congestion and potential game formulation. We also derive analytical conditions for Nash equilibrium solutions for a special network topology.

3. In Chapter 5, we introduce distributed joint transmit beamformer and power adaptation algorithms for multi-user MIMO wireless ad-hoc networks. For a cooperative network, we study the probabilistic convergence to an optimal transmit beamformer action profile that minimizes the total network power consumption. For a noncooperative network, we study a regret-matching learning algorithm where the better transmission parameters (i.e. power and transmit beamformers) are selected or learned throughout iterations based on the "regret" function.

Chapter 2

Game Theory Background

In this chapter, we will give background information about the game theoretic concepts used in this dissertation. The readers can also refer to [12] [13] [14] (and references therein) for a detailed review of concepts.

2.1 Finite Strategic-Form Games

In this dissertation, we only assume finite strategic-form games. A finite strategic-form game Γ consists of an N-player set $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_N\}$ where each player $\mathcal{P}_i \in \mathcal{P}$ has an action set $\mathcal{A}_i \in \mathcal{A}$ and a utility function $u_i : \mathcal{A} \rightarrow \mathbb{R}$ where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_N$.

In a game, each player $\mathcal{P}_i \in \mathcal{P}$ chooses an action $a_i \in \mathcal{A}_i$, and receives an utility $u_i(\mathbf{a})$ depending on the strategy profile $\mathbf{a} = (a_1, a_2, ..., a_N) \in \mathcal{A}$. Let a_{-i} denote the collection of actions of players other than player P_i, i.e.

$$a_{-i} = \{a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_N\}.$$

Note that the action profile $\mathbf{a}$ and the utility $u_i(\mathbf{a})$ can also be written as (a_i, a_{-i}) and $u_i(a_i, a_{-i})$ respectively. Similarly, let $\mathcal{P}_{-i} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_{i-1}, \mathcal{P}_{i+1}, ...\mathcal{P}_N\}$ denote the set of players other than player $\mathcal{P}_i$ and let $\mathcal{A}_{-i} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, ...\mathcal{A}_N\}$ the set of all collective actions of all players other than player $\mathcal{P}_i$.

2.2 Forms of equilibrium

2.21 Nash equilibrium

An action profile $\mathbf{a}^* = (a_i^*, a_{-i}^*) \in \mathcal{A}$ is called a *pure Nash equilibrium* if for all players $\mathcal{P}_i \in \mathcal{P}$,

$$u_i(a_i^*, a_{-i}^*) = \max_{a_i \in \mathcal{A}_i} u_i(a_i, a_{-i}^*).$$

In a Nash equilibrium point, no players have an incentive to unilaterally deviate.

Mixed-strategy Nash equilibrium

A *mixed strategy* is a probability distribution over pure strategies. In mixed-strategies, each player's randomization is statistically independent of those of his opponents, and the utilities of *mixed strategy profiles* are the expected values of the corresponding pure strategy utilities. Let $\Phi(\mathcal{A}_i)$ be the set of probability distributions over the action set $\mathcal{A}_i$ and the *mixed-strategy* of player $\mathcal{P}_i$ be defined as $\chi_i \in \Phi(\mathcal{A}_i)$. We denote $\chi \in \Phi(\mathcal{A})$ to represent a *mixed-strategy profile*. We denote $\chi_i(a_i)$ as the probability that player $\mathcal{P}_i$ will select action a_i and $\sum_{a_i \in \mathcal{A}_i} \chi_i(a_i) = 1$. When all players $\mathcal{P}_i \in \mathcal{P}$ are playing independently, according to their own strategy $\chi_i \in \Phi(\mathcal{A}_i)$, then the expected utility of player $\mathcal{P}_i$ for action probability χ_i becomes

$$u_i(\chi_i, \chi_{-i}) = \sum_{\mathbf{a} \in \mathcal{A}} u_i(\mathbf{a}) \chi_1(a_1) \chi_2(a_2) \ldots \chi_N(a_N)$$

where $\chi_{-i} = \{\chi_1, \ldots \chi_{i-1}, \chi_{i+1}, \ldots, \chi_N\}$ denotes the collection of action probabilities of players other than $\mathcal{P}_i$.

Definition 2.2.1. *Mixed Strategy Nash Equilibrium:* A strategy profile $\chi* = (\chi_1^*, \chi_2^*, \ldots, \chi_N^*)$

is called a mixed-strategy Nash equilibrium for all players $\mathcal{P}_i \in \mathcal{P}$ if,

$$u_i(\chi_i^*, \chi_{-i}^*) \geq u_i(\chi_i, \chi_{-i}^*), \ \ \forall \chi_i \in \Phi(\mathcal{A}_i). \tag{2.1}$$

2.22 Correlated equilibrium

Nash equilibrium is an equilibrium point under which the players choose their actions or strategies independently. Assume that, before the game is played, players build a “signaling device” that sends *different but correlated signals* (which does not affect the utilities) to each of them. Then, each player may then choose his action in the game depending on this “signal”. Notice that if the signals are independent across the players, it is a Nash equilibrium (in mixed or pure strategies) of the original game. But the signals could also be correlated, in which case the equilibria is the correlated equilibrium [15]. Therefore, correlated equilibrium is a broader class of equilibria and contains the set of Nash equilibria.

Let $\chi(\mathbf{a}) \in \Phi(\mathcal{A})$ denote any probability distribution of joint action $\mathbf{a}$ and $\sum_{\mathbf{a}\in\mathcal{A}} \chi(\mathbf{a}) = 1$. In the special case when all players $\mathcal{P}_i \in \mathcal{P}$ play independently according to their personal strategy $\chi_i(\mathbf{a}_i) \in \Phi(\mathcal{A}_i)$ as is the case for the definition of Nash equilibrium, then the joint probability distribution is simplified into,

$$\chi(\mathbf{a}) = \chi_1(a_1)\chi_2(a_2)\ldots\chi_N(a_N)$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ and $\chi_i(a_i)$ represents the probability that player $\mathcal{P}_i$ will select action a_i.

Definition 2.2.2. *Correlated Equilibrium:* The probability distribution $\chi(\mathbf{a}) \in \Phi(\mathcal{A})$

is a correlated equilibrium if for all players $\mathcal{P}_i \in \mathcal{P}$ and for all actions $a_i, a_i' \in \mathcal{A}_i$,

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i, a_{-i})\chi(a_i, a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i', a_{-i})\chi(a_i, a_{-i}), \tag{2.2}$$

or

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i, a_{-i})\chi(a_{-i}|a_i) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i', a_{-i})\chi(a_{-i}|a_i'), \tag{2.3}$$

where $\chi(.|.)$ is the conditional probability.

The condition for correlated equilibrium in (2.3) states that at a probability distribution $\chi(a_i, a_{-i})$ that is a correlated equilibrium, each player $\mathcal{P}_i'$s conditional expected payoff for action a_i is at least as good as his conditional expected payoff for any other action $a_i' \neq a_i$. In other words, each player $\mathcal{P}_i$ does not gain by disobeying the recommendation to play a_i if every other player plays the recommendation.

2.23 Coarse correlated equilibrium

Coarse correlated equilibrium is used to describe a probability distribution on pure strategy profiles by relaxing the correlated equilibrium condition [13]. In coarse-correlated equilibrium, the players do not receive any recommendation on how to play the game, each of them can choose to either adhere to and get the corresponding correlated expected payoff or to deviate based on predictions, by picking some strategy. The coarse correlated equilibrium conditions state that no player can gain by unilaterally deviating based on predictions.

Definition 2.2.3. *Coarse Correlated Equilibrium:* The probability distribution $\chi(\mathbf{a}) \in \Phi(\mathcal{A})$ is a coarse correlated equilibrium if for all players $\mathcal{P}_i \in \mathcal{P}$ and for all actions $a_i, a_i' \in \mathcal{A}_i$,

$$\sum_{\mathbf{a} \in \mathcal{A}} u_i(a_i, a_{-i})\chi(a_i, a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i', a_{-i})\chi_{-i}(a_{-i}) \tag{2.4}$$

where

$$\chi_{-i}(a_{-i}) = \sum_{a_i' \in \mathcal{A}_i} \chi(a_i', a_{-i}).$$

Note that $\chi_{-i}(a_{-i}) \in \Phi(\mathcal{A}_{-i})$ denotes the marginal distribution of all players other than player $\mathcal{P}_i$.

Note that, all correlated equilibria and Nash equilibria are in fact coarse correlated equilibria. If all players select their actions independently, then the definitions of correlated, coarse correlated, and Nash equilibria become equivalent.

2.3 Classes of games

In this dissertation, we will focus on three different classes of games: identical interest games, congestion games and potential games.

2.31 Identical Interest Games

In identical interest games, the players' utilities $u_i, \;\; i = 1, 2, ..., N$, are same. In other words, for some function $\phi : \mathcal{A} \to \mathbb{R}$,

$$u_i(a_i, a_{-i}) = \phi(a_i, a_{-i}),$$

for all $\mathcal{P}_i \in \mathcal{P}$ and $\mathbf{a} \in \mathcal{A}$. All identical interest games have at least one Nash equilibrium, namely any action profile $\mathbf{a}$ that maximizes $\phi(\mathbf{a})$.

2.32 Potential Games

A potential game is a normal form game such that any changes in the utility function of any player in the game due to an unilateral deviation by the player is reflected in a global function. An exact potential game is a special case of potential game.

A game that has an exact potential function is called an exact potential game. An exact potential function $\mathbb{P}(.)$ is defined as

$$\begin{aligned} \mathbb{P} : \mathcal{A} \to \mathbb{R},\ \ \forall \mathcal{P}_i \in \mathcal{P} \ \text{ and } \ a_i, a_i' \in \mathcal{A}_i, \\ u_i(a_i, a_{-i}) - u_i(a_i', a_{-i}) = \mathbb{P}(a_i, a_{-i}) - \mathbb{P}(a_i', a_{-i}). \end{aligned} \tag{2.5}$$

The above result shows that the gain (or loss) caused by any players unilateral move is exactly the same as the gain (or loss) in the potential function, which may be viewed as a global objective function. Note that in potential games, any action profile maximizing the potential function is a pure Nash equilibrium, hence every potential game possesses at least one such equilibrium. Some of the properties of potential games are:

- All NE points are the maximizers of the potential function, either locally or globally,
- Potential games have at least one pure NE,
- At each step of the potential games, a *better response* or *best response* strategy converges to a NE if each player investigates its strategy space and takes actions to maximize its utility [16] [17].

2.33 Congestion Games

The congestion game Γ is defined as a tuple $(\mathcal{P}, \mathcal{F}, (\mathcal{A}_i)_{i \in \mathcal{P}}, (w_f)_{f \in \mathcal{F}})$ where $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ denotes the set of players, $\mathcal{F} = \{1, \ldots, m_f\}$ is the set of facilities and m_f is the number of facilities, $\mathcal{A}_i \subseteq 2^{\mathcal{F}}$ is the strategy space of player $\mathcal{P}_i$, and $w_f : \mathbf{a} \to \mathbb{R}$ is a cost function associated with using the facility f where $\mathbf{a} = (a_1, \ldots, a_N)$ is a *state of the game* in which player $\mathcal{P}_i$ chooses strategy $a_i \in \mathcal{A}_i$. We define $\theta_f(\mathbf{a})$ as the subset of players directly connected to facility f including the player at facility f, that is $\theta_f(\mathbf{a}) = \{\mathcal{P}_i | f \in a_i\}$. The players aim at choosing strategies $a_i \in \mathcal{A}_i$ minimizing their individual cost, where the total cost $\Phi_i(\mathbf{a})$ of player $\mathcal{P}_i$ is given by $\Phi_i(\mathbf{a}) = \sum_{f \in a_i} w_f(\theta_f(\mathbf{a}))$. We define utility function for source Y_i in our congestion game as

$$\begin{aligned} u_i : \mathbf{a} \to \mathbb{R}, \quad u_i(a_i, a_{-i}) &= -\Phi_i(\mathbf{a}), \\ &= -\sum_{f \in a_i} w_f(\theta_f(\mathbf{a})), \end{aligned} \tag{2.6}$$

where $a_{-i} = (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$ is the strategy space of player $\mathcal{P}_i$'s opponents. The game performance is influenced by the selection of cost functions $w_f(\theta_f(\mathbf{a}))$ for facilities. In fact, every congestion game is a potential game and every finite potential game is isomorphic to a congestion game [16].

2.4 Repeated Games

In a repeated game, at each iteration $n = 0, 1, 2, ...$, each player $\mathcal{P}_i \in \mathcal{P}$ chooses an action $a_i(n) \in \mathcal{A}_i$ based on an iterative update strategy and receives the utility $u_i(\mathbf{a}(n))$ where $\mathbf{a}(n) = (a_1(n), ..., a_N(n))$. Different learning algorithms can be designed to update the actions of each player based on the information obtained in the past observations. For example, in Chapter 5, we will discuss a learning algorithm called regret-matching based learning in which the players choose the actions based on their "regret" for not choosing other actions.

2.41 Iterative Updating Strategies

Let $\mathbf{T} : \mathcal{A} \rightarrow \mathcal{A}$ be any mapping from a subset $\mathcal{A} \subseteq \mathbb{R}^N$ to itself. The updating scheme of actions for all players $\mathcal{P}_i \in \mathcal{P}$ at iteration n can be defined as

$$\mathbf{a}(n+1) = \mathbf{T}(\mathbf{a}(n)) \tag{2.7}$$

where $\mathbf{a}(n)$ is the vector of actions of the players at iteration n. The equilibria of the system, if they exists, are the fixed points of mapping $\mathbf{T}$, i.e. they are the vectors $\mathbf{a}^*$ resulting as solution of $\mathbf{a}^* = \mathbf{T}(\mathbf{a}^*)$.

Nonlinear fixed-point problems can be solved by iterative methods using distributed algorithms [18] [19]. The most common updating strategies for the actions $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ based on mapping $\mathbf{T}$ are [18] [19]:

- *Jacobi scheme:* All components of $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ are updated *simultaneously,*

- *Gauss-Seidel scheme:* All components of $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ are updated *sequentially,* in a round-robin fashion, i.e. one after the other

- *Totally asyncronous scheme:* All components of $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ are updated *totally asynchronous* way. In this scheme, some components of a_k may be updated more frequently than others, and possibly outdated information on the other components can be used.

Note that *Totally asyncronous scheme* contains the *Jacobi* and *Gauss-Seidel schemes* as special cases. In general, these schemes converge to the fixed point under different conditions [18] [19].

Chapter 3

Energy-Efficient Routing for Wireless Sensor Networks

3.1 Introduction

In wireless sensor networks (WSNs), data from different nodes in a region are highly correlated. For example, if the data are random variables, e.g. temperature measurements, the measured values at nodes are spatially and temporally correlated. Transmitting all sensor data can increase traffic and data redundancy at the destination nodes. This may result in high energy consumption for the overall network. WSNs can benefit from multi-hop routing by using data aggregation along the route from the furthest sensor node to the network sink. In such a scenario, besides energy, correlation awareness is a significant feature that should be taken into consideration in routing in multi-hop WSNs. The routing decisions can significantly change when data aggregation is involved [20].

Most routing algorithms in WSNs aim to minimize the total transmission cost of transporting the data collected by nodes in a distributed manner. Routing protocols have been proposed for WSNs emphasizing various metrics depending on the application and design specifications. Energy-efficient routing algorithms allow WSNs to be deployed with smaller battery packs and to achieve longer lifetime for a given battery size [1] [2]. One classical energy-efficient routing algorithm is minimum energy routing (MER). The MER algorithm has been used to minimize transmission energy [21] [22] [23].

Routing with data aggregation aims to find the optimum network topology for maximum correlated data gathering to reduce the cost function in resource-limited sensor networks. Not surprisingly, correlation awareness has a great impact on routing algorithms and hence deserves a careful study in multi-hop WSNs. Recent work has looked at exploiting the data correlation by using data aggregation along the multi-hop path [3] [24] [25] [26] [27].

Scaglione and Servetto [24] study the interdependence of routing and data aggregation based on information theoretic analysis. Luo et al. [25] consider both transmission cost and aggregation cost during the decisions of each node on the routing process. Cristescu et al. [3] construct an optimal network correlated data gathering tree. The general cost minimizing optimization problem is shown to be NP-complete even for the simplifying assumption of the *self coding* model. The optimal solution is found to be between the shortest path tree (SPT) and the solution to the traveling salesman path. The authors propose different heuristic approximation algorithms to construct correlated data gathering trees and compare them with a simulated-annealing based algorithm which is proposed as a performance benchmark since the optimal routing solution is NP-hard.

Von Rickenbach and Wattenhofer [26] propose an optimal minimum energy gathering algorithm (MEGA) for *foreign coding* data aggregation model and an approximation algorithm called low energy gathering algorithm (LEGA) for *self coding* data aggregation model. In these data aggregation models, there is a restricted context of data aggregation. Once data is aggregated, it is not possible to alter the packet again in another node throughout the route, hence re-encoding is not possible. Based on the foreign coding model, the authors are proposing MEGA which requires maintenance of two trees, the coding tree which is simply a directed minimum spanning tree for raw data and the SPT for aggregated data. On the other hand, using the self coding model, the authors propose LEGA which uses a shallow light tree as the data gathering algorithm and achieves a $2(1 + \sqrt{2})$ approximation ratio. Although both MEGA and LEGA are optimal and near optimal approximation algorithms under their respective data aggregation models, their performance deteriorates in highly dense networks or in high correlation environments. The reason is that data redundancy among all nodes can not be exploited efficiently. Each source node can only aggregate once, hence its redundancy with other nodes (other than the next hop node) can not be eliminated.

Pattem et al. [27] investigate several tree construction schemes like routing-driven aggregation, aggregation-driven routing and static cluster-based routing for WSNs. Routing-driven aggregation performs opportunistic data aggregation over SPT, maximum data aggregation with minimum Steiner tree like routing is performed using aggregation-driven routing, and a static cluster-based routing is proposed that achieves near-optimal performance for diverse correlation levels. Unlike data aggregation models like foreign coding and self coding, in [20] [25] and [27], the authors also explore the in-network aggregation at several hops. The proposed models allow data aggregation (or *data compression*) at several hops, allowing greater reduction of the data.

This chapter extends the work above by proposing an energy-efficient routing strategy for correlated data gathering using a *multi-hop* data aggregation model which relies on the source data correlation between neighbor nodes. It differs from prior work by taking into consideration both the energy metric, interference and opportunity for multi-hop data aggregation. The routing algorithm incorporates correlation awareness into the system to minimize energy expenditure. A simple game-theoretic model with utility functions that account for data correlation is developed. Numerical studies are used to evaluate the performance of different routing algorithms and the potential advantages of our proposed approaches are presented.

3.2 System Model

Our focus is on applications where joint data aggregation and routing is applicable for data gathering in WSNs. Images collected by different sensor nodes in a video surveillance and image based tracking systems, or the collection of environment data measurements like temperature, humidity, vibration, sound or light from a field of sensors are possible application scenarios.

The collected data can be aggregated and routed in order to decrease redundancy. Therefore, we consider the problem of maximum correlated data gathering with a single sink, to which all the data has to be sent. All source nodes in the network collect, transmit and aggregate data. The total number of source nodes is N. Let V be the set of all nodes, which includes source nodes and one sink node, and E be the set of edges, or possible links among nodes. An edge is assumed to exist between two nodes if they are within communication range. Let $|A|$ denote the number of elements of set A. Then $|V| = N + 1$, and $|E| \leq N(N + 1)/2$. Define the network graph $G = (V, E)$. We assume that $R \subset V$ is the set of source nodes, where $|R| = N$. The sources are labeled Y_1 through Y_N. For the sake of simplicity, we assume there is only single sink labeled D where the data from all the nodes has to arrive. The algorithm proposed in this chapter can be extended to the case where there are multiple sinks and data from different subsets of nodes has to be sent into different sinks. All nodes are randomly distributed over a specified region.

In the following, we develop an algorithm that minimizes the total network energy. For the algorithm, we assume that there is a target bit error rate (BER) which ensures successful communication across a link. Our system has perfect error detection but no error correction capability. Automatic retransmission request is used so that a packet with error is retransmitted until received correctly. Suppose that the packet length is M. Then the probability of correct reception of the packet is $P_c(\gamma) = (1 - 2BER(\gamma))^M$ where $BER(\gamma)$ is bit error rate corresponding to a signal to interference and noise ratio (SINR) γ. The $BER(.)$ function will depend on the modulation scheme and the noise and interference environment. In this chapter, we assume a CDMA system, for which cumulative interference can be assumed to be Gaussian. We use non-coherent frequency shift keying modulation for which $BER(\gamma) = 0.5 \exp(-0.5\gamma)$ under Gaussian noise and interference. This equation can be used to find a target SINR γ^* for the system.

Our system uses synchronous direct-sequence CDMA (DS-CDMA) where nodes use variable spreading sequences. The spreading factor for each transmitter, L, can be adjusted to meet the target quality-of-service (QoS) (or target SINR) requirements. The minimum spreading gain between the nodes Y_i and Y_j to reach a certain target SINR, γ^*, is [28]

$$L_{i,j} = \frac{\gamma^* \left[\sum_{k=1,k\neq i,j}^{N} h_{k,j} P_k\right]}{h_{i,j} P_i - \gamma^* \sigma^2}, \tag{3.1}$$

where the link gain $h_{i,j} = 1/d_{i,j}^p$, $d_{i,j}$ is the distance of between the nodes Y_i and Y_j, p is the path loss exponent, which is usually between 2 and 4 for free-space and short-to-medium-range radio communication and σ^2 is the thermal noise power. The transmission rate, or bit throughput between nodes Y_i and Y_j in bits-per-second (bps) is determined by the spreading rate $L_{i,j}$: $\Omega_{i,j} = W/L_{i,j}$ where W is the system bandwidth.

The energy per bit per packet transmission, which has units of Joule/bits, represents the total amount of energy consumed in order to deliver one data bit to the destination. This chapter considers only the energy used for transmission, neglecting the energy used with reception and data processing. The energy per bit $E_b^{i,j}$ for packet transmissions between nodes Y_i and Y_j can be defined similarly as in [28]:

$$E_b^{i,j} = \frac{MP_i}{m\Omega_{i,j}P_c(\gamma)}, \tag{3.2}$$

where M is the packet length, m is the information bits in a packet, P_i is the constant transmit power for all $Y_i,\ i = \{1, 2, ..., N\}$.

To quantify the amount of data generated by each sensor node and data aggregation along the route, sources are associated with their data rates or weights as defined in [25]. It is assumed that data collected by the sensor nodes is correlated over geographical regions. Therefore, depending on the density of the WSNs in the field, the readings from nearby nodes may be highly correlated and hence contain data redundancies. Each source node Y_i generates data at a certain rate $\Psi(Y_i)$, where $\Psi(Y_i)$ is the data rate of source Y_i. It is important to note that data rate $\Psi(Y_i)$ refers to rate of symbol encoding of source Y_i, not the rate of data transmission. The data rate $\Psi_i(Y_i)$ shows the average number of bits per symbol used to encode the data source and has units bits/symbol. As shown in Fig. 3.1, the source nodes represented with circles in the network can either send their own raw data directly into the sink, or if there are other source nodes connecting to it, they can use the raw or aggregated data from other nodes to aggregate. Then, the aggregated data is sent to the sink along the node's route. Our main focus is on minimizing total network energy expenditures.

3.21 Data Aggregation Model

In this chapter, correlated data from multiple nodes is aggregated into one compressed data stream in order to reduce the network load. For data aggregation, we adopt the lossless step-by-step multi-hop aggregation model introduced in [25]. In this approach, each source node aggregates its data with that of the child nodes in a step-by-step

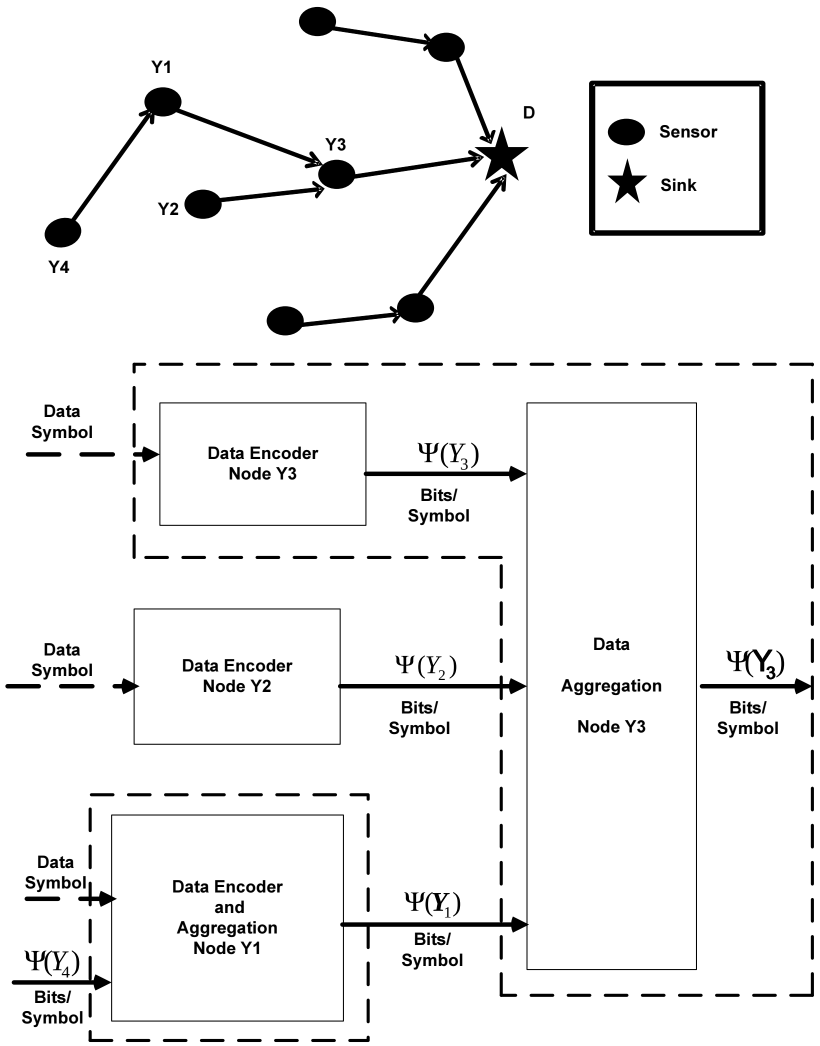


Figure 3.1: An example of data gathering tree. Data generated by encoder of each source node arrives at sink D by data aggregation through intermediate source nodes.

fashion. Let $\Psi(Y_i)$ denote the rate of data generated by node Y_i. Let $\mathbf{Y}_i$ be the set of node Y_i and all child nodes of node Y_i. Define $\Psi_i(\mathbf{Y}_i)$ as the net data rate at node Y_i due to data generated by node Y_i and its child nodes, after the aggregation process is done. For instance, suppose that node Y_i has q child nodes Y_1, Y_2, ..., Y_q. Then $\mathbf{Y}_i = \{Y_i, Y_1, Y_2, ..., Y_q\}$. The data received from a child node Y_k, $k = 1, .., q$ will be data aggregated from node Y_i and all of its child nodes, $\Psi_i(\mathbf{Y}_i)$. If Y_i is a leaf node with no child nodes, then it will forward only data generated by itself, $\Psi_i(\mathbf{Y}_i) = \Psi(Y_i)$.

The aggregation of q multiple inputs with source node Y_i is performed step by step with each node taking their turns depending on their arrival times. The parent node first aggregates its own data with newly received data. Then, the parent node aggregates the result with another node's data that arrives next. Therefore, for example in Fig. 3.1, node Y_3 first aggregates data from node Y_2 with its original data because it has no subtree and will probably arrive first. Then, node Y_3 saves the aggregated data as its temporary data rate as $\Psi^{temp}(Y_3)$; and so node Y_3 will aggregate it again with the data at node Y_1 and send the final result with data rate $\Psi_3(\mathbf{Y}_3)$, along its path for possible further aggregation until sink D. At the end of the aggregation process, the total data rate of node Y_3 becomes $\Psi_3(\mathbf{Y}_3)$. Multi-hop data aggregation discussed above is repeated at every intermediate source node along the route.

