國立交通大學

網路工程研究所

碩士論文

無線感測網路下用以支援多種型態流量之 TDMA-over-CSMA 鏈結層協定

A TDMA-over-CSMA Link Layer Protocol for Supporting Multi-type Traffics in a Wireless Sensor Network

研 究 生：李亭瑩

指導教授：曾煜棋　教授

黃俊龍　教授

中華民國九十七年六月

無線感測網路下用以支援多種型態流量之 TDMA-over-CSMA 鏈結層協定

A TDMA-over-CSMA Link Layer Protocol for Supporting Multi-type Traffics in a Wireless Sensor Network

研究生：李亭瑩　　Student：Ting-Ying Li

指導教授：曾煜棋　　Advisor：Yu-Chee Tseng
黃俊龍　　Jiun-Long Huang

國立交通大學
網路工程研究所
碩士論文

A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

無線感測網路下用以支援多種型態流量之 TDMA-over-CSMA 鏈結層協定

研究生：李亭瑩　　　　　　　　　指導教授：曾煜棋 教授
黃俊龍 教授

國立交通大學網路工程研究所碩士班

摘　要

無線感測網路中存在著多種不同類型的流量，因此設計一個滿足各種流量需求的媒介存取控制協定是個具有挑戰性的議題。例如：緊急事件的回報在資訊傳遞的時間上較為緊要，而規律型的回報則是在感測節點上的省電考量較重要。在本論文中，我們提出了一個 TDMA-over-CSMA 的鏈結層協定，可支援無線感測網路中的資料匯集、緊急事件回報及需求型的流量。我們的主要目標有：使多數的流量可達到較低的傳輸延遲、對於匯集型的流量可達到資料聚集及省電的功效、及降低同一緊急事件區域中同時偵測到資訊之節點的資料冗餘度。另外，我們也提出了一個重複利用時槽的方法以支援需求型的流量。模擬結果呈現出我們的協定具有較佳的或可抗衡現有協定的表現，而這些現有協定可做為處理單一類型流量的代表。我們也在 Jennic—以 ZigBee 協定為基礎的平台上完成實作的部分。

關鍵詞：資料匯集、鏈結協定、媒介存取控制、排班、TDMA、無線感測網路

A TDMA-over-CSMA Link Layer Protocol for Supporting Multi-type Traffics in a Wireless Sensor Network

Student：Ting-Ying Li　　　Advisor：Prof. Yu-Chee Tseng
Prof. Jiun-Long Huang

Department of Computer Science
National Chiao-Tung University

ABSTRACT

Traffics in wireless sensor networks are typically of multiple types. To design a suitable medium access control (MAC) protocol to satisfy the requirements of all traffic types is a challenging issue. For example, reports of emergency events are more time-critical, while regular reports could be more energy-critical. In this paper, we present a TDMA-over-CSMA link protocol for supporting *convergecast*, *event-reporting*, and *on-demand* traffics in a wireless sensor network. The main goals are to achieve low transmission latency for most types of traffics, to facilitate data aggregation and energy efficiency for convergecast traffics, and to reduce data redundancy in an event area when multiple nodes detect the same event at the same time. In addition, an opportunistic slot reuse scheme is proposed to exploit spatial reuse and to support on-demand traffics. Simulation results show that our protocol performs better than or comparable to existing protocols, which can typically take care of only one type of traffics. Implementation work has also been done on Jennic, a ZigBee-based platform.

Keywords: convergecast, link protocol, MAC, scheduling, TDMA, wireless sensor network

誌 謝

誠摯地感謝指導教授曾煜棋老師，感謝老師讓我有機會成為您的研究團隊的一份子，並提供我們優良的實驗室環境及資源，也感謝這兩年來悉心地教導，讓我學到了許多做研究的精神和態度。感謝我的共同指導教授黃俊龍老師，老師總是可以像個朋友般地和學生相處，在我的研究過程中遇到困難時，老師會與我分享自己的經驗並給予我建議，讓我在這兩年裡，多了一份自在感。感謝 HSCC 實驗室及黃老師實驗室的所有學長姊、同學、學弟妹們，也特別感謝協助過我完成論文的學長姊們，及共同努力過的碩二伙伴：志誠、子文、婉婷、怡廷、小晉、書賢、文杰、雁智；欣怡、壬禾、國和、建平等，感謝有你們才能讓我研究所的學業及論文得以順利完成。

