On Wireless Sensor Networks

by

Xiao Liang

B.Eng., University of Electronic Science & Technology of China, 2000

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master Of Applied Science

in the Department of Electrical and Computer Engineering

University of Victoria

On Wireless Sensor Networks

by

Xiao Liang

B.Eng., University of Electronic Science & Technology of China, 2000

Supervisory Committee

Dr. T. Aaron Gulliver, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Wei Li, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Xiaodai Dong, Member

(Department of Electrical and Computer Engineering)

Dr. Jianping Pan, Outside Member

(Department of Computer Science)

Supervisory Committee

Dr. T. Aaron Gulliver, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Wei Li, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Xiaodai Dong, Member

(Department of Electrical and Computer Engineering)

Dr. Jianping Pan, Outside Member

(Department of Computer Science)

Abstract

Wireless sensor networks (WSNs) are a valuable technology to support a huge range of applications spanning from military to commercial, such as battlefield surveillance, habitat monitoring, forest fire detection, disaster salvage, and inventory control management. It has the capability to reveal previously unobservable phenomena in the physical world and significant flexibility of installation and manipulation. The major challenges of WSNs

design come from the requirements of energy constraint, distributed control and scalability.

In this thesis we studied three topics on WSNs. The first one was energy-efficient medium access control (MAC) protocol design. We proposed a MAC protocol which schedules the send and receive times for all nodes within the network. As all nodes only wake up when they send or receive, significant energy is saved by reducing idle listening, collisions and overhearing. And we evaluated the protocol performance by using the GloMoSim simulator [20].

The second topic was about distributed transmission power control of wireless sensor nodes. We proposed a simple scheme which employs request-to-send (RTS) and clear-to-send (CTS) frames to exchange channel gain information, based on which concurrent senders determine their own transmission power without a central controller to coordinate them. Simulation results showed that our scheme can save 30% to 50% of the energy, and also reduce transmission latency.

The third topic was on event detection. Due to the fact that wireless sensor nodes are spatially distributed throughout the area of interest, this application has to been implemented in distributed form. We proposed a scheme which does not need a pre-designated fusion node, and this greatly improves network scalability. We provided close-form expressions to estimate the probabilities of detection failure and false alarm, and validated them by extensive simulation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of Wireless Sensor Networks

Recent advances in the development of integrate circuit (IC) technology have made it possible to fabricate low-cost, small-size sensor devices equipped with a microprocessor and wireless transceiver. These tiny devices, called sensor nodes, along with the capability of computation and wireless communication, invoke a revolutionary way to explore the physical world. When a huge number of sensor nodes are scattered throughout the area of interest, the very close proximity to an object ensures each sensor node gets high accuracy sensing data. In addition, the dispersion of the nodes provides high resolution to analyze phenomena that exhibit spatial and temporal variations at multiple scales [1]. Distributed sensor data flows through wireless links to the gateway of the network, which is called sink, and then the aggregated information is transferred to the user by the sink. Wireless links provide the network with great flexibility of installation and manipulation. The potential to reveal previously unobservable phenomena in the physical world as well as significant flexibility makes wireless sensor networks (WSN) very attractive to a huge range of applications spanning from military to commercial, such as battlefield surveillance, environmental and habitat monitoring, forest fire detection, disaster salvage, and inventory control management [2].

1.2 Challenges and Motivations

The lack of infrastructure makes a WSN much different from most prevalent wireless networks such as cellular systems and wireless LANs. In cellular systems, base stations are intentionally deployed in the geographic area of interests and connected to a backbone wired network. Base stations and the backbone network perform all networking functions. All mobile nodes only communicate with the base station in a single hop. There is no peer-to-peer communications among mobile nodes. Most wireless LANs have this similar, centralized, single hop architecture [3, 4]. In contrast, in most WSN applications, sensor nodes are deployed randomly, e.g., scattered from aircraft or shot by a cannon. There is no means for an operator to install nodes in a defined position or to construct an infrastructure in the area of interest beforehand. Thus, it is impossible for sensor nodes to enjoy the powerful services provided by a backbone network. The sensor nodes handle the necessary control and networking functions among themselves generally through the use of a distributed control algorithm. Nearby nodes communicate in a peer-to-peer manner; distant nodes communicate in a multihop manner, whereby intermediate nodes relay packets from the source node to the destination. Furthermore, the lack of infrastructure forces sensor nodes to have autonomous intelligence to organize themselves, to establish the network without any prior knowledge of the number and location of their peers.

On the other hand, the broadcast nature of the wireless channel fundamentally differentiates WSNs from its wired counterparts, such as metropolitan area networks (MANs), wide area networks (WANs), and the Internet. Particularly, the wireless channel allows a node to transmit a signal directly to any other node, given sufficient transmission power. As a consequence, all other nodes within its radio radius will be interfered with. The link signal-to-interference-plus-noise power ratio (SINR) between two nodes varies randomly due to the variance of transmission distance, propagation environment and interference characteristics. The great complexity and dynamics of wireless links eliminates the possibility to directly apply the research results of wired networks to WSNs.

Multi-hop ad-hoc networks (MANETs) probably are the nearest counterpart to WSNs. However, there still are several major differences between MANETs and WSNs. The most important is that WSNs are embedded in the real world rather than in a virtual data world. Data in a WSN is produced by detecting the world's physical phenomena, such as light intensity, temperature, or object movement. In contrast, data in MANETs is typically originated by human, such as Email or digital pictures. Thus, users of a

WSN are not as aware of the data as in MANETs. Instead, they are normally informed with the highest-level conclusion or results after in-network data processing. Furthermore, it is usually impossible for a user of a WSN to manage and maintain individual sensor nodes. The close dependance on the physical environment and the requirement of in-network data processing creates in the WSN unique data traffic and communication patterns.

In most WSN applications, particularly for environment monitoring and object detection, the data traffic within the network is bursty and event-driven. Sensor nodes are usually idle for long periods of time and suddenly burst into activity for a brief duration when events occur. Thus, a WSN is often assumed to have a low duty cycle, i.e, just for one percent or less of its lifetime it is working and all other times it is idle [2]. Besides being temporally bursty, sensor data is spatially correlated. Nodes in geographical proximity normally detect the same event, thus there is much redundancy among their data. Therefore, reducing the redundancy and distilling the high-level information is an essential task for in-network data processing.

The number of nodes within a single sensor network is generally huge, from hundreds to thousands. This huge number facilitates the sensing resolution and robustness of the network. However, it also brings the challenge of scalability to protocol design, i.e., the protocol should maintain its performance no matter how big the network is. The key to scalability lies in the use of distributed algorithm. Generally, distributed algorithm means an algorithm makes a decision at each step or part without complete knowledge of the whole object, but at the end a global result can be reached. By using decentralized information and resource control, protocols can scale as the network grows since they only rely on local information. The conflict between obtaining a global optimal result and local information and control is the essential challenge for distributed algorithm design.

Without an infrastructure, most sensor nodes are powered by batteries with finite energy. Furthermore, due to the lack of user maintenance, it is costly to recharge the battery for individual nodes. Therefore, energy-conservation becomes the most important design objective for WSNs. A WSN is well known as an energy-constrained network. Besides the energy constraint, other secondary but important constraints come from the restriction of bandwidth, the moderate computation capability and the requirement of low price.

More challenges come from the dynamics in the networks. For example, a node failure due to power depletion or physical damage will result in a change in the network topology. As it is impossible to recover the network structure by manually fixing the failed nodes, the network should be able to automatically adjust itself, changing the network connectivity in order to resume proper operation. Therefore, the ability to adapt to network dynamics is an indispensable feature of sensor networks, and also a formidable challenge to designers.

In summary, WSNs are a special subset of MANETs, with the unique features of being energy-constrained and embedded in a physical world. The major challenges come from the requirements of energy constraint, scalability and adaptivity. Innovative ideas are needed to design intelligent algorithms and protocols, which are energy-efficient with distributed control and adaptivity at each layer to compensate for and exploit network dynamics.

1.3 Scope of the Thesis

The rest of this thesis is organized as follow. In Chapter 2, we investigate energy-efficient MAC protocol design, and propose a method which exploits the unique tree structure communication pattern in WSNs to schedule the sending and receiving times for all nodes within the network and thus saves significant energy by reducing idle listening, collisions and overhearing. In Chapter 3, we propose a distributed approach for concurrent transmission nodes to adjust their transmission power individually, and consequently save energy as well as decrease transmission latency. In Chapter 4, we propose a scalable and energy-saving scheme for WSN detection applications, and provide close-form expressions to estimate its performance. The analysis is validated by extensive simulations.

Chapters 2 to 4 consider different aspects of on WSNs. However, all these topics have the same design objective: saving energy. Furthermore, the research results can be integrated together. For example, the transmission power control in Chapter 3 can be used in Chapter 2 to improve the performance of the MAC protocol by reducing the group interference.

Finally Chapter 5 summarizes the results presented in this thesis and suggests some related topics for future work.

Chapter 2

An Energy-Efficient MAC Protocol Exploiting The Tree Structure In Wireless Sensor Networks

2.1 Introduction

Data aggregation is a necessary technique in most WSN applications for two main reasons: 1) A user typically does not or cannot process the raw sensor data, instead, he would like to obtain higher level results distilled from this raw data. Thus, a WSN must have an in-network processing capability to aggregate raw data into high-level information [1]. 2) Data aggregation dramatically reduces the data that must be transferred, and consequently conserves the network's most precious resource, energy [7]. In a large scale network, communication between source nodes and the sink is usually via multiple-hops, so data aggregation along with multiple-hopping forms a unique communication paradigm, a tree structure. In this case, the sink is the tree root, source nodes are the leaves and intermediate nodes that conduct local data aggregation are the branch joints. Fig. 2.1 shows an example of this tree structure. The tree structure communication paradigm has a significant impact on the MAC protocol design, since it entails coordination of node traffic in the tree. For example, an intermediate node which conducts data aggregation has to collect all data from its children before sending it out. However, this feature can be exploited to save energy in the MAC layer by reducing idle listening. Thus, we propose a novel MAC protocol named EMAC, which explicitly exploits the tree structure to schedule the send and receive times for each node. Thus, during data retrieval from source nodes to the sink, every node wakes up only when it has data to send or receive, and consequently conserves energy by eliminating idle listening, collisions and overhearing.

2.2 Related Work

An algorithm was proposed in [9] which jointly constructs the tree and assigns a schedule to each node. It assumes that the sink knows all node positions and computes schedules for all nodes, which is impractical in large scale networks as the number of nodes is huge, because it will be a formidable challenge to the sink's memory and computation capability. In addition, the method by which the sink collects position information and assigns the schedules to all nodes remains unexplained. The chain feature of the tree

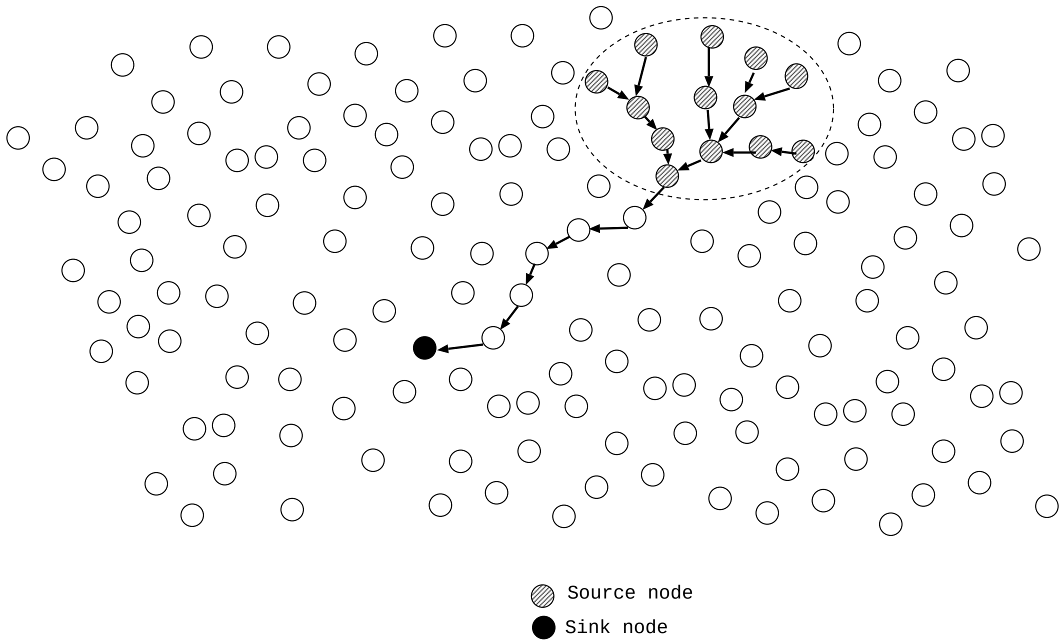

Figure 2.1: An example of the sensor network tree structure.

was noted in [11], where it was assumed that all nodes know their offset from the sink. However, a strategy for each node to determine its offset was not given. Moreover, since nodes that have the same offset are likely to send at the same time, contention among them becomes an issue. A more important problem is data aggregation. In order to conduct data aggregation, a parent node would wait to gather packets from all its children before it sends out its own packet. However, in [11], a parent node immediately relays its child data, and so doesn't wait until the data is gathered from its all children, which is inefficient for data aggregation.

In this thesis, we provide an explicit explanation of the entire procedure from gathering information to assigning a schedule to each node. The schedule algorithm computation is distributed on multiple nodes rather than being done only by the sink, so the computational load for each node is much lower and practical.