The reason for step by step aggregation of multiple inputs rather than joint aggregation schemes like the differential entropy model [29] is that storing multiple sources' data and aggregating them at once requires large memory and power for sensors. Moreover, the data reported with different nodes will arrive at the parent node at different times due to intermediate nodes' signal processing, distance between nodes, wireless medium characteristics, error-control schemes, interference, noise or transmitted powers.

Let $\Psi^{temp}(Y_i, Y_j)$ be the temporary data rate of a node after data generated by node Y_i is aggregated with data from one of its child nodes Y_j. This data rate is calculated as [25]:

$$\Psi^{temp}(Y_i, Y_j) = \max\left(\Psi(Y_i), \Psi(Y_j)\right) + (1 - \rho_{i,j}) \min\left(\Psi(Y_i), \Psi(Y_j)\right), \tag{3.3}$$

where $\rho_{i,j}$ is the correlation coefficient between nodes Y_i and Y_j. Similar to [25], in order to distinguish the correlation between the raw data and aggregated data, we use a "forgetting" factor for aggregated data. The correlation between aggregated data at two parent nodes is only a fraction of the data correlation calculated according to their distance.

3.3 Efficient routing framework for energy minimization

3.31 Energy per symbol and symbol throughput

Two components of the network will determine the energy consumption per symbol transmission of a node, namely the data rate of sources and transmission energy. Given the correlation models in Section 3.21, the energy per symbol transmission between nodes Y_i and Y_j, accounting for data redundancy through correlation can be defined as

$$\begin{aligned} E_s^{i,j}(\Psi_i(\mathbf{Y}_i)) &= E_b^{i,j}\left[\frac{\text{Joule}}{\text{bits}}\right]\Psi_i(\mathbf{Y}_i)\left[\frac{\text{bits}}{\text{symbol}}\right], \\ &= \frac{MP_i}{m\,\Omega_{i,j}\,P_c(\gamma)}\Psi_i(\mathbf{Y}_i)\left[\frac{\text{Joule}}{\text{symbol}}\right], \end{aligned} \tag{3.4}$$

where $\Psi_i(\mathbf{Y}_i)$ is the aggregated data rate at node Y_i and $\mathbf{Y}_i$ is the set of all q sources using node Y_i including Y_i, i.e. $\mathbf{Y}_i = \{Y_i, Y_1, Y_2, \ldots, Y_q\}$. The energy per symbol required has units Joule/symbol and indicates the total amount of per joule of energy consumption in order to deliver one data symbol to the destination without an error.

Similar to the definition of energy per symbol, the symbol throughput of a link between nodes Y_i and Y_j is defined as

$$\begin{aligned}\zeta_{i,j}(\Psi_i(\mathbf{Y}_i)) &= \frac{W}{L_{i,j}}\left[\frac{\text{bits}}{\text{second}}\right]\frac{1}{\Psi_i(\mathbf{Y}_i)}\left[\frac{\text{symbol}}{\text{bits}}\right], \\ &= \frac{\Omega_{i,j}}{\Psi_i(\mathbf{Y}_i)}\left[\frac{\text{symbol}}{\text{second}}\right].\end{aligned} \tag{3.5}$$

The symbol throughput has units symbol/second and symbolizes the total amount of symbol transmitted per second to the destination. Note that in a routing problem, the symbol throughput of data ζ_i generated by source Y_i is defined as the minimum of the symbol throughput of all links on the route from source Y_i to the data sink D, since the link with the least symbol throughput determines the symbol throughput of each source. This link is the bottleneck link for that particular source node. Note that the bottleneck throughput of nodes that are connecting to sink via same node is same and is the minimum of all bottleneck throughput of all nodes whose paths are connected to sink via same node.

3.32 Optimization Problem

The energy minimization problem in the network and physical layers can be formulated as follows:

$$\begin{aligned}&\underset{S_i \in X_i}{\text{Minimize}}\ \sum_{i=1}^{N}\sum_{k,l\in S_i} E_s^{k,l}(\Psi_k(\mathbf{Y}_k)), \\ &\text{subject to}\ \ \text{SINR}_{k,l} \geq \gamma^*,\ \ P_k = C,\end{aligned} \tag{3.6}$$

where $\text{SINR}_{k,l}$ is the received SINR at node Y_l for the link between nodes Y_k and Y_l, $P_k = C$ is the constant transmit power of node Y_k, $k = \{1, 2, ..., N\}$, $\mathbf{Y}_k$ is the set of all sources connected directly into node Y_k and including node Y_k, i.e. $\mathbf{Y}_k = \{Y_k, Y_1, Y_2, \ldots, Y_q\}$, S_i is the set of relaying and aggregating nodes used for source Y_i, and X_i is the set of all possible relaying and aggregating nodes for a route of source Y_i.

The optimum routing solution is hard to determine when each sensor uses the data aggregation model in Section 3.21 since at each route, multi-hop aggregation is employed. The joint optimization of transmission cost and data aggregation is shown to be NP-complete even for simplistic assumption of self-coding data aggregation model [3]. Therefore, finding the optimal energy minimization algorithms is a NP-hard optimization problem. Consequently, the solution we propose is a decentralized energy minimization algorithm using correlation structure of the network. We propose a game theoretic formulation which can be shown to converge to a local optimal solution with relatively low complexity and in a distributed fashion.

Game theoretic interpretation

The above problem can be formulated as a congestion game model which can be shown to be isomorphic with a potential game. In this game, the players are the source nodes in quest for routes, the source nodes used throughout the route are the shared facilities, the action of the players is the selection of a group of facilities that form a route into the sink, and costs can be associated with various route selections.

Formally, the proposed game-theoretic routing model for correlation aware routing considers the route selection of each sensor node as a congestion game Γ as defined in Section 2.33 with a player set $\mathcal{P} = \mathcal{N}$ where $\mathcal{N} = \{Y_1, \ldots, Y_N\}$ denotes the set of players, i.e. the source nodes in our game, a set of facilities $\mathcal{F} = \{1, \ldots, m_f\}$, the strategy (or action) space $\mathcal{A}_i = X_i$ of player (or source node) Y_i, and the cost function $w_f : \mathbf{S} \to \mathbb{R}$ associated with using the facility f where $\mathbf{a} = \mathbf{S} = (S_1, \ldots, S_N)$ is a *state of the game* in which player Y_i chooses strategy (or action) $S_i \in X_i$. We define $\theta_f(\mathbf{S})$ as the subset of *sources nodes* directly connected to facility f including the source node at facility f, that is $\theta_f(\mathbf{S}) = \{Y_i | f \in S_i\}$. The players aim at choosing strategies $S_i \in X_i$ minimizing their individual cost, where the cost $\delta_i(\mathbf{S})$ of player Y_i is given by $\delta_i(\mathbf{S}) = \sum_{f \in S_i} w_f(\theta_f(\mathbf{S}))$. We define utility function for source Y_i in our congestion game as

$$\begin{aligned} u_i : \mathbf{S} \to \mathbb{R}, \;\; u_i(S_i, S_{-i}) &= -\delta_i(\mathbf{S}), \\ &= -\sum_{f \in S_i} w_f(\theta_f(\mathbf{S})), \end{aligned} \tag{3.7}$$

where $S_{-i} = (S_1, S_2, \ldots, S_{i-1}, S_{i+1}, \ldots, S_N)$ is the strategy space of player Y_i's opponents. The game performance is influenced by the selection of cost functions $w_f(\theta_f(\mathbf{S}))$ for facilities. We propose and compare several metrics in the next section.

3.4 Facility cost selection for the congestion game

We consider the problem of constructing the *minimum energy correlated data gathering tree*. In setting up the costs for facilities, we can consider the following parameters:

1. Energy spent for relaying bits or symbols on outgoing links from the facility,
2. Opportunity for aggregation.

3.41 Minimum Energy Routing

The classic approach is to consider only energy minimization. We denote this classic approach as MER (e.g. [21] [22]). For MER, the following utility function is used

$$u_i(S_i, S_{-i}) = -\sum_{f \in S_i} E_b^f, \tag{3.8}$$

where E_b^f is the cost of using the link of facility f, i.e. energy per bit required on ongoing links from facility f, through the strategy (or route) S_i and S_{-i}.

3.42 Correlation-aware routing for energy minimization

In WSNs, constructing the correlated data gathering routes is an important task for cost minimization [3] [29]. To account for data correlation and potential for data aggregation in the network, we propose correlation-aware routing (CAR) game formulation for the solution of energy per symbol minimization problem of (3.6). For CAR, given a network, the problem is to induce the formation of a maximal correlated data aggregation tree from each reporting sensors (sources) to the sink. For CAR, we define the cost of using the link of facility f

$$\begin{aligned} w_f(\Psi_f(\theta_f(\mathbf{S}))) &= E_b^f \Psi_f(\theta_f(\mathbf{S})), \\ &= E_s^f(\Psi_f(\theta_f(\mathbf{S}))), \end{aligned} \tag{3.9}$$

where $\Psi_f(\theta_f(\mathbf{S}))$ is the total aggregated data rate at facility f. The utility function $u_i(S_i, S_{-i})$ of source Y_i is given by

$$\begin{aligned} u_i(S_i, S_{-i}) &= -\sum_{f \in S_i} \left(w_f(\Psi_f(\theta_f(\mathbf{S}))) - w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) \right), \\ &= -\sum_{f \in S_i} E_b^f [\Psi_f(\theta_f(\mathbf{S})) - \Psi_f(\theta_f^{-i}(\mathbf{S}))], \end{aligned} \tag{3.10}$$

where $\Psi_f(\theta_f^{-i}(\mathbf{S}))$ is the total aggregated data rate at facility f when source Y_i is not present. Note that when Y_i is not present on facility f, all the sub-trees rooted at Y_i are pruned because we are interested in the correlation contribution of only node Y_i (see Fig. 3.2). Each player aims to choose the strategy $S_i \in X_i$ to maximize its utility function, to find the best routes that will result in maximum data aggregation.

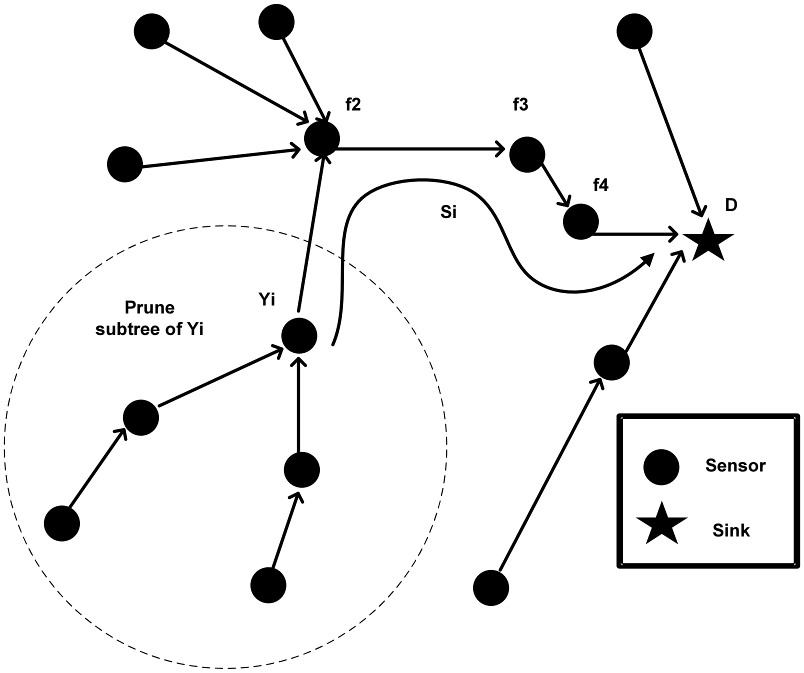


Figure 3.2: Pruning of the subtree rooted at Y_i to calculate $\Psi_f(\theta_f^{-i}(\mathbf{S}))$ over facilities $\{Y_i, f_2, f_3, f_4\} \in S_i$.

Potential Game Formulation for CAR

Nash equilibrium (NE) represents a stable state in which no player wishes to leave unilaterally its own strategy in order to improve the value of its utility function. In certain classes of games, the game converges to a NE when a *best* or *better response* adaptive strategy is employed. In what follows, we show that the congestion game associated with CAR is isomorphic with a potential game, for which a best response strategy is shown to converge to a NE. More specifically, we can show that games defined within these algorithms are exact potential games, by defining an exact potential function which is the optimization target minimization of the game, that exactly reflects changes in individual utility functions.

A potential game as defined in Section 2.32 is a normal form game such that any changes in the utility function of any player in the game due to an unilateral deviation by the player is reflected in a global function. We assume that in the normal form game each player takes actions sequentially and at each stage of the game players choose their actions which improve their utility functions.

Generally in potential games, the updating procedure is carried out sequentially. Note that, while sequential updates may require additional synchronization overhead, a simple approximation implementation may be based on randomized access which on average will result in sequential updates. This can be shown experimentally to have minimal impact on convergence properties [30]. We will demonstrate that correlation aware routing with utility functions given by (3.10) is an exact potential game (EPG) with the exact potential function,

$$\mathcal{P}(S_i, S_{-i}) = -\sum_{f=1}^{m_f} w_f(\Psi_f(\theta_f(\mathbf{S}))), \tag{3.11}$$

where m_f is the total number of facilities used in the directed graph G and $m_f = N$ since all nodes are used as a facility in the network.

Theorem 3.4.1. *CAR defined by utility function (3.10) and the potential function (3.11) is an EPG.*

Proof. See Appendix 3.A. □

Corollary 3.4.2. *CAR always has NE and converges to NE strategies by using a best response adaptive strategy.*

Corollary 3.4.3. *Any NE of the CAR is locally optimal, i.e. in a NE, all the players can't reduce the energy unilaterally and thus the game reaches a local minimum total energy for CAR.*

Each source $Y_i \in \mathcal{N}$ in CAR algorithm updates its strategy S_i through maximization of its corresponding utility (3.10). Hence, at each iteration, each user finds the best routes (sequential updates) that will increase its utility. From Theorem 3.4.1, the potential function $\mathcal{P}(S_i, S_{-i})$ will continue to increase until it reaches a local maximum point using best response dynamics. Since the potential function of any strategy profile is finite, it follows that every sequence of improvement steps and the number of iterations to converge is finite due to the finite improvement property (FIP) of best response dynamics in congestion games [12] [16].

One of the primary drawback of the most best response strategies like CAR algorithm introduced here, is the computational complexity, which grows linearly with the cardinality of the strategy space i.e. the network size N. To address this issue, for large network sizes, the search for better routes can be restricted on k-hop neighborhood nodes $\mathcal{N}_k(Y_i)$ of each source node Y_i in order to reduce the computational complexity. This is based on the observation that a node is highly correlated within a certain radius of its neighborhood. This is a natural assumption for sensor networks, since the correlation decreases with the increase of the distance between nodes, hence the local correlation is dominant. Better response can be another improvement of the best response, where at each step, the player updates as long as the randomly selected strategy yields a better performance. The dramatically reduced computation is the tradeoff with the convergence speed. Moreover, there may be multiple NE in a potential game and the performance of different equilibria may vary.

The procedure for sequential updates of routes with best response dynamics of CAR can be summarized as following:

Initialization : Construct minimum energy routes using the distributed algorithms like Dijsktra's Algorithm [18]. Then, use the following iterations for energy minimization.

Repeat : At each iteration n.

- For each of the source node $Y_i,\ i \in \{1, 2, ..., N\}$,

    1. Select $S_i \in X_i$ that maximizes the utility function in (3.10).
    2. Update routing strategy S_i for source Y_i.

Until : The stopping criteria Δ is met.

The stopping criterion Δ is the minimum number of iteration steps κ for the algorithm to converge, where κ is a counter which adds one after each updating process.

The total network energy consumption per symbol transmission for CAR algorithm is

$$E_s^{total,CAR} = \left(\sum_{f=1}^{m_f} w_f(\Psi_f(\theta_f(\mathbf{S}))) \right). \tag{3.12}$$

The symbol throughput of source Y_i is

$$\lambda_i = \min_{\forall k,l \in S_i} \quad \zeta_{k,l} \text{ and } S_i \in X_i. \tag{3.13}$$

The total symbol throughput in the network is the symbol throughput of sum of bottleneck throughput of each source, i.e.

$$\zeta^{total} = \sum_{i=1}^{N} \lambda_i. \tag{3.14}$$

3.43 Minimum Energy Data Gathering Algorithm (MEGA)

MEGA is an algorithm that tries to minimize the aggregation and raw data costs jointly using *foreign coding* aggregation model [26]. In *foreign coding* aggregation model, once aggregated data is not aggregated again over the next facilities. The resulting topology is a combination of two tree constructions, namely the coding tree and the shortest path tree (SPT) (or in our results MER tree) (also see Fig. 3.3). The coding tree is for the aggregation of the raw data and the MER (or SPT) tree is for the aggregated data. Define the set of facilities used for MER tree of node Y_i as M_i and coding tree as C_i and the strategy of each source node S_i. With foreign coding data aggregation model, the utility function of MEGA is

$$u_i(S_i, S_{-i}) = -E_b^{i,j}\Psi_i(Y_i) - [(1-\rho_{i,j})\Psi_i(Y_i)] \sum_{f \in M_j} E_b^f, \tag{3.15}$$

where $\rho_{i,j}$ is the correlation coefficient between nodes Y_i and Y_j and $Y_j \in C_i$, where Y_j the next hop node of Y_i in the coding tree. In other words, MEGA tries to find the next hop Y_j that minimizes total energy consumption over the coding tree C_i and MER tree M_j of node Y_j.

After constructing the coding trees according to utility function of (3.15) for MEGA, the total energy per symbol consumption in the network is

$$E_s^{total,MEGA} = \sum_{\substack{i=1,\\ j \in C_i}}^{N} \left(E_b^{i,j}\Psi_i(Y_i) + \left([(1-\rho_{i,j})\Psi_i(Y_i)] \sum_{f \in M_j} E_b^f \right) \right). \tag{3.16}$$

The symbol throughput of each source Y_i is calculated as the minimum throughput of multi-hop links from each source to the sink. The links of each source Y_i consists of the the links in the coding tree C_i and MER tree M_j of next hope node Y_j. Therefore, the symbol throughput of source Y_i is given in (3.13) and the total symbol throughput in the network is given in (3.14).

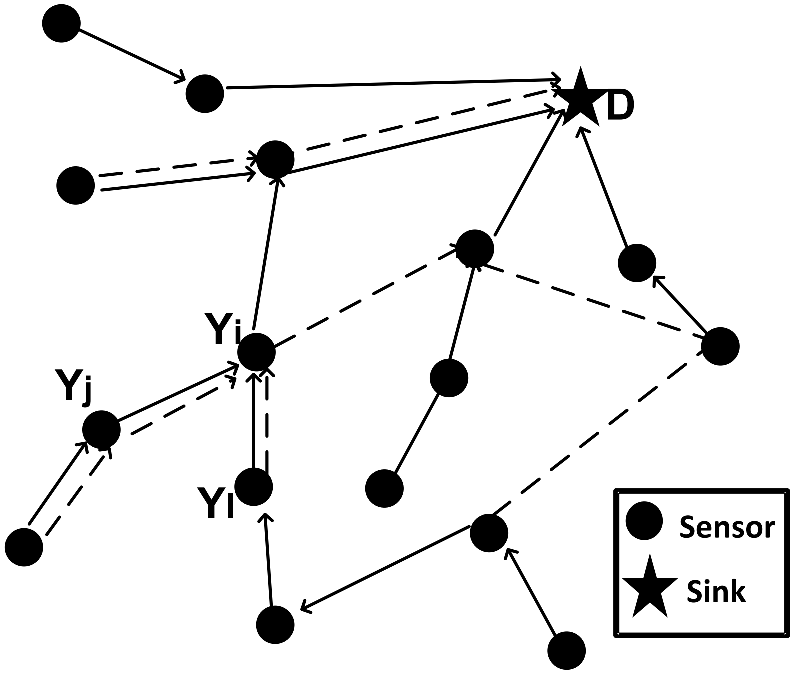


Figure 3.3: Multiple transmissions using two different links from node Y_i in MEGA. The coding tree is shown with dashed arrows and MER tree is with solid arrows.

3.5 Simulation Results

In this section, we present an extensive set of numerical results to evaluate the performance of our proposed CAR algorithm with the other classical approaches, MER and MEGA. For MER and CAR algorithms, the data generated by each source is aggregated in multiple nodes throughout the route to the sink using the aggregation model in Section 3.21. For nodes randomly deployed in a $2D$ field, the impact of network size and correlation coefficient and number of iterations required to converge for different algorithms are compared. In accordance with design parameter, the performance of our proposed CAR algorithm is compared in terms of improvements in energy savings with MER and MEGA.

For fair comparison of the all algorithms, the total network energy per symbol is compared under a constant total symbol throughput $\zeta^{total} = \sum_{i=1}^{N} \lambda_i$ requirement. In other words, the total *effective* energy per symbol of all algorithms is compared when the total amount of symbol obtained per each second is same for all the algorithms. Therefore, by this way, the energy and throughput gains are incorporated into one performance metric of *effective* energy improvements.

3.51 Simulation Setup

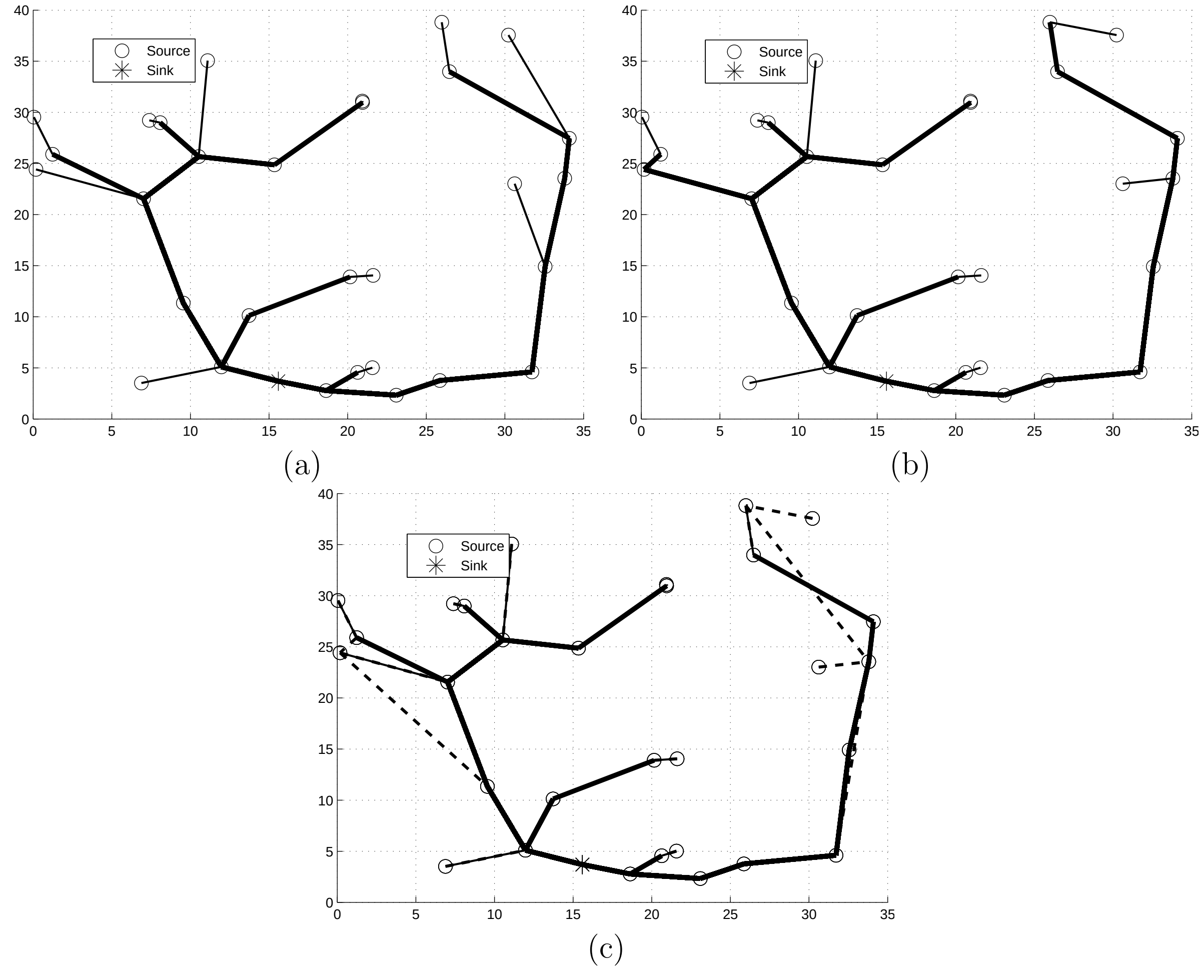
The number of sensor nodes in the network is varied between $N = 10$ to $N = 40$, which is uniformly distributed over a square area of dimension $40m \times 40m$. We adopt *Gaussian random field* data correlation model that is frequently encountered in practice [29]. In this model, correlation coefficient $\rho_{i,j}$ between nodes Y_i and Y_j decreases exponentially with the increase of the distance between nodes $d_{i,j}$, i.e. $\rho_{i,j} = \exp(-d_{i,j}^2/c)$ where c is the correlation constant where $c = 0m^2$ corresponds to no aggregation, $c = 100m^2$ corresponds to low correlation and $c = 1000m^2$ corresponds to high correlation environment. The path loss exponent $p = 2$. In simulations, we use a “forgetting” factor of 0.8 per link, that is, if data is aggregated at a node Y_i, then the correlation $\rho_{i,j}$ between that node's aggregated data and its parent node Y_j in the routing tree is reduced to 0.8 of its original value. The noise power is $\sigma^2 = 10^{-13}$ Watts, which corresponds to thermal noise power for a bandwidth of $W = 1$ Mhz. We choose the equal transmit powers of all nodes to be 110 dB above the noise floor ($P_i = 10^{-2}$ Watts, $\forall i \in \mathcal{N}$). The target SINR is selected to be $\gamma^* = 5$ (7 dB). We assume that each packet contains 80 bits of information and no overhead (i.e., $m = M = 80$). The generated raw data rate of each source, $\Psi(Y_i)$, is assumed to be constant for all $Y_i \in \mathcal{N}$ and without loss of generality, each symbol is represented with 1 bit of information, i.e. $\Psi(Y_i) = 1$ bits/symbol. The total effective energy consumption of all algorithms are compared when the required symbol throughput is $\zeta^{total} = 100$ kbps. The results are simulated and averaged over 100 different network configurations for each routing algorithm.

The MER algorithm uses Dijkstra's algorithm to find the best routes from sink to source nodes with one iteration. Although MER was not proposed in the context of data aggregation, we set-up the paths according to their corresponding utility functions, and then aggregate data opportunistically based on the routes set-up, this approach is called routing-driven aggregation [27]. MEGA is obtained after MER is constructed, hence the required number of iterations for convergence is two. CAR is implemented iteratively based on the best response strategy described in the previous section. Note also that, we start CAR algorithm with the same tree structure as MER respectively, hence at first iteration their total effective energy values are equal. MEGA uses foreign coding model, i.e. the aggregation is performed only at next hop, while CAR and MER performs the multi-hop aggregation model described in Section 3.21.

Through multiple iterations, the algorithms change the initial routing tree. Fig. 3.4 demonstrates the branches of the constructed trees for MER, MEGA and CAR algorithms under the same network topology for $N = 30$, $c = 1000$. Thick lines indicate the regions where the data aggregations are performed. Note that for MEGA, coding tree is shown with dashed lines, whereas MER tree is shown with solid lines. The results show that different routing metrics with different utility functions lead to paths with significantly different trees or network connectivity. For example, MER tends to discover paths with lower energy while CAR searches for minimum energy routing paths to aggregate more efficiently.