感謝口試委員賴添煌教授、吳世琳教授及趙志民教授，在口試時給予我許多寶貴的建議，讓本論文得以更加完善，非常感謝各位老師。

感謝這兩年帶給我歡笑及感動的好朋友們：交大的博濟、翊庭、頌文、振揚、世男、立先、文賢、柏亨、貴笠、幸芸、岳展、睿裕等；清大的崇豪、俊忠等；竹教大的雨臻等伙伴們，你們讓我看見化不可能為可能的力量，也讓我對這個世界多了份關懷與熱誠。謝謝博濟一直以來對我的照顧，也替我分擔了許多生活中的大小事。謝謝同樣是碩二的翊庭在這一年中互相勉勵打氣，你的細心經常提醒我許多重要的事。也感謝我大學及高中的好朋友們，謝謝你們無論是在有形或無形中給予我的支持與鼓勵及多年來的情誼。

特別感謝我的母校東吳大學資訊系的郭育政老師，感謝郭老師從我大學時期開始就不斷地鼓勵我，當我遇到學業或生活上的困難時，老師總是不厭其煩地教導我並與我分享人生的經驗，畢業後也經常透過 e-mail 與老師聯繫，這兩年來許多的不如意也在老師的鼓勵下逐一化解，由衷地感謝老師您對學生的用心，您不只是個良師，也是個益友，非常感恩您這些日子來的照顧。

最後感恩我的父母，這二十多年來真是辛苦了，您們辛勤地工作養家及對孩子們無微不至的照顧，讓我能夠無憂無慮地渡過人生的每個階段，也謝謝所有支持我的家人們，有你們的陪伴我才能有今日的成就。感恩所有幫助過我的人、事、物。祝福所有人都能平安喜樂。

李亭瑩　謹誌
于 國立交通大學網路工程研究所 2008.06

Contents

摘要 i

Abstract ii

誌謝 iii

Contents iv

List of Tables vi

List of Figures vi

1 Introduction 1

2 Related Works 3

3 System Model 5

4 Protocol Details 8

4.1 Convergecast Scheme 8
4.2 Event-reporting Scheme 9
4.3 Opportunistic Slot-reuse Scheme 11
4.4 Energy-Efficiency 12

5 Implementation 13

6 Simulation Results 16

6.1 Protocol parameters 16
6.2 Traffic generation models 17
6.3 Performance metrics 18
6.4 Success Rate 19
6.5 Delay 19

6.6 Convergecast, Event and Broadcast Coverage 20

7 Conclusions **26**

Bibliography **27**

List of Tables

6.1 Simulation parameters 16
6.2 Simulation Cases 18
6.3 The ranking of the importance of metrics. 19
6.4 Performances for the three protocols 21

List of Figures

3.1 The frame structure of our protocol. 6
3.2 Classification of slots. 7
4.1 (a) The result of the bottom-up manner. (b) After the top-down manner, V3and V4 are assigned to new slots. (c) The final result of the slot assignment scheme. 9
4.2 An example of an emergency event reporting and pipeline effect. 11
5.1 User interface. 13
5.2 The environment of demonstration. 14
5.3 The environment of demonstration. 14
5.4 The remaining energy (Volt) of power-saving mode and non-power saving mode. 15
5.5 The energy conservation (Am) of node 3. 15
5.6 The network topology and slot allocation of Fig. 5.5. 15
6.1 The results of case 1. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f)

Broadcast coverage. 22
6.2 The results of case 2. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage. 23
6.3 The results of case 3. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage. 24
6.4 The results of case 4. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage. 25

Chapter 1

Introduction

Wireless sensor networks (WSNs) have been studied intensively [3, 8, 4]. Traffics in WSNs can be categorized into *periodical* traffics and *sporadic* traffics. In many WSN applications, sensor nodes have to report sensed data periodically. This type of traffics is more predictable and thus can be handled by a convergecast scheme or a scheduling scheme [17, 11, 6]. On the other hand, sporadic traffics are more unpredictable and can be further categorized as *time-critical* and *non-time-critical* ones. A typical instance of time-critical traffics is emergent event reporting. Non-time-critical traffics include all other communications, such are broadcast, queries, and control messages; here we refer to them as *on-demand* traffics. A WSN normally has to handle more than one type of traffics. For example, in a temperature monitoring system, nodes report their readings periodically and exchange control packets sporadically. When a fire occurs, the event has to be reported as quickly as possible.