2.3 System Model and Problem Description

When queried by the sink or when an event is detected, a number of nodes in a local geographical region will generate and attempt to deliver their data. The data flow from multiple source nodes towards the sink, including aggregation on intermediate nodes, is illustrated in Fig. 2.2. The shape of the tree is determined by the positions and number of source nodes, and the routing algorithm by which each node determines its next hop. We partition all the nodes into groups according to their roles. The nodes that share the same next hop or parent node are siblings. Siblings and their parent form a group, in which the parent node acts as the data aggregation agent as well as intermediate relay. These groups are quite different from the clusters formed in hierarchical protocols such as LEACH [12]. The groups here have a chain relationship. The parent node in a lower group will be a child in its upper group. Every group interacts with its upper and lower groups. An upper group includes the group heads of its lower groups as children. Thus, an upper group is also called a parent group, and a lower group is called a child group. Two groups are called sibling groups if their group heads are siblings. For example in Fig. 2.2, there are 7 groups. Group 1 has group 0 as its parent group, and has groups 3, 4 and 5 as its child groups, and has group 2 as its sibling group. The tree can also be divided into tiers according to the number of hops to the sink. In Fig. 2.2, there are three tiers. Group 0 is in tier 0, groups 1 and 2 are in tier 1, and groups 3, 4, 5 and 6 are in tier 2.

Within a group, the parent node can act as a coordinator to schedule the send and receive times for its children. A parent node is also a child node in its parent group, as it acquires its schedule from its own parent. Based on the chain relationship, starting from the sink, each node obtains a schedule from its parent and then assigns a schedule to its children until every node in the tree has a schedule. For example, in Fig. 2.2, node 0 assigns the send times for nodes 1 and 2. Then node 0 goes to sleep until its receive time. Nodes 1 and 2 use their send times to arrange the send times for their children. As nodes 1 and 2 know the transmission times in group 0, they can schedule

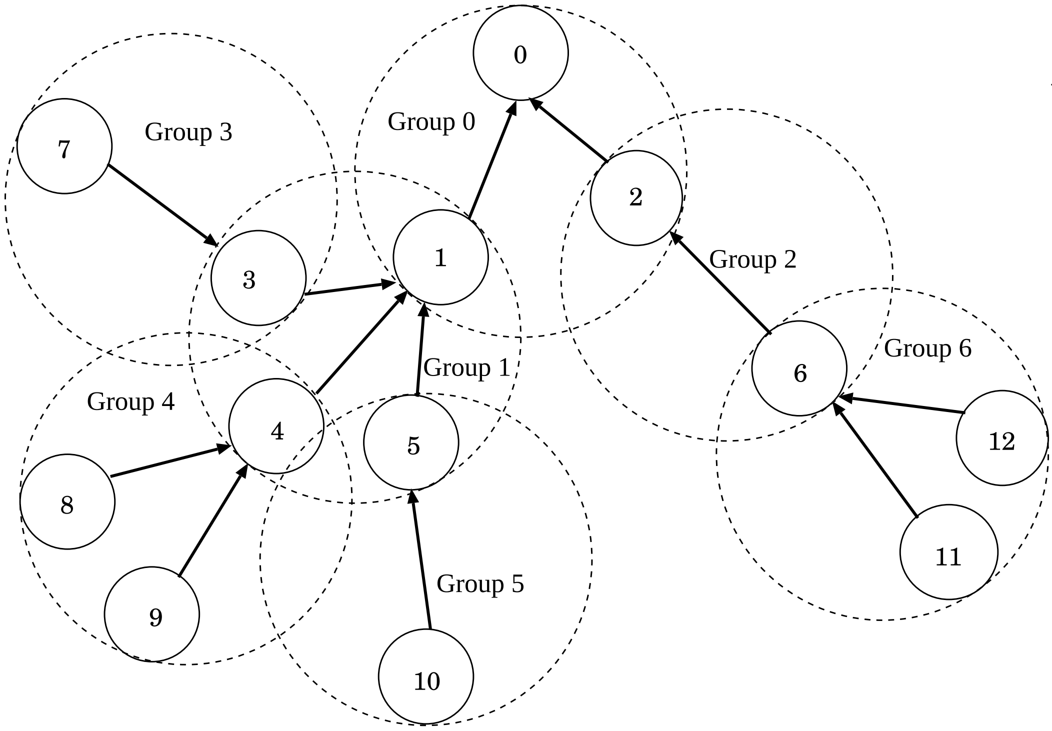

Figure 2.2: Groups within a tree structure.

transmission times within their own groups so as to not collide with group 0. Of course, inter-group interference can be eliminated by means of multiple access modulation such as CDMA or FDMA, as used in LEACH, or as in [9]. However the use of CDMA or FDMA greatly increases the complexity of the transceiver hardware, which may lead to much more energy consumption and counteract the energy saved in the MAC layer. Therefore, we prefer to avoid inter-group interference by scheduling group transmissions at different times. Due to the limited transmission power of each node, the interference range is limited. Moreover, if power control is applied, the interference range will be better controlled, which normally is shorter than a two-hop distance. Thus, we assume that the data transmissions of nonadjacent groups will not interfere with each other. For example, in Fig. 2.2, groups 3 and 5 are not adjacent sibling groups, so the data transmitted from node 10 to node 5 may interfere with nodes in group 4 but not with node 3, which is the parent node of group 3. This assumption is justified since the nonadjacent siblings of a parent are very likely to be beyond the two hop range of a node, and thus are beyond the radio interference range. Furthermore, if the transmission power control techniques, such as proposed in Chapter 3, are employed, the assumption of no interference among nonadjacent groups can be more likely achieved.

In some cases [10], the route from a source node to the sink needs to change from time to time in order to balance the energy consumption of intermediate nodes. However, the frequency of route changes is much less than the data transmission rate. Thus, it is reasonable to assume that the tree structure is static long enough for a number of packets to completely flow from the source nodes to the sink.

Source nodes usually deliver their data packets at a rate which is predefined by the application [13]. The delivery rate is related to the "sampling rate" of the physical phenomena. The higher the rate, the higher the temporal resolution obtained of the physical phenomena. Although the delivery rate is the same for all source nodes, the packet lengths for the nodes may be different. Regardless of the variation, there exists an upper bound on the packet length. In the tree structure, data flowing from multiple sources accumulates at intermediate nodes. The function of data aggregation is to compress this data so that the output of an intermediate node is not excessive. We assume the maximum packet length is known *a priori*, and denote the transmission duration for the maximum length packet as T_U. Since the packet length for each node is unpredictable, we assign T_U for every transmission to ensure there are no overlaps with neighbor nodes.

In summary, the constraints and assumptions for our protocol are as follows:

1. adjacent sibling groups may interfere with each other, others will not;
2. the tree structure will not change during the data retrieval period;
3. data packets have a predefined maximum length, and nodes send their packets at the same predefined rate;

4. the delay from source nodes to the sink is always within the application requirement, and this is estimated later in this chapter.

2.4 EMAC Protocol Design

EMAC consists of four phases: *register*, *load report, schedule assignment*, and *data transmission*. The register phase starts at all nodes that know their next hop or parent. By registering, each node will find out how many neighbor nodes are its children. Then in the load report phase, every node tell its parent how many children it has. Since the maximum packet length is predefined, the number of children is equivalent to the maximum time a node will spend on receiving data from its children. In the schedule assignment phase, each node gets a schedule from its parent. Based on this schedule and the transmission load of its children, a node determines and broadcasts the schedule for its children, and then goes to sleep. In the data transmission phase, each node wakes up to send or receive according to its schedule.

2.4.1 Register Phase

When the parent is known, a node sends a short frame named *Register Request* to its parent. If the requester receives an ACK from its parent, it will not send the request again. If it does not receive an ACK after a predefined duration, it sends the request again. When a node receives a request, it will check whether it has received it from the same node before. If not, it will consider it as a new child and assign a unique serial number, which starts from zero, to this child in the ACK frame reply. If yes, the parent will reuse

Fame Type	Serial Number	Dst Addr	Scr Addr	Child Number	Position

(a) Frame format for Register Request or Load report

Fame Type	Serial Number	Dst Addr

(b) Frame format for ACk

Fame Type	Src Addr	Tx Begin Time	First Rx Time	Second Rx Time	Bitmap

(c) Frame format for Schedule Assignment

Figure 2.3: The control frame formats.

the serial number it has previously assigned to that node in the ACK frame. The request and ACK frame formats are shown in Fig. 2.3. In this phase, carrier sense multiple access with collision avoidance (CSMA/CA) and binary slotted exponential back-off are used to avoid collision. As each node only has one frame to send, the collision probability will decrease dramatically as nodes successfully send their requests. Thus there must be a time when all nodes have registered with their parent. By the end of the register request phase, each node knows how many children it has by the number of requests it has received. If a node has no children, it is a pure source node, and will not act as an intermediate node.

2.4.2 Load Report Phase

The load report phase starts from pure source nodes sending their child number and location information to their parents. When an intermediate node receives a load report from a child, it sends an ACK immediately. Similar to the register phase, the parent node will check whether the report has been received previously. The load report and responding ACK frame formats are the same as those used in the register phase, as shown in Fig. 2.3.

An intermediate node collects all its children's reports before it sends its own child number and location information to its parent. In this way, when the sink collect all load reports from its children, it knows that the load report phase is finished. Then the schedule assignment phase can begin.

2.4.3 Schedule Assignment Phase

In the schedule assignment phase, a parent node tells its children when they should start receiving and sending. The arrangement of receive times for the nodes is based on the tier architecture. To eliminate interference between two successive tiers, and to support data aggregation, a lower tier always finishes transmission before the upper tier. By the end of the load report phase, a node knows its children. The receiving duration of a node is determined by multiplying T_U by its child number. If a node has no children, it will not receive data, and thus does not need a receive time scheduled by its parent. Otherwise, a node's receive times are determined by its parent. If two groups of children in the same tier do not interfere, they can receive in the same time duration. This parallel receiving reduces delay without increasing inter-group interference. A node uses its children's position information to tell whether two child groups interfere with each other, and arranges non-interfering child groups in the same duration. Thus all potentially interfering groups are assigned different durations. As we assume inter-group interference only occurs between adjacent sibling groups, all nonadjacent groups can share time durations. Therefore, two sets of non-overlapping time durations are enough to avoid inter-group interference in a tier. To explain this mechanism clearly, consider the example shown in Fig. 2.2. Node 0 tells node 1 and node 2 their receive times. As groups 1 and 2 can interfere with each other, their receiving durations cannot overlap. However, in the next tier, groups 4 and 6 can share the same receive times, while groups 3 and 5 share other receive times.

The schedule is broadcast by parents to their children tier by tier in the format shown in Fig. 2.3. Children nodes look up the bit value in the "Bitmap" field corresponding to their serial numbers. If the value is '0', the child node choose "First Rx Time" as its receive time; otherwise choose "Second RX Time" as its receive time. In the schedule frame, "Tx Begin Time" indicates the parent's receive time, based on which the children determine their individual send times, since a node's receive time is actually its children's send time. To avoid send collisions among multiple children to a same parent, child nodes send their data sequentially according to their serial numbers. As in our scheme the sending duration for a node is fixed at T_U, a child node determines its send time as its serial number times of T_U later than its parent's receive time.

Since all children are likely to get the schedule at the same time, collisions will occur if they try to broadcast their schedules immediately after this reception. Therefore, we require each node to wait for a specified time according to its serial number before broadcast. Define the time duration for broadcasting a schedule frame as T_{sf}. Node 0 can broadcast its schedule immediately after reception of the parent schedule, while node 1 has to wait for T_{sf}. The uniqueness of the serial numbers for each node ensures schedule broadcasting collisions are avoided.

2.4.4 Data Transmission Phase

A node only wakes at its scheduled time, at all other times it is sleeping. The schedules define the relative temporal transmission sequence for all nodes in the tree. Since all nodes generate data at a predefined rate, they can periodically send and receive in the sequence defined by schedules until the tree structure changes. When the send time arrives, a node wakes up to send its packet in the following T_U seconds. When the receive time arrives, a node wakes up to receive data from its children. When reception from all children is completed, the node goes to sleep until its send time arrives. Since the transmission time for a node may be much shorter than T_U, the receiver can turn off its radio immediately after the transmission is done. The radio is turned on again before the next child begins transmitting. This process is done as long as the energy cost for radio transition does not exceed the energy saved by reducing idle listening.

2.5 Delay Estimation

Our protocol allows for data aggregation as packets flow from the pure source nodes to the sink tier by tier. The maximum number of tiers in a tree is the maximum number of hops from a source node to the sink. In other words, if the farthest node in the tree is H hops from the sink, then the tree has H tiers. A tier consists of a number of groups. Denote the number of children inside a group as n_c. Then the group latency, L_g, can be estimated as $L_g = n_c \times T_U$. Since we assume that nonadjacent groups will not interfere, twice the maximum value of L_g is sufficient to prevent inter-group interference. Thus, the maximum latency within a tier is $max(L_t) = 2max(L_g) = 2 \times max(n_c) \times T_U$ and the maximum latency from source nodes to the sink is $max(L) = 2 \times T_U \times (\sum_{i=1}^{H} max(n_c)_i)$ This includes transmission time, aggregation delay, and the delay to avoid inter-group interference.

2.6 Performance Results

The proposed MAC protocol was implemented using GloMoSim [20]. As shown in Fig. 2.2, we randomly deploy 13 nodes in a tree structure with a only limitation of the node deployment that nonadjacent groups do not interfere each other. Additionally, to capture the impact of data aggregation, an intermediate node does not relay immediately when it receives a packet

Table 2.1: Radio Parameters

Radio bandwidth	2M bps
Radio Transmission Range	100 m
Radio Interference Range	200 m
Packet Length	512 bytes
Transmit Power	18mW
Receive or Idle Power	9mW
Sleep Power	0.5mW

from its children, instead, it sends out a packet of 512 bytes after collecting packets from all its children. We choose 3 metrics to evaluate performance: Energy consumption is the total energy cost to deliver 20 packets from every pure source node; Latency is the average delay of all packets received by the sink from pure source nodes; The delivery ratio is the ratio of the packets actually received by the sink to the number it is supposed to receive. The radio parameters are listed in Table 2.1.