3.52 Effective Energy Improvements

Fig. 3.5 shows the total effective energy per symbol in the network for CAR, MER and MEGA versus increasing number of nodes from $N = 10$ to $N = 40$ in the



(a) (b) (c)

Figure 3.4: Selected paths of each source and the tree structures of different routing strategies for $N = 30$, $c = 1000$. (a) MER. (b) CAR. (c) MEGA.

network for three data correlation settings ($c = 0$, $c = 100$ and $c = 1000$). The energy improvements returned by CAR algorithm over MER and MEGA algorithms increase gradually as the network size grows. The reason is that MER algorithm is optimized only for routing whereas MEGA and CAR algorithms are optimized for both data aggregation and routing. However, MEGA does not perform multi-hop aggregation and its performance deteriorates at large network sizes. For example for $N = 10$, the percentage improvement of CAR algorithm compared to MER is 4.31 percent and compared to MEGA is 14.30 percent, whereas for $N = 40$ the improvements are around 25.35 and 52.18 percent compared to MER and MEGA respectively, at $c = 100$. Note also that MEGA's performance increases as the correlation constant increases and is almost same with MER algorithm at $c = 1000$ even though MER algorithm performs multi-hop data aggregation. The reason is that it can optimize both data aggregation and routing more efficiently at higher correlation levels.

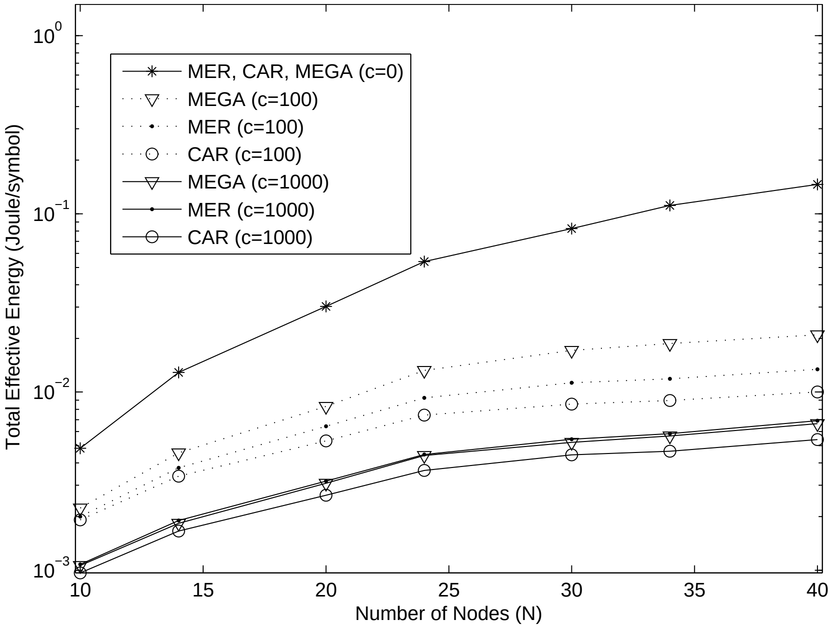


Figure 3.5: Total effective energy versus number of nodes, N.

Note that effective energy incorporates both the energy and throughput gains of the above mentioned algorithms into one performance metric. At low correlations, effective energy improvements of CAR algorithm over MER algorithm are higher than the effective energy improvements at high correlations. The reason is that at high correlation levels, correlation between all nodes is so high that no matter which path is used for data aggregation and routing, the transmitted data rate can be reduced significantly. Therefore, there is not much room for further improvement by choosing a better path using the CAR algorithm after MER is established at strong correlation levels. For example, if full data aggregation is performed, i.e. when $c \to \infty$, the performance of all algorithms are expected to perform same. From Fig. 3.5, we also see that compared to no data aggregation case, i.e. $c = 0$, the improvements of CAR algorithm is 93.15 percent for $c = 100$ and 96.29 percent for $c = 1000$ when $N = 40$. These results show the significant performance gain and advantage of performing in-network data aggregation of correlated data compared to no aggregation schemes in the network.

3.53 Impact of Correlation Coefficient

Fig. 3.6 shows the total effective energy of CAR, MER and MEGA algorithms as the correlation constant c increases from 0 to 1000 for different network sizes. The performances of all algorithms improve as the data correlation becomes larger for the same network size. This shows that the routing algorithm performances can greatly benefit from data aggregation by reducing redundancy among correlated data. Note that MEGA and CAR algorithms benefit more from the correlation increments than the MER algorithm since both algorithms are optimized for data aggregation. For example, for correlation parameter of $c = 200$, the percentage energy improvements of CAR over MER and MEGA are on the order of 16.73 and 31.46 for network size of $N = 20$ respectively, whereas, for correlation parameter of $c = 800$, the percentage improvements are 16.36 and 16.61 respectively. Moreover, the percentage improvements of CAR at $c = 1000$ are 16.70 and 10.40 compared to $c = 0$ (no aggregation) case when $N = 20$ and $N = 10$ respectively. MER and CAR algorithms achieve less energy consumptions for the smaller network size ($N = 10$) than larger network size ($N = 20$) under all correlation situations. In this case, both the additional number of transmissions and the interference in the network increase with increasing number of nodes which is observed from Fig. 3.5.

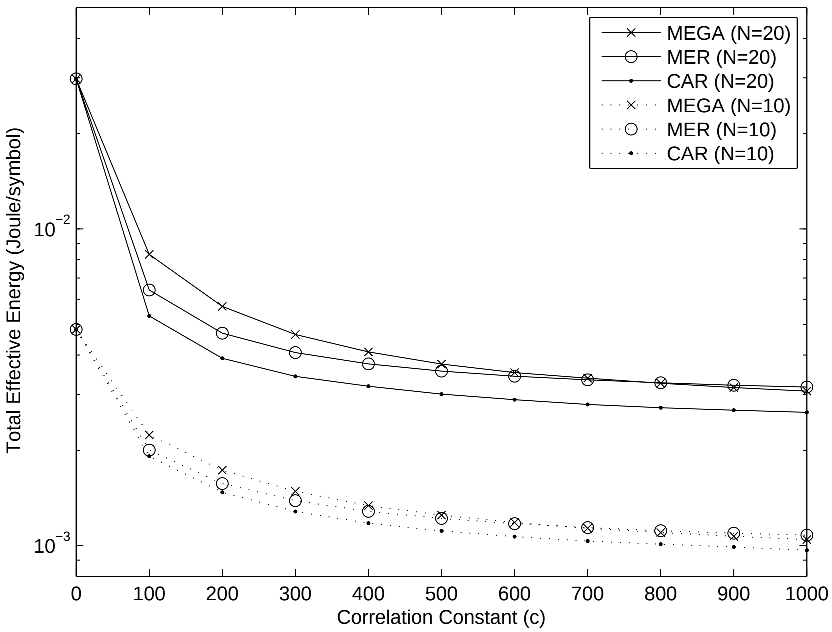


Figure 3.6: Total effective energy versus correlation constant (c).

3.54 Convergence of the Algorithm

In addition to the effectiveness of CAR, the number of iterations for the convergence of the proposed distributed algorithm is also important. In this section, we show the minimum number of required iterations for the convergence of the CAR algorithm for different network sizes. Fig. 3.7 shows the normalized effective energy consumptions of CAR compared to MER versus number of iterations. The minimum number of iterations κ required for the total effective energy to converge is 3 to 4 iterations for the network size ranging from 10 to 40 nodes.

The existence of NE for CAR algorithm is supported by the convergence of curve of Fig. 3.7. Furthermore, CAR algorithm involve small number of iterations after MER is established, and it can be implemented efficiently in a distributed fashion.

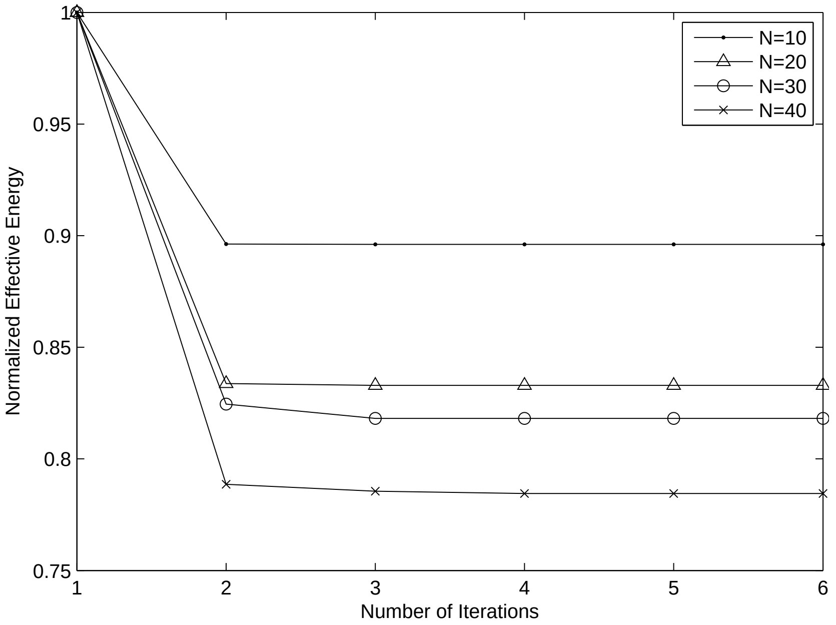


Figure 3.7: Normalized effective energy consumption of CAR with respect to MER versus number of iterations for $c = 1000$.

3.6 Conclusions

In this chapter, we addressed the problem of efficient transmission structure in wireless sensor networks where each source transmits and aggregates the correlated data over intermediate nodes to the sink. We have investigated the impact of efficient data aggregation in establishing routing paths towards the sink for energy minimization problem. For correlation aware routing, we have proposed a distributed iterative protocol based on a game theoretic framework, which is shown to converge within a couple of iterations. We have also shown that, by accounting for correlation structure and multi-hop aggregation in constructing routes, significant effective energy gains over classic approaches can be achieved.

Appendix 3.A: Proof of Theorem 3.4.1

Suppose there exists a potential function of the congestion game Γ:

$$\mathcal{P}(S_i, S_{-i}) = -\sum_{f \in \mathcal{F}} w_f(\Psi_f(\theta_f(\mathbf{S}))), \tag{3.17}$$

where f is the facility and $\mathcal{F}$ is the set of facilities as defined in Section 3.32. Let $S_i \in \mathbf{S}$ be the strategy of source $Y_i, i = 1, ..., N$, i.e. the collection of nodes used for relaying and aggregating and $S_i' \in \mathbf{S'}$ be another strategy where $\mathbf{S'} = (S_1, ..., S_{i-1}, S_i', S_{i+1}, ..., S_N)$. Then,

$$\begin{aligned} \mathcal{P}(S_i, S_{-i}) &= -\sum_{f \in \mathbf{F}} w_f(\Psi_f(\theta_f(\mathbf{S}))), \\ &= \left(-\sum_{f \in S_i \backslash S^*} w_f(\Psi_f(\theta_f(\mathbf{S}))) \right) + \\ &\quad \left(-\sum_{f \in S_i' \backslash S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) - \sum_{f \in S^*} w_f(\Psi_f(\theta_f(\mathbf{S}))) \right) + \\ &\quad \left(-\sum_{f \in \mathcal{F} \backslash \{S_i \bigcup S_i'\}} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) \right), \end{aligned} \tag{3.18}$$

where S^* denotes the common facilities used by the strategies S_i and S_i', i.e. $S^* = S_i \bigcap S_i'$. Define,

$$Q(S_{-i,-i'}) = -\sum_{f \in \mathcal{F} \backslash \{S_i \bigcup S_i'\}} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))). \tag{3.19}$$

Then,

$$\begin{aligned}\mathcal{P}(S_i, S_{-i}) = &\left(-\sum_{f\in S_i\setminus S^*} w_f(\Psi_f(\theta_f(\mathbf{S})))\right)\\ &+\left(-\sum_{f\in S_i'\setminus S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) - \sum_{f\in S^*} w_f(\Psi_f(\theta_f(\mathbf{S})))\right)\\ &+Q(S_{-i,-i'}).\end{aligned} \tag{3.20}$$

If source Y_i changes its strategy from S_i to S_i', then the potential function becomes,

$$\begin{aligned}\mathcal{P}(S_i', S_{-i}) = &\left(-\sum_{f\in S_i\setminus S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}')))\right)+\\ &\left(-\sum_{f\in S_i'\setminus S^*} w_f(\Psi_f(\theta_f(\mathbf{S}'))) - \sum_{f\in S^*} w_f(\Psi_f(\theta_f(\mathbf{S}')))\right)\\ &+Q(S_{-i,-i'}).\end{aligned} \tag{3.21}$$

Note that $Q(S_{-i,-i'})$ and $-\sum_{f\in S^*} w_f(\Psi_f(\theta_f(\mathbf{S}')))$ are not affected by the strategy changing of source Y_i. Therefore,

$$\begin{aligned}\mathcal{P}(S_i', S_{-i}) - \mathcal{P}(S_i, S_{-i}) = &\left(-\sum_{f\in S_i\setminus S_i^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))) - \sum_{f\in S_i'\setminus S^*} w_f(\Psi_f(\theta_f(\mathbf{S}')))\right)-\\ &\left(-\sum_{f\in S_i\setminus S^*} w_f(\Psi_f(\theta_f(\mathbf{S}))) - \sum_{f\in S_i'\setminus S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S})))\right).\end{aligned} \tag{3.22}$$

From (3.10) and the definition for $w_f(.)$ in (3.9),

$$\begin{aligned}u_i(S_i', S_{-i}) - u_i(S_i, S_{-i}) &= \left(- \sum_{f \in S_i'} \big(w_f(\Psi_f(\theta_f(\mathbf{S}'))) - w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))) \big) \right) - \\ &\quad \left(- \sum_{f \in S_i} \big(w_f(\Psi_f(\theta_f(\mathbf{S}))) - w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) \big) \right), \\ &= \left(- \sum_{f \in S_i' \backslash S^*} \big(w_f(\Psi_f(\theta_f(\mathbf{S}'))) - w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))) \big) \right) - \\ &\quad \left(- \sum_{f \in S_i \backslash S^*} \big(w_f(\Psi_f(\theta_f(\mathbf{S}))) - w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) \big) \right),\end{aligned}$$

$$\begin{aligned}&= \left(- \sum_{f \in S_i \backslash S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) - \sum_{f \in S_i' \backslash S^*} w_f(\Psi_f(\theta_f(\mathbf{S}'))) \right) \\ &\quad - \left(- \sum_{f \in S_i \backslash S^*} w_f(\Psi_f(\theta_f(\mathbf{S}))) - \sum_{f \in S_i' \backslash S_i} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))) \right).\end{aligned}$$

Note that

$$- \sum_{f \in S_i \backslash S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))) = - \sum_{f \in S_i \backslash S^*} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))),$$

and

$$- \sum_{f \in S_i' \backslash S_i} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}'))) = - \sum_{f \in S_i' \backslash S_i} w_f(\Psi_f(\theta_f^{-i}(\mathbf{S}))).$$

Hence,

$$u_i(S_i', S_{-i}) - u_i(S_i, S_{-i}) = \mathcal{P}(S_i', S_{-i}) - \mathcal{P}(S_i, S_{-i}). \tag{3.23}$$

Then, $\mathcal{P}(S_i, S_{-i})$ defined in (3.11) is an EPG of game Γ.

Chapter 4
Throughput Maximization for Correlated Data Routing in Wireless Sensor Networks

4.1 Introduction

Wireless sensor networks (WSNs) are used primarily for collecting environmental information. In large scale WSNs, a large number of small-sized and low-powered sensor nodes are deployed over a geographical area. Each sensor node consists of data sensing, processing and communication parts. By sensing a geographical area, the data generated by each sensor has to be transmitted into the sink nodes, possibly in a multi-hop fashion.

Sensor nodes in a certain region have knowledge of only nodes in their neighborhoods, so efficient optimization of communication patterns, like resource allocation and network connectivity is best accomplished in a distributed manner. Efficient decentralized routing algorithms are required for WSNs which use minimum communication overhead. The main goal of most routing algorithms in WSNs is to minimize the total transmission cost of transporting the data collected by nodes. Different objectives can be assigned for many routing protocols emphasizing various metrics depending on the application requirements in WSNs. For example, low-latency can be an important task for early disaster warning applications and in timely detection of events where the data generated by sensors need to be delivered to the sink nodes as quickly as possible, energy-efficiency may be crucial for network lifetime of sensor nodes for periodic monitoring situations. Reliable communication, support for mobility, robustness or fault-tolerance can be other application requirements.

One classical approach for an energy-efficient routing algorithm is minimum energy routing (MER). The MER algorithm has been used to minimize transmission energy [23]. Interference-aware routing (IAR) strategies are shown to achieve better performance results than MER for ad hoc networks [28] [31]. Mahmood et al. [28] studied the impact of interference on performance of multi-hop wireless ad-hoc network. They showed that by taking the interference into consideration, routes derived from the optimization problem often yield noticeably better throughput than the minimum energy path route. Kwon et al. [31] take into account the interference created by existing flows in the network. The improvement in overall network performance is obtained by finding paths to detour automatically around a congested hot-spot area in the network. However, interference aware routing has not been explicitly studied in the context of WSNs.

In-network data aggregation is a technique used in WSNs to eliminate data redundancy [11], minimize energy [26] or improve the network lifetime [4]. Using proper in-network data aggregation techniques may significantly reduce the amount of data transmission and communication load, hence improve networks' overall performance. WSNs can also benefit from multi-hop routing by using multi-hop data aggregation en-route to the sink. Therefore, besides energy and interference awareness, correlation awareness is an significant feature that should be taken into consideration in routing for multi-hop WSNs. Multi-hop data aggregation is more suitable for applications when a strong correlation exists in the sensor readings, or for collecting summarized data when some of the measurements are more important, for example highest temperature sensor reading (for fire alarm warnings). The routing decisions can significantly change when data aggregation is involved [5] [20].

The network load and redundancy among sensed data can be reduced by exploring the data correlation and using in-network processing through sensor routing algorithms. In network processing of sensor data increases the amount of data processing within the network, but can reduce the amount of traffic in the network significantly. The relationships between data aggregation and routing have been studied extensively under various circumstances [3] [6] [11] [25] [32].

Goel et al. [32] present a randomized logarithmic approximation algorithm when the joint entropy of multiple sources if a concave function of the number of sources. Luo et al. [25] consider both transmission cost and aggregation cost during the decisions of each node on the routing process. General cost minimizing optimization problem is shown to be NP-complete in [3]. Duarte-Melo et al. [8] present trivial upper bound of *data throughput per source* as W/N (W is the transmission capacity and N is the number of nodes), which can be achieved when the sink is 100% busy in receiving for many-to-one communications. This upper bound is achievable under certain conditions such as when all sources can directly (via single hop) transmit to the sink node. The authors also present different upper bounds if communication occurs in multi-hop for a randomly deployed network communication [8]. In [6] [20] and [25], the authors are exploring the in-network aggregation at several hops. The proposed models allow data aggregation at several hops which allow greater data compression.

In many large and dense WSN applications, it is of interest to maximize the total aggregated data rate at which the network can support efficient delivery of large amounts of data. Moreover, in time-critical applications, sensor data must be delivered with time constraints so that appropriate actions can be taken in real-time (after time deadline the data may be useless) [33]. In certain delay constraints, sensor network should be able to maintain a certain throughput in order to satisfy the quality-of-service (QoS) requirements and to guarantee the stability of the network while satisfying latency constraints in a practical system. Note that maximizing throughput and minimizing energy consumption are usually conflicting to each other and a trade-off is involved for data collection in WSN applications. It is important to evaluate the energy-throughput trade-offs for data gathering in WSNs. For example, Yu et al. [6] explores the energy-latency trade-offs of packet transmission by minimizing overall energy cost of sensor nodes subject to latency constraints.

In most related papers discussed above, either by using simplistic aggregation models, correlation-awareness is not exploited efficiently for in-network data aggregation or no explicit solutions are proposed for throughput efficiency taking into consideration energy, interference impact and opportunity for data aggregation. In this chapter of the dissertation, we propose an efficient throughput maximization routing strategy for data gathering in sensor networks where correlated data is periodically flowed from a set of sources to a common sink over a tree-based routing topology. Toward this end, correlated data gathering tree is constructed using a multi-hop data aggregation model for the routing problem. The routing algorithm developed can also be combined for interference and correlation awareness to maximize the throughput. We design a suitable utility function such that when each node optimizes its own utility function, this leads to optimization of the global objective function during the route construction process. Extensive simulations were conducted for both proposed and other classical routing algorithms and significant throughput gains are obtained under a wide range of system setups. We also compare our proposed routing algorithms with the optimal routing solution for a small network size. Although the optimal routing solution requires a centralized implementation and is NP-complete, it provides a performance benchmark for the proposed routing algorithms' results.

4.2 System Model

In WSNs, the data collection can be triggered either by external sources like *queries* or *events* or by continuous monitoring without external triggering. The collected correlated data can be aggregated during routing in order to decrease redundancy of released traffic. For continuous monitoring, communication is sustained over long periods of time, and throughput or bandwidth utilization are often not the biggest concern in the network due to the slow rate at which data is collected. Some applications, however, may require higher data rates or generate bursty traffic which must reach the sink node by a deadline. For example, in query or event-driven models throughput and bandwidth utilization can be an important concern for the timely delivery of the data.

We consider the problem of *correlated data gathering for throughput maximization* with a single sink, to which all the data has to be sent in a static wireless sensor network. This many-to-one communication where data from a set of sources are collected at a single node is known as *data gathering tree* [11]. We consider a system with two types of nodes: source nodes, $Y_i \in \mathcal{N}_S$, which collect and transmit data and have the capability to aggregate data, and relay nodes, $Y_i \in \mathcal{N}_R$, which do not generate data themselves but transmit data from other nodes in the network where $\mathcal{N}_S$ is the set of source nodes and $\mathcal{N}_R$ is the set of relay nodes. For the sake of simplicity, we assume there is only single sink labeled D which is the destination for the data from all the nodes. Let $\mathcal{N}$ be the set of all nodes, which includes source nodes, relay nodes and one sink node, and E be the set of edges, or possible links among nodes. Define the network graph $G = (\mathcal{N}, E)$. Let $|A|$ denote the number of elements of set A. Then $|\mathcal{N}| = N + 1$, and $|E| \leq N(N + 1)/2$. There are $|\mathcal{N}_S| = \alpha$ source nodes, and $|\mathcal{N}_R| = N - \alpha$ relay nodes in the network. The ratio of source nodes to all nodes is defined as $\varphi = \alpha/N$. The sources are labeled Y_1 through Y_α and the relays are labeled $Y_{\alpha+1}$ through Y_N. We assume that the network graph is connected, i.e. there always exists a communication link between any pair of nodes Y_i and Y_j in $\mathcal{N}$.

In the following, we develop an algorithm that maximizes the total network throughput. For the algorithm, we assume that there is a target bit error rate (BER) which ensures successful communication across a link. Automatic retransmission request is used so that a packet with error is retransmitted until received correctly. Suppose that the packet length is M. Then the probability of correct reception of the packet is $P_c(\gamma) = (1 - BER(\gamma))^M$ where $BER(\gamma)$ is bit error rate corresponding to a signal-to-interference and noise ratio (SINR) level γ. The BER function will depend on the modulation scheme and the noise and interference environment. In this chapter, we assume a code-division multiple access (CDMA) system, for which cumulative interference can be assumed to be Gaussian. We use non-coherent frequency shift keying modulation for which $BER(\gamma) = 0.5\exp(-0.5\gamma)$ under Gaussian noise and interference. This equation can be used to find a target SINR γ^* for the system.

Our system uses synchronous direct-sequence CDMA (DS-CDMA) where nodes use variable spreading sequences. The spreading factor for each transmitter, L, can be adjusted to meet the target quality-of-service (QoS) (or target SINR) requirements. The minimum spreading gain between the nodes Y_i and Y_j to reach a certain target SINR, γ^*, is [28]

$$L_{i,j} = \frac{\gamma^* \left[\sum_{k=1,k\neq i,j}^{N} h_{k,j} P_k\right]}{h_{i,j} P_i - \gamma^* \sigma^2}, \tag{4.1}$$

where the link gain $h_{i,j} = 1/d_{i,j}^2$ and $d_{i,j}$ is the distance of between the nodes Y_i and Y_j and σ^2 is the thermal noise power. The transmission rate, or the bit throughput between nodes Y_i and Y_j in bits-per-second (bps) is determined by the spreading rate

$L_{i,j}$: $\Omega_{i,j} = W/L_{i,j}$ where W is the system bandwidth and same for all Y_i.

Most of the traffic generated by sensor nodes is correlated data destined for the sink. The energy per bit per packet transmission, which has units of Joule/bits, represents the total amount of energy consumed in order to deliver one data bit to the destination. The energy per bit $E_b^{i,j}$ for packet transmissions between nodes Y_i and Y_j can be defined as [28]:

$$E_b^{i,j} = \frac{MP_i}{m\Omega_{i,j}P_c(\gamma)}, \tag{4.2}$$

where M is the packet length, m is the information bits in a packet, P_i is the constant transmit power for all $Y_i,\ i = \{1, 2, ..., N\}$.

To quantify the amount of data generated by each sensor node and data aggregation along the route, sources are associated with their data rates or weights as defined in [25]. It is assumed that data collected by the sensor nodes is correlated over geographical regions. Therefore, depending on the density of the WSNs in the field, the readings from nearby nodes maybe highly correlated and hence contain data redundancies. Each source node Y_i generates data at a certain rate $\Theta_i(Y_i)$, where $\Theta_i(Y_i)$ is the data rate of source Y_i. The data rate $\Theta_i(Y_i)$ shows the average number of bits per symbol used to encode the data source and has units bits/symbol. Any source node can aggregate data. In Fig. 4.1 source nodes are represented with circles and relay nodes are represented with plus signs. Source nodes can either send their own raw data directly to the sink, or they can forward their data to *intermediate* source nodes or relay nodes. Source nodes will aggregate all incoming data from other nodes with their own data, relay nodes do not perform aggregation. Then, the aggregated or relayed data is sent to the sink along the receiving node's route. The focus of this chapter is on increasing total network throughput.

4.21 Data Aggregation Model

In this chapter, correlated data from multiple nodes is aggregated into a single stream of data in order to reduce the network load for the set of wireless sensors organized as a tree. For data aggregation, we adopt the lossless step-by-step multi-hop aggregation model introduced in [25]. In this approach, each source node aggregates its data with that of the child nodes sequentially. Relay nodes only convey the data and do not perform data aggregation. Note that $\Theta_i(Y_i)$ denotes the rate of data generated by node Y_i. Node Y_j becomes a child node of node Y_i if node Y_j forwards its data to node Y_i. Let $\mathbf{Y}_i$ be the set of node Y_i and all child nodes of node Y_i. Without loss of generality, the generated raw data rate of each source, $\Theta_i(Y_i)$, is assumed to be constant for all $Y_i \in \mathcal{N}$ and each symbol is represented with μ bit of information, i.e. $\Theta_i(Y_i) = \mu$ bits/symbol. We assume that data aggregation at an intermediate source node is performed only after all input data is received from its children. Define $\Theta_i(\mathbf{Y}_i)$ as the net data rate at node Y_i due to data generated by node Y_i and its child nodes, after the aggregation process is done. For instance, suppose that node Y_i has q child nodes Y_1, Y_2, ..., Y_q. Then $\mathbf{Y}_i = \{Y_i, Y_1, Y_2, ..., Y_q\}$. The data received from a child node Y_j, $j = 1, .., q$ will be data aggregated from node Y_j and all of its child nodes, $\Theta_j(\mathbf{Y}_j)$. If Y_j is a leaf node with no child nodes, then it will forward only data generated by itself, $\Theta_j(\mathbf{Y}_j) = \Theta_j(Y_j)$. The aggregated data is transmitted to the parent node.

The aggregation of q inputs with source node Y_i is performed step by step with each node taking their turns depending on their arrival times. The parent node first aggregates its own data with newly received data. Then, the parent node aggregates the result with another node's data that arrives next. Therefore, for example in Fig. 4.1, assume that data from node Y_m arrives at node Y_1 before data from Y_i.

Node Y_1 first aggregates data from node Y_m with its original data. Node Y_1 saves the aggregated data to temporary storage and aggregates it again with data from node Y_i. The resulting data, with data rate $\Theta_1(\mathbf{Y}_1)$ is sent along its path to the sink D. If there are other source nodes on the path, multi-hop data aggregation is repeated at every intermediate source node along the route.