Different MAC protocols may facilitate different traffic types. For periodical traffics, a schedule-based MAC protocol, such as TDMA, may be more appropriate. However, a schedule-based MAC protocol is less flexible and may suffer from higher delays. For event reporting, a contention-based MAC protocol, such as CSMA, may be more appropriate. It is clear that a hybrid MAC protocol is needed if multi-type traffics have to be handled simultaneously.

A lot of efforts have been dedicated to MAC protocols for WSNs [11, 21, 13, 18, 15, 19, 20, 16, 2, 7, 22, 23]. Reference [11] proposes a schedule-based protocol called FlexiTP to handle periodical traffics. Unfortunately, sporadic traffics are not addressed. Several MAC protocols have been proposed to handle event-reporting traffics [20, 7, 22, 23]. These protocols exploit spatial correlation of sensor readings to reduce redundant reports. Unfortunately, these protocols can not handle periodical traffics well. TRAMA [15] is a schedule-based TDMA MAC protocol. It allows a node with heavier traffics to borrow slots from nodes with lighter traffics. So, it can also handle on-demand traffics well. But TRAMA is not tailored to convergecast and event-reporting traffics. In [16], a hybrid TDMA/CSMA protocol called Z-MAC is proposed. Nodes experiencing lower contention will work in a CSMA mode, while those experiencing higher contention will work in a TDMA mode. Thus, Z-MAC enjoys the low latency of CSMA and the high channel utilization of TDMA. But it does not address periodical convergecast traffics.

Our work is motivated by the observation that no existing protocol can simultaneously handle convergecast, event-reporting, and on-demand traffics well. We propose a TDMA-over-CSMA link protocol to address these needs. The underlying protocol is a CSMA-like protocol and we build on top of it a TDMA-like protocol. Nodes follow a collision-free schedule to transmit convergecast traffics, but contend for slots to transmit event-reporting traffics whenever needed to reduce transmission delay. An opportunistic slot reuse scheme is proposed to exploit spatial reuse and to transmit non-time-critical on-demand traffics. Our protocol can provide low transmission latencies for most types of traffics, facilitate data aggregation (and thus energy efficiency) for convergecast traffics, and help reduce data redundancy for event-reporting traffics.

The rest of this paper is organized as follows. Chapter 2 reviews related work. Our system is given in Chapter 3. The protocol details are presented in Chapter 4. Our implementation and simulation results are in Chapter 5 and Chapter 6, respectively. Chapter 7 concludes this paper.

Chapter 2

Related Works

Several MAC protocols [21, 13, 18, 15, 19, 16, 2, 5] have been developed for WSNs. S-MAC [21] proposes sleep schedules to conserve energy. By maintaining loose synchronization, nodes follow the periodical sleep/listen schedules to avoid idle listening. To improve energy-efficiency, T-MAC [18] extends S-MAC by shortening the listen period. B-MAC [13] allows nodes to follow their own sleep schedules independently by using a long preamble sampling mechanism to eliminate the synchronization overhead. The sender will send a preamble before sending data, and the receiver will periodically wake and sample the channel. Once the receiver listens to the preamble, it will keep awake until the end of the transmission.

TDMA-based protocols achieve energy-efficiency by providing collision-free transmissions, and have better performance than that of contention-based protocols in heavy load networks. In LMAC [19], nodes choose the time slot randomly according to the neighbor information in a distributed way. TDMA-W [5] allows nodes to be self-organize, and conserve energy by using a Wakeup slot to prior notify the receiver to wake and listen to the senders' Send slot. TRAMA [15] divides a time-slotted channel into random-access periods and scheduled-access periods. In a random-access period, nodes contend the slots to exchange the neighbor information. In scheduled-access period, nodes transmit collision-free data and propagate schedules. TRAMA prevents to allocate slot to a node which does not have data to send to conserve energy, and provides dynamic slot-reuse to improve the slot utilization.