IEEE 802.11 is one of the most dominant MAC protocols used in current wireless networks. However, the power control function of IEEE 802.11 cannot satisfy the requirements of WSN applications. SMAC [6] is a modification of IEEE 802.11 protocol specifically designed for WSNs. It lets nodes sleep for a period of time to avoid unnecessary energy waste. We implemented a simplified SMAC and 802.11 protocols as benchmarks for comparison. The SMAC protocol has a fixed schedule of listening for 0.3s and sleeping for 0.7s periodically. The performance of the three protocols is compared. The average of 5 simulations is plotted in Figs. 2.4, 2.5 and 2.6.

The simulation results show that EMAC outperforms SMAC and 802.11

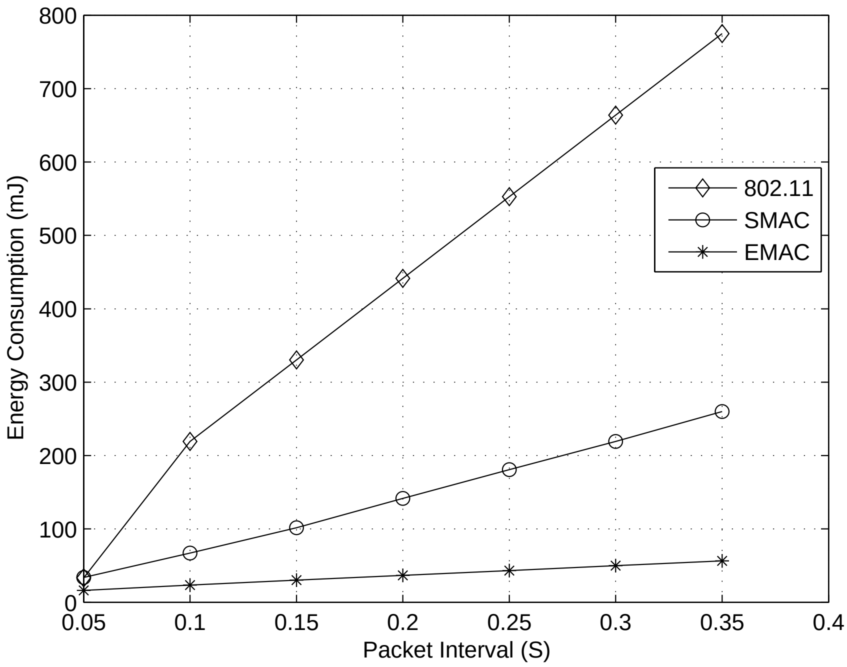

Figure 2.4: Total energy consumption on radios in all nodes.

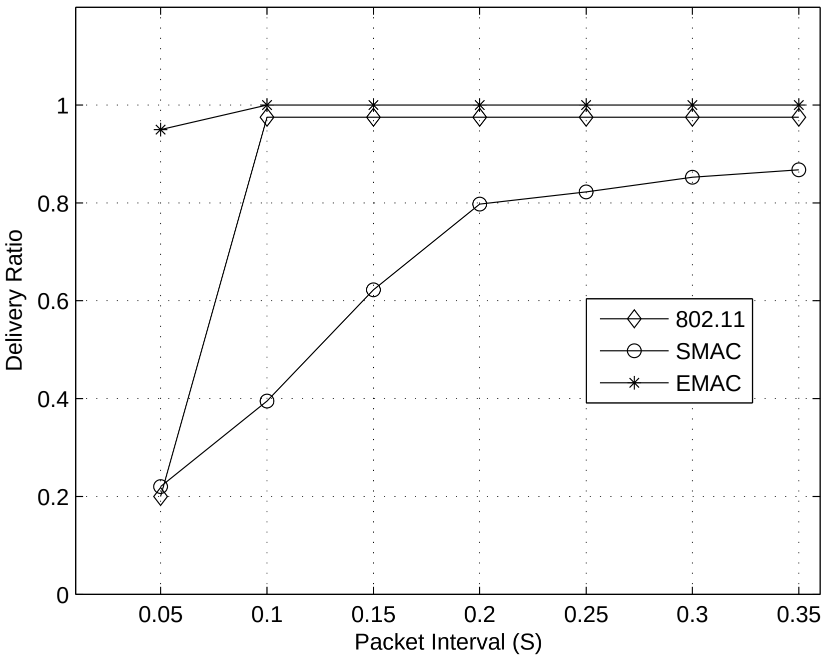

Figure 2.5: Delivery ratio of the packets from pure source nodes to the sink.

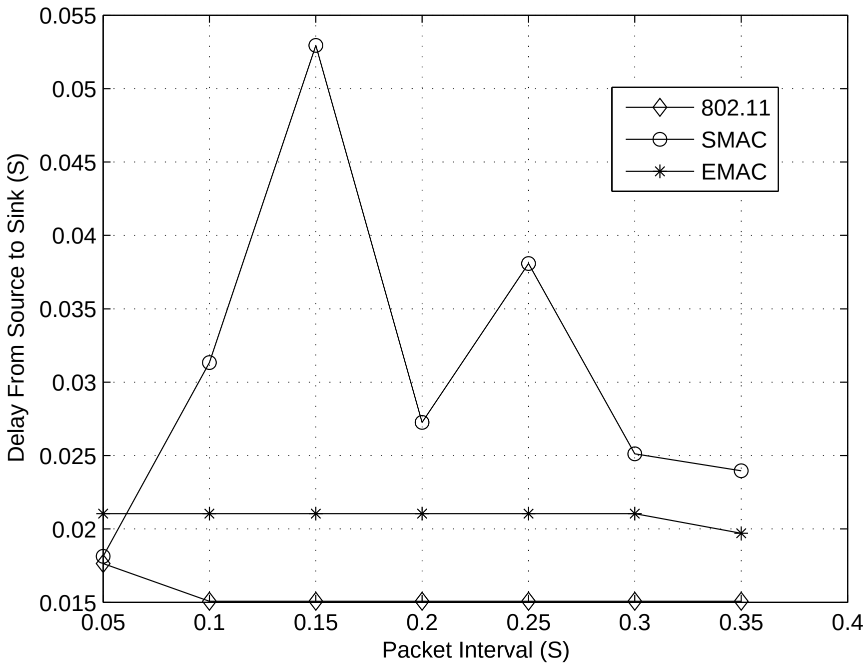

Figure 2.6: Average packet latency from pure source nodes to the sink.

both in delivery ratio and energy efficiency. Packet loss is mainly due to collisions. In 802.11 and SMAC, when a packet fails to be received after 3 transmissions, it is dropped. In our simulation deployment, since a node not only interferes with its siblings but also the nodes in adjacent groups, thus, many collisions occur when nearby nodes attempt to send at the same time. This situation is exacerbated when the packet interval is small. The periodic sleep with SMAC actually reduces the available time of a parent node, thus competition among child nodes to transmit data is more severe. Therefore, the delivery ratio of SMAC is much lower than the other two. In contrast, since EMAC arranges access time for all the nodes to avoid collision, the delivery ratio is always near 100%. However, this advantage comes at a cost of increased latency. Therefore 802.11 latency is better than that with EMAC, but EMAC latency is fairly constant as it is only related to the tree structure. However, the latency of SMAC is very dynamic and random, and so is hard to estimate. Therefore, the variances for different packet intervals are listed in Table 2.2. EMAC diminishes idle listening as much as possible, so energy consumption is always the least among the three protocols.

2.7 Summary

A unidirectional tree structure is the major communication paradigm in WSNs when data is transferred from source nodes to the sink. To save energy, the design of the MAC protocol should fully exploit the characteristics of this communication paradigm. Thus we propose the EMAC protocol,

Table 2.2: SMAC Latency Variance

Packet Interval (S)	Minimum Latency (S)	Maximum Latency (S)
0.05	0.0109	0.0258
0.1	0.0228	0.0418
0.15	0.0300	0.124
0.2	0.0197	0.0323
0.25	0.0269	0.0582
0.3	0.0169	0.0343
0.35	0.0159	0.0401

which arranges send and receive times for each node by exploiting the tree structure. Since all nodes only wake up when they send or receive, significant energy is saved by reducing idle listening, collision and overhearing.

Since EMAC is designed for data retrieval, it may not be suitable for other communication paradigms. Thus, a node may adopt a general-purpose MAC protocol, such as 802.11 or SMAC, when it is not in data retrieval phase.

Chapter 3

Distributed Transmission Power Control for Wireless Sensor Networks

3.1 Introduction

In wireless networks, transmission power control (TPC) is used to limit the transmission power so that the required signal-to-interference-plus-noise power ratio (SINR) is satisfied. Compared to conventional fixed power transmission, TPC not only saves energy, but also decreases the transmission latency. For example, as shown in Fig. 3.1, node A sends to node C, and node B sends to node D. If A and B use their maximum power, they will interfere with the other receivers, i.e., A interferes with D and B interferes with C, which could result in transmission failure on both links. However, if they can reduce their power to a suitable level, they not only save transmission power but also limit the interference. This increases the probability of successful concurrent transmission, and consequently decreases the transmission latency, since node B need not postpone its transmission after A finishes sending.

Both energy savings and latency reduction are desirable goals in the design of wireless sensor networks (WSNs). Due to the inherent feature of spatial distribution of sensor nodes, TPC in a WSN has to be distributed, in the sense that each sender determines its own transmission power rather than a central controller to coordinate multiple sender-receiver pairs. Distributed TPC research started a decade ago for cochannel interference control in cellular radio systems [14]. Those algorithms satisfied the distributed requirement by an approach that each sender iteratively exchanges information with its corresponding receiver while gradually adjusting its power to a proper level. Hopefully multiple sender-receiver pairs converge to an equilibrium. However, this approach is unsuitable for WSNs, since the extra communication overheads will cost far more energy than that saved by TPC. Recently, distributed TPC has also been investigated for ad hoc networks [15, 16]. However, the algorithms for ad hoc networks cannot properly apply to WSNs, since nodes in WSNs cannot always be awake as is the case with their counterparts in ad hoc networks. Furthermore, other WSN communication features, such as sleeping and network synchronization may facilitate the distributed TPC design. Thus, we propose in this chapter a new distributed TPC scheme dedicated to WSNs, which exploits the medium access control (MAC) frames to gather the path gains. The transmission powers are determined in one step, which saves energy as well as decreases transmission latency.

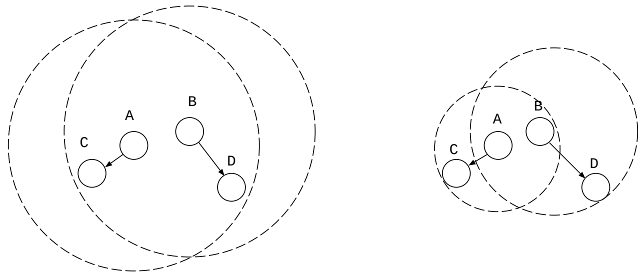

Figure 3.1: An example of concurrent transmission with TPC.

3.2 Problem Formulation

Suppose that in a vicinity there are N interfering transmitter-receiver pairs. We denote the SINR of the ith receiver as

$$R_i = \frac{G_{ii}P_i}{\eta_i + \sum_{i \neq j} G_{ji}Pj} \tag{3.1}$$

where $G_{ji} > 0$ is the channel gain from the jth transmitter to the ith receiver, P_i is the transmission power of the ith sender, and η_i is the thermal noise power at the ith receiver. There is a minimum SINR requirement $\gamma_i > 0$ such that when $R_i \geq \gamma$ the receiver has acceptable signal quality. Thus, for N pairs to simultaneously communicate successfully, the following constraint must be met

$$(\mathbf{I} - \mathbf{F})\mathbf{P} \succeq \mathbf{u} \quad \text{with} \quad \mathbf{P}_{max} \succeq \mathbf{P} \succ 0 \tag{3.2}$$

where $\mathbf{P} = (P_1, P_2, \ldots, P_n)^T$ is the vector of transmission powers for the N transmitters, and

$$\mathbf{u} = (\frac{\gamma_1\eta_1}{G_{11}}, \frac{\gamma_2\eta_2}{G_{22}}, \ldots, \frac{\gamma_N\eta_N}{G_{NN}})^T \tag{3.3}$$

is the vector of noise powers scaled by the SINR thresholds and the channel gains. $\mathbf{F}$ is a matrix with

$$F_{ji} = \begin{cases} 0 & \text{if } i = j \\ \frac{\gamma_i G_{ji}}{G_{ii}} & \text{if } i \neq j \end{cases} \tag{3.4}$$

for $i, j \in 1, 2, \ldots, N$.

The components of $\mathbf{P} = (\mathbf{I} - \mathbf{F})^{-1}\mathbf{u}$ are the minimum feasible transmission powers to ensure successful transmission on all links concurrently, if $\mathbf{P}_{max} \succeq \mathbf{P} \succ 0$. Since G_{ji} denotes the channel gain which is non-negative, and we do not have multiple disjoint networks, the matrix $\mathbf{F}$ has non-negative elements and is irreducible. According to the Perron-Frobenius theorem and standard matrix theory [17], if the Perron-Frobenius eigenvalue $\lambda_F < 1$, there exists a vector $\mathbf{P} \succ 0$ such that $(\mathbf{I} - \mathbf{F})\mathbf{P} \succeq \mathbf{u}$. However, computing the Perron-Frobenius eigenvalue is too complex for a sensor node. Thus, we restrict and simplify the conditions for solving (3.2) as if for all $j \neq i$ $F_{ji} < 1$, then $\mathbf{P} \succ 0$ exists.