The reason for step by step aggregation of multiple inputs rather than joint aggregation schemes like the differential entropy model [29] is that storing multiple sources' data and aggregating them at once requires large memory and power for sensors. Moreover, the data reported from different nodes will arrive at the parent node at different times due to processing and transmission delays at intermediate nodes.

Let $\Theta^{temp}(Y_i, Y_j)$ be the temporary data rate of a node after data generated by node Y_i is aggregated with data from one of its child nodes Y_j. This data rate is calculated as [25]:

$$\Theta^{temp}(Y_i, Y_j) = \max\left(\Theta_i(Y_i)\,, \Theta_j(Y_j)\right) + (1 - \rho_{i,j}) \min\left(\Theta_i(Y_i), \Theta_j(Y_j)\right), \tag{4.3}$$

where $\rho_{i,j}$ is the correlation coefficient between nodes Y_i and Y_j and is a function of distance between these two nodes. In order to distinguish the correlation between the raw data and aggregated data, we use a "forgetting" factor for aggregated data [25]. The correlation between aggregated data at two parent nodes is only a fraction of the data correlation calculated according to their distance.

In this chapter, we assume only all the generated and aggregated data around the network is forwarded towards the sink node to ensure more uniform data collection across the network. This means that a source nodes' throughput is constrained not only by the throughput on the path from that node to the sink, but also by the throughput at which it receives data from its child nodes towards the edges of the network. In most scenarios, the bottleneck link for a route is the link connecting it to the sink, therefore this assumption does not affect the network throughput in most scenarios.

The *branches* of the network are distinct clusters of nodes rooted at node Y_i that are one hop from the sink D (see Fig. 4.1). A branch is denoted by $B_i,\ i = \{1, 2, 3, ..., \Upsilon\}$ where Υ is the total number of branches in the network. A branch B_i represents the set of nodes in the tree. The total throughput in the network is the sum of the throughput of all sources in all branches. The throughput of every source node is limited by the bottleneck throughput on the branch it is attached to.

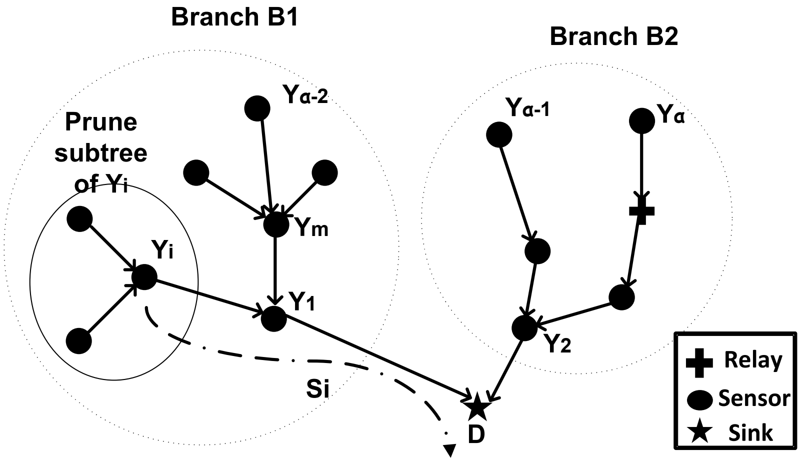


Figure 4.1: An example of data aggregation tree structure with two branches and pruning of the subtree rooted at Y_i to calculate $\lambda^{-i}(\mathbf{S})$ over facilities $\{Y_i, Y_1\} \in S_i$.

4.3 Efficient routing framework for throughput maximization

4.31 Symbol Throughput and Energy per Symbol

Two components of the network will determine the symbol throughput of a node, namely the data rate of sources (or the weight of sources) and bit throughput or transmission rate. Given the correlation models in Section 4.21, the symbol throughput of a link between nodes Y_i and Y_j is defined as

$$\begin{aligned}\zeta_{i,j}(\Theta_i(\mathbf{Y}_i)) &= \frac{\Omega_{i,j}}{\Theta_i(\mathbf{Y}_i)}, \\ &= \frac{W}{L_{i,j}\Theta_i(\mathbf{Y}_i)}\left[\frac{\text{symbol}}{\text{second}}\right],\end{aligned} \tag{4.4}$$

where $\Theta_i(\mathbf{Y}_i)$ is the aggregated data rate at node Y_i and $\mathbf{Y}_i$ is the set of all q sources using node Y_i including Y_i, i.e. $\mathbf{Y}_i = \{Y_i, Y_1, Y_2, \ldots, Y_q\}$. The symbol throughput has units symbol/second and symbolizes the total amount of symbol transmitted per second to the destination.

The energy per symbol transmission between nodes Y_i and Y_j can be defined as

$$\begin{aligned}E_s^{i,j}(\Theta_i(\mathbf{Y}_i)) &= E_b^{i,j}\left[\frac{\text{Joule}}{\text{bits}}\right]\Theta_i(\mathbf{Y}_i)\left[\frac{\text{bits}}{\text{symbol}}\right], \\ &= \frac{MP_i}{m\ \Omega_{i,j}\ P_c(\gamma)}\Theta_i(\mathbf{Y}_i)\left[\frac{\text{Joule}}{\text{symbol}}\right].\end{aligned} \tag{4.5}$$

The energy per symbol required has units Joule/symbol and indicates the total amount of per joule of energy consumption in order to deliver one data symbol to the destination without error.

4.32 Optimization Problem

Let $\mathbf{S} = (S_1, S_2, ..., S_\alpha)$ and S_i represent the set of all sources and relays used for source Y_i along its path into the sink node in a connected tree network topology. Let $\mathbf{X} = (X_1, X_2, ..., X_\alpha)$ and X_i be the set of all possible source and relay nodes for a route of source Y_i. SINR$_{i,j}$ is the received SINR at node Y_j for the link between nodes Y_i and Y_j, and $P_i = C$ is the constant transmit power of node $Y_i \in \mathcal{N}$. $\mathbf{Y}_i$

denotes the set of all sources or relays connected directly to node Y_i including Y_i, i.e. $\mathbf{Y}_i = \{Y_i, Y_1, Y_2, \ldots, Y_q\}$. Let there be n_{B_i} source nodes in a branch B_i and assume node Y_i uses the branch B_i. Note that $B_i = B_j$ if nodes Y_i and Y_j are using the same branch.

The throughput maximization problem in the network and physical layers can be formulated as

$$\begin{aligned} &\underset{\mathbf{S}\in\mathbf{X}}{\text{Maximize}} \ \sum_{i=1}^{\Upsilon} n_{B_i}\lambda_{B_i}(\mathbf{S}), \\ &\text{subject to } \ \text{SINR}_{i,j} \geq \gamma^*, \ \forall (Y_i, Y_j) \in \mathbf{S}, \\ &\qquad\qquad P_i = C, \ \forall Y_i \in \mathcal{N}, \end{aligned} \tag{4.6}$$

where λ_{B_i} is the *bottleneck throughput* of branch B_i for the set of active links $\mathbf{S}$,

$$\lambda_{B_i}(\mathbf{S}) = \min_{Y_i \in B_i} \ \min_{(Y_m, Y_n)\in S_i} \ \zeta_{m,n}(\Theta_m(\mathbf{Y}_m)), \tag{4.7}$$

and $\zeta_{m,n}(\Theta_m(\mathbf{Y}_m))$ is the symbol throughput between nodes Y_m and Y_n defined in (4.4). Note that every branch in the tree may have a different number of sources and communication structures between those sources and may have different bottleneck throughput. However, the bottleneck throughput of all sources in a branch will be the same.

As the *min* or *max* function is not differentiable, distributed solutions based on gradient algorithms are not directly applicable to the optimization problem in (4.6). One solution is to use sub-gradient algorithms to solve the *min-max* problem in a distributed manner [34]. However, the number of iterations required for convergence is substantial. Another solution is to approximate the *min-max* function using smoothing functions [4], but this approach requires special network topologies and data correlation models, and has not yet been extended to general network structures.

Optimal Routing Solution

The optimum routing tree in a wireless sensor network with a single sink can be found using the brute force approach. A centralized agent selects all possible routes for each node to maximize the overall throughput,

$$(S_1^*, S_2^*, ..., S_\alpha^*) = \arg\max_{\mathbf{S}\in\mathbf{X}} \sum_{i=1}^{\Upsilon} n_{B_i}\lambda_{B_i}(\mathbf{S}), \tag{4.8}$$

where $\mathbf{X}$ and $\mathbf{S}$ are defined above. A combination $\mathbf{S} = (S_1, S_2, ..., S_\alpha) \in \mathbf{X}$ is a *profile* and the brute force approach to solve the problem is to investigate all profiles exhaustively.

The optimum solution is hard if each sensor uses the data aggregation model in Section 4.21 since at each route, multi-hop aggregation is employed along each route. Moreover, finding the optimal solution is harder if the interference level in the network changes dynamically, for example when users use relay nodes in the network. Note that $\zeta_{i,j}(\Theta_i(\mathbf{Y}_i))$, the symbol throughput between nodes Y_i and Y_j defined in (4.4), will change if the source node uses one of the relaying nodes. This is because activating a relay node will change the interference level on all the links in the network. The joint optimization of transmission cost and data aggregation is shown to be NP-complete even for simple data aggregation models [3]. Therefore, finding the optimal throughput maximization algorithm is a NP-hard combinatorial optimization problem. Next, we will introduce a decentralized throughput maximization algorithm using correlation structure of the network. We propose a game theoretic formulation which can be shown to converge to a local optimal solution with relatively low complexity in a distributed fashion.

Game theoretic interpretation

The above problem can be formulated as a congestion game model which can be shown to be isomorphic with a potential game. In this game, the players are the source nodes in quest for routes, the shared facilities are source nodes and relaying nodes, the actions of the players are the selection of a group of facilities that form a route, and costs can be associated with various route selections. Formally, the proposed game-theoretic routing model for throughput maximizing correlation aware routing considers the route selection of each sensor node as a congestion game Γ as defined in Section 2.33. The game Γ is a tuple $(\mathcal{N}_s, \mathcal{F}, (X_i)_{i\in\mathcal{N}_s}, (w_f)_{f\in\mathcal{F}})$ where $\mathcal{N}_s = \{Y_1, \ldots, Y_\alpha\}$ denotes the set of players, i.e. the source nodes in our game, $\mathcal{F} = \{1, \ldots, m_f\}$ is the set of facilities, $X_i \subseteq 2^{\mathcal{F}}$ is the strategy space of player $Y_i \in \mathcal{N}_s$, and $w_f : \mathbf{S} \to \mathbb{R}$ is a cost function associated with using the facility f. $\mathbf{S} = (S_1, \ldots, S_\alpha)$ is a *state of the game* in which player Y_i chooses strategy (or route) $S_i \in X_i$. We define $\Delta_f(\mathbf{S})$ as the subset of nodes directly connected to facility f including the node at facility f, that is $\Delta_f(\mathbf{S}) = \{Y_i \in \mathcal{N} | f \in S_i\}$. The players choose strategies $S_i \in X_i$ to minimize their individual cost, where the cost $\delta_i(\mathbf{S})$ of player Y_i is given by $\delta_i(\mathbf{S}) = \sum_{f\in S_i} w_f(\Delta_f(\mathbf{S}))$.

We define utility function for source Y_i in our congestion game as

$$\begin{aligned} u_i : \mathbf{S} \to \mathbb{R},\;\; u_i(S_i, S_{-i}) &= -\delta_i(\mathbf{S}), \\ &= -\sum_{f\in S_i} w_f(\Delta_f(\mathbf{S})), \end{aligned} \tag{4.9}$$

where $S_{-i} = (S_1, S_2, \ldots, S_{i-1}, S_{i+1}, \ldots, S_\alpha)$ is the strategy space of player Y_i's opponents. The game performance is influenced by the selection of cost functions $w_f(\Delta_f(\mathbf{S}))$ for facilities. We propose and compare several metrics in the next section.

4.4 Facility cost selection for the congestion game

We consider the problem of constructing the *maximum correlated data gathering with minimum interference tree* for throughput maximization. In setting up the costs for facilities, we can consider several parameters like symbol throughput obtained on links along the route, interference and energy impact of the facility of the neighborhood network, or correlation awareness and opportunity for data aggregation.

4.41 Minimum Energy Routing (MER)

The classical approach to routing in WSNs is to consider only energy minimization. We denote this classic approach as MER (e.g. [21] [22]). For MER, the following utility function is used

$$u_i(S_i, S_{-i}) = -\sum_{f \in S_i} E_b^f, \tag{4.10}$$

where E_b^f is the cost of using the link of facility f, i.e. energy per bit required on ongoing links from facility f, through the strategy (or route) S_i and S_{-i}.

4.42 Interference Aware Routing (IAR)

Adding relays to a network complicates the interference pattern. Unlike source nodes, relays only transmit and cause interference to other links if they are part of the route of one or more source nodes in the network. In forming minimum energy routes, the MER algorithm does not take into account the interference that a relay node might generate if it is used. For this reason, Interference Aware Routing (IAR) algorithm have been proposed in the context of ad hoc networks [28]. For IAR, the following utility function is attempted to be maximized by each source Y_i

$$u_i(S_i, S_{-i}) = -\sum_{f \in S_i} \eta_f E_b^f, \tag{4.11}$$

where η_f is the normalized density of the nodes in a certain region. If the neighborhood is defined as circular region with radius D_r, η_f approximates the density of the nodes in a node's vicinity and represents an estimate of the interference impact a node has on its neighbors [28]. Note that this approach results in simple implementation and requires only local data. It was shown in [28] [31] that considering interference impact gives higher average throughput for overall network compared with MER for ad hoc networks. However, it is not clear if this is an optimum solution for WSNs, since sources generate correlated data and aggregation occurs at intermediate source nodes.

4.43 Throughput maximizing correlation-aware routing algorithm (T-CAR)

To account for data correlation and potential for data aggregation in the network for the solution of the bottleneck throughput maximization problem described in (4.6), we propose throughput maximizing correlation aware routing (T-CAR). Note that n_{B_i} is the number of source nodes in a branch B_i. The cost of using the branch B_i is defined as:

$$w_{B_i}(\lambda_{B_i}(\mathbf{S})) = -n_{B_i}\lambda_{B_i}(\mathbf{S}). \tag{4.12}$$

Let $\lambda_{B_i}^{-i}(\mathbf{S})$ denote the *bottleneck throughput* of branch B_i when source node Y_i is not present for the same set of state $\mathbf{S}$. When Y_i is not present on facilities $f \in S_i$, all the sub-trees rooted at Y_i are pruned (see Fig. 4.1). Note that, $\lambda_{B_i}^{-i}(\mathbf{S})$ is calculated only when the data aggregation weights are updated in the absence of source Y_i and its sub-tree. On the other hand, for this calculation, the spreading gains between links will not be affected. The reason is that we are interested in the correlation contribution in terms of generated data rate of only node Y_i and its sub-tree. Then, the utility function $u_i(S_i, S_{-i})$ of source Y_i is defined as

$$\begin{aligned} u_i(S_i, S_{-i}) &= -\left(w_{B_i}(\lambda_{B_i}(\mathbf{S})) - w_{B_i}(\lambda_{B_i}^{-i}(\mathbf{S}))\right), \\ &= \quad n_{B_i}\lambda_{B_i}(\mathbf{S}) - \ (n_{B_i} - n_i)\lambda_{B_i}^{-i}(\mathbf{S}), \end{aligned} \tag{4.13}$$

where n_i is the number of source nodes on the subtree rooted at source node Y_i including Y_i. Hence, each player Y_i in the network wants to find the best strategy $S_i \in X_i$ searching over all branches $B_i,\ i = \{1, 2, ..., \Upsilon\}$ that gives maximum utility defined in (4.13).

An alternative approach, which considers both opportunities for aggregation and interference impact, is throughput maximizing interference and correlation aware routing (T-ICAR). T-ICAR is a combination of T-CAR and IAR. The solution provided by this algorithm consists of constructing the maximum correlated data gathering using the idea of T-CAR algorithm and minimum interference impact relaying nodes, as in the IAR algorithm. So, T-ICAR is simply adding correlation-awareness into IAR which is constructed at the start of iterations. The utility function for T-ICAR algorithm is same with T-CAR, i.e. defined by (4.13). The only difference between T-ICAR and T-CAR algorithms is the starting trees in the first iteration. T-ICAR starts with IAR tree and T-CAR algorithm starts with DIRECT tree where all nodes transmit directly into sink node or with MER tree.

4.44 Potential Game Formulation

In certain classes of games, the game converges to a NE when a *best* or *better response* adaptive strategy is employed. In what follows, we show that the congestion game associated with T-CAR (or T-ICAR) is isomorphic with a potential game, for which a best response strategy is shown to converge to a NE.

Potential games as defined in Section 2.32 are games in which players converge to a pure strategy NE by following a sequence of improvement paths. We assume that in the normal form game each player takes actions sequentially and at each stage of the game players choose their actions which improve their utility functions. Potential games have the fundamental property that a NE always exists. Most of the learning algorithms for a potential game guarantee convergence into a (pure) NE point.

In a potential game, as defined in Section 2.32, if a player unilaterally changes its strategy, the change in the objective function would be equal to the change in the potential function. Generally in potential games, the updating procedure is carried out sequentially. Note that, while sequential updates may require additional synchronization overhead, a simple approximation implementation may be based on randomized access which on average will result in sequential updates. This can be shown experimentally to have minimal impact on convergence properties. Next, we demonstrate that T-CAR with utility functions given by (4.13) is an exact potential game (EPG).

Theorem 4.4.1. *T-CAR with utility function defined by (4.13) is an EPG with potential function,*

$$\mathcal{P}(S_i, S_{-i}) = -\sum_{i=1}^{\Upsilon} w_{B_i}(\lambda_{B_i}(\mathbf{S})), \tag{4.14}$$

where $w_{B_i}(\lambda_{B_i}(\mathbf{S})) = -n_{B_i}\lambda_{B_i}(\mathbf{S})$.

Proof. See Appendix 4.A. □

Therefore, from Theorem 4.4.1, T-CAR always has at least one (pure) NE and converges to NE strategies by using a best response adaptive strategy. Each source $Y_i \in \mathcal{N}_s$ in T-CAR algorithm updates its strategy S_i through maximization of its corresponding utility (4.13). Hence, at each iteration, each user finds the best routes (sequential updates) that will increase its utility. From Theorem 4.4.1, the potential function $\mathcal{P}(S_i, S_{-i})$ will continue to increase until it reaches a local maximum point using best response dynamics [12] [16].

In summary, T-CAR algorithm constructs the routing tree in two parts: first, an arbitrary initial routing tree is constructed, and second, routing decisions of each node are updated with nodes adapting in turns. The procedure for sequential updates of routes with best response dynamics of T-CAR can be summarized as follows:

Input : A connected network G.

Output : A throughput maximizing data gathering tree of G.

Initialization : Find an initial spanning tree T. We assume that DIRECT routing tree where all sensor nodes can be initialized to connect to the sink node directly or MER tree can be employed at the initialization stage for the initial spanning tree. Then, use the following iterations for throughput maximization (T-CAR).

Repeat : At each iteration n.

- For each of the source node $Y_i \in \mathcal{N}_s$,

    1. Select $S_i \in X_i$ that maximizes the utility function in (4.13) for T-CAR.
    2. Update routing strategy S_i for source Y_i and form the corresponding tree structure.

Until : The stopping criteria Δ is met.

The stopping criterion Δ is the minimum number of iteration steps κ for the algorithm to converge, where κ is a counter which adds one after each updating process.

The total *symbol throughput* in the network is defined as the total bottleneck throughput of each source in the tree,

$$\zeta^{total} = \sum_{i=1}^{\alpha} \lambda_i(\mathbf{S}), \tag{4.15}$$

where $\lambda_i(\mathbf{S}) = \lambda_{B_i}(\mathbf{S})$ for $Y_i \in B_i$ and λ_{B_i} is defined in (4.7).

4.45 Dynamic Network Management

Although, we have focused on how to build an efficient throughput maximizing routing tree for aggregating data in a stationary environment, the proposed algorithm can also be used to accommodate the inherent dynamic topology of the sensor network such as sink mobility or duty schedules. We assume that each node retrieve up-to-date topology information within a certain region with cell radius D_r, for example the information about the corresponding total cost functions of using each individual node, or the duty schedules of nodes which determines whether they are active or not around this region. Note that in sensor networks, data correlation decreases with distance between nodes, hence local correlation is dominant. Therefore, in large network sizes, a node is highly correlated within a certain radius of its neighborhood. When each node is making decisions about its next hop at the beginning of each iteration, the nodes can determine if the nodes around has new, broken or modified links. Afterwards, the nodes can execute the routing decisions using updated link costs and lists at the end of the iteration, hence the proposed algorithm can dynamically reconfigure the network to maximize the throughput.

4.46 Condition for the Nash equilibrium strategy

Thus far, we have shown that a NE strategy exists for T-CAR when players use the proposed utility function. In this section, we want to address the question under which conditions the NE strategy exists and give an upper bound for the correlation coefficient for the NE condition. It is difficult to address the NE point for random network configurations, therefore we focus on NE conditions for a special configuration namely, DIRECT transmission in which case all nodes transmit to sink node directly.

Proposition 4.4.2. *: Let $\mathbf{S_{direct}}$ be the selected strategy for each source Y_i, $\forall i \in \mathcal{N}_s$ in a given graph G where all nodes communicate directly with sink node. Then, the set of state $\mathbf{S_{direct}} = \{S_1, S_2, ..., S_\alpha\}$ is a NE solution when*

$$L_{i,j} < L_{j,D}(2-\rho_{i,j}) \;\; \textit{and} \;\; \rho_{i,j} < \frac{2L_{j,D}}{L_{i,D}+L_{j,D}} = \eta_{i,j}, \quad \forall\, Y_i, Y_j \in \mathcal{N}_s \; \textit{and} \, Y_i \neq Y_j, \tag{4.16}$$

and when

$$L_{i,j} > L_{j,D}(2-\rho_{i,j}) \;\; \textit{and} \;\; \rho_{i,j} > \frac{2L_{j,D}}{L_{i,D}+L_{j,D}} = \eta_{i,j}, \quad \forall\, Y_i, Y_j \in \mathcal{N}_s \; \textit{and} \, Y_i \neq Y_j, \tag{4.17}$$

where $L_{i,D}$ and $L_{i,j}$ are the minimum spreading gains between nodes Y_i and the sink node D and nodes Y_i and Y_j respectively for all sources $Y_i, Y_j \in \mathcal{N}_s$.

Proof. Let us assume that there exists a strategy $S_i' \in X_i$ where node Y_i changes its strategy from direct transmission and connects through node Y_j into sink node D. The corresponding strategy set is denoted by $\mathbf{S}' = \{S_1, S_2, ..., S_i', S_j, ..., S_\alpha\}$ and the utility function of Y_i for the new strategy S'_i is $u_i(S'_i, S_{-i})$. If $\mathbf{S_{direct}}$ is the NE strategy, then $u_i(S'_i, S_{-i}) < u_i(S_i, S_{-i})$ must hold for all $Y_i \in \mathcal{N}_s$. Note that in $\mathbf{S}'$, the bottleneck link can be either the link between nodes Y_j and D (case I condition) or the link between nodes Y_i and Y_j (case II condition).

In case I condition, the corresponding utilities for these two different strategies are $u_i(S_i, S_{-i}) = T_i$ and $u_i(S'_i, S_{-i}) = 2T'_j - T_j$ where $T_i = \lambda_{B_i}(\mathbf{S}_{direct})$ and $T_j = \lambda_{B_j}(\mathbf{S}_{direct})$ are the corresponding symbol throughput of nodes Y_i and Y_j respectively when the strategy set is $\mathbf{S_{direct}}$ and $T'_j = \lambda_{B_j}(\mathbf{S}')$ is the symbol throughput for node Y_j when the strategy set is $\mathbf{S}'$ where the bottleneck link is between nodes Y_j and D. Denote $T'_i = W/(L_{i,j}\mu)$ as the symbol throughput between nodes Y_i and Y_j. From case I condition, $T'_j < T'_i$ hence $L_{i,j} < L_{j,D}(2 - \rho_{i,j})$. Moreover, from the condition $u_i(S'_i, S_{-i}) < u_i(S_i, S_{-i})$,

$$\begin{aligned}
2T'_j - T_j &< T_i, \\
\frac{2W}{L_{j,D}(\mu + (1-\rho_{i,j})\mu)} - \frac{W}{\mu L_{j,D}} &< \frac{W}{\mu L_{i,D}}, \\
\frac{2}{L_{j,D}(2-\rho_{i,j})} &< \frac{1}{L_{i,D}} + \frac{1}{L_{j,D}}, \\
\frac{2L_{i,D}}{L_{j,D} + L_{i,D}} &< (2 - \rho_{i,j}), \\
(\rho_{i,j} - 2) &< -\frac{2L_{i,D}}{L_{j,D} + L_{i,D}}, \\
\rho_{i,j} &< \frac{2L_{j,D}}{L_{j,D} + L_{i,D}} \\
&= \eta_{i,j}.
\end{aligned} \tag{4.18}$$

This completes the proof of (4.16).

In case II condition, the bottleneck link is between nodes Y_i and Y_j. Therefore, the corresponding utilities for these two different strategies are $u_i(S_i, S_{-i}) = T_i$ and

$u_i(S'_i, S_{-i}) = 2T'_i - T_j$ where $T'_i = \lambda_{B_j}(\mathbf{S}')$ is the symbol throughput between node Y_i and Y_j when the strategy set is $\mathbf{S}'$. The symbol throughput between nodes Y_j and D is $T'_j = W/(L_{i,j}\mu(1+(1-\rho)))$. From case II condition, $T'_i < T'_j$ which gives $L_{i,j} > L_{j,D}(2-\rho_{i,j})$. Moreover, from the condition $u_i(S'_i, S_{-i}) < u_i(S_i, S_{-i})$,

$$\begin{aligned} 2T'_i - T_j &< T_i, \\ \frac{2W}{\mu L_{i,j}} - \frac{W}{\mu L_{j,D}} &< \frac{W}{\mu L_{i,D}}, \\ \frac{2}{L_{i,j}} &< \frac{1}{L_{i,D}} + \frac{1}{L_{j,D}}, \\ \frac{2L_{i,D}L_{j,D}}{L_{j,D}+L_{i,D}} &< L_{i,j}. \end{aligned} \tag{4.19}$$

Combining (4.19) and the condition $L_{j,D}(2-\rho_{i,j}) < L_{i,j}$, we get

$$\rho_{i,j} > \frac{2L_{j,D}}{L_{i,D}+L_{j,D}}, \tag{4.20}$$

which completes the proof of (4.17) and hence the proof. □

Using Proposition 4.4.2, we have the following corollary:

Corollary 4.4.3. *In the case when there is no correlation among nodes, i.e.* $\rho_{i,j} = 0$*, state* $\mathbf{S_{direct}}$ *is a NE point when* $L_{i,j} < 2L_{j,D}$, $\forall\, Y_i, Y_j \in \mathcal{N}_s$ *and* $Y_i \neq Y_j$.

4.5 Simulation Results

In this section, we present the numerical results and draw some observations on the comparative performance of our proposed routing algorithms with the other classical approaches. Namely, we compare the performance of T-CAR and T-ICAR with other algorithms like IAR, MER, DIRECT and optimal routing solution (OPT). The impact of network size and correlation coefficient, the size of cell radius and number of iterations required to converge for different algorithms are compared. In accordance with design parameter, the performance of our proposed algorithms are compared in terms of energy and throughput with other classical algorithms.