Several hybrid MAC protocols which combine CSMA and TDMA mechanism were also developed. In Z-MAC [16], nodes in the low-contention areas will adopt a CSMA-based MAC protocol and those in the high-contention areas will adopt a TDMA-based MAC protocol. Thus, Z-MAC takes the advantages of low latency of CSMA and high channel utilization of TDMA. Funneling-MAC [2] was proposed to solve the funneling effect problem. Localized TDMA is performed in the region near to the sink to avoid collision.

All of above protocols do not consider the characteristics of difference traffics of WSNs explicitly. Some protocols [11, 6, 12, 14] have been developed for data aggregation application. DMAC [12] proposed a *staggered* wakeup/sleep schedule to reduce transmission latency. Through the data gathering tree, child nodes always can transmit packets before its parent. [11] follows the same concept of DMAC to allocate time slots to per aggregation path. FLAMA [14] is a variation of TRAMA which is dedicated to data gathering application. These three protocols do not consider event-reporting and on-demand traffics.

The authors of [9] proposed some TDMA algorithms to handle three common traffic patterns: broadcast, convergecast, and local gossip in sensor networks. Nodes are self-stabilizing according to the neighbor information. The sink locates at the left-top position of the network. When a node receives the message from its left or top neighbor, it will determine its slots. For convergecast traffics, the uplink delay (nodes to the sink) will be reduced according to the slot allocation. For local gossip traffics, each node maintains two slots in a TDMA cycle to reduce bi-directional (uplink/downlink) delay, and the slot allocation approach can also be used to reduce broadcast delay. However, the TDMA algorithms in [9] are designed for a rectangular or a hexagonal grid topology, and they cannot be adapted to the topology over which the nodes are uniform distributed. Besides, the requirements of the event-reporting traffics do not be considered in [9].

Several MAC protocols [7, 20, 10] have been proposed for event-driven networks. In [7], the authors observe that the data readings which are sensed by the nodes in the event area are highly correlated. Thus, it is unnecessary that all sensors report the event in order. CC-MAC [20] also takes the spatial correlation into consideration. CC-MAC consists of two components: E-MAC which aims to filter the correlation records and N-MAC which is used for sensors not in the event area to forward reporting packets. LLMAC [10] is also a MAC protocol used in event-driven WSNs. The authors argue that nodes should have different priorities to use medium in event-driven WSNs.

As far as we know, no existing protocol can simultaneously handle convergecast, event-reporting, and on-demand traffics well. Thus, one contribution of this work is proposing a MAC protocol that can support different traffic types simultaneously.

Chapter 3

System Model

This work considers three types of traffics with the following criteria.

- Convergecast: This is to monitor the environment by asking sensor nodes to periodically report their readings. However, under emergency events, such reports can be temporarily suspended.
- Event-reporting traffic: It is to report unusual events to the sink. Such traffics are sporadic but time-critical.
- On-demand traffic: These include broadcast, local gossip, query, and other control packets. They are sporadic but not time-critical.

In order to handle these types of traffics, we propose a TDMA-over-CSMA approach. Time is divided into frames, each led by a CSMA period followed by k cycles. Each cycle contains some time slots, which has a convergecast period and an event-reporting period. Fig. 3.1 shows the frame structure. The CSMA period is for transmitting control and management packets, such as slot assignment, synchronization, and hello packets. Typical backoff and contention mechanisms will be adopted during such periods. However, during the convergecast and the event-reporting periods, a TDMA-like mechanism is adopted. Note that the slots in the convergecast period and the event-reporting period can be permuted so that the slots belonging to the event-reporting period can be inserted into the convergecast period evenly to achieve low event-reporting latency.

To handle convergecast traffics, we will develop a convergecast scheme to determine the length of the convergecast period and to allocate slots to nodes. A tree will be formed from the network with the sink as the root. Our approach will facilitate data aggregation and take reporting latency into account.