Suppose η and γ are the same for all nodes. To determine its transmit power, each sender has to collect all the channel gains, G_{ji}, and construct and check **F** and **u**. If a solution for (3.2) cannot be obtained, all senders modify their matrices until a solution is obtained. By solving (3.2), each sender determines whether it will send or not, and how much power it should use if it sends. Given that **F** and **u** constructed by all senders are the same, all senders determine their own transmission powers based on the same matrices. Thus, the transmission equilibrium will be reached without the exchange of additional information.

In the next section, we propose a MAC protocol named TPMAC which integrates a scheme allowing senders to collect the channel gains and determine their own transmission powers individually.

3.3 The Proposed TPMAC Protocol

3.3.1 Assumptions

We define the duration for transmitting a data packet as a period. A node which has packets to send in a period is called a potential sender. The corresponding receiver for a potential sender is called the intended receiver. We assume that during a period the wireless channel gains do not change, and nodes in a vicinity are synchronized.

In a WSN, nodes in proximity detect an event at the same time, generate source data and try to send this to their corresponding relay nodes. Thus, a vicinity in which there are multiple interfering sender-receiver pairs is formed. Relay nodes obtain data, and process and forward it to the next relay nodes, forming a new vicinity. Geometrically, the new vicinity is adjacent to the first one. However, due to the requirement of in-network data processing [7], transmission in the new vicinity usually occurs after the transmission has finished in the previous one. Thus, although there may exist a number of vicinities along a route from source to sink, it is unlikely that interference will occur between adjacent vicinities. Thus, we assume that within a vicinity a potential sender can receive a signal successfully from all intended receivers, and there is no interference from a node outside the vicinity.

3.3.2 Protocol Overview

As shown in Fig. 3.2, a period consists of four phases: RTS contention, CTS reply, concurrent data transmission, and ACK reply. In the first phase, all potential senders compete to send their RTS signals with maximum power P_{max}. A node calculates the channel gain from a potential sender by comparing the received RTS signal power with P_{max}. In contrast to 802.11 and SMAC [6], when an intended receiver gets an RTS signal, it does not reply with a CTS immediately. Rather it continues listening and receiving RTS signals from other potential senders. Thus, by the end of the RTS contention phase, a receiver collects the channel gains from most potential senders. Then, in the next CTS reply phase, all intended receivers sequentially broadcast their recorded channel gains using CTS signals with power P_{max}. By collecting the CTS signals, all potential senders construct the channel gain matrix $\mathbf{F}$ and $\mathbf{u}$. In the data transmission phase, a potential sender determines whether it will send or not, and if so its transmission power. Then in the ACK phase,

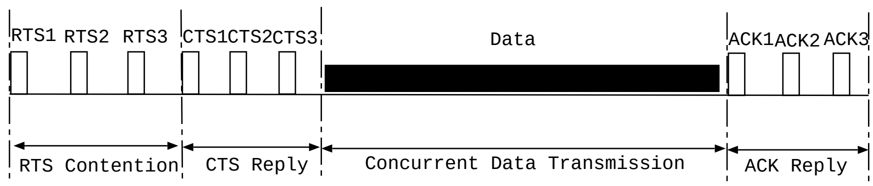

Figure 3.2: An example of three concurrent transmission pairs during a period.

receivers sequentially reply with an ACK to their corresponding senders in order to avoid collisions.

3.3.3 Protocol Details

RTS Contention

At the beginning of a period, a potential sender randomly sets its own back-off timer. If no other user sends before the timer expires, the potential sender sends out its RTS signal using power P_{max}. Otherwise a potential sender detects an RTS signal. In this case, it pauses its timer, and records the number of detected RTS signals before sending its own RTS signal. This number is added in the RTS, and will be used as a serial number to control the sequence of CTS replies.

When a node receives an RTS signal, it checks the source and destination addresses in the RTS frame. If a node has data to send, under the following three conditions, it will abandon sending in this period: 1) the source node is its intended receiver; 2) its intended destination is the same as the destination in the RTS frame; 3) the RTS frame is for itself. It will also check whether it has already received RTS frames including the same destination address.

If it has, the node ignores the current RTS frame, because this situation indicates that more than one sender has sent an RTS signal to the same intended receiver. If it has not received RTS frames including the same destination address, it calculates and records the channel gain based on the RTS signal.

The remaining potential senders resume their back-off timers and compete to send out their RTS signals when the channel returns to idle. After sending its RTS signal, a potential sender goes to sleep until the CTS reply duration begins. Here, we utilize a CSMA/CA-like scheme with random back-off to avoid collisions among RTS signals.

CTS Reply

If a node receives an RTS signal destined for itself, it knows that it has been chosen as an intended receiver. This RTS is called a primary RTS to the receiver. All other nodes in the vicinity which are neither potential senders nor intended receivers then go to sleep until the next period begins. Intended receivers include their recorded channel gains in CTS frames, and send them using power P_{max}. The send time of an intended receiver is determined according to the serial number, which begins at zero, obtained from its primary RTS signal. For example, if a potential sender is the third who sent RTS in the RTS contention duration, it will put serial number 2 in the RTS frame to its intended receiver. After obtaining the serial number from its primary RTS, the intended receiver knows it is the third one to send CTS in the CTS reply duration. In this way, all intended receivers sequentially broadcast their CTS signals, and collisions are avoided. After the CTS signal is sent, an intended receiver goes to sleep until data transmission begins.

All CTS signals are collected by the senders to construct the channel gain matrix. A sender records the number of received CTS signals before its primary CTS was received. It uses this number to determine its place in the N transmitter-receiver pairs. For instance, if a sender records i CTS signals before obtaining its primary CTS, it is the $(i+1)$th sender in the N pairs.

Data Transmission

Suppose a potential sender receives m CTS signals. Based on these signals, it constructs an m by m matrix $\mathbf{F}$ and a vector $\mathbf{u}$ of length m. According to the assumptions in Section 3.3.1, all potential senders receive the same number of CTS signals and construct the same $\mathbf{F}$ and $\mathbf{u}$. The matrix check starts from the upper left 2×2 submatrix. If there is no entry greater than or equal to 1 in an $n \times n$ submatrix, the check is extended to the $n+1 \times n+1$ submatrix. Otherwise, the entries in the nth column and nth row are deleted, so $\mathbf{F}$ and $\mathbf{u}$ size are reduced. The corresponding nth sender can no longer transmit in this period and goes to sleep until the next period. The remaining senders resume checking the nth column and row of the new reduced matrix. To illustrate this procedure, we provide an example below. Suppose there are 4 pairs of senders and receivers. The gain matrix and corresponding vector $\mathbf{u}$

are

$$\left(\begin{array}{cccc|c} 0 & F_{1,2} & F_{1,3} & F_{1,4} & u_1 \\ F_{2,1} & 0 & F_{2,3} & F_{2,4} & u_2 \\ F_{3,1} & F_{3,2} & 0 & F_{3,4} & u_3 \\ F_{4,1} & F_{4,2} & F_{4,3} & 0 & u_4 \end{array}\right)$$

The check starts from $F_{1,2}$ and $F_{2,1}$. If these two numbers are both smaller than 1, the process continues with $F_{1,3}, F_{2,3}, F_{3,1}$ and $F_{3,2}$. Suppose $F_{2,3} \geq 1$, then the matrix and vector are reduced to

$$\left(\begin{array}{ccc|c} 0 & F_{1,2} & F_{1,4} & u_1 \\ F_{2,1} & 0 & F_{2,4} & u_2 \\ F_{4,1} & F_{4,2} & 0 & u_4 \end{array}\right)$$

and the third sender goes to sleep. The remaining senders resume checking $F_{1,4}, F_{2,4}, F_{4,1}, F_{4,2}$ in the new matrix. This procedure repeats until the $m \times m$ matrix shrinks to an $m' \times m'$ matrix with all $0 \leq F_{ij} < 1$.

The remaining senders use the matrix and vector of size m' in (3.2) to get $\mathbf{P}$. In addition, the senders check the values of $\mathbf{P}$. If any element is greater than P_{max}, the last row and column of $\mathbf{F}$ and $\mathbf{u}$ are deleted, and the last sender goes to sleep. The reduced matrix is solved again. This procedure is repeated until all $P_i \leq P_{max}$.

Although all senders determine their own subscripts in the CTS relay duration, some may need to be changed if the matrix is reduced. When the procedure to obtain feasible $\mathbf{P}$ terminates, the concurrent senders determine

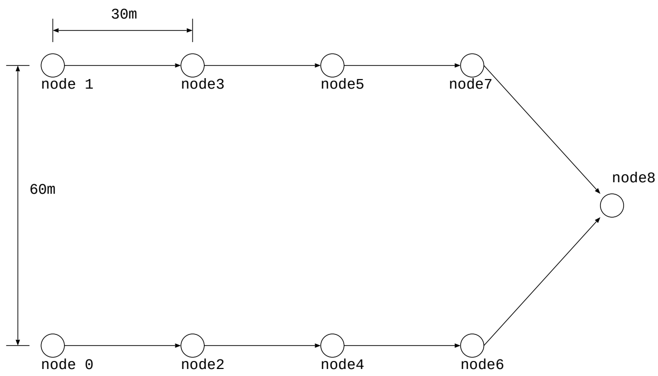

Figure 3.3: Node deployment 1.

their final serial numbers, which are put in the frame head of the data packets to control the ACK reply sequence.

The intended receivers wake up at the beginning of the data transmission phase, and wait for a predefined duration according to their timers. If it does not obtain data before timeout, it will go to sleep until the next period. If it does receive data, it will reply with an ACK in the next phase.

3.4 Numerical Results

We implemented TPMAC using the GloMosim simulator [20]. The propagation model has a path loss factor of 4 [18]. Other parameters are listed in Table 3.1.

We consider two basic scenarios with the node deployments shown in Figs. 3.3 and 3.4 [19] . In the first, since the interference channel gain, G_{ji} ($j \neq i$), is smaller than the signal channel gain, G_{ii}, concurrent transmission can successfully take place between two sender-receiver pairs. However, with

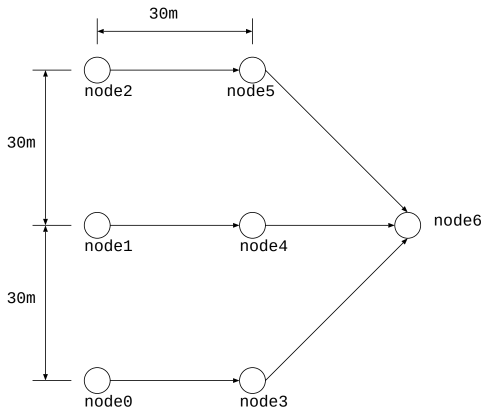

Figure 3.4: Node deployment 2.

Table 3.1: Simulation Parameters

Radio bandwidth	2M bps
Maximum Transmission Power	10 dBm
Packet Length	1024 bytes
Circuit Power in Tx Mode	14.75 mW
Circuit Power in Rx Mode	13.5 mW
Power in Idle Mode	0.5 mW
Power in Sleep Mode	15 μ W
Receive SINR Threshold	11 dB
Noise Figure	10
Environment Temperature	290 K

the second deployment, since the node density is higher, some interference channel gains are high so that concurrent transmission of all sender-receiver pairs cannot succeed. Thus, some senders have to defer sending. More complicated or random scenarios can be viewed as a mixture of these two. We evaluate the performance of TPMAC and compare it with SMAC in terms of average energy-consumption. The results are shown in Figs. 3.5 and 3.6. In SMAC, nodes use a fixed transmission power of 10dBm. TPMAC saves 30% to 50% of the energy compared to SMAC in both cases. This is due to the transmission power reduction of senders, and the reduction of overhearing probability of receivers.

We also investigated the latency of TPMAC for the topology shown in Fig. 3.7. To be fair, we set the periodic sleep duration of SMAC short enough so that no sleep delay occurs. Fig. 3.8 shows that concurrent transmission reduces the point to point latency by half independent of the numbers of hops.

3.5 Summary

We proposed a simple scheme which employs RTS and CTS frames to exchange channel gain information between sender-receiver pairs in a vicinity. Based on the channel gain information, a sender determines an appropriate transmission power which satisfies its minimum required SINR. Even though each sender determines its power individually without negotiation among its peers, multiple sender-receiver pairs can achieve concurrent transmission. It

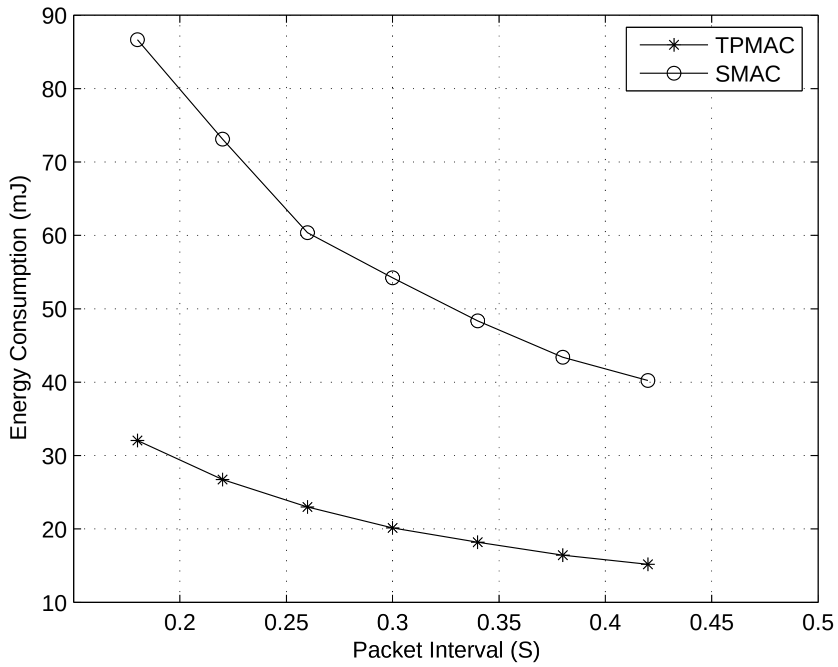

Figure 3.5: Energy consumption with deployment 1.