4.51 Simulation Setup

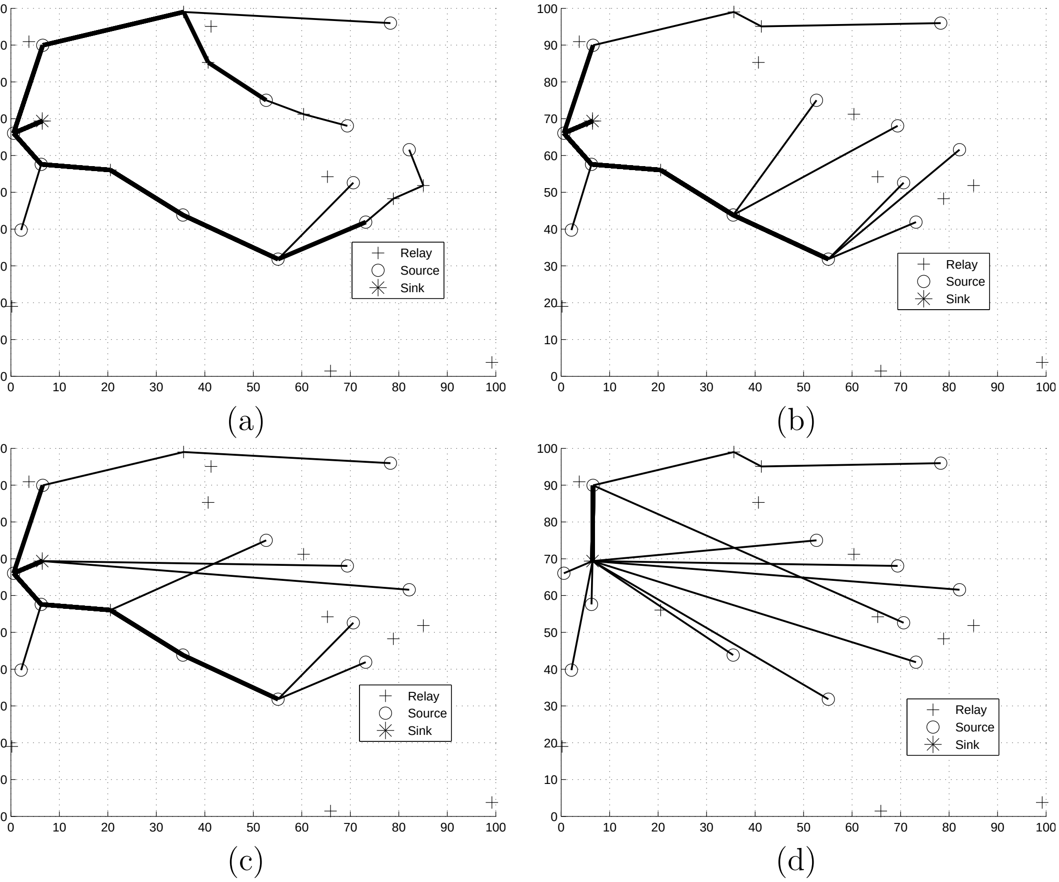
The number of sensor nodes in the network is varied between $N = 4$ to $N = 40$, which is uniformly distributed over a square area of dimension $100m \times 100m$. We randomly deploy each node in the network such that the distance between each node in the network is larger than $3m$. The ratio of all nodes to source nodes is selected to be $\varphi = \alpha/N = 1$ (i.e. all nodes are sources) or $\varphi = \alpha/N = 0.5$ (i.e. the number of relay nodes and sensor nodes are same). We adopt *Gaussian random field* data correlation model that is frequently encountered in practice [29]. In this model, correlation coefficient $\rho_{i,j}$ between nodes Y_i and Y_j decreases exponentially with the distance between nodes $d_{i,j}$, for instance, we use $\rho_{i,j} = \exp(-d_{i,j}^2/c)$ which corresponds to a singular continuous-space process [29] and c is the correlation constant. In simulations, we use a "forgetting" factor of 0.8 per link, that is, if data is aggregated at a node Y_i, then the correlation $\rho_{i,j}$ between that node and its parent node Y_j in the routing tree is reduced to 0.8 of its original value. The noise power is $\sigma^2 = 10^{-13}$ Watts, which corresponds to thermal noise power for a bandwidth of $W = 1$ Mhz. We choose the equal transmit powers of all nodes to be 110 dB above the noise floor ($P_i = 10^{-2}$ Watts, $\forall i \in \mathcal{N}$). The target SINR is selected to be $\gamma^* = 5$ (7 dB). We assume that each packet contains 80 bits of information and no overhead (i.e., $m = M = 80$). Each symbol is represented with $\mu = 1$ bit of information, i.e. $\Theta(Y_i) = 1$ bits/symbol. We use $D_r = 16m$ for our simulations to evaluate the performance of IAR and T-ICAR. The results are simulated and averaged over 1000 different network configurations for each routing algorithm.

MER and IAR algorithms are used as a performance benchmark to compare with our proposed algorithms and to measure the trade-off involved in energy and throughput. Both MER and IAR algorithms use Dijkstra's algorithm [18] to find the best routes from sink to source nodes. Although MER and IAR were not proposed in the context of data aggregation, we set-up the paths according to their corresponding utility functions, and then aggregate data opportunistically based on the routes set-up, this approach is called routing-driven aggregation [27]. T-CAR and T-ICAR are implemented iteratively based on the best response strategy described in the previous sections. Note also that, we start T-ICAR algorithm with IAR and T-CAR algorithm with DIRECT when $\varphi = 1$ and with MER algorithm when $\varphi = 0.5$.

Through multiple iterations, the algorithms change the initial routing tree significantly. Fig. 4.2 demonstrates the branches of the constructed trees for different routing strategies discussed previously, namely for MER, IAR, T-CAR and T-ICAR under the same network topology for $N = 24$, $c = 100$ and $\varphi = 0.5$. Thick lines indicate the regions where the data aggregations are performed. The sensor nodes that are distant from the sink has lower data rates and therefore, have thinner lines. The results show that different routing metrics with different utility functions lead to paths with significantly different trees or network connectivity. For example, T-CAR tends to discover paths with higher throughput while exploiting the data correlation whereas MER searches for minimum energy routing paths.

4.52 Symbol Throughput Improvements

In this section, we study the impact of network size on the performances of the above algorithms. Fig. 4.3 shows the total throughput and energy of T-CAR, DIRECT and MER algorithms versus number of nodes for $c = 1000$ when $\varphi = 1$. We see that the throughput of T-CAR increases with network size at the expense of energy loss



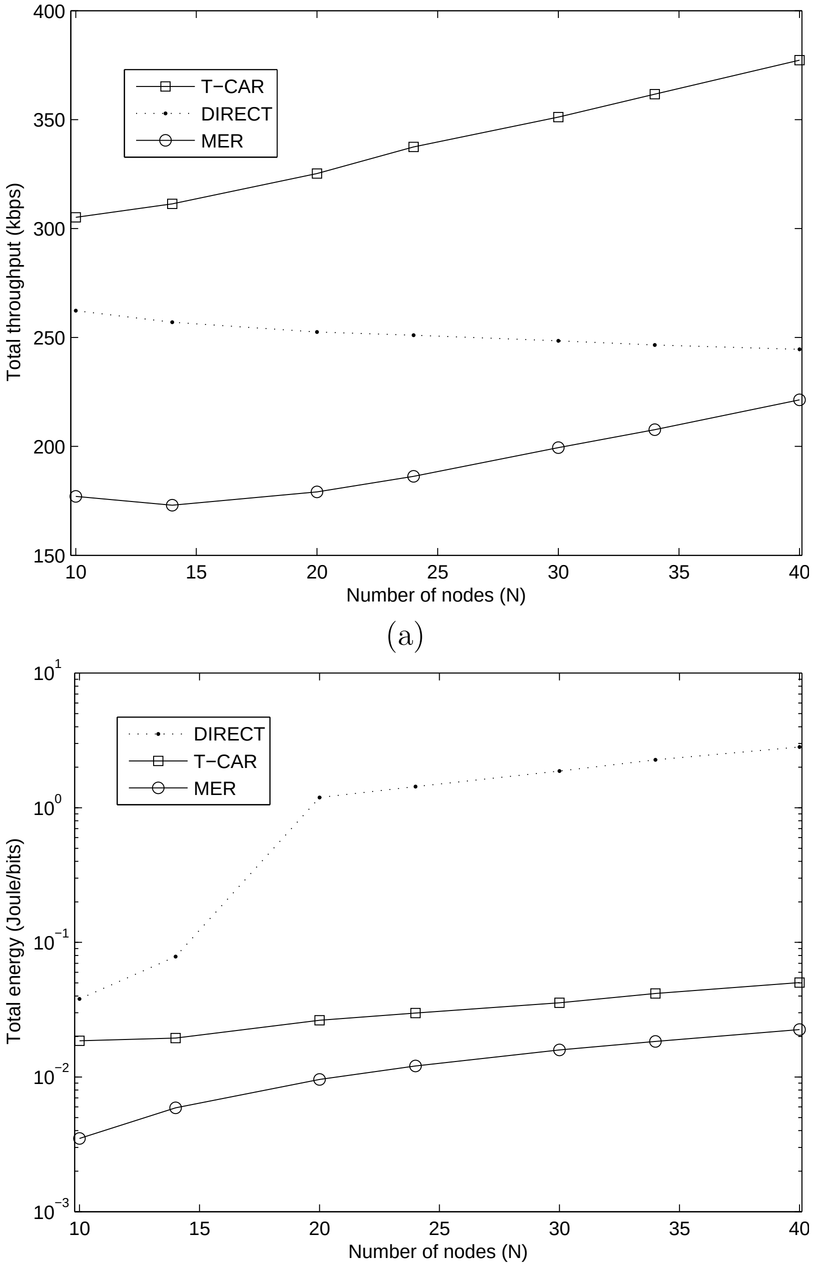
(a) (b) (c) (d)

Figure 4.2: Selected paths of each source and the tree structures of different routing strategies for $N = 24$, $c = 100$ and $\varphi = 0.5$. (a) MER. (b) IAR. (c) TCAR (d) TICAR.

due to the trade-off between energy and throughput gains. This can be explained as follows: the network topology changes from sparse to dense when the network size increases. The data correlation between neighboring nodes becomes higher as the distance between nodes decreases, therefore more redundant data can be reduced with data aggregation. The increased number of nodes has mainly two affects on the network throughput of both T-CAR and MER algorithms: First, it increases the number of simultaneous transmissions which also increases the interference in the network. As a result, the bottleneck throughput of each node decreases. Second, increasing number of nodes clearly increases the total network throughput by increasing the data correlation between nodes. The data correlation and interference have reverse effects on network throughput. Note also that MER algorithm is optimized for energy, not for throughput. However, as the network size N increases, the throughput of MER algorithm increases too. This result indicates that MER algorithm benefits more from data correlation than network interference as the number of nodes increase.

The percentage throughput improvements of T-CAR with respect to initial starting tree of DIRECT increases with network size. In large network sizes, T-CAR has more potential and opportunity to give higher throughput improvements since DIRECT doesn't exploit data correlation. Hence, the percentage gain of throughput is higher. We also observe that T-CAR has also significant throughput improvements over MER. As an example, for $N = 10$, the throughput improvements of T-CAR over DIRECT and MER are approximately 16 and 72 percent respectively. On the other hand, for $N = 40$, the corresponding throughput improvements are 54 and 70 percent over DIRECT and MER respectively. Fig. 4.3 also shows the total energy expenditures of the above algorithms. Compared to MER algorithm, due to the trade-off between energy and throughput, the energy loss of T-CAR is on the order of 81 and 55 percent for $N = 10$ and $N = 40$ respectively.

Numerical studies also show important total throughput improvements of T-ICAR algorithm over T-CAR, IAR, and MER algorithms at the expense of energy loss as shown in Fig. 4.4 for $N = 24$, $\varphi = 0.5$, $D_r = 16m$ and $c = 100$. In terms of total throughput, T-ICAR and T-CAR outperform IAR and MER, because their utility functions are designed for throughput maximization by exploiting data correlation. The total throughput improvement of T-ICAR is on the order of 4.36 percent over T-CAR, 70 percent over IAR and 84 percent over MER. On the other hand, the energy loss of T-ICAR is on the order of 97.06 percent over IAR and 97.32 percent over MER. Note also that T-CAR and T-ICAR have approximately same energy consumptions.

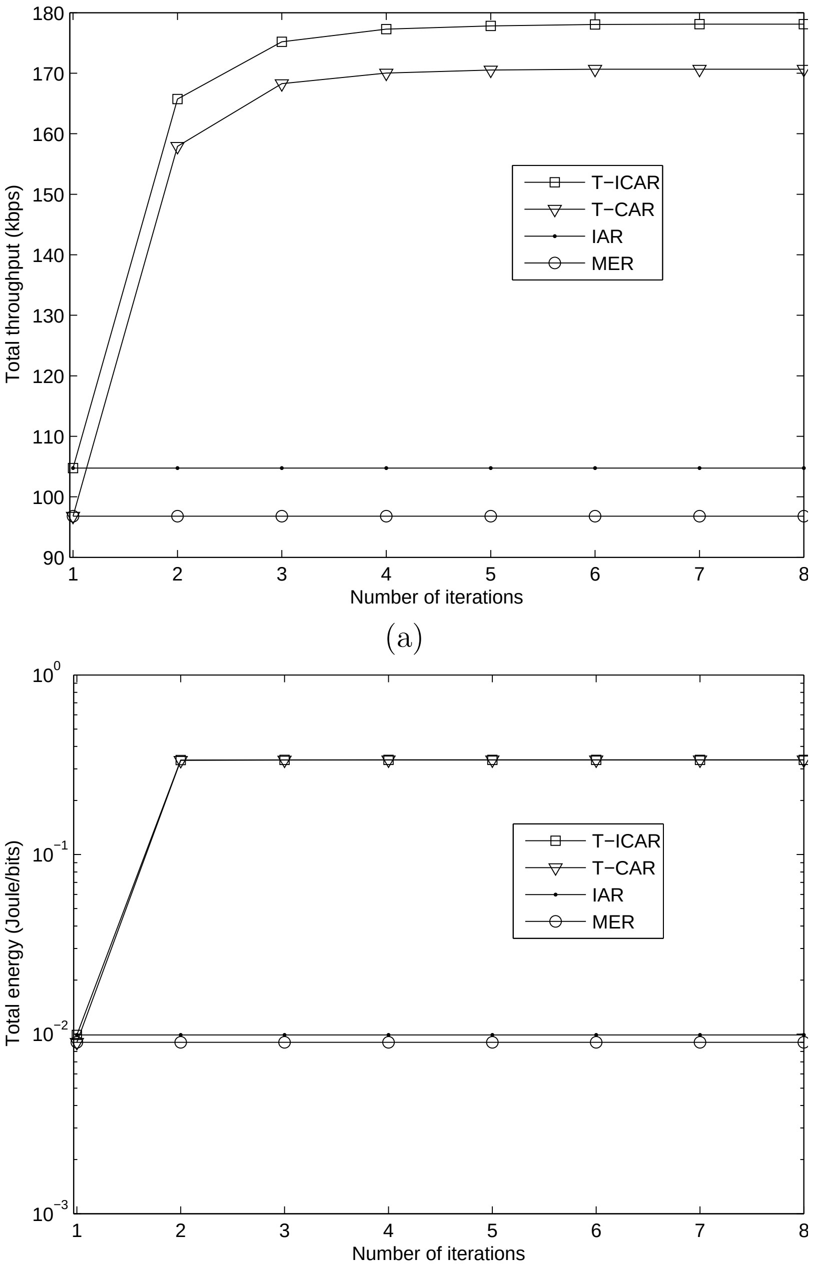


(a)

(b)

Figure 4.3: Total throughput and energy versus number of nodes, N, for $c = 1000$ and $\varphi = 1$. (a) Total throughput. (b) Total energy.

From Fig. 4.4, we also see that T-CAR performs 62.91 percent better than IAR at this moderate interference environment. However, it has been shown in [28], that the gains of IAR can also diminish for very low or very high interference environments



(a)

(b)

Figure 4.4: Total throughput and energy of MER, IAR, CAR and ICAR throughout the iteration process for $N = 24$, $\varphi = 0.5$, $c = 100$ and $D_r = 16m$. (a) Total throughput. (b) Total energy.

which can increase the performance gap even more. IAR is designed to eliminate "hot spots" in the network and reduce interference. There exists a energy-throughput trade-off involved in selecting the IAR algorithm compared to MER algorithm. IAR algorithm is known to outperform the MER algorithm in terms of throughput at the expense of slight increase in the average energy per bit transmission for wireless ad-hoc networks [28]. In WSNs, the throughput gain when cell radius $D_r = 16m$ is 8.23 percent, whereas there is an 9.09 percent increase in energy expenditures at the same time as can be observed from Fig. 4.4. Therefore, IAR algorithm is a better algorithm than MER in terms of throughput by avoiding hot-spot areas and controlling the interference in the network.

4.53 Impact of Correlation Coefficient

Fig. 4.5 shows the total throughput for T-CAR and MER algorithms as the correlation constant c increases from 0 to 1000 for different network sizes. Intuitively, the throughput obtained for both algorithms at higher correlation is much higher compared with the throughput obtained at low correlation levels.

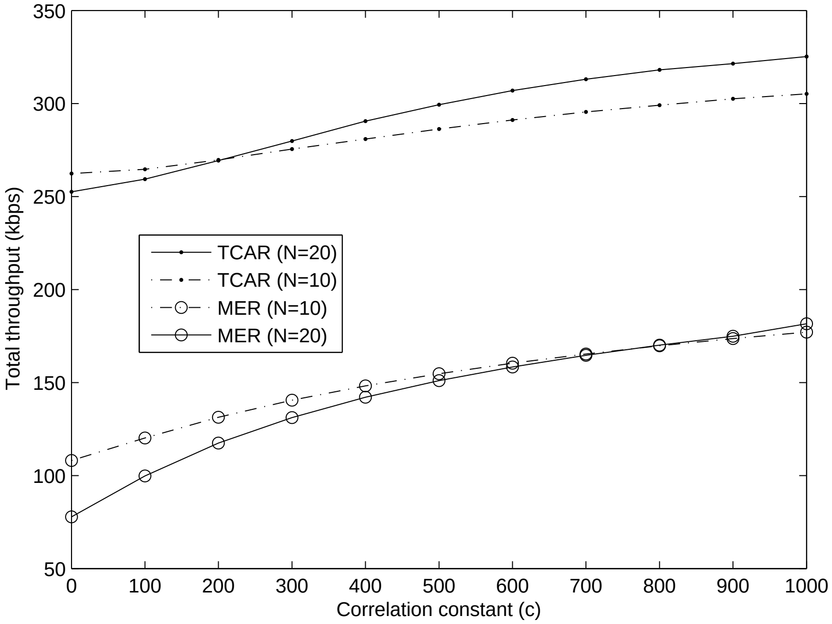


Figure 4.5: Total throughput versus correlation constant (c) for $\varphi = 1$.

T-CAR algorithm gives significant throughput improvements for all network size and under all the correlation constants compared to MER algorithm. The percentage improvements of T-CAR compared to MER decrease with increasing correlation constant. The reason is that at low correlation levels, T-CAR has more potential and opportunity to give higher throughput improvements. Hence, the percentage improvement for throughput is higher. As the correlation constant increases, the room for further improvement of T-CAR algorithm is not so high as before, hence percentage improvements are lower. For example, for correlation constant of $c = 200$, the percentage throughput improvements of T-CAR over MER are 56.38 and 51.30 for network sizes of $N = 20$ and $N = 10$ respectively, whereas, for correlation constant of $c = 800$, the percentage improvements are 46.52 and 43.24 for $N = 20$ and $N = 10$ respectively. Note also that at low correlations, the throughput of T-CAR is lower at high network size $N = 20$ compared with low network size $N = 10$. This is due the fact that at higher network sizes and low correlation, the interference in the network diminishes the total throughput in the network. This shows that interference impact is dominant at low correlations and hence reduces the total throughput. However, as the correlation constant increases, the correlation impact dominates the interference impact resulting in higher total network throughput.

The results in Fig. 4.5 implies that T-CAR algorithm can achieve significant throughput gains compared to MER algorithm for a wide range of data correlation by utilizing only local information. Finally, similar conclusions may also be drawn for T-ICAR algorithms' throughput as the correlation coefficient changes.

4.54 Convergence of the Algorithm

In addition to the effectiveness of T-CAR, the number of iterations for the convergence of the distributed algorithms is also important. In this section, we show the number of

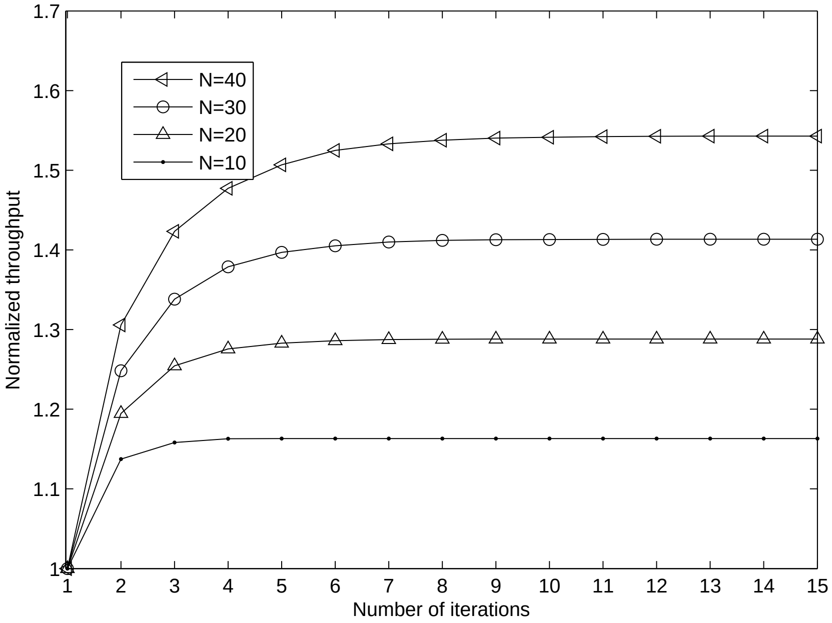


Figure 4.6: Normalized throughput of T-CAR with respect to MER versus number of iterations for $\varphi = 1$ and $c = 1000$.

required iterations for the convergence of the T-CAR algorithm for different network sizes. Fig. 4.6 shows the normalized throughput of T-CAR compared to DIRECT versus number of iterations for $\varphi = 1$ and $c = 1000$. The percentage improvements in terms of throughput of the T-CAR algorithm converge in $\kappa = 5, 9, 14$ and 15 iterations for network sizes of $N = 10, 20, 30$ and 40 respectively. Fig. 4.6 indicates that T-CAR algorithm is able to achieve convergence to NE. Moreover, T-CAR algorithm involves small number of iterations, and it can be implemented efficiently in a distributed fashion. The convergence speed of the algorithm is associated with the network size N since longer convergence iteration is needed when N is large. Similar convergence results may also be shown for T-ICAR algorithms after IAR is established.

4.55 Comparisons with Optimal Routing Solution

Constructing and optimal routing tree to minimize a total cost function in a WSN is an NP hard problem [3]. Therefore, we can only solve the problem for a small topology in a reasonable time. Note that optimal routing solution can provide a measure how good the proposed routing algorithms really are. Consider a small wireless sensor network with $N = 4$ for $\varphi = 1$ and $c = 1000$. As a performance benchmark, the global optimum routing solution is obtained by considering all possible feasible strategies with a single sink, which is 125 different *set of states* or *data-gathering trees of* G.

Table 4.1 shows the energy per symbol and symbol throughput of MER, DIRECT, T-CAR and OPT algorithms. Note that the global optimum routing solution, indicated by OPT, functions as the upper bound of the overall throughput. We observe that T-CAR algorithm's performance is within 97 percent of the total throughput of OPT while the MER algorithm's performance is within 76 percent of OPT. Moreover, in our experiments, T-CAR algorithm converges in 4 iterations. Note that the performance improvement of T-CAR is highly affected by the selection of the initial starting tree. If the starting tree is selected based on the performance metric it is designed for, proposed correlated-aware algorithms tend to give closer performance results to OPT. Note also the trade-off involved in energy and throughput for T-CAR algorithm and OPT. The energy loss is 82 percent for T-CAR algorithm and 80 percent for OPT algorithm compared to MER algorithm.

Table 4.1: Comparisons with centralized optimization at the end of iterations for $N = 4$, $c = 1000$ and $\varphi = 1$

	OPT	T-CAR	MER	DIRECT
Throughput (kbps)	311.6	303.26	236.47	280.03
Energy (mJ/bits)	3.0	3.4	0.6	5.3

4.6 Conclusions

In this chapter, we have presented a detailed investigation of efficient throughput maximizing transmission structure in wireless sensor networks where each source transmits and aggregates the correlated data over intermediates source nodes to the sink. We have considered the impact of interference, as well as efficient data aggregation in establishing routing paths towards the sink for throughput maximization problem. For correlation aware routing, we have proposed a distributed iterative protocol based on a game theoretic framework, which is shown to converge within a couple of iterations. We have also shown that, by accounting for both correlation structure and interference impact in constructing routes, significant throughput gains over classic approaches can be achieved. Due to the NP-completeness of the optimum routing solution, the performance of the proposed algorithms is compared for a small network size and is shown to be within 97% of the optimal throughput when the initial starting tree is within 76% of the optimal throughput solution.

Appendix 4.A: Proof of Theorem 4.4.1

Suppose there exists a potential function of the congestion game Γ:

$$\mathcal{P}(S_i, S_{-i}) = -\sum_{i=1}^{\Upsilon} w_{B_i}(\lambda_{B_i}(\mathbf{S})) = \sum_{f\in\mathcal{F}} \lambda_f(\mathbf{S}), \tag{4.21}$$

where f is the facility, $\mathcal{F}$ is the set of all facilities as defined in Section 4.32 and $\lambda_f(\mathbf{S}) = \lambda_{B_i}(\mathbf{S})$ for $f \in B_i$. Let $S_i \in \mathbf{S}$ be the strategy of source $Y_i, i = 1, ..., \alpha$, i.e. the collection of nodes used for relaying or aggregating using the branch B_i and $S_i' \in$ **S'** be another strategy for source Y_i using the branch B_i' where **S'** $= (S_1, ..., S_{i-1}, S_i', S_{i+1}, ..., S_\alpha)$. Note that $B_i = B_i'$ if strategies S_i and S_i' are using the set of nodes in the same branch. Then,

$$\begin{aligned}\mathcal{P}(S_i, S_{-i}) &= \sum_{f\in\mathbf{F}} \lambda_f(\mathbf{S}),\\ &= \sum_{f\in B_i} \lambda_f(\mathbf{S}) + \sum_{f\in B_i'} \lambda_f^{-i}(\mathbf{S}) - \sum_{f\in B^*} \lambda_f(\mathbf{S}) + \sum_{f\in\mathcal{F}\backslash\{B_i\bigcup B_i'\}} \lambda_f(\mathbf{S}),\end{aligned} \tag{4.22}$$

where B^* denotes the common facilities used by the strategies S_i and S_i', i.e. $B^* = S_i \bigcap S_i'$. Define,

$$Q(B_{-i,-i'}) = -\sum_{f\in B^*} \lambda_f(\mathbf{S}) + \sum_{f\in\mathcal{F}\backslash\{B_i\bigcup B_i'\}} \lambda_f(\mathbf{S}). \tag{4.23}$$

Then,

$$\mathcal{P}(S_i, S_{-i}) = \sum_{f\in B_i} \lambda_f(\mathbf{S}) + \sum_{f\in B_i'} \lambda_f^{-i}(\mathbf{S}) + Q(B_{-i,-i'}). \tag{4.24}$$

If source Y_i changes its strategy from S_i to S_i', then the potential function becomes,

$$\mathcal{P}(S_i', S_{-i}) = \sum_{f\in B_i} \lambda_f^{-i}(\mathbf{S}') + \sum_{f\in B_i'} \lambda_f(\mathbf{S}') + Q(B_{-i,-i'}). \tag{4.25}$$

Note that $Q(B_{-i,-i'})$ is not affected by the strategy changing of source Y_i. Therefore,

$$\mathcal{P}(S_i', S_{-i}) - \mathcal{P}(S_i, S_{-i}) = \left(\sum_{f\in B_i} \lambda_f^{-i}(\mathbf{S}') + \sum_{f\in B_i'} \lambda_f(\mathbf{S}')\right) - \left(\sum_{f\in B_i} \lambda_f(\mathbf{S}) + \sum_{f\in B_i'} \lambda_f^{-i}(\mathbf{S})\right). \tag{4.26}$$

Note that the utility function in (4.13) can be rewritten as:

$$u_i(S_i, S_{-i}) = \sum_{f\in B_i} \lambda_f(\mathbf{S}) - \sum_{f\in B_i} \lambda_f^{-i}(\mathbf{S}). \tag{4.27}$$

From above,

$$u_i(S_i', S_{-i}) - u_i(S_i, S_{-i}) = \left(\sum_{f \in B_i'} \lambda_f(\mathbf{S}') - \sum_{f \in B_i'} \lambda_f^{-i}(\mathbf{S}') \right) - \left(\sum_{f \in B_i} \lambda_f(\mathbf{S}) - \sum_{f \in B_i} \lambda_f^{-i}(\mathbf{S}) \right).$$

Hence,

$$u_i(S_i', S_{-i}) - u_i(S_i, S_{-i}) = \mathcal{P}(S_i', S_{-i}) - \mathcal{P}(S_i, S_{-i}). \tag{4.28}$$

Then, $\mathcal{P}(S_i, S_{-i})$ defined in (4.14) is an EPG of game Γ.