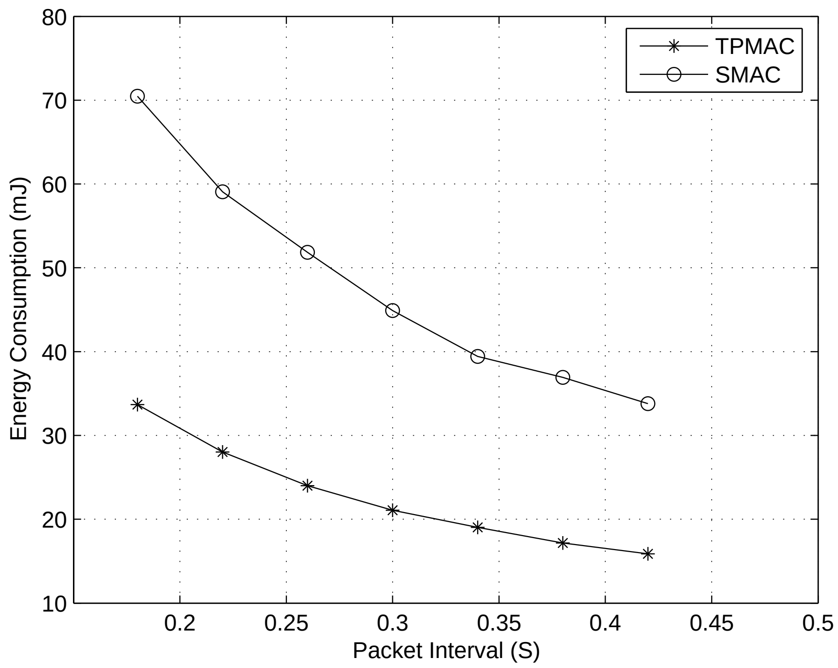

Figure 3.6: Energy consumption with deployment 2.

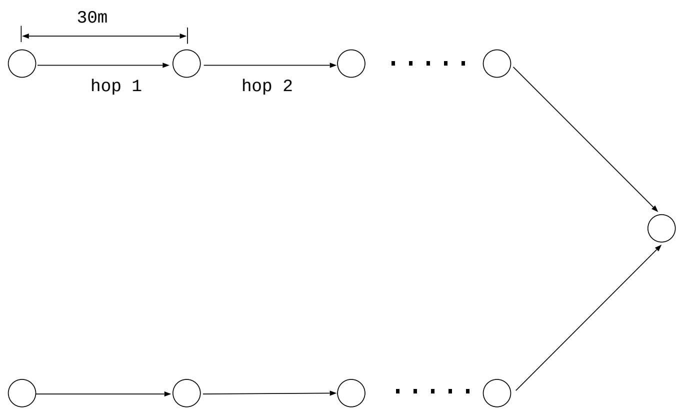

Figure 3.7: Node deployment for delay testing.

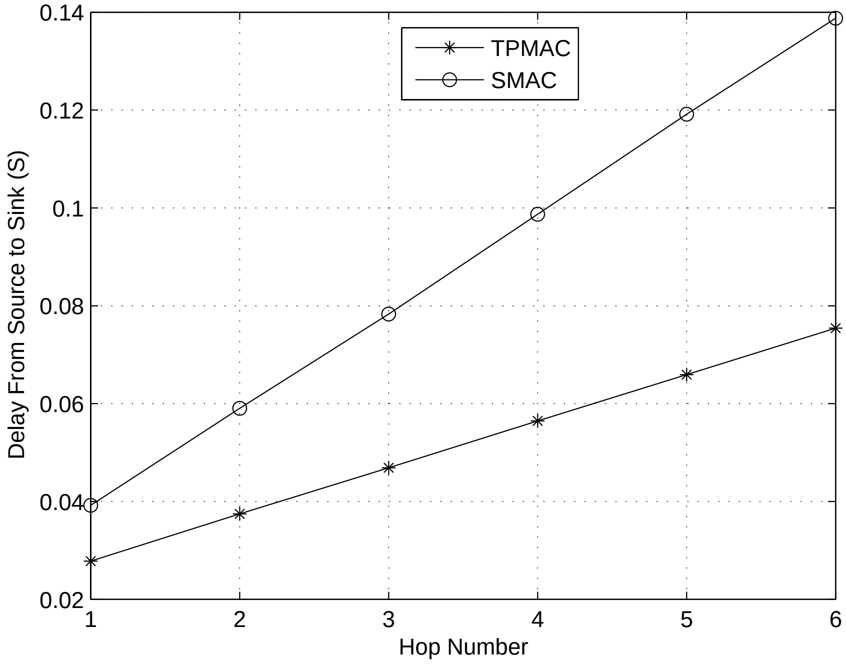

Figure 3.8: Transmission delay vs number of hops.

was shown that transmission power control saves energy, and reduces latency due to concurrent transmissions.

Chapter 4

Random Distributed Detection

4.1 Introduction

Detecting events in an area of interest is the most prevalent envisioned application for a WSN. This is exemplified by the great duck island project conducted by UC Berkeley, which is monitoring elusive seabirds in Maine [21]. Due to the fact that wireless sensor nodes are spatially distributed throughout the area of interest, detection has to been implemented in a form of distributed detection (DD), as depicted in Fig. 4.1. When an event occurs, for example a bird appearing, a number of sensor nodes make local decisions on whether an event has taken place, and then those local decisions are fused to generate a global decision.

The issue of DD has been investigated since the 1980s [22, 23]. However, these classical DD research results cannot be directly applied to WSNs, due to the facts that some fundamental assumptions of classical DD are impractical

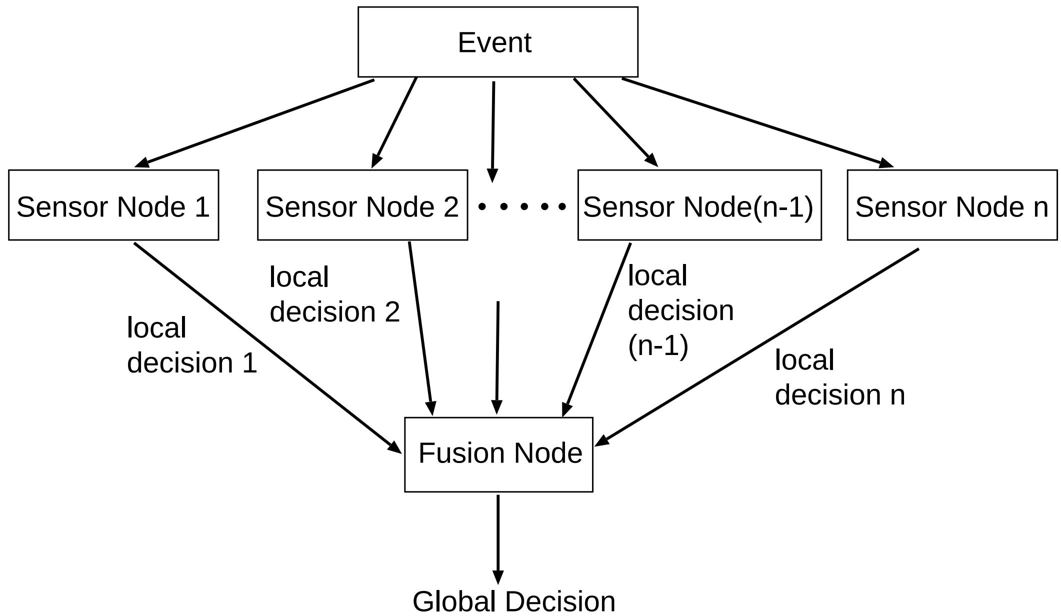

Figure 4.1: Abstract representation of the distributed detection framework.

for WSNs.

Classical DD usually assumes that all nodes in the interest area have independently identical distributed (iid) observations. However, this fundamental assumption is unsuitable for WSNs, due to the fact that when an event takes place within a WSN, its signal usually only covers a portion of the sensor nodes rather than all nodes. Sensor nodes near the event will have better quality observations from those that are far from the event.

In classical DD, the number of nodes involved in the detection is predefined and known [22]. However, in WSNs, sensor nodes are usually randomly deployed [2]. In addition, an event can occur randomly anywhere. Therefore, when an event happens, the number of the nodes involved is random and unknown.

In classical DD, the fusion node is pre-designated. However, this proposition is very impractical for WSNs. Since sensor nodes send their local decision to the fusion node through wireless communication, with significant energy constraints, the fusion node should be close to the local decision nodes. The local decision nodes are always close to the location of events. Thus, the fusion node should always be close to events. However, since events occur randomly anywhere, it is impractical to pre-deploy the fusion center near where the event occurs. The author of [24] utilizes the cluster structure to solve this problem of random fusion node by making the cluster head the fusion node. Obviously, this excludes applications involving non-hierarchical WSNs which do not have a cluster structure. For hierarchical WSNs, the cluster structure reduces scalability of the DD application. The author of [25] assumes that data center nodes are mobile, and can move to a desired place if necessary. The requirement of mobility of nodes puts much challenge and limits the application range of this method.

In this thesis, we propose a new detection scheme named random distributed detection (RDD), which does not need pre-designated fusion node as well as satisfy the requirements of scalability and energy-saving.

4.2 The Problem Model

We consider a large number N of identical sensors randomly deployed over a wide area of size A. The average node density D in this area is defined as N/A. Our purpose is to detect whether an event is occurring within this area at a given time instance.

4.2.1 Assumptions

We assume that an event can occur anywhere within the area with equal probability, and when it occurs, only sensor nodes near the event within radius r receive a signal with power p from the event. As illustrated in Fig. 4.2, we define the circle of radius r around the event as the signal intensity region a. Nodes beyond this region are assumed to be too far to receive a signal from the event. The region a is a simplified model of an event being an isotropic signal source with gradually fading signal intensity [25]. Therefore,

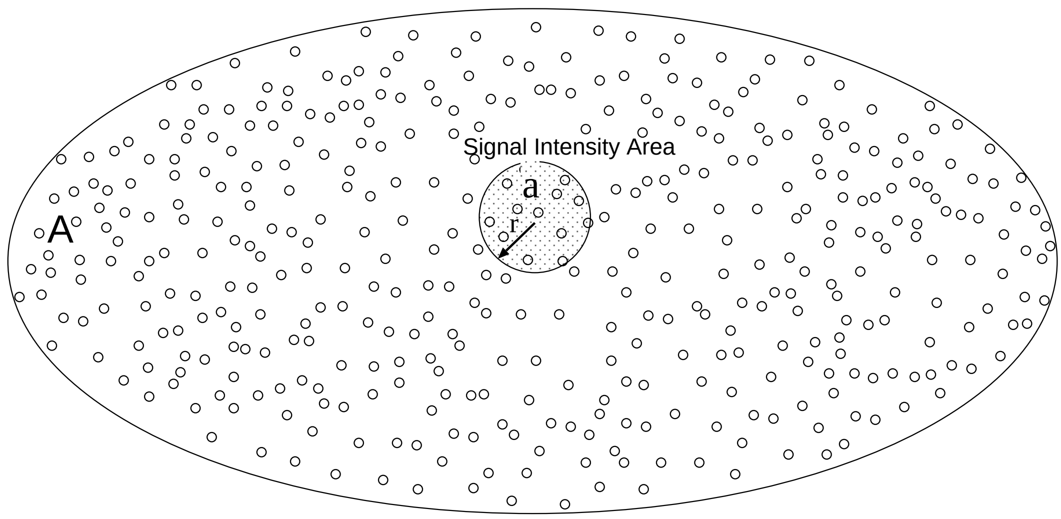

Figure 4.2: Illustration of the node deployment and signal intensity region.

the observation of a node i at a given time instance is modeled as

$$m_i = \begin{cases} \sqrt{p} + z_i & \text{an event is present and node } i \text{ is inside the signal intensity region} \\ z_i & \text{otherwise} \end{cases} \quad (4.1)$$

where z_i is independent observation random noise modeled as a zero-mean Gaussian random variable with variance σ^2.

Node i makes a local decision by comparing its measurement m_i with a local threshold τ, which is predefined and identical for all nodes. If $m_i \geq \tau$, node i makes the decision that an event has been detected; and if $m_i < \tau$, node i makes the decision that an event has not been detected.

A node sends out its local decisions by wireless broadcast. We assume that neighbors within a distance r from the sender can receive the broadcasts.

When a node receives a local decision, it can determine whether the sender is within distance r. Many algorithms for localization and ranging have been developed for wireless sensor networks [26].

4.2.2 RDD scheme

A detection procedure is launched by a command from the sink. Each node observes its measurement and compares it with the local threshold τ. If the measurement is greater than τ, it makes a local detection decision. We call a node which makes a detection decision an activated node, which tries to broadcast its decision to its neighbors. Non-activated nodes do not need to send out their decisions, and so can go to sleep to save energy. The broadcasts from multiple activated nodes may lead to collisions. This problem should be solved by the MAC protocol. When a node hears a local detection decision from its neighbors, it compares the distance from the sender to itself. If the distance is shorter than the signal intensity region radius r, the node takes the local detection decision as a valid local decision. If an activated node collects $K-1$ valid local decisions, it determines that there are at least K activated nodes within a region of πr^2 and then makes a global detection decision. A node which makes a global detection decision broadcasts a suppression signal to stop any local decision broadcasts from other activated nodes to save energy.