Chapter 5

Joint Iterative Beamforming and Power Adaptation for Wireless Ad-Hoc Networks

5.1 Introduction

Using multiple-input multiple-output (MIMO) techniques in wireless communications has always attracted an increasing interest. The use of multiple antennas boosts up the capacity and the spectral efficiency of communication systems [35] [36]. Factors such as large number of simultaneous transmissions and reduced cell-sizes necessitates the interference management for wireless MIMO systems [37]. More efficient interference management and higher performance results can be achieved by exploiting adaptation capability in wireless MIMO systems. System adaptability can be achieved by allowing transmitters and receivers to adapt their transmitter parameters (e.g. powers, beamformers, frequency, modulation, rate etc.) or receive filters according to communication environment over time. Rather than using fixed system parameters with complex transmitter and receiver designs, the adaptation capability of the wireless system can increase the system efficiency with less computation complexities.

Transmit beamforming has been the focus of extensive research in the literature [38] [39] [40] [41] [42] [43] [44] [45] and designing optimum signaling at the transmitter has shown important improvements for systems operating in varying interference environments [19] [38] [40] [46] [47] [48] [49]. In spatial transmit beamforming, each communicating node's symbol stream is multiplied by a preselected transmit beamforming weight vector for transmission through multiple antennas such that the overall interference due to other multiple nodes is minimized. Optimization of the signal and beamforming done at the transmitter improves the efficiency by steering the beam towards the intended receiver while placing nulls toward the unintended receivers in order to avoid causing excessive interference to them. The transmitters may adapt their signals through a low rate feedback channel, based on the received system information from the receiver [50]. Power control mechanism can also be combined with limited rate feedback from the receiver in order to satisfy certain Quality-of-Service (QoS) requirements at the receiver [51] [52] [53].

In general, MIMO beamforming techniques in communication systems are addressed in three different systems: point-to-point, cellular, and ad hoc networks. The great potential of MIMO in point-to-point communication is shown in [35] [38] [40] [54] and designing linear precoders (eigencoders) and beamformers are well studied for point-to-point MIMO links [39] [41]. In cellular networks, beamforming algorithms are designed as a means to minimize the total power or to enhance the capacity using MIMO techniques for array-equipped base station and single antenna mobile transmitters [42] [43] [44] [45]. In ad hoc networks, without a central controller, distributed beamforming techniques are used to overcome lower system throughput and higher energy consumption [46] [55] [56] [57]. The optimization solutions designed in ad-hoc networks need careful study, because the environment is interference limited and the performance of MIMO techniques depends significantly on the overheads introduced by the proposed algorithms.

Distributed spatial beamforming algorithms are proposed for multi-user ad-hoc MIMO networks under channel reciprocity conditions in [56] [57]. Channel reciprocity holds when the channel matrix in both directions of a MIMO link are matrix transpose of each other which is usually assumed in time-division duplex (TDD) systems [57]. Bromberg et al. [57] consider the capacity maximization problem and proposes a locally enabled global optimization (LEGO) algorithm for distributed beamforming update under Gaussian other-user interference. Iltis et al. [56] formulate the problem as a noncooperative game for overall power minimization of the network under a constant QoS constraint (target signal-to-interference plus noise ratio (SINR)). The proposed iterative minimum mean-square error (IMMSE) algorithm solves an optimization problem by computing transmit/receive beamformer pairs and transmit powers in a distributed manner [56]. In IMMSE algorithm, the receive beamformer is enforced to be equal to the conjugate of the transmit beamformer and the algorithm relies on the channel reciprocity condition. Hence, the IMMSE algorithm does not demand explicit feedback schemes for channel state information (CSI) at the transmitters. However, during the updating procedure of the IMMSE algorithm, transmitting overheads of training sequences and power control commands are incurred. The amount of overheads increases with iterations, since the algorithm performs transmit/receive beamformer and power updates iteratively. Moreover, if the transmitter and receiver are communicating by using different channels or frequencies for its transmission and reception, i.e. when the channel reciprocity is not valid, CSI is required to feedback to the transmitter, which sometimes needs high overhead.

In order to lower communication overhead between transmitter and receiver and for conditions in which channel reciprocity does not hold, quantized transmit beamforming codebook design using limited feedback beamforming scheme for single user MIMO systems is proposed in [54]. The concept is based on selecting a codeword in a predetermined codebook that is known to both transmitter and receiver. The transmit beamformer selection from a predefined codebook can reduce the latency in highly mobile and unstable communication networks. Moreover, when the communication system needs low rate or has bandwidth constraints, feedback overhead in situations like non-reciprocal channels are substantially reduced using the proposed codebook design approach. In this scenario, the receiver only feeds back the index of the selected transmit beamformer to the transmitter. In the case when there is no channel reciprocity between transmitters and receivers, an iterative limited feedback beamforming algorithm is proposed in [58]. The algorithm is studied for rate maximization problem in MIMO multi-user ad-hoc networks using sequential discrete transmit beamformer selection update algorithm. The algorithm performs iterative best response strategy which is the maximization of the received SINR by each node at each iteration. However, the convergence analysis of the proposed algorithm is not followed.

Using game theory has given useful solutions for efficiency and convergence proofs of some of the important problems in wireless communications such as distributed power control algorithm design [59], joint code-division multiple access (CDMA) waveform and power control design [52] [53] [60] and optimum transmission signaling strategies [61] [62]. On the other hand, application of game theory for distributed beamforming update is problematic [56] which will also be noted in this chapter. Lacatus et al. [53] and Popescu et al. [52] study joint CDMA codeword (or sequence) and power adaptation using noncooperative game model. The problem is formulated as a separable game using noncooperative convex games, with corresponding sub-games: power control and codeword control game. However, in contrast to our joint optimization problem, the joint optimization of powers and CDMA codewords is investigated only over *convex games* (i.e. the set of action space is non-empty, compact and convex [12] [59]), and therefore the decision variables (i.e. the powers and codeword sequences) are continuous, not discrete in these games.

Optimum transmit signaling for MIMO interference systems for rate maximization problem using game theory has been studied in [19] [46] [47] [48] [49]. In these papers, the system is modeled as noncooperative game where every MIMO link is a player and computes against the others by choosing the transmit covariance matrix as the transmission strategy to maximize their own rate as the utility function. In [47], rate maximization for MIMO ad-hoc networks is performed by power control and the existence of Nash equilibrium (NE) solution is shown using the concave game analysis. However, the convergence analysis of the proposed algorithms is not followed. In [19] [46] [48] [49] [61] [62], the decision parameters are selected as the covariance matrices of the transmitted signal vector. Arslan et al. [48], show that individual mutual information maximization game is a concave game [63] in MIMO interference channels, which implies the existence of a NE for arbitrary channel matrices. The uniqueness of the equilibrium is proved when the multi-user interference (MUI) is almost negligible or sufficiently small. Decentralized algorithms using local information are proposed as update strategies to determine the link parameters. As an extension of their work and for more general conditions, the uniqueness of the NE solution is provided in [19]. Scutari et al. [19] provide a unified framework for the noncooperative mutual information maximization problem for MIMO interference systems. A unified set of sufficient conditions guaranteeing the uniqueness of the NE and the convergence of asynchronous water-filling algorithm is provided for *square non-singular* channel matrices. The analysis is based on interpretation of MIMO waterfilling operator as a matrix projection onto the convex and closed set of covariance matrices. In [49], same authors extend their results for *arbitrary* channel matrices. However, in these papers the selection of *discrete* optimized signaling has not been exclusively investigated. The existence (or uniqueness) of the NE solution that is proved in [47] [48] is valid either for *convex or concave games* or for positive definite covariance matrices that are well defined as a convex and closed set [19] [49]. On the other hand, cooperative and noncooperative algorithms for joint channel and power allocation chosen from a “discrete” strategy space are studied in [64] in the context of wireless mesh networks. However, the proposed noncooperative algorithm is suboptimum and one of the adaptation parameters (i.e. channel adaptation) is not followed after the first iteration.

In MIMO ad-hoc networks in MUI environments, the transmission scheme of each user also depends on that of other users, since the interferences at each user depends on the transmission parameters of the other users. The decision of each user also reshapes the interference emitted to other links. Therefore, power minimization problem using distributed algorithms with transmit beamformer selection game is challenging especially in ad-hoc networks. Unlike power control games, the questions of optimal power minimization algorithms are difficult to address for the beamforming games, due to the lack of a natural ordering of the actions [56]. For example, the transmit beamformer and power of one node pair affects the SINR of other node pairs, and vice versa. Moreover, if the node pairs belong to different regulation entities, the non-cooperative node pairs may only want to minimize their own transmit power rather than the overall one. Therefore, finding the optimal distributed transmit beamformer solution for the power minimization problem is not straightforward.

The analysis for the selection of actions from the discrete set and convergence analysis is still missing for joint transmit beamforming and power adaptation in the literature. To the best of authors' knowledge, the problem of joint discrete transmit beamforming and power adaptation has not been formalized in multi-user MIMO ad-hoc networks. In this chapter, we study a decentralized approach for optimizing the transmit beamformer and power levels using local information and reasonable computational burden. We consider the total power minimization under constant received target SINR constraint. Our contributions in this chapter are twofold: First, we study an efficient cooperative beamforming algorithm for global power minimization problem with convergence analysis. Note that when performing cooperative algorithm, the amount of information to be exchanged between nodes will grow with the number of iterations. Second, we study a noncooperative regret-matching learning algorithm for joint transmit beamformer and power update for the total network power minimization. The noncooperative update solution reduces the amount of overhead by using only local information. We compare the performance of proposed algorithms with the optimal global solution which is found by searching all feasible strategy space.

5.2 System Model and Concepts

In this chapter, we consider a wireless ad hoc network shown in Fig. 5.1. The ad hoc network consists of multiple transmit and receive antenna node pairs. All nodes are assumed to be using same channel. The interference comes from the other node pairs which operate simultaneously on the same channels. In this ad hoc network model, there are N node pairs and each node pair $m \in \{1, 2, ..., N\}$ consists of one transmitter node and one receiver node. Each transmitter and receiver node is equipped with T antennas. The complex symbol stream transmitted is $b_m \in \mathbb{C}$ with $E\{|b_m|^2\} = 1$. Each node has a unit-norm receive/transmit beamformer pair $(\mathbf{w}_m, \mathbf{t}_m)$ with $\mathbf{w}_m, \mathbf{t}_m \in \mathbb{C}^T$.

The received signal vector $\mathbf{r}_m \in \mathbb{C}^T$ at the $m-$th receiving node is given by

$$\mathbf{r}_m = \sqrt{P_m}\mathbf{H}_{m,m}\mathbf{t}_m b_m + \sum_{i \neq m} \sqrt{P_i}\mathbf{H}_{m,i}\mathbf{t}_i b_i + \mathbf{n}_m, \tag{5.1}$$

where $\mathbf{H}_{m,i}$ denotes $T \times T$ MIMO channel between the i'th transmitting node and the $m-$th receiving node and is quasi-static and P_m is the power of the $m-$th transmitting node. The additive white Gaussian noise terms $\mathbf{n}_m \in \mathbb{C}^T$ have identical covariance matrices $\sigma^2\mathbf{I}_T$ where σ^2 is the noise power and $\mathbf{I}_T$ is the $T \times T$ identity matrix. Note that the first term of the right-hand side of (5.1) is the desired signal, whereas the

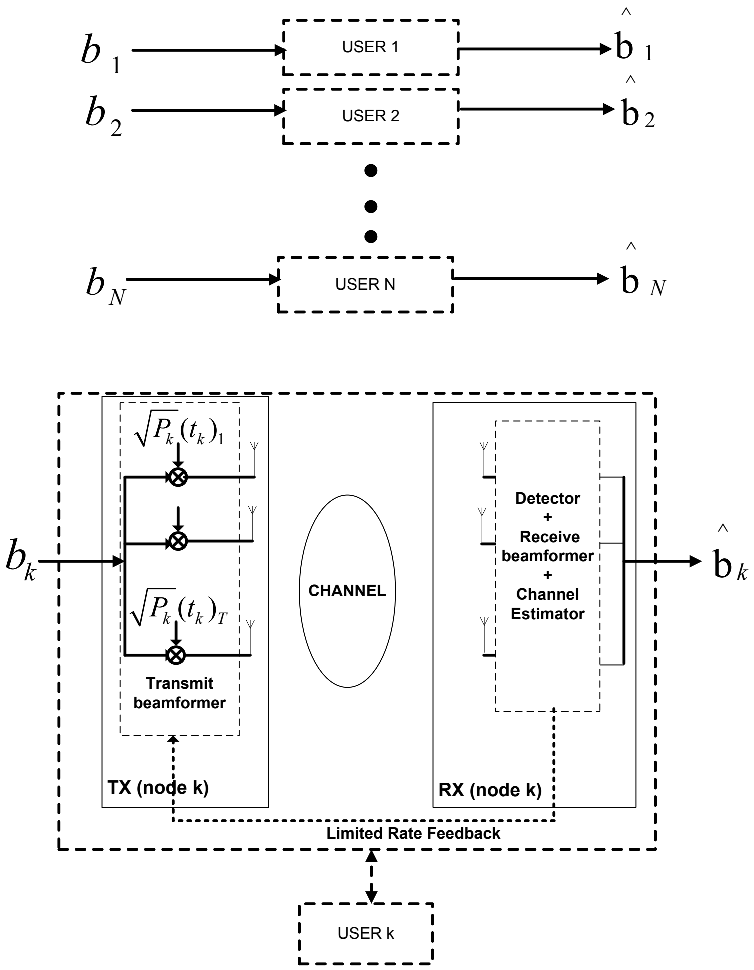


Figure 5.1: Multi-user power control and limited feedback transmit beamforming scheme for MIMO ad-hoc networks. $(\mathbf{t}_k)_m$ represents the m'th row of the k'th user's transmitter vector $\mathbf{t}_k$.

second term is the interference from the other transmitting nodes.

As we are interested in the minimum achievable power, we consider the worst case where all node pairs always have some packets to transfer and all nodes in the network can transmit simultaneously. The network is assumed to be synchronous The set of available codebook beamformers for the $m-$th transmitting and receiving node pair is denoted by $\Delta_m = \{\mathbf{t}_m^1, \mathbf{t}_m^2, ..., \mathbf{t}_m^\Upsilon\}$ with cardinality Υ. In a limited feedback beamforming system, the receiving node selects a transmit beamformer among the codebook feeds back the index of the selected beamformer. Each node can select between Υ transmit beamformer vectors. We assume that there is only one way communication link for all $m-$th node pairs. Let $\mathbf{t}_m \in \Delta_m$ be the selected transmit beamformer for the $m-$th transmitting and receiving node pair. Denote $\Theta = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N]^T$ and $\mathbf{P} = [P_1, P_2, ..., P_N]^T$ as the transmit beamformer selection and transmission power vectors for N nodes respectively. The $T \times T$ the interference plus noise covariance matrix at $m-$th receiving node is

$$\mathbf{R}_m(\Theta_{-m}, \mathbf{P}_{-m}) = \sum_{i \neq m} P_i \mathbf{H}_{m,i} \mathbf{t}_i \mathbf{t}_i^H \mathbf{H}_{m,i}^H + \sigma^2 \mathbf{I}, \tag{5.2}$$

where Θ_{-m} and $\mathbf{P}_{-m}$ are the transmit beamfomers and powers of nodes other than m.

An antenna beam pattern that adjusts the antenna gains to form nulls towards the direction of the interferers while keeping a constant gain towards the directions of the multi-path of the intended receiver can be designed using receive antenna arrays. The minimum variance distortionless response beamformer [56] [65] can adjust the array weights properly such that the sum of interference and noise is minimized. The normalized receive beamformer at $m-$th receiving node is

$$\mathbf{w}_m = \frac{\hat{\mathbf{w}}_m}{||\hat{\mathbf{w}}_m||}, \tag{5.3}$$

where $\hat{\mathbf{w}}_m = \mathbf{R}_m^{-1} \mathbf{H}_{m,m} \mathbf{t}_m$. The resulting received SINR at the $m-$th receiving node due to desired transmitter of $m-$th node pair is

$$\mathbf{\Gamma}_m = \frac{P_m |\mathbf{w}_m^H \mathbf{H}_{m,m} \mathbf{t}_m|^2}{\sum_{i \neq m} P_i |\mathbf{w}_m^H \mathbf{H}_{m,i} \mathbf{t}_i|^2 + \sigma^2}, \tag{5.4}$$

where $||\mathbf{w}_m||^2 = ||\mathbf{t}_m||^2 = 1$ for all m.

The proposed distributed algorithms attempt to achieve a target SINR by adjusting transmit powers. To construct a distributed iterative limited feedback beamforming scheme, let us first consider the case when there is only one node pair in the wireless network. The receiver selects the transmit beamformer from the codebook Δ_1 as

$$\mathbf{t}_1^* = \arg\max_{\mathbf{t}_1 \in \Delta_1} \Gamma_1, \tag{5.5}$$

where $\mathbf{t}_1^*$ is the optimal transmit beamformer selection for one node pair. Then, the receiver returns the index of the beamformer for transmit beamformer selection $\mathbf{t}_1^*$ and the received "normalized" SINR, $(\mathbf{t}_1^*)^H \mathbf{H}_{1,1}^H \mathbf{R}_1^{-1} \mathbf{H}_{1,1} \mathbf{t}_1^*$, through low-rate feedback channel. The transmitter selects the transmitter beamformer in order to minimize its own transmission *power* P_1, where P_1 is updated as

$$P_1 = \frac{\gamma_0}{(\mathbf{t}_1^*)^H \mathbf{H}_{1,1}^H \mathbf{R}_1^{-1} \mathbf{H}_{1,1} \mathbf{t}_1^*}, \tag{5.6}$$

where γ_0 is the target SINR value.

Consider now the case where N node pairs coexist in the wireless network. Note that for each node pair m, the value of received SINR, i.e. Γ_m is a function of $(\Theta, \mathbf{P})$. Therefore, the transmit power of one node pair depends not only on the transmit beamformer selection of itself, but also those of other node pairs' transmit power and transmit beamformer selections in the network. Furthermore, in beamforming, if user $i \neq m$ changes its transmit beamformer $\mathbf{t}_i$ to increase its own SINR Γ_i, it can either increase or *decrease* Γ_m, the SINR of link m, depending on the relative positions of the nodes. Therefore, designing an optimal distributed algorithm which converges to a set of beamformers to minimize the overall transmit power while meeting target

SINRs for all node pairs is not a straightforward task.

5.3 Optimization Problem and Game Theoretical Interpretation

The goal is to minimize the transmit power of all nodes $m \in \{1, 2, ..., N\}$ under constant target SINR γ_0. The optimization problem can be defined as,

$$\begin{aligned} \underset{\Theta,\mathbf{P}}{\text{Minimize}} \quad & \sum_{m=1}^{N} P_m, \\ \text{subject to} \quad & \Gamma_m \geq \gamma_0, ||\mathbf{w}_m|| = ||\mathbf{t}_m|| = 1, \\ & P_{min} < P_m \leq P_{max}, \;\; m \in \{1, 2, ..., N\}, \end{aligned} \tag{5.7}$$

where P_{min} and P_{max} are the minimum and maximum transmit powers, respectively. We consider the above problem as a normal form game, which can be mathematically defined by the triplet $\prod = \langle \mathcal{N}, \mathcal{C}, \{u_m\}_{m=1}^{N} \rangle$ where $\prod$ is a game, $\mathcal{N} = \{1, 2, ..., N\}$ is the finite set of players of the game, $\mathcal{C} = C_1 \times C_2 \times, ..., C_N$ represents the set of all available actions for all the players and $\{u_m\}_{m=1}^{N} : \mathcal{C} \rightarrow \mathbb{R}$ is the set of utility functions that the players associate with their strategies. Actions $c_m \in C_m$ for a player m are the transmit powers $P_m \in [P_{min}, P_{max}]$ and the transmit beamformer selections $\mathbf{t}_m \in \Delta_m$.

Players select actions to maximize their utility functions. We want to determine if there exists a convergence point (i.e. a Nash equilibrium (NE)) from which no player would deviate anymore for analyzing the outcome of the game see Section 2.21 for the definition of NE. One of the questions that arise is whether the beamforming selections $\Theta = [\mathbf{t}_1, ..., \mathbf{t}_m, ..., \mathbf{t}_N]^T$ and eventually power allocations $\mathbf{P} = [P_1, P_2, ..., P_N]^T$ will converge to NE solution. In the following section, we will discuss the scenarios where the node pairs are cooperative or non-cooperative in order to search for best results and convergence guarantees.

5.31 System Feasibility Region

In order to ensure that every user should have SINR no less than the required target SINR γ_0, i.e. $\Gamma_m \geq \gamma_0, \;\; m \in \{1, 2, ..., N\}$, we will define a system feasibility region Ω. We rewrite the SINR in (5.4) as

$$\mathbf{\Gamma}_m = \frac{P_m G_{mm}}{\sum_{i \neq m} P_i G_{mi} + \sigma^2}, \tag{5.8}$$

where $G_{mm} = |\mathbf{w}_m^H \mathbf{H}_{m,m} \mathbf{t}_m|^2$ and $G_{mi} = |\mathbf{w}_m^H \mathbf{H}_{m,i} \mathbf{t}_i|^2$. The inequality $\Gamma_m \geq \gamma_0, \;\; m \in \{1, 2, ..., N\}$ can be written in matrix form as,

$$(\mathbf{I} - \mathbf{DG})\mathbf{P} \geq \mathbf{v}, \tag{5.9}$$

where $\mathbf{I}$ is a $N \times N$ identity matrix, $\mathbf{v} = [v_1, v_2, ..., v_N]^T$ with $v_m = \sigma^2 \gamma_0 / G_{mm}$, $\mathbf{D} = diag\{\gamma_0, ..., \gamma_0\}$ is a $N \times N$ matrix and

$$[\mathbf{G}_{mi}] = \begin{cases} 0 & \text{if } i = m, \\ G_{mi}/G_{mm} & \text{if } i \neq m. \end{cases}$$

By Perron-Frobenius Theorem [66], there exists a positive power allocation to achieve the desired targeted SINRs if and only if the maximum eigenvalue of $\mathbf{DG}$, i.e. spectrum radius $\rho(\mathbf{DG})$ is inside the unit circle [67]. When $|\rho(\mathbf{DG})| < 1$, the optimal power solution is

$$\mathbf{P} = \begin{cases} (\mathbf{I} - \mathbf{DG})^{-1}\mathbf{v} & \text{if } |\rho\mathbf{DG}| < 1, \\ +\infty & \text{otherwise.} \end{cases}$$

Note that the system feasibility region Ω is the supporting domain where there exist positive power vector solutions. The above condition for optimality is a necessary condition for existence of feasible region Ω. In this chapter, we assume that we are in the system feasibility region for all $\Theta \in \Upsilon^N$

5.4 Cooperative and Noncooperative Beamforming for MIMO ad-hoc networks

5.41 Optimal (Centralized) Solution

In a wireless ad hoc network, the centralized agent can select the transmit beamformers and the corresponding transmit powers to minimize the total transmit power of all transmitting antennas as,

$$(\Theta^*, \mathbf{P}^*) = \arg\min_{\Theta, \mathbf{P}} \sum_{m=1}^{N} P_m(\Theta, \mathbf{P}_{-m}), \tag{5.10}$$

where $\Theta^* = [\mathbf{t}_1^*, \mathbf{t}_2^*, ..., \mathbf{t}_N^*]^T$ and $\mathbf{P}^* = [P_1^*, P_2^*, ..., P_N^*]^T$ are the optimal transmit beamformer and power solutions respectively. The transmit power P_m of $m-$th node pair is defined as

$$P_m(\Theta, \mathbf{P}_{-m}) = \frac{\gamma_0}{\mathbf{t}_m^H \mathbf{H}_{m,m}^H \mathbf{R}_m^{-1} \mathbf{H}_{m,m} \mathbf{t}_m}, \tag{5.11}$$

where $\mathbf{R}_m$ is a function of $(\Theta_{-m}, \mathbf{P}_{-m})$ as shown in (5.2). A naive approach to solve the problem is to investigate all strategy profiles $\Theta = (\mathbf{t}_1, ..., \mathbf{t}_m, ..., \mathbf{t}_N)^T$ exhaustively (Note that for a given fixed strategy profile Θ, the corresponding power profile $\mathbf{P}$ can be computed using (5.11) for each individual node pair m). In order to compute (5.10), the centralized agent evaluates the total network power for Υ^N possible beamforming vector combinations. For example, for a network size with 10 node pairs where each user has to select from a codebook of size $\Upsilon = 16$ beamformers, the search space is

16^{10} strategy profiles. Obviously, finding the centralized transmit beamformer is cumbersome in large-scale wireless ad-hoc network. Next, we will introduce decentralized power minimization algorithms using cooperative and noncooperative techniques.

5.42 Cooperative Power Minimization using Beamforming

In this section, we consider the scenarios where all node pairs in wireless network are cooperative. In a cooperative game, nodes in the network are able to coordinate and select the transmit beamformer accordingly. We want to find the transmit beamformer and power assignments such that the overall power in the whole network is minimized. The objective function can be written as

$$u_{network}(\mathbf{\Theta}, \mathbf{P}) = -\sum_{m=1}^{N} P_m(\mathbf{\Theta}, \mathbf{P}_{-m}). \tag{5.12}$$

We assume that each user's utility function is (5.12). That is,

$$u_i(\mathbf{\Theta}, \mathbf{P}) = u_{network}(\mathbf{\Theta}, \mathbf{P}) = -\sum_{m=1}^{N} P_m(\mathbf{\Theta}, \mathbf{P}_{-m}),\ \forall i \in \mathcal{N}. \tag{5.13}$$

In other words, we model the game as *identical interest game* which is a special case of potential games [14] [68] as described in Section 2.31. It is easy to verify that all identical interest games have at least one pure NE, namely any action profile that maximizes $u_{network}(\mathbf{\Theta}, \mathbf{P})$ [48] [64]. We analyze a cooperative power minimization algorithm (COPMA) which can converge to the optimal NE with arbitrarily high probability. This method is analogous to the decentralized negotiation method called *adaptive play* [48]. The key characteristic of COPMA is the randomness deliberately introduced into the decision making process to avoid reaching a local solution. In COPMA, the choices of players (in our case transmit beamformer selections) lead the system to the optimal NE solution with arbitrarily high probability [48].

Inspired by [64], a decentralized implementation of COPMA can be done as follows: Assume that each node pair m in the network has an unique ID_m and maintains two variables $P_m^{current}$ and $P_m^{updated}$ which are the $m-$th node pairs transmit power in the network previously and after the random change of transmit beamformer, respectively. The node pairs can be chosen randomly or in a round-robin order for updating of the transmit beamformers. Whenever a node pair changes its strategy or transmit beamformer, it broadcasts a vector $\{ID_m, P_m^{current}, P_m^{updated}\}$ via a backbone network. After that, all the other node pairs $i \in \mathcal{N} \backslash m$ will set $P_i^{current} = P_i^{updated}$, recalculate $P_i^{updated}$ as the new transmit power and will send the vector $\{ID_i, P_i^{current}, P_i^{updated}\}$ to the updating node pair m. Finally, $m-$th node pair will decide whether the new transmit beamformer should be kept or not with some probability which depends on $P_{current}$ and $P_{updated}$ which are the total transmit power in the network previously and after the random change of transmit beamformer respectively. The detailed description of COPMA is provided as follows:

Initialization: For each transmitting and receiving pair m, the initial index of transmit beamformer is selected as one and the initial transmit powers are set as $P_m = P_{max}$, $\forall m \in \mathcal{N}$.

Repeat: Randomly choose a node pair m as the updating one with probability $1/N$. Denote $\mathbf{t}_m(n) \in \Delta_m$ as the current transmit beamformer of $m-$th node pair in iteration n.

1. Set $\mathbf{t}_m(n) = \mathbf{t}_m(n-1)$, $\forall m \in \mathcal{N}$. Calculate $P_m^{current}$ as in (5.11) $\forall m \in \mathcal{N}$.

2. To update node pair m, randomly choose a transmit beamformer, $\mathbf{t}_m^{updated} \in \Delta_m$ and calculate $P_m^{updated}$ as in (5.11). Then, broadcast a data vector $\{ID_m, P_m^{current}$, $P_m^{updated}\}$ to all other node pairs $i \in \mathcal{N} \backslash m$.

3. After receiving the data vector, for each i,

    - If P_i changes (due to change in interference perceived at the $i-$th receiver), every other node pair $i \in \mathcal{N} \backslash m$ sets $P_i^{current} = P_i^{updated}$ and calculates its new transmit power from (5.11) and sets it to $P_i^{updated}$.
    - If P_i does not change, $P_i^{current}$ and $P_i^{updated}$ remain unchanged.

    After $P_i^{current}$ and $P_i^{updated}$ are updated for every other node pairs $i \in \mathcal{N} \backslash m$ in the network, send back the vector $\{ID_i, P_i^{current}, P_i^{updated}\}$ to node pair m.

4. Node pair m computes the total current total network power as $P_{current} = \sum_{m=1}^{N} P_m^{current}$ and total updated network power as $P_{updated} = \sum_{m=1}^{N} P_m^{updated}$ with $\mathbf{t}_m^{updated}$ based on the received power values from all other node pairs $i \in \mathcal{N} \backslash m$.

5. For a *smoothing factor* $\tau > 0$, $\mathbf{t}_m(n) = \mathbf{t}_m^{updated}$ for the $m-$th node pair with probability
$$\frac{1}{1 + \exp\left((P_{updated} - P_{current})/\tau\right)}, \tag{5.14}$$
    i.e. the updating node pair m selects $\mathbf{t}_m^{updated}$ with probability (5.14).

6. The $m-$th node pair broadcasts a notifying signal that contains the decision about whether the new transmit beamformer is kept. If not kept, every other node pair $i \in \mathcal{N} \backslash m$ keeps $P_i^{updated} = P_i^{current}$

Until : Predefined number of iteration steps $n = \kappa$.

Note that step-5 of the updating rule implies that if $\mathbf{t}_m^{updated}$ yields a better performance, i.e. $(P_{updated} - P_{current}) < 0$, the $m-$th node pair will change to updated beamformer $\mathbf{t}_m^{updated}$ with high probability. Otherwise, it will keep the current transmit beamformer with high probability. Note also that the tradeoff between COPMA's performance and convergence speed is controlled by the parameter τ. Large τ represents extensive space search with slow convergence, whereas small τ represents restrained space search with fast convergence. The smoothing factor τ is selected to be a function of n such that as n increases, $\tau \downarrow 0$. For example, we choose τ inversely proportional to n^2 in our simulations. The long term behavior of COPMA is characterized in the following theorem.

Theorem 5.4.1. *Assume that the objective of each node pair is defined as sum power minimization in the network as defined in (5.10). Let $\Theta(k) = [\mathbf{t}_1(k), \mathbf{t}_2(k), ..., \mathbf{t}_N(k)]^T$ denote the profile of choices at step k in COPMA and $\Theta^* = [\mathbf{t}_1^*, \mathbf{t}_2^*, ..., \mathbf{t}_N^*]^T$ the optimal profile. COPMA converges to the optimal NE with arbitrarily high probability. In other words,*

$$\lim_{\tau \to 0} \lim_{k \to +\infty} \mathbb{P}_\tau^*(\Theta(k) = \Theta^*) \ = 1. \tag{5.15}$$

Proof. The proof of Theorem 5.4.1 follows along the similar lines of the proof in [48] [64] and [69]. Notice that the transmit beamformer selection with N players each with Υ codebook size generates an N-dimensional Markovian chain on a finite state space with Υ^N states or different *profiles*. First, we study the analysis for two player games, i.e. $N = 2$ dimensional case as shown in Fig. 5.2.

Let $\mathbf{t}_m \in \Delta_m$ and $\mathbf{t}_k \in \Delta_k$ be the choices of two players say m and k, and assume that $\Delta = \Delta_m = \Delta_k = [\mathbf{t}_m^1, \mathbf{t}_m^2, .., \mathbf{t}_m^\Upsilon]$. The players m and k can choose a transmit beamformer from Δ. Let Θ_{ij} denote the state $[\mathbf{t}_m^i, \mathbf{t}_n^j] \in \Upsilon^2$ where $m-$th user selects $i-$th transmit beamformer $\mathbf{t}_m^i$ and $n-$th user selects $j-$th transmit beamformer $\mathbf{t}_n^j$. At an arbitrary time instant, for any state of the Markovian chain, only one of the players can update their transmit beamformer. Therefore, for example in Fig.5.2, state $\Theta_{ij} = [\mathbf{t}_m^i, \mathbf{t}_n^j]^T$ can only transit into a state either in the same row or the same column. For any fixed $\tau > 0$, the transition probability from state

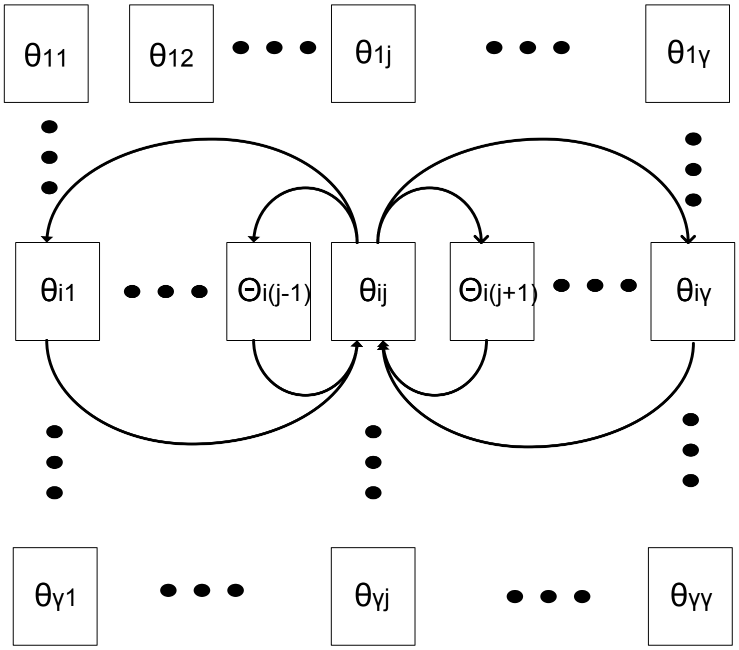


Figure 5.2: Markov chain with two players for COPMA.

$\Theta_{ij} = [\mathbf{t}_m^i, \mathbf{t}_n^j]^T \in \Upsilon^2$ to state $\Theta_{lp} = [\mathbf{t}_m^l, \mathbf{t}_n^p]^T \in \Upsilon^2$ is given by

$$\mathbb{P}_\tau(\Theta_{lp}|\Theta_{ij}) = \frac{1}{2\Upsilon(1 + e^{(\mathbf{P}(\Theta_{lp}) - \mathbf{P}(\Theta_{ij}))/\tau})}, \tag{5.16}$$

where Θ_{ij} and Θ_{lp} differ in exactly one transmit beamformer selection, i.e. $\Theta_{ij} \neq \Theta_{lp}$ for $i = l$ or $j = p$, τ is the smoothing factor of COPMA and $\mathbf{P}(\Theta_{ij})$ is the minimum total network power required to reach target SINR γ_0 for both users at state Θ_{ij} calculated using (5.11) for each user. If Θ_{ij} and Θ_{lp} are different in more than one position, then $\mathbb{P}_\tau(\Theta_{lp}|\Theta_{ij}) = 0$. In addition $\mathbb{P}_\tau(\Theta_{ij}|\Theta_{ij}) > 0$ is always true. Therefore, for any fixed $\tau > 0$, the induced Markov chain is irreducible and aperiodic.

The stationary distribution $\mathbb{P}_\tau^*$ for each state can be obtained from the following balanced equations (using the arrows in Fig. 5.2):

$$\sum_{p=1,p\neq j}^{\Upsilon} \mathbb{P}_\tau^*(\Theta_{ij}) \times \mathbb{P}_\tau(\Theta_{ip}|\Theta_{ij}) = \sum_{p=1,p\neq j}^{\Upsilon} \mathbb{P}_\tau^*(\Theta_{ip}) \times \mathbb{P}_\tau(\Theta_{ij}|\Theta_{ip}), \tag{5.17}$$

for all $\Theta_{ij} \in \Upsilon^2$ and $\Theta_{ip} \in \Upsilon^2$. Substituting (5.16) into (5.17) gives

$$\begin{aligned} &\sum_{p=1,p\neq i}^{\Upsilon} \mathbb{P}_\tau^*(\Theta_{ij}) \times \frac{1}{2\Upsilon(1+e^{(\mathbf{P}(\Theta_{ip})-\mathbf{P}(\Theta_{ij}))/\tau})} \\ &= \sum_{p=1,p\neq i}^{\Upsilon} \mathbb{P}_\tau^*(\Theta_{ip}) \times \frac{1}{2\Upsilon(1+e^{(\mathbf{P}(\Theta_{ij})-\mathbf{P}(\Theta_{ip}))/\tau})}. \end{aligned} \tag{5.18}$$

Then, the stationary distribution of the induced Markov chain at step k is obtained as

$$\mathbb{P}_\tau^*(\Theta(k)) = \frac{e^{-\mathbf{P}(\Theta(k))/\tau}}{\sum_{\bar{\Theta}(k)\in\Upsilon^2} e^{-\mathbf{P}(\bar{\Theta}(k))/\tau}}, \tag{5.19}$$

for arbitrary state $\Theta(k) \in \Upsilon^2$. Hence from irreducibility and aperiodicity of the Markovian chain, we have

$$\lim_{\tau\to 0} \lim_{k\to+\infty} \mathbb{P}_\tau^*(\Theta(k) = \Theta^*) \ = 1, \tag{5.20}$$

where $\Theta^* \in \Upsilon^2$. The result validates that COPMA converges to the optimal state with arbitrarily high probability for two-player ($N = 2$) case. Finally, the analysis can easily be extended for general multi-player ($N > 2$), N dimensional Markovian chain cases as well. □

With the above theorem, the transmit beamformer selection is shown to reach the optimal Nash equilibrium solution with arbitrarily high probability. One disadvantage of cooperative based algorithms is that communication overhead incurred to calculate the total network power increases with iterations. In the next section, we study a noncooperative learning algorithm using local information with less computations.

5.5 Regret-Matching based joint transmit beamformer and power Selection Game (RMSG)

In this section, we want to obtain a distributed learning algorithm for joint transmit beamformer and power selection scheme in MIMO ad-hoc networks. We will use an utility function for noncooperative users. Note that the interaction among N "selfish" node pairs can be defined as non-cooperative power minimization game where each node pair is attempting to find their own transmit beamformers. In the noncooperative joint iterative beamforming and power adaptation, the N node pairs care only about their own power minimizations exclusively, rather than the overall network power. Again, each player's utility function depends on the transmit beamformer and power of itself as well as those of other transmit beamformers and powers. Note that noncooperative distributed beamforming algorithms for multi-user MIMO ad-hoc networks lacks the quality of "strategic complementarities" [70] that are found in power control-only games [59]. Therefore, it is thus not clear to design an ordered set of actions for noncooperative beamforming games and deterministic convergence analysis has not been followed yet. Hence, instead we study a noncooperative learning algorithm called the regret matching adaptive algorithm from [15], in which the players choose their actions based on the regret for not choosing particular actions in the past. The steady-state solution of the regret matching based learning algorithm exhibits "no regret" and the probability of choosing a strategy is proportional to the "regret" for not having chosen other strategies.

Denote $\bar{\mathfrak{t}}_m$ as the vector of all strategies or actions for user m, i.e. $\bar{\mathfrak{t}}_m =$

$[\mathbf{t}_m^1, \mathbf{t}_m^2, ..., \mathbf{t}_m^{\Upsilon}]$ and $\mathbf{t}_m(i)$ as the current selected transmit beamformer vector for the $m-$th user at iteration i. Define the average regret vector $Q_m^{\bar{\mathbf{t}}_m}(k)$ of user m for an action vector $\bar{\mathbf{t}}_m$ at iteration (or time) k as

$$Q_m^{\bar{\mathbf{t}}_m}(k) = \frac{1}{k-1}\sum_{i=1}^{k-1}(u_m(\bar{\mathbf{t}}_m, \mathbf{t}_{-m}(i)) - u_m(\mathbf{t}_m(i))). \tag{5.21}$$

In regret matching based joint transmit beamformer and power selection game (RMSG), each user m computes $Q_m^{\bar{\mathbf{t}}_m}$ for every action $\mathbf{t}_m \in \Delta_m$ in all past steps when all other player's actions remain unchanged. Each player m updates its regret $Q_m^{\bar{\mathbf{t}}_m}(k)$ for every set of actions $\bar{\mathbf{t}}_m$ based on the following recursion formula,

$$Q_m^{\bar{\mathbf{t}}_m}(k+1) = \frac{k-1}{k}Q_m^{\bar{\mathbf{t}}_m}(k) + \frac{1}{k}(u_m(\bar{\mathbf{t}}_m, \mathbf{t}_{-m}(k)) - u_m(\mathbf{t}_m(k))). \tag{5.22}$$

At every step $k > 1$, each user m updates its own average regret vector $Q_m^{\bar{\mathbf{t}}_m}(k)$ for every strategy in $\bar{\mathbf{t}}_m$. In regret matching, after computing the average regret vector, $Q_m^{\bar{\mathbf{t}}_m}(k)$, each user m chooses an action or strategy $\mathbf{t}_m(k), k > 1$, according to probability distribution $\chi_m(k)$ defined as

$$\chi_m^{\bar{\mathbf{t}}_m}(k) = \text{Prob}(\mathbf{t}_m(k) = \bar{\mathbf{t}}_m) = \frac{[Q_m^{\bar{\mathbf{t}}_m}(k)]^+}{\sum_{\bar{\mathbf{t}}_m \in \Delta_m}[Q_m^{\bar{\mathbf{t}}_m}(k)]^+}, \tag{5.23}$$

where $[x]^+$ denotes the non-negative part of x. Notice that in regret matching game, each user m chooses a strategy $\mathbf{t}_m \in \Delta_m$ at any step with probability proportional to the average regret for not choosing that strategy $\mathbf{t}_m \in \Delta_m$ in the past steps. The detailed summary of RMSG in using Gauss-Seidel updating scheme [19] is given in Table 5.1 where κ is the predefined number of iterations.

Note that every finite strategy game has a mixed strategy Nash equilibrium [12]. Therefore, for all finite games, using a proper learning algorithm, the game can be

Table 5.1: Regret-Matching based joint transmit beamformer and power selection game (RMSG) algorithm

```
Initialization: For each transmitting and receiving pair $m$, the initial transmit
beamformers are selected with equal probability, the initial transmit powers
are set as $P_m = P_{max}$ and the initial average regret vector is set as
$Q_m^{\bar{\mathbf{t}}_m}(1) = \mathbf{0}$, $\forall m \in \mathcal{N}$.

Iteration:
For $k = 1, 2, ..., \kappa$
    For $m = 1, 2, ..., N$
        – Update the average regret vector $Q_m^{\bar{\mathbf{t}}_m}(k)$ using the recursion
          in (5.22)
        – Update the probability distribution $\chi_m^{\bar{\mathbf{t}}_m}(k)$ in (5.23) and
          select the transmit beamformer $\mathbf{t}_m(k)$ based on updated $\chi_m^{\bar{\mathbf{t}}_m}(k)$.
        – Calculate the new transmit power $P_m$ based on selected $\mathbf{t}_m(k)$
          using (5.11).
    Next m
Next k
```

shown to converge to the fixed points of probability. The advantage of regret matching based selection is that it is distributed and requires limited information exchange between the users if the utility function is properly selected. It is also known that the time-averaged behavior of regret-matching game converges almost surely (with probability one) to the set of coarse-correlated equilibrium [14] [68] [13] which is described in Section 2.23. Therefore, the joint transmit beamformer and power selections converges to fixed points of probability. In fact, in our joint transmit beamformer and power selection game, the average regret of a user using regret matching becomes asymptotically zero, which is confirmed by our simulations and can be modified for future analytical convergence analysis. The utility function of noncooperative or "selfish" users for the transmit beamformer and power selection at iteration k is

$$U_m\left(\mathbf{t}_m, \mathbf{t}_{-m}(k)\right) = \log(\mathbf{t}_m^H \mathbf{H}_{m,m}^H \mathbf{R}_m^{-1} \mathbf{H}_{m,m} \mathbf{t}_m). \tag{5.24}$$

Note that by using the above utility function each user is trying to maximize its own "normalized" SINR, $\mathbf{t}_m^H \mathbf{H}_{m,m}^H \mathbf{R}_m^{-1} \mathbf{H}_{m,m} \mathbf{t}_m$, by selecting an appropriate transmit beamformer $\mathbf{t}_m \in \Delta_m$. Note also that during selection process of the best transmit beamformer using the above utility, each user can easily update the average regret in the recursion formula (5.22) locally.

5.6 Simulation Results

In this section, we investigate the performance results of centralized optimization, COPMA and RMSG. We assume that wireless ad hoc network has N homogeneous pairs where each pair has one transmitter node and one receiving node. Each entry in the channel matrix $\mathbf{H}_{m,k}$ $\forall m, k \in \mathcal{N}$ is assumed to be independent identically distributed complex Gaussian distribution with zero mean and unit variance. We consider a radio propagation with a path-loss exponent $\nu = 4$. This implies that the fading power is attenuated by d_m^{-4} where d_m is the distance between transmitter and receiver for $m-$th node pair. The target SINR γ_0 is selected to be 10 dB. We assume that channels don't vary during the iterations and the Grassmannian codebook of [54] is used for the simulation results. The codebook size is selected to be $\Upsilon = 16$ with $T = 3$ antennas for all users. $P_{max} = 100$ mW (20 dBm) and $P_{min} = 1$ mW (0 dBm) in our simulations. We assume six different transmit power levels: 1mW, 5mW, 20mW, 30mW, 50mW and 100mW motivated by the IEEE 802.11b standard in [71] (Note that the transmit powers are selected from this discrete power level set which corresponds to ceiling function of (5.11)). The selected network topologies are feasible for the given power levels, i.e. the power levels are between P_{min} and P_{max} for all $\Theta \in \Upsilon^N$ in the network (see also Section 5.31 for the necessary condition for the existence of system feasibility region). The noise power is $\sigma^2 = 3.16 \times 10^{-13}$ W $(-95$

dBm) which corresponds to approximate thermal noise power for a bandwidth of 20 MHz.

5.61 Small Networks

We first consider a small wireless ad-hoc network with 4 users, i.e. $N = 4$. All transmitting and receiving nodes are randomly located in a square of $30m \times 30m$ area. We choose $\tau = 0.1/n^2$ in our simulations, where n denotes the iteration step. The global optimum solution is obtained by enumerating all feasible strategies, i.e. 16^4 profiles, as the performance benchmark. The maximum number of iterations is $\kappa = 120$ for COPMA and RMSG. The performances of total power is shown in Fig. 5.3. As indicated by the "Centralized" curve, the global minimum power solution obtained by this approach functions as the lower bound of the overall power. We observe that COPMA's performance increases with iterations and settles at the global optimum combination after 92 iterations. Note that 68% and 76% of the gain from using COPMA algorithm is realized within the first 59 and 83 iterations respectively, although further improvement results from more iterations.

RMSG algorithm discussed in Section 5.5 is minimizing the total transmit power in the network defined by (5.10) using the utility function (5.24) in a noncooperative manner. Fig. 5.3 also shows how the total power in the network varies over 120 iterations using RMSG. Note that RMSG yields inferior performance compared to COPMA in terms of overall power. However, the updating procedure is noncooperative and requires less overhead as the iterations continue. The total network power converges to a value of 135 mW from the $68-$th iteration whereas the centralized case has 65 mW total network power. The steady state is reached when all the users select a transmit beamformer index with probability one.

Fig. 5.4 and Fig. 5.5 depict the trajectories of transmit beamformer selection

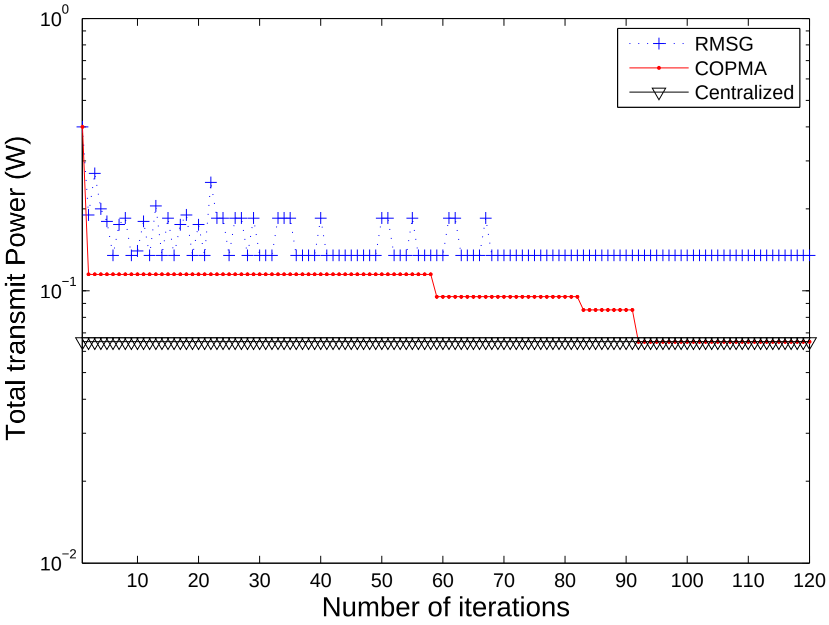


Figure 5.3: Total transmit power versus number of iterations with $N = 4$, $T = 3$ and $\Upsilon = 16$.

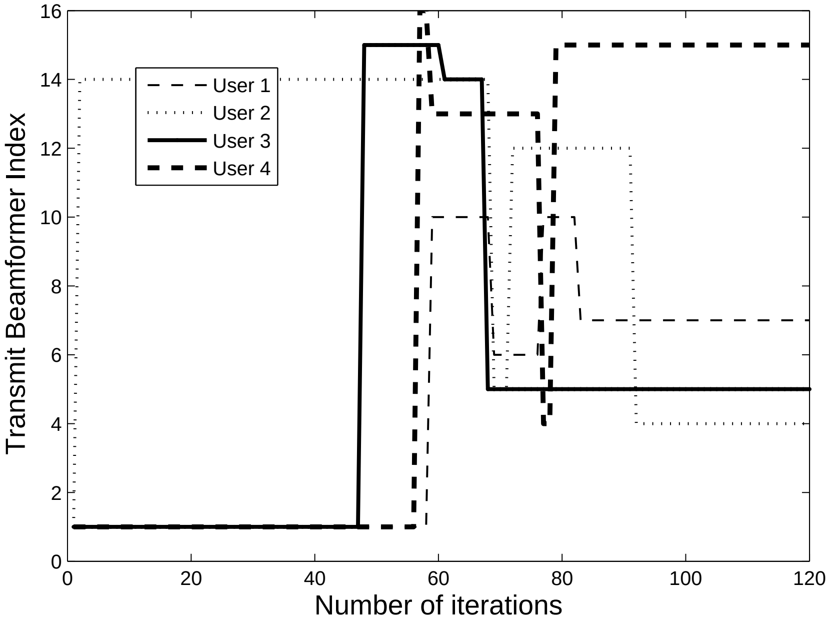


Figure 5.4: Transmit beamformer indexes versus number of iterations in COPMA with $N = 4$, $T = 3$ and $\Upsilon = 16$.

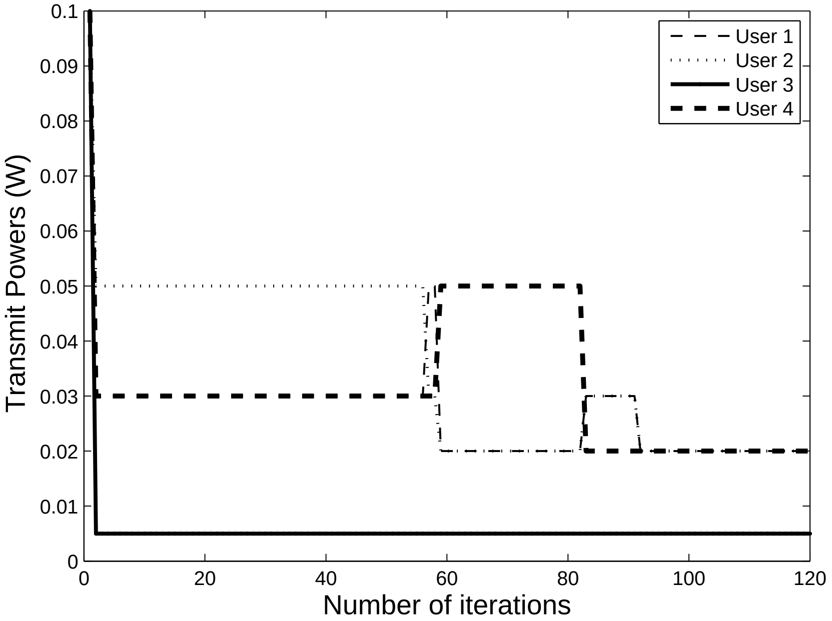


Figure 5.5: Transmit powers versus number of iterations in COPMA with $N = 4$, $T = 3$ and $\Upsilon = 16$.

indices and power trajectories in COPMA for each user in the network topology. At the initialization step, each user starts with maximum power levels and first index of transmit beamformer selections. Then, each user updates iteratively following COPMA algorithm, until the optimum Nash equilibrium is achieved. Note that when the transmit beamformers, Θ and power level vectors $\mathbf{P}$ converge in Fig. 5.4 and Fig. 5.5, the corresponding overall transmit powers obtained by COPMA is shown in Fig. 5.3. Therefore, the existence of NE and convergence into NE in COPMA is corroborated by curves in Fig. 5.4 and Fig. 5.5.

Probability mass function (p.m.f): In this subsection, we take a look at the probability mass function $\chi_m^{\bar{\mathbf{t}}_m}$ of the RMSG algorithm calculated in (5.23). Fig. 5.6 represents the change in the probability mass function after $1, 12, 50$ and 100 iterations for one of the user. Initially, the users choose the strategies, i.e. transmit beamformers, with equal probability where the strategies are represented by the indexes 1 to $\Upsilon = 16$ in the x-axis and the probabilities of selecting these indexes are on the y-axis. It is seen that after 12 iterations, the probability of choosing transmit beamformer index 9 is higher than that for other transmit beamformer index, although the other probabilities for indexes $3, 4$ and 12 are not totally eliminated. After 25 iterations, all other probabilities except those of 4 and 9 are eliminated. A stationary point is reached when user 1 chooses transmit beamformer index 9 at iteration 100.