4.3 Theoretical Analysis

The probability of failing to make a global detection decision when an event actually occurs is defined as the probability of detection error (PDE); the probability of making a false alarm decision when no event occurs is defined as the probability of false alarm (PFA). From (4.1), the local detection probability of a node within the signal intensity area for a given τ is $\rho_d = Q(-(p-\tau)/\sigma)$. The local detection probability of a node outside the signal intensity area, also named the false alarm probability, is $\rho_{fa} = Q(\tau/\sigma)$, where $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$. Given an $SNR = p/\sigma^2$ and node density D, our goal is to analyze the performance of the random distributed detection scheme in terms of PFA and PDE.

4.3.1 PDE

Since the nodes are randomly scattered through the area with a uniform distribution, the number of nodes n covered by the signal intensity area $a = \pi r^2$ is a homogeneous Poisson process with intensity D [27], where $D = N/A$. Thus, the probability of n sensor nodes within region a is given by

$$p(n) = \frac{(D \times a)^n}{n!} e^{-D \times a} \tag{4.2}$$

Given a specific number of nodes n within the signal intensity area, the probability that j of n nodes become activated is $\binom{n}{j}(\rho_d)^j(1-\rho_d)^{n-j}$. If $j \geq K$, a global detection decision would be made. However, this is not always the case. As an example, consider Fig. 4.3(a) which shows the circle region centered at an event which covers 4 active nodes. If $K = 4$, a global detection decision is supposed to be made. However, in our scheme, there are no pre-designated fusion nodes. When an activated node counts valid local decisions from its neighbors, it always takes itself as the center of the intensity area. However, since the regions centered at the activated nodes do not fully overlap with the actual signal intensity region, none of them can cover 4 or more activated nodes, thus, a detection error occurs. We call the phenomenon region mismatch, which obviously relates to the the geometric deployment of the nodes. We define $\xi(j, K)$ as the function to estimate the PDE when there are j activated nodes within the signal intensity area with global threshold K. If $(D \times \rho_{fa}) < 0.5$, ξ can be estimated as as follow

$$\xi(j, K) = \begin{cases} \frac{0.62 - \rho_{fa} \cdot D \cdot \theta(K)}{(j-K+1)!} & \text{when } j \geq K \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

which is derived from the observations below. Firstly, if $j < K$, PDE is one. Secondly, when $j \geq K$, the region mismatch will result in a PDE, which from the simulation is estimated as 0.62. On the other hand, as shown in Fig. 4.3(b), a false alarm node near the signal intensity area may contribute to making a global decision of detection. This shows that false alarm activated nodes near the intensity area will decrease PDE. Nonetheless, a smaller K is more susceptible to false alarms than a bigger K. Let $\theta(K)$ indicate the susceptibility coefficient of a certain K to a nearby false alarm node, and is defined as

$$\theta = \begin{cases} 1 & K \leq 4 \\ 0.4 & K = 5 \\ 0 & K > 5 \end{cases} \tag{4.4}$$

which implies that when the global threshold $K \leq 4$, the global decision is very susceptible to nearby false alarm nodes, and when threshold $K > 5$ the susceptibility is ignorable. In (4.3), $\rho_{fa} \cdot D$ represents the normalized average number of nearby false alarm nodes, which combines with the susceptibility coefficient $\theta(K)$ to reduce the PDE. Thirdly, the more j is greater than K, the smaller PDE is. Thus, ξ is inversely proportional to the factorial of $(j - K + 1)$. Consequently, given the PDE for a specific value j, the PDE averaged on all j is calculated as:

$$PDE = \sum_{j=K}^{N} \sum_{n=0}^{N} \xi p(n) \binom{n}{j} (\rho_d)^j (1 - \rho_d)^{n-j} \tag{4.5}$$

4.3.2 PFA

Given the local false alarm possibility ρ_{fa}, the probability that m out of the N nodes will become activated nodes is given by $\binom{N}{m}(\rho_{fa})^m(1 - \rho_{fa})^{N-m}$. The m activated nodes are randomly scattered throughout the whole area. We divide the whole area into A/a pieces of regions. If there are more than K active nodes among the m nodes within any piece of region, a global false alarm will be made. The probability that more than K nodes will be within

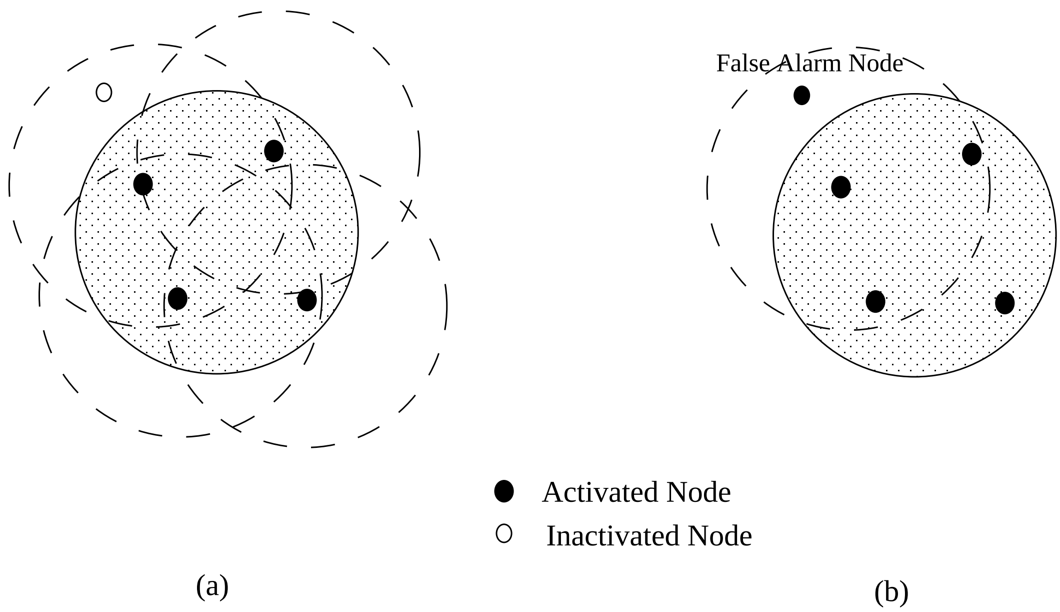

Figure 4.3: (a) An example of region mismatch leading to PDE. (b) An example of false alarm nodes leading to correct detection.

a specific region is $1 - \sum_{i=0}^{K-1} \frac{(ma/A)^i}{i!} e^{-ma/A}$. Thus, the probability that any piece of region makes a global false alarm is estimated as

$$pfa(m) = (A/a - 10ln(m+1))(1 - \sum_{i=0}^{K-1} \frac{(ma/A)^i}{i!} e^{-ma/A}) \tag{4.6}$$

Then, the PFA is

$$PFA = \sum_{m=0}^{N} p(m) \binom{N}{m} (\rho_{fa})^m (1 - \rho_{fa})^{N-m} pfa(m) \tag{4.7}$$

4.3.3 Determine Optimal τ and K

According to (4.5) and (4.7), PDE goes up and PFA goes down monotonically with increasing of τ. We define the performance criterion C as a linear weighted combination of PFA and PDE, $C = \alpha PFA + \beta PDE$, where $\alpha + \beta = 1$. For the parameters SNR, D and K considered in this thesis, C is a convex function with respect to τ, as shown in Fig. 4.4. Thus, it is easy to apply a classical optimization algorithm to find the minimum of C, i.e., where the gradient of $C(\tau)$ is equal to zero.

For a small value of K, PFA stays at a relatively high level for all values of τ. On the contrary, for a large value of K, PDE stays at a relatively high level. So, there may exist a proper K, for which both PFA and PDE can be relatively low. As an example, Fig. 4.4 shows that the minimum C for $K = 5$ is smaller than those of other values of K. There is another consideration for choosing K with respect to energy savings in terms of the

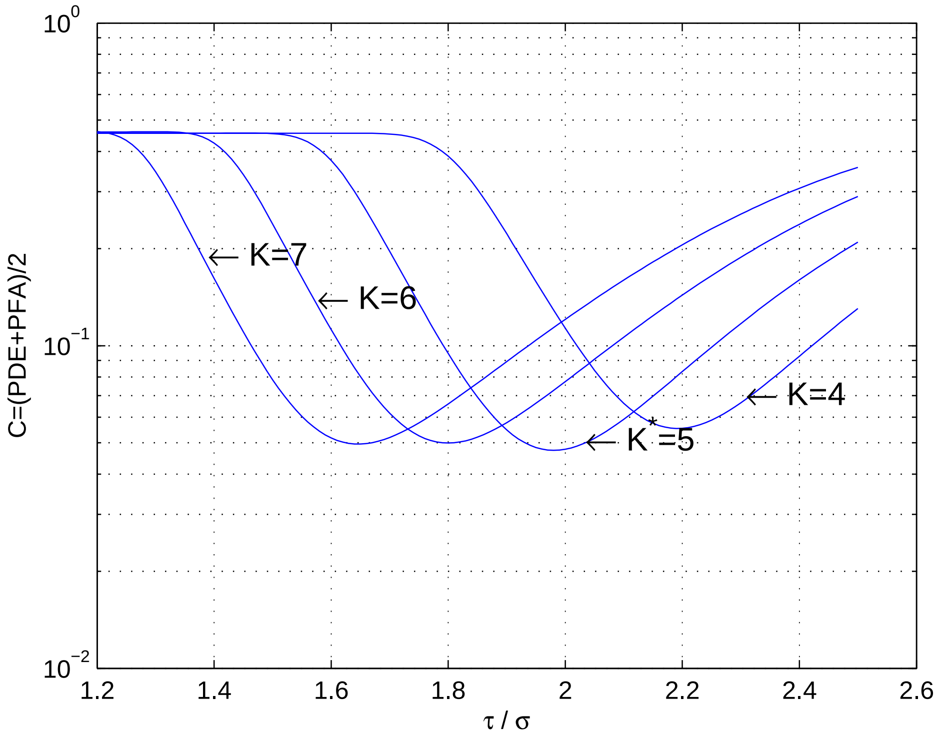

Figure 4.4: An example of minimum C for different K.

number of node broadcasts. Since K determines the minimum number of node broadcasts before a global decision is made, K is proportional to the average number of node broadcasts. Thus, a larger K indicates a higher energy costing due to more broadcasts. Furthermore, since with a large K, a high PDE is the dominant problem, τ should be a small to reduce the PDE, but this will increase the number of false alarm activated nodes, and consequently, increase the number of node broadcasts.

4.4 Numerical Results

We consider a two-dimensional space A of size of $100\pi m^2$. At any time instant, an event may happen anywhere within this area with equal probability. When an event occurs, nodes within a radius of $1m$ of the event will receive a signal from the event. For each set of simulation parameters of SNR, N, K, and τ, 100,000 Monte Carlo runs were executed to get the average PDE and PFA.

Figs. 4.5 to 4.10 compare the estimated PDE results from (4.5) and the simulated results for different sets of parameters. The solid lines represent the estimated results, and the points represent the simulated results. From the figures we can tell that simulated results fairly agree with the estimated results, except that in Figs. 4.5 to 4.7, the simulated PDE at the smallest τ appear much lower than the estimated ones. It is because that at small τ the number of false alarm nodes will be large, which will decrease PDE as well as dramatically increase the PFA. Although (4.5) considers the impact of false alarm node, we limit the the impact by the condition of $(D \times \rho_{fa}) < 0.5$. However, this condition is violated at the smallest τ in Figs. 4.5 to 4.7. Thus, the simulated results at those small τ are less than the estimated ones. On the other hand, although the estimation at small τ is inaccurate, those τ hardly will be chosen as a parameter in practical applications, due to the fact the at those τ the PFA will be huge and unacceptable.

The PDE goes up with τ increasing. For a given SNR, a bigger K results in a larger PDE at a same τ. To reach a required PDE, a bigger K has to take a much smaller τ. However, if the SNR or node density increases, the required PDE can be achieved even on a relative large τ.

Figs. 4.11 and 4.12 compares the estimated PFA results from (4.7) and the simulated results for different sets of parameters. The PFA goes down with τ increasing. A smaller K results in a larger PFA at a same τ. To reach a required PFA, a smaller K has to take a much larger τ. If the node density increases, the τ has to increase more to achieve the required PFA.

Table 4.1 summarizes the simulation results for different sets of parameters of SNR, D and K. The τ value was optimized with respect to the performance criterion C, where $\alpha = \beta = 0.5$. By comparing the simulation results, we can tell that for a given set of SNR, D, the optimal τ decreases as K increases. However, there is a K, for which the C on its optimal τ is smaller than others. For example, $K = 6$ gets the smallest C for $N = D \times A = 2000$; $K = 7$ gets the smallest C for $N = D \times A = 4000$. The best value of K appears to be only determined by D.

Since the energy for detection is primarily cost on the wireless broadcasts for local decisions, we use the number of node broadcasts as an indication of energy cost. The simulation results show that although there are thousands of nodes, only tens of node broadcasts are required to get a fairly good performance. If the SNR or node density increases, the PDE and PFA can simultaneously decrease as well as the number of node broadcasts decreases.

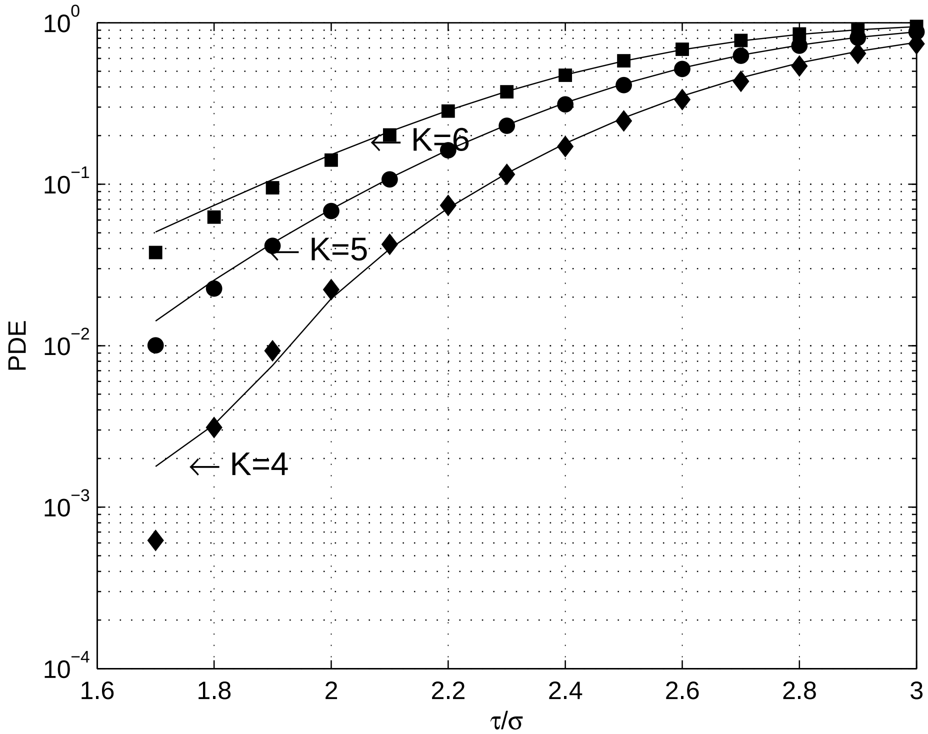

Figure 4.5: Comparison of theoretical and simulated PDE when N=2000, SNR = 6dB.

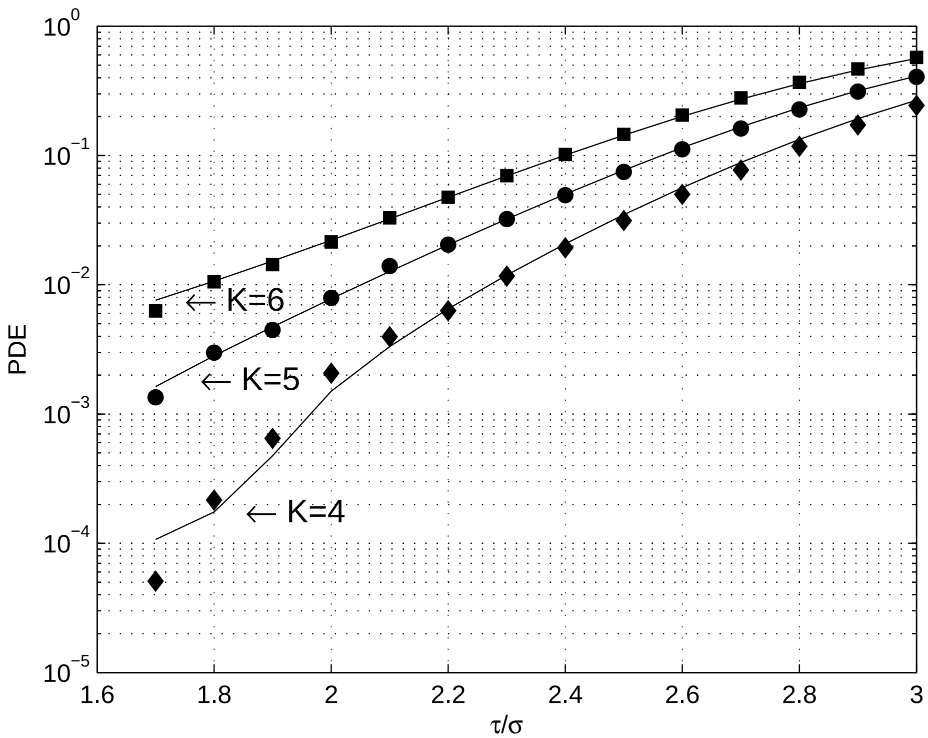

Figure 4.6: Comparison of theoretical and simulated PDE when N=2000, SNR = 8dB.

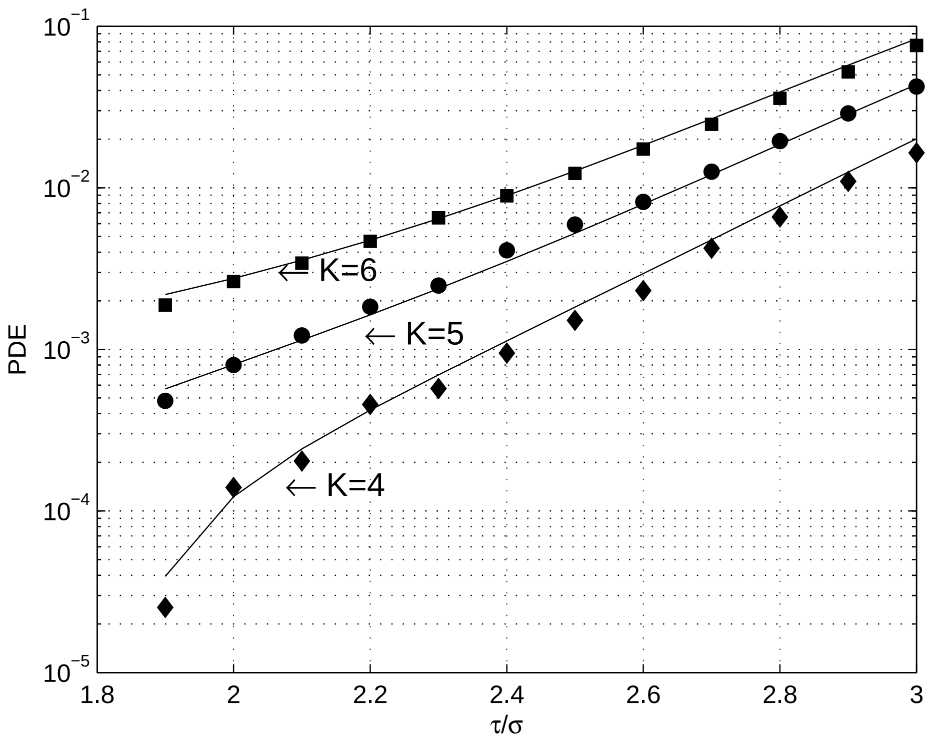

Figure 4.7: Comparison of theoretical and simulated PDE when N=2000, SNR = 10dB.

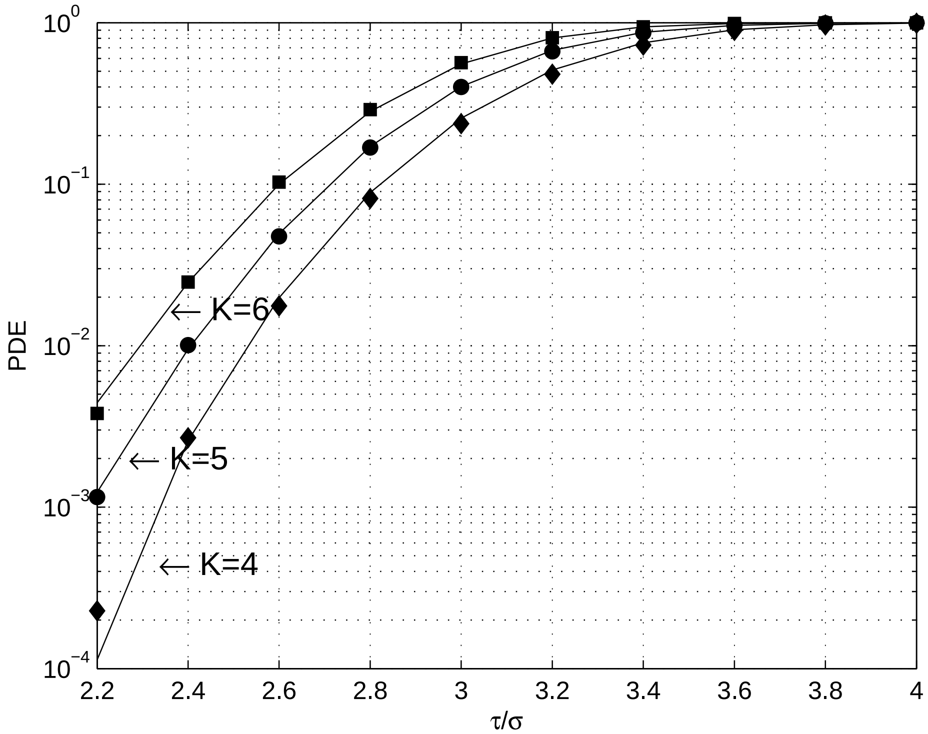

Figure 4.8: Comparison of theoretical and simulated PDE when N=4000, SNR = 6dB.

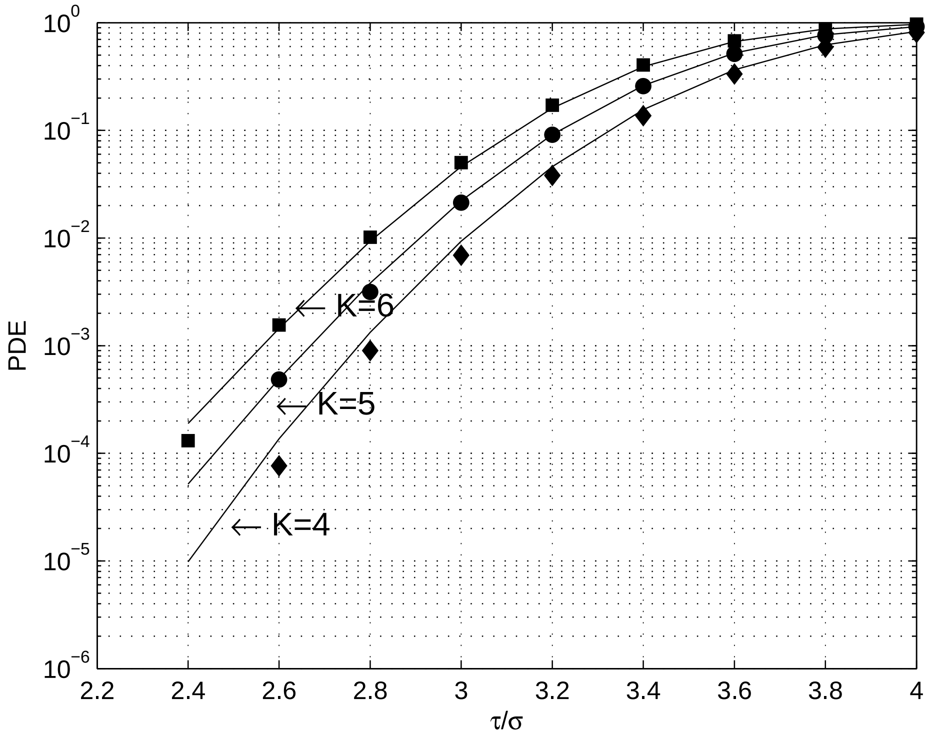

Figure 4.9: Comparison of theoretical and simulated PDE when N=4000, SNR = 8dB.

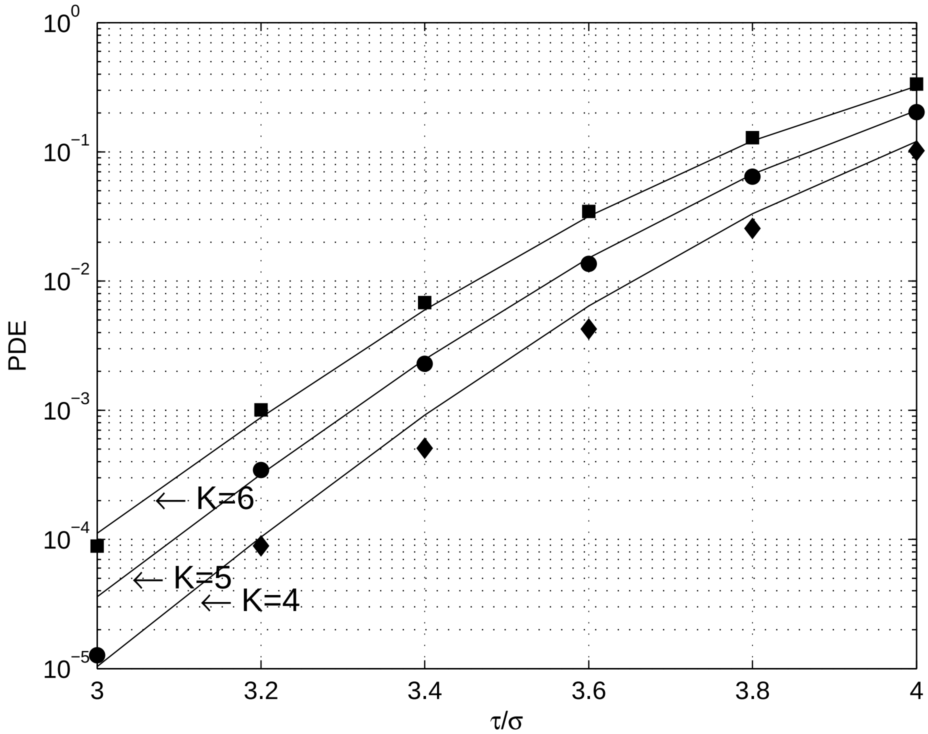

Figure 4.10: Comparison of theoretical and simulated PDE when N=4000, SNR = 10dB.