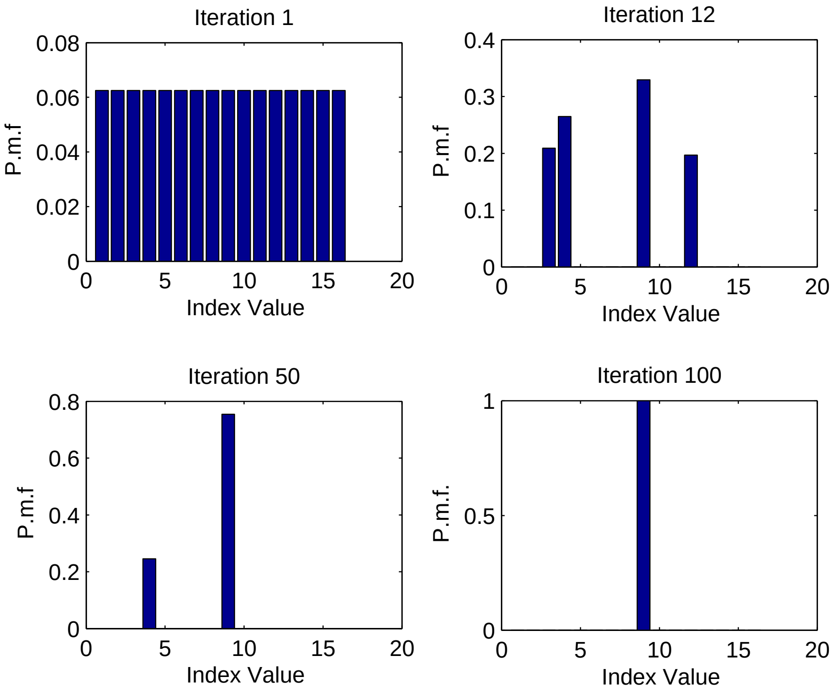


Figure 5.6: The probability distribution of RMSG for one of the users when $N = 4$.

5.62 Large Networks

We now consider a large wireless ad-hoc network with $N = 10$ node pairs randomly located on a $100m \times 100m$ area and smoothing factor for COPMA is selected as $\tau = 200/n^2$ in order to search more efficiently in this huge strategy space. The other simulation parameters are the same. Fig. 5.7 shows the network topology and transmit beampatterns of COPMA with $N = 10$ users. Note that the centralized approach is no longer feasible in this scenario due to the enormous strategy space (i.e. 16^{10} profiles).

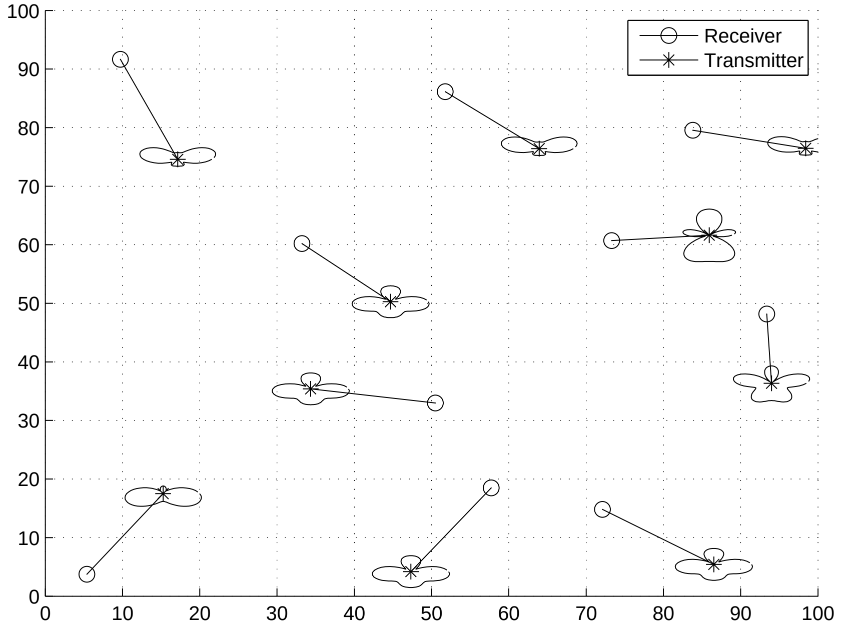


Figure 5.7: Node configuration and transmit beampatterns of COPMA with $N = 10$ users.

We again investigate both cooperative and regret-matching learning algorithms represented by COPMA and RMSG curves, where the maximum number of iterations is set to $\kappa = 400$ for COPMA and $\kappa = 1500$ for RMSG. COPMA's performance is close to optimal solution, thus provides a good benchmark to test the performance of RMSG. Fig. 5.8 shows the total network power versus number of iterations for COPMA and RMSG over increasing number of iterations. This figure shows that RMSG's performance is within 75.56% of the COPMA value at the end of iterations. Furthermore, RMSG needs larger amount of iterations compared to COPMA for the convergence. However, note that RMSG performs noncooperative update for transmit beamformer and powers at each iterations and the amount of overhead is minimum. In RMSG algorithm, the total power converges to total network power of 0.2250 W from the 1296−th iteration. The steady state for the joint selection of transmit beamformer indexes and transmit powers is reached when "all" the users in the network do not deviate from their chosen strategies. Note also that the majority of users reach a steady state within 115 iterations. However, one of the user takes longer than 1000 iterations to reach steady state.

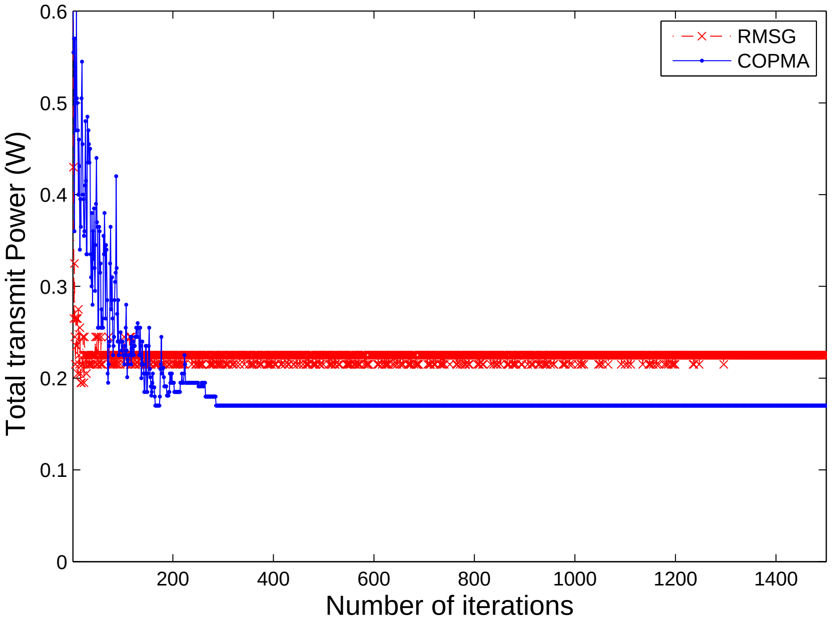


Figure 5.8: Total transmit power versus number of iterations with $N = 10$, $T = 3$ and $\Upsilon = 16$.

Probability mass function (p.m.f): A similar figure for the probability mass function of RMSG for one of the user that takes longer convergence than others is shown in Fig. 5.9 for large network size with $N = 10$. As can be seen in Fig. 5.9, the probability of choosing index 16 is higher than other indexes at iteration 500, but the probability of choosing index 5 and 12 is not totally eliminated even after 1000 iterations. Since the network size is large, the learning process takes a longer time to converge (around 1332 iterations) to steady-state transmit beamformer indexes,

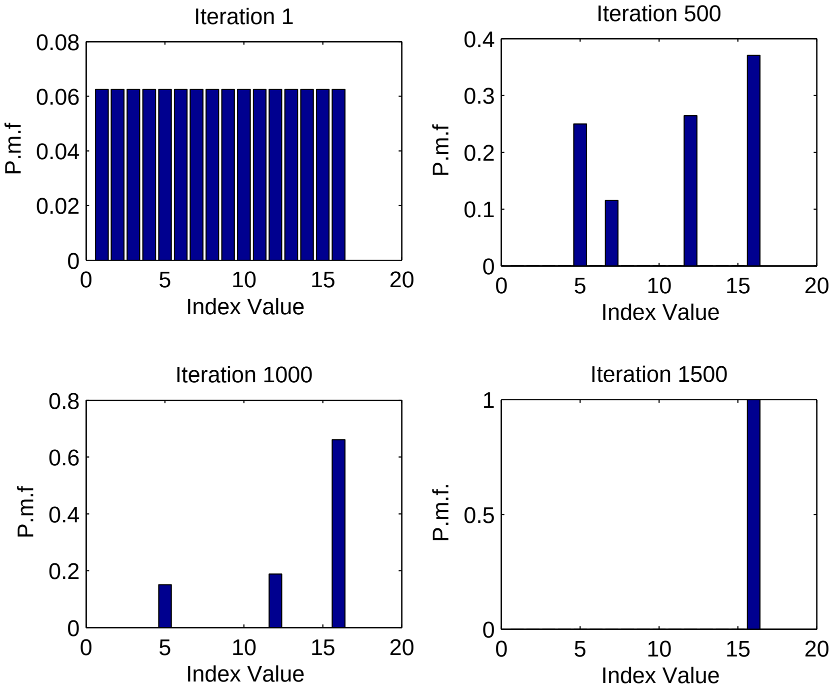


Figure 5.9: The probability distribution for one user when $N = 10$ in RMSG.

compared to small network case for RMSG algorithm.

5.7 Conclusions

In this chapter, we considered cooperative and noncooperative joint power control and beamforming algorithms in multi-user MIMO ad-hoc networks. A cooperative power minimization algorithm (COPMA) with high probability of convergence is investigated for the cooperative scheme. For noncooperative users, we studied an adaptive learning algorithm called RMSG, where the users updated their probabilities of choosing a transmit beamformer and power based on the regrets of not choosing the other strategies. Numerical results corroborated the convergence results of COPMA, and the effectiveness of the COPMA's performance in terms of comparison with RMSG and the centralized case.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we investigated cost minimization problems for wireless networks and studied three main contributions in terms of network energy minimization and network throughput maximization for wireless sensor networks (WSNs) and network power minimization for wireless ad-hoc networks.

In our first contribution, we addressed the problem of energy-efficient transmission structure in WSNs where each source transmits and aggregates the correlated data over intermediate nodes to the sink. We have investigated the impact of efficient data aggregation in establishing routing paths towards the sink for energy minimization problem. For correlation aware routing, we proposed a distributed iterative protocol based on a game theoretic framework, which is shown to converge within a couple of iterations. We have also shown that, by accounting for correlation structure and multi-hop aggregation in constructing routes, significant effective energy gains over classic approaches can be achieved.

In our second contribution, we presented a detailed investigation of efficient throughput maximizing transmission structure in wireless sensor networks where each source transmits and aggregates the correlated data over intermediates source nodes to the sink. We have considered the impact of interference, as well as efficient data aggregation in establishing routing paths towards the sink for throughput maximization problem. For throughput maximizing correlation aware routing, we have proposed a distributed iterative protocol based on a game theoretic framework, which is shown to converge within a couple of iterations. We have also shown that, by accounting for both correlation structure and interference impact in constructing routes, significant throughput gains over classic approaches can be achieved.

In our third contribution, we considered cooperative and noncooperative joint power control and beamforming algorithms in multi-user multiple-input-multiple-output (MIMO) ad-hoc networks using a game-theoretic approach. Under constant signal-to-interference plus noise ratio requirements, the transmit beamforming selection algorithms from a predefined codeword are being studied in the context of total network power minimization. We first considered a cooperative case where all users collaborate with each other in order to minimize the overall power of the network. A cooperative power minimization algorithm, (COPMA) with high probability of convergence is investigated for the cooperative scheme. For noncooperative users, we studied an adaptive learning algorithm called RMSG, where the users updated their probabilities of choosing a transmit beamformer and power based on the regrets of not choosing the other strategies. Numerical results corroborated the convergence results of COPMA, and the effectiveness of the COPMA's performance in terms of comparison with the centralized case or optimal global solution which is found by searching all feasible strategy space.

6.2 Future Work

Although we have investigated the optimization problems with total network energy consumption reduction in Chapter 3 and throughput improvements in Chapter 4 for routing of correlated data in WSNs, the ideas of the proposed game theoretic approach for correlated data aggregation scheme can be modified to favor other routing metrics like end-to-end transmission delay minimization using mutual information accumulation in the network [72] by decreasing the load over overwhelmed bottleneck nodes, total data accuracy improvements, data latency and security problems and capacity maximization schemes. The joint optimization of multiple objectives, such as minimizing delay or latency as well as maximizing network lifetime or throughput can be another application area of the proposed methodology. Moreover, the utility function can be modified to incorporate data aggregation cost into the network performance for complex operations of data aggregation [25]. Instead of using the best response dynamics, some of the distributed learning algorithms (e.g. regret-matching learning algorithm discussed in Chapter 5 or simulated-annealing and genetic algorithms) can also be applied to the energy minimization and throughput maximization problems using the proposed utility functions to improve the efficiency and efficacy of WSNs. Another possible area of study can be to extend the Nash equilibrium tree configuration studied in our analysis and find Pareto optimal tree configuration for the correlated data routing game problem.

The joint beamforming and power adaptation algorithm discussed in Chapter 5 can be extended by simultaneously adjusting the initial selected codebook according to interference in the network. The codebook used in our analysis is designed for single user scenarios. The codebook adaptation can be done by combining the initial selected codeword with different weights and adjusting the weights iteratively depending on the interference each receiver observes in the environment. The evolutionary algorithms, e.g. genetic algorithm [73] can be used to generate better codebook solutions considering the multi-user interference. Other transmitter/receiver adaptation parameters like modulation, frequency, rate, waveform etc. can also be appended into the utility function of the proposed cooperative and noncooperative learning algorithms. Moreover, the convergence proof to pure strategy NE of RMSG algorithm can be studied for general game models.

Bibliography

[1] I. F. Akyildiz, W. Sue, Y. Sankarasubarmaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 50, pp. 102–114, August 2002.

[2] C. Jones, K. Sivalingam, and P. Agrawal, “A survey of energy efficient network protocols for wireless networks,” *ACM Journal of Wireless Networks (WINET)*, vol. 7, pp. 343–358, July 2001.

[3] R. Cristescu, B. B. Lozano, M. Vetterli, and R. Wattenhofer, “Network correlated data gathering with explicit communication: NP completeness and algorithms,” *IEEE/ACM Transactions on Networking*, vol. 14, pp. 41–54, February 2006.

[4] C. Hua and T. P. Yum, “Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 16, pp. 892–903, August 2008.

[5] E. Zeydan, D. Kivanc, C. Comaniciu, and U. Tureli, “Bottleneck throughput maximization for correlated data routing: A game theoretic approach,” in *IEEE 44th Annual Conference on Information Sciences and Systems (CISS’10)*, March, 2010.

[6] Y. Yu, V. K. Prasanna, and B. Krishnamachri, “Energy minimization for real-time data gathering in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 5, pp. 3087–3096, November 2006.

[7] A. Boulis, S. Ganeriwal, and M. B. Srivastana, “Aggregation in sensor networks: An energy-accuracy trade-off,” in *Proceedins of 1st IEEE Int’l. Wksp. Sensor Network Protocols and Applications*, May 2003.

[8] E. J. Duarte-Melo and M. Liu, “Data-gathering wireless sensor networks: organization and capacity,” *Computer Networks*, vol. 43, no. 4, pp. 519–537, 2003.

[9] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “AIDA: Adaptive application-independent data aggregation in wireless sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 2, pp. 426–457, 2004.

[10] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: security protocols for sensor networks,” *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.

[11] B. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks,” in *IEEE 22nd International Conference on Distributed Systems Workshop (ICDCSW’02)*, 2002.

[12] D. Fudenberg and J. Tirole, *Game Theory.* Cambridge, MA: MIT Press, 1991.

[13] H. P. Young, “Strategic learning and its limits,” in *Oxford University Press*, 2005.

[14] J. Marden, *Learning in Large-Scale Games and Cooperative Control.* UCLA, Los Angeles: PhD Thesis, 2007.

[15] S. Hart and A. Mas-Colell, “A simple adaptive procedure leading to correlated equilibrium,” *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[16] D. Monderer and L. Shapley, “Potential games,” *Games and Economic Behavior*, vol. 14, pp. 124–143, May 1996.

[17] R. Rosenthal, “A class of games possessing pure-strategy nash equilibria,” *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973.

[18] D. Bertsekas and R. Gallager, *Data Networks.* Upper Saddle River, NJ: Prentica Hall, 1992.

[19] G. Scutari, D. P. Palomar, and S. Barbarossa, “Competitive design of multiuser MIMO systems based on game theory: A unified view,” *IEEE Journal on Selected Areas of Communications*, vol. 26, pp. 1089–1103, September 2008.

[20] E. Zeydan, D. Kivanc, and C. Comaniciu, “Efficient routing for correlated data in wireless sensor networks,” in *IEEE MILCOM’08*, November, 2008.

[21] S. Banerjee and A. Misra, “Minimum energy paths for reliable communication in multi-hop wireless networks,” in *MobiHoc ’02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 146–156, ACM, 2002.

[22] V. Rodoplu and T. Meng, “Minimum energy mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1443, 1999.

[23] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Energy-efficient broadcast and multicast trees in wireless networks,” *Mob. Netw. Appl.*, vol. 7, no. 6, pp. 481–492, 2002.

[24] A. Scaglione and S. Servetto, “On the interdependence of routing and data compression in multi-hop sensor networks,” in *ACM Wireless Networks*, vol. 11, pp. 149–160, 2005.

[25] H. Luo, Y. Liu, and S. K. Das, “Routing correlated data with fusion cost in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1620–1632, November 2006.

[26] P. von Rickenbach and R. Wattenhofer, “Gathering correlated data in sensor networks,” in *DIALM-POMC ’04: Proceedings of the 2004 joint workshop on Foundations of Mobile Computing.*

[27] S. Pattem, B. Krishnamachari, and R. Govindan, “The impact of spatial correlation on routing with compression in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 1–33, 2008.

[28] H. Mahmood and C. Comaniciu, “Interference aware cooperative routing for wireless ad hoc networks,” *Ad Hoc Networks*, vol. 7, no. 1, pp. 248–263, 2009.

[29] R. Cristescu, B. B. Lozano, and M. Vetterli, “Networked Slepian-Wolf theory: Theory, algorithms and scaling laws,” *IEEE Transactions on Information Theory*, vol. 51, pp. 4057–4073, December 2005.

[30] N. Nie and C. Comaniciu, “Adaptive channel allocation spectrum etiquette for cognitive radio networks,” *Mobile Networks and Applications*, vol. 11, pp. 779–797, December 2006.

[31] S. Kwon and N. B. Shroff, “Energy efficient interference-based routing for multihop wireless networks,” in *IEEE INFOCOM’06*, April, 2006.

[32] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk," *ACM-SIAM Symp. Discrete Algorithms*, pp. 499–507, 2003.

[33] J. Ryu, C. G. Lee, T. T. Kwon, and J. Han, "Combined scheduling and routing for deterministic guarantee of en-to-end deadlines in cell structured sensor networks," *IEEE Sensors Journal*, vol. 9, pp. 1291–1301, October 2009.

[34] R. Madan and S. Lall, "Distibuted algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 2185–2193, August 2006.

[35] I. E. Telatar, "Capacity of multi-antenna Gaussian channel," *European Transactions on Telecommunications*, vol. 10, p. 589595, November/December 1999.

[36] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, pp. 198– 218, February 2004.

[37] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *to appear in IEEE Journal on Selected Areas in Communications*, December 2010.

[38] H. Sampath, P. Stoica, and A. Paulraj, "Generalized linear precoder and decoder design for MIMO channels using the weighted MMSE criterion," *IEEE Transactions on Communications*, vol. 49, pp. 2198–2206, December 2001.

[39] D. P. Palomar, J. M. Cioffi, and M. A. Lagunas, "Joint Tx-Rx beamforming design for multicarrier MIMO channels: a unified framework for convex op-

timization," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2381–2401, September 2003.

[40] D. P. Palomar, J. M. Cioffi, and M. A. Lagunas, "Uniform power allocation in MIMO channels: A game theoretic approach," *IEEE Transactions on Information Theory*, vol. 49, pp. 1707–1727, July 2003.

[41] D. P. Palomar, M. A. Lagunas, and J. M. Cioffi, "Optimum linear joint transmit-receive processing for MIMO channels with QoS constraints," *IEEE Transactions on Signal Processing*, vol. 52, pp. 1179–1197, May 2004.

[42] F. Rashid-Farrokhi, K. J. R. Liu, and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1437–1450, October 1998.

[43] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 18–28, January 2004.

[44] K. Wong, R. D. Murch, and K. B. Letaief, "Performance enhancement of a multiuser MIMO wireless communication system," *IEEE Transactions on Communications*, vol. 50, pp. 1960–1970, December 2002.

[45] F. Farrokhi, L. Tassiulas, and K. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna arrays," *IEEE Transactions on Communications*, vol. 46, pp. 1313–1324, October 1998.

[46] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2839–2848, November 2003.

[47] C. Liang and K. P. Dandekar, “Power management in MIMO ad hoc networks: A game-theoretic approach,” *IEEE Transactions on Wireless Communications*, vol. 6, pp. 1164–1170, April 2007.

[48] G. Arslan, M. F. Demirkol, and Y. Song, “Equilibrium efficiency improvement in MIMO interference systems: a decentralized stream control approach,” *IEEE Transactions on Wireless Communications*, vol. 6, pp. 2984–2993, August 2007.

[49] G. Scutari, D. P. Palomar, and S. Barbarossa, “The MIMO iterative waterfilling algorithm,” *IEEE Transactions on Signal Processing*, vol. 57, pp. 1917–1935, May 2009.

[50] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, “An overview of limited feedback in wireless communication systems,” *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1341–1365, October 2008.

[51] E. Zeydan, D. K. Tureli, and U. Tureli, “Iterative beamforming and power control for MIMO ad hoc networks,” in *Proceedings of IEEE GLOBECOM’10*, December 2010.

[52] D. C. Popescu, D. B. Rawat, O. Popescu, and M. Saquib, “Game-theoretic approach to joint transmitter adaptation and power control in wireless systems,” *IEEE Transactions on Systems , Man and Cybernetics–Part B: Cybernatics*, vol. 40, pp. 675–682, June 2010.

[53] C. Lacatus and D. C. Popescu, “Adaptive interference avoidance for dynamic wireless systems: A game theoretic approach,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, pp. 189–202, June 2007.

[54] D. J. Love and R. W. Heath, “Grassmannian beamforming for multiple-input multiple-output wireless systems,” *IEEE Transactions on Information Theory*, vol. 49, pp. 2735–2747, October 2003.

[55] M. C. Bromberg and B. G. Agee, “Optimization of spatially adaptive reciprocal multipoint communication networks,” *IEEE Transactions on Communications*, vol. 51, pp. 1254–1257, August 2003.

[56] R. Iltis, S. Kim, and D. Hoang, “Noncooperative iterative MMSE beamforming algorithms for ad hoc networks,” *IEEE Transactions on Communications*, vol. 54, pp. 748–759, April 2006.

[57] M. C. Bromberg, “Optimizing MIMO multipoint wireless networks assuming Gaussian other-user interference,” *IEEE Transactions on Information Theory*, vol. 49, pp. 2352–2362, October 2003.

[58] J. Lee and Y. G. Li, “Iterative limited feedback beamforming for MIMO ad-hoc networks,” in *Proceedings of IEEE GLOBECOM’09*, December 2009.

[59] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, “Efficient power control via pricing in wireless data networks,” *IEEE Transactions on Communications*, vol. 50, pp. 291–303, February 2002.

[60] S. Buzzi and H. V. Poor, “Joint receiver and transmitter optimization for energy-efficient CDMA communications,” *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 459–472, April 2008.

[61] G. Scutari, D. P. Palomar, and S. Barbarossa, “Optimal linear precoding strategies for wideband non-cooperative systems based on game theory part I: Nash

equilibria," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1230–1249, March 2008.

[62] G. Scutari, D. P. Palomar, and S. Barbarossa, "Optimal linear precoding strategies for wideband non-cooperative systems based on game theory part II: Algorithms," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1250–1267, March 2008.

[63] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, pp. 529–534, July 1965.

[64] Y. Song, C. Zheng, and Y. Fang, "Joint channel and power allocation in wireless mesh networks: A game theoretical perspective," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1149– 1159, September 2008.

[65] J. Chang, L. Tassiulas, and F. Rashid-Farrokhi, "Joint transmitter receiver diversity for efficient space division multiaccess," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 16–27, January 2002.

[66] G. H. Golub and C. F. V. Loan, "Matrix computations," in *Johns Hopkins University Press*, (Baltimore, MD), 1996.

[67] Z. Han, F. R. Farrokhi, and K. J. R. Liu, "Joint power control and blind beamforming over wireless networks: A cross layer approach," *EURASIP Journal on Applied Signal Processing*, vol. 5, pp. 751–761, 2004.

[68] J. R. Marden, G. Arslan, and J. S. Shamma, "Regret based dynamics: convergence in weakly acyclic games," in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, (New York, NY, USA), pp. 1–8, ACM, 2007.

[69] H. P. Young, “Individual strategy and social structure,” in *Princeton University Press*, (Princeton,NJ), 1998.

[70] P.Milgrom and J. Roberts, “Rationalizability, learning and equilibrium in games with strategic complementarities,” *Econometrica*, pp. 1255–1277, November 1990.

[71] M. Gruteser, A. Jain, J. Deng, F. Zhao, and D. Grunwald, “Exploiting physical layer power control mechanisms in IEEE 802.11b network interfaces,” *Tech. rep. CU-CS-924-01, Dept. of Comp. Sci., Univ. of CO Boulder*, December 2001.

[72] S. C. Draper, L. Liu, A. F. Molisch, and J. Yedidia, “Routing in cooperative wireless networks with mutual-information accumulation,” in *Proceedings of IEEE ICC'08*, May 2008.

[73] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning.* New York: Addison Wesley, 1989.

Vita

Engin Zeydan

Address

401 Monroe Street, Hoboken, NJ 07030

Education

Stevens Institute of Technology, Hoboken, NJ
Doctoral Candidate in Electrical and Computer Engineering
expected date of graduation, December 2010

Middle East Technical University, Ankara, TURKEY
MS in Electrical and Electronics Engineering
August 2006

Middle East Technical University, Ankara, TURKEY
BS in Electrical and Electronics Engineering
June 2004

Professional Experience

Department of ECE, Stevens Institute of Technology, NJ
Teaching Assistant (August 2010 - December 2010)

Lane Department of CSEE, West Virginia University, WV
Graduate Research Assistant (January 2009 - July 2010)

Department of ECE, Stevens Institute of Technology, NJ
Research Assistant (January 2007 - December 2008)

Department of ECE, Stevens Institute of Technology, NJ
Teaching Assistant (August 2006 - December 2006)

Publications

E. Zeydan, D. Kivanc, U. Tureli (2010).
Iterative Beamforming and Power Control for MIMO Ad Hoc Networks.
IEEE GLOBECOM'10, Miami, FL, December 2010

E. Zeydan, D. Kivanc, C. Comaniciu (2010).
Bottleneck Throughput Maximization for Correlated Data Routing:
A Game Theoretic Approach.
IEEE CISS'10, Princeton, NJ, March 2010

E. Zeydan, D. Kivanc, C. Comaniciu, U. Tureli (2008).
Efficient Routing for Correlated Data in Wireless Sensor Networks
IEEE MILCOM'08, San Diego, CA, Nov. 2008

E. Zeydan, D. Kivanc, U. Tureli (2008).
Cross Later Interference Mitigation using a convergent two-stage

game for Ad-Hoc Networks
IEEE CISS'08, Princeton, NJ, March 2008

E. Zeydan, D. Kivanc, U. Tureli (2008).
Unitary and Non-unitary Differential Space Frequency coded OFDM
IEEE WCNC'08, Las Vegas, NV, April 2008

E. Zeydan, D. Kivanc, U. Tureli (2007).
Joint Iterative Channel Allocation and Beamforming Algorithm
for Interference Avoidance in Multiple-Antenna Ad Hoc Networks
IEEE MILCOM'07, Orlando, FL, Oct. 2007

E. Zeydan, U. Tureli (2007).
Differential Space-Frequency Group Codes for MIMO-OFDM
IEEE CISS'07, Baltimore, MD, March 2007

Honors

Best Research Poster Award, Stevens Inst. of Tech., Dec. 2007
Senior Design Project, certificate of achievement,”good use
of engineering”, June 2004