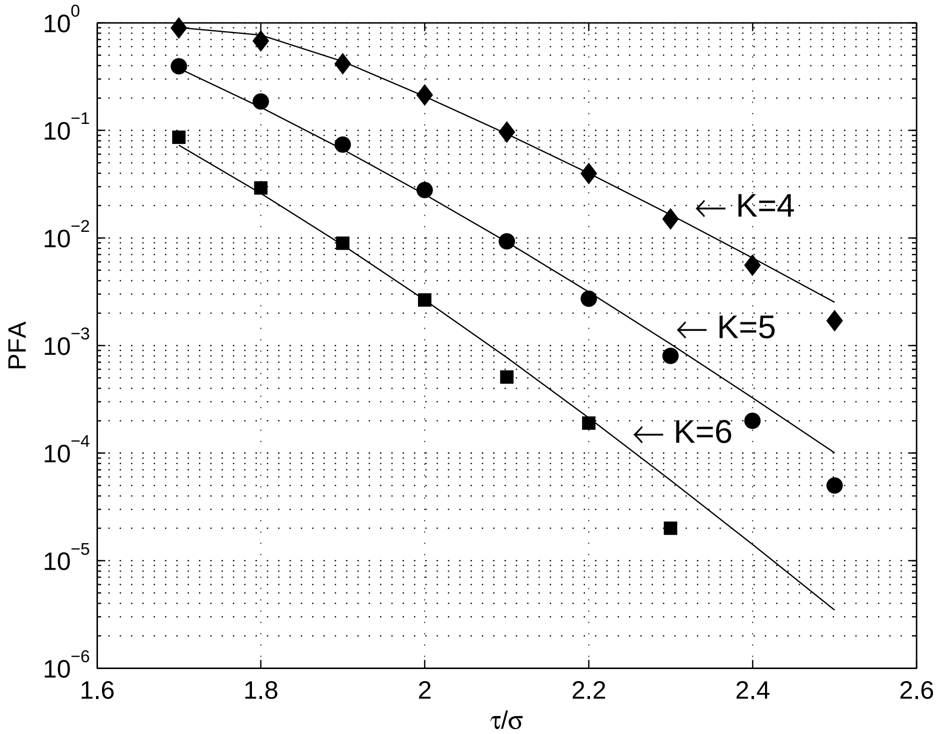

Figure 4.11: Comparison of theoretical and simulated PFA when N=2000.

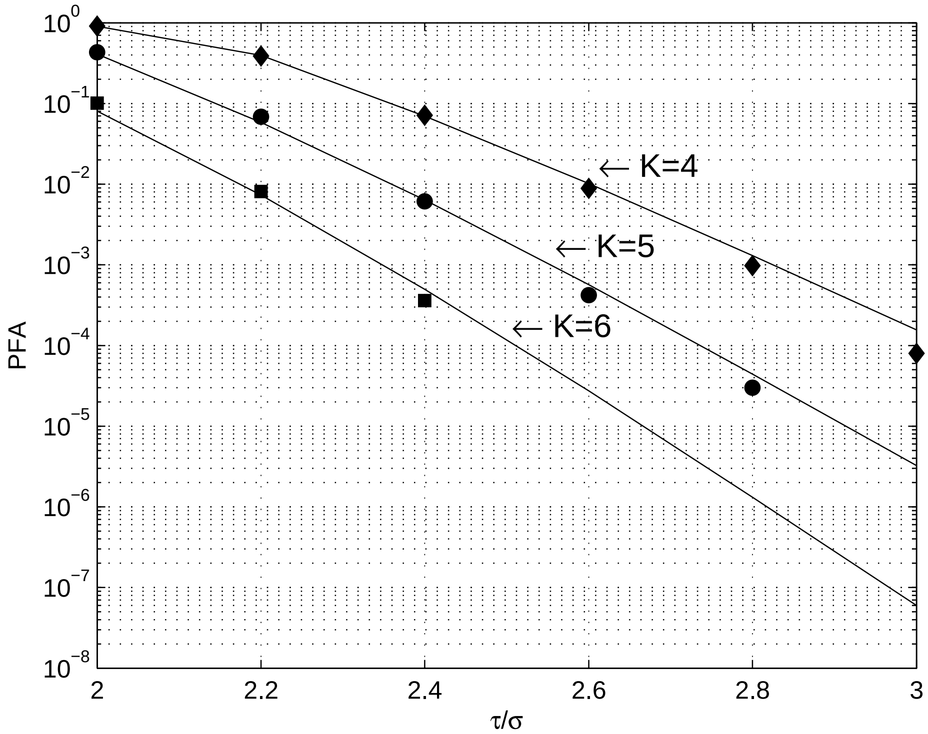

Figure 4.12: Comparison of theoretical and simulated PFA when N=4000.

Table 4.1: Simulation Results for Various Optimal τ

$N = 2000, SNR = 6dB$				
K	τ / σ	PDE	PFA	Average Number of Node Broadcasts
4	2.1920	0.0696	0.0463	21.5583
5	1.9804	0.0597	0.0376	36.6710
6	1.8001	0.0606	0.0328	56.2272
7	1.6994	0.0839	0.0148	72.8030
8	1.6972	0.1363	0.0026	77.1444
9	1.6963	0.2165	0.0002	81.5817
$N = 2000, SNR = 8dB$				
K	τ / σ	PDE	PFA	Average Number of Node Broadcasts
4	2.3552	0.0154	0.0087	13.6259
5	2.1289	0.0114	0.0069	24.4075
6	1.9354	0.0098	0.0063	39.3114
7	1.7700	0.0219	0.0061	58.4800
8	1.6996	0.0256	0.0015	70.2122
9	1.6991	0.0510	0.0002	74.2782
$N = 4000, SNR = 6dB$				
K	τ / σ	PDE	PFA	Average Number of Node Broadcasts
4	2.5689	0.0132	0.0133	14.7916
5	2.3910	0.0074	0.0072	24.1643
6	2.2387	0.0042	0.0051	36.1428
7	2.1111	0.0045	0.0033	50.7302
8	2.0000	0.0049	0.0028	66.8105
9	1.9994	0.0109	0.0004	70.5397
10	1.9989	0.0218	0.0000	73.4152
11	1.9982	0.0393	0.0000	76.9932

4.5 Summary

We proposed a practical, simple to implement scheme for WSN detection applications. We provided two close-form expressions to estimate PDE and PFA and validated them by extensive simulations, which shows that PDE goes up and PFA goes down monotonically with the local threshold τ increasing. We defined the overall performance as a linear combination of PDE and PFA. For the parameters of SNR, K and D considered in this thesis, the overall performance is a convex function with respect to τ. We employed a simple optimization algorithm to find the optimal threshold τ, on which the overall performance is best for a given set of SNR, K and D. By comparing the simulation results, we can tell that for a given set of SNR, D, there is a K, for which the overall performance on its optimal τ is better than others. Nonetheless, the best value of K appears to be only determined by D. The simulation results also show that although there are thousands of nodes, only tens of node broadcasts are required to get a fairly good performance. And if the SNR or node density increases, the PDE and PFA can simultaneously decrease as well as energy cost decreases.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we studied three topics on WSNs. The first one was energy-efficient MAC protocol design. A unidirectional tree structure is the major communication paradigm in WSNs when data is transferred from source nodes to the sink. It is formed by data aggregation combined with multiple-hopping. We proposed a medium access control (MAC) protocol which takes advantage of this structure to schedule the send and receive times for all nodes within the network. As all nodes only wake up when they send or receive, significant energy is saved by reducing idle listening, collisions and overhearing. And then we evaluated the protocol performance by using the GloMoSim simulator.

The second topic was about transmission power control of wireless sensor nodes. In wireless networks, transmission power control (TPC) can save energy and decrease transmission latency, which are desirable goals in the design of WSNs. Due to the inherent feature of spatial distribution of sensor nodes, TPC in a WSN has to be distributed, in the sense that senders determine their transmission power by themselves rather than a central controller to coordinate them. We proposed a simple scheme which employs RTS and CTS frames to exchange channel gain information between sender-receiver pairs in a vicinity. Based on the channel gain information, a sender determines an appropriate transmission power which satisfies the SINR requirement on its responding receiver. Even though each sender determines its power individually, multiple senders can achieve concurrent transmission. Simulation results showed that our scheme can save 30% to 50% of the energy, and also reduce transmission latency.

The third topic was on one of the most prevalent envisioned applications for WSNs – event detection. Due to the fact that wireless sensor nodes are spatially distributed throughout the area of interest, this application has to been implemented in distributed form. Unlike other related distributed detection schemes, we proposed a scheme which does not need a pre-designated fusion node, and this greatly improves network scalability. We provided close-form expressions to estimate the probabilities of detection failure and false alarm, and validated them by extensive simulations. Given these two expressions, a user can easily determine suitable parameters to satisfy the performance requirements.

5.2 Future Work

There are several possible future research directions for the work presented in this thesis. Here we briefly discuss extensions on the second and third thesis topics.

In the second topic, we assumed that within a concurrent transmission pairs vicinity, a potential sender can receive a signal successfully from all intended receivers, and there is no interference from a node outside the vicinity. This assumption ensure that all potential senders can collect complete channel information within the vicinity. For a more practical scenario, we may have to relax this assumption, and assume that potential senders sometimes only collect incomplete channel information. Thus, these senders may have to make a probabilistic sending decision based on the incomplete information.

In the third topic, we assumed the signal intensity region is a circle, within which all nodes receive the same signal power from the event. This is a simplified model of the signal power distribution. A more realistic assumption would be an isotropic signal source with fading signal power. This should be studied in the future.

Bibliography

[1] C.S. Raghavendra, Krishna M. Sivalingam, and Taieb Znati, *Wireless Sensor Networks*, Kluwer Academic Publishers, 2004.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] A.J. Goldsmith, and S.B. Wicker, “Design challenges for energy-constrained ad hoc wireless networks,” *IEEE Wireless Communications*, vol. 9, Issue 4, pp. 8–27, Aug. 2002.

[4] A.J. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.

[5] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” Proc. *ACM Conf. on Embedded Networked Sensor Systems*, pp. 181–192, Nov. 2003.

[6] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” Proc. *IEEE INFOCOM*, pp. 1567–1576, June 2002.

[7] B. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks,” *Distributed Computing Systems Workshop*, pp. 575–578, July 2002.

[8] S. Kulkarni, A. Iyer, and C. Rosenberg, "An address-light, integrated MAC and routing protocol for wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 4, pp. 793–806, Aug. 2006.

[9] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," *Wireless Commun. and Networking*, vol. 3, pp. 1942–1947, Mar. 2003.

[10] T.-S. Chen, H.-W. Tsai, and C.-P. Chu, "Gathering-load-balanced tree protocol for wireless sensor networks," Proc. *IEEE Int. Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 8–13, June 2006.

[11] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," Proc. *Parallel and Distributed Processing Symp.*, pp. 224, Apr. 2004.

[12] W. B. Heinzelman, and A. P. Chandrakasan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[13] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[14] J. Zander, "Distributed cochannel interference control in cellular radio systems,"*IEEE Trans. Vehic. Tech.*, vol. 41, no. 3, pp. 305–311, Aug. 1992.

[15] J.P. Monks, V. Bharghavan, and W.-M.W. Hwu, "A power controlled multiple access protocol for wireless packet networks," Proc. *IEEE INFOCOM*, pp. 219–228, Apr. 2001.

[16] Fan Wang, O. Younis, and M. Krunz, “GMAC: A Game-theoretic MAC protocol for mobile ad hoc networks,” *Int. Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1–9, Apr. 2006.

[17] E. Seneta, *Nonnegative Matrices and Markov Chains*, New York: Springer, 1981.

[18] J.M. Molina-Garcia-Pardo, A. Martinez-Sala, M.V. Bueno-Delgado, E. Egea-Lopez, L. Juan-Llacer, and J. Garca-Haro, “Channel model at 868 MHz for wireless sensor networks in outdoor scenarios,” Proc. *IWWAN*, May 2005.

[19] Zhiwei Zhao, Xinming Zhang, Peng Sun, and Pengxi Liu, “A transmission power control mac protocol for wireless sensor networks,” Proc. *International Conference on Networking*, pp. 5–9, Apr. 2007.

[20] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: A library for parallel simulation of large-scale wireless networks,” Proc. *Workshop on Parallel and Distributed Simulation*, pp. 154–161, May 1998.

[21] J. Kumagai, “Life of birds [wireless sensor network for bird study],” *IEEE Spectrum*, vol. 41, pp. 42–49, Apr. 2004.

[22] A.R. Reibman, and L.W Nolte, “Optimal detection and performance of distributed sensor systems,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 23, pp. 24–30, Jan. 1987.

[23] J.-F. Chamberland, and V.V. Veeravalli, “Wireless sensors in distributed detection applications,” *IEEE Signal Processing Mag.*, vol. 24, pp. 16–25, May 2007.

[24] Tian Qingjiang, and E.J. Coyle, “Optimal distributed detection in clustered wireless sensor networks,” *IEEE Trans. Signal Processing*, vol. 55, pp. 3892–3904, July 2007.

[25] Y. Sung, L. Tong, and A. Swami, "Asymptotic locally optimal detector for large-scale sensor networks under the Poisson regime," *IEEE Trans. Signal Processing*, vol. 53, pp. 2005–2017, June 2005.

[26] A. Boukerche, H.A.B.F. Oliveira, E.F. Nakamura, and A.A.F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, issue 6, pp. 6–12, Dec. 2007.

[27] Michael Mitzenmacher, Eli Upfal, *Probability and Computing*, Cambrige Press, 2005.

University of Victoria

Partial Copyright License

Title of Thesis: On Wireless Sensor Networks.

Author: ________________________________

Xiao Liang

May 1, 2008