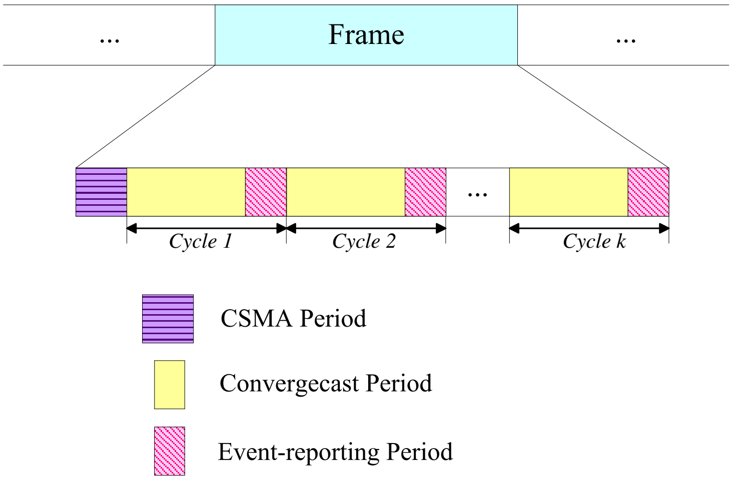


Figure 3.1: The frame structure of our protocol.

To handle event-reporting traffics, we will develop an allocation scheme to assign one slot to each node. Our design intentionally allows a node's slot to be the same as its one-hop neighbors'. So a slot may not be exclusively used by a particular node. When a node detects an emergency event, it can perform contention-based channel access to use any of such slots to quickly report the event. Also, because of the contention behavior, redundant reports of the same event can be reduced.

It remains to discuss the slots for on-demand traffics. In fact, these traffics do not have dedicated slots. We will develop an opportunistic slot-reuse scheme to identify some opportunistic slots from the convergecast period. Note that such slots exist due to the way that we allocate convergecast slots. Nodes will contend for such slots based on their traffic loads. A node can claim or disclaim such a slot during the CSMA period, and exchange its schedule with its neighbors.

To summarize, we categorize traffics into convergecast, event-reporting, and on-demand ones. A TDMA-over-CSMA link protocol is proposed to support these traffics. Fig. 3.2 shows the period and slot allocation. Each slot is for one packet transmission. For each slot, we give the highest priority to the type of traffic designed for that slot. However, if there is no such traffics, the other types of traffics can use that slot. Finally, our protocol allows nodes going to sleep for some slots in which they do not transmit/receive packets.

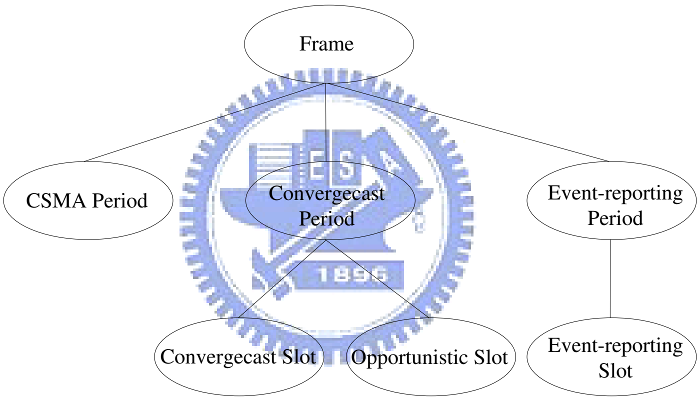


Figure 3.2: Classification of slots.

Chapter 4

Protocol Details

4.1 Convergecast Scheme

This scheme will assign one convergecast slot to each node. The result will also determine the length of the convergecast period. Here, we will apply and modify the scheme in [17]. The main differences from [17] are as follows. First, the slot selection in [17] is receiver-based, while ours is transmitter-based. Second, we do not consider slot reuse because later on we will still exploit slot reuse for on-demand traffics (opportunistic slots).

The scheme works as follows.

1. First, a BFS tree rooted from the sink is constructed from the network. Then we traverse nodes of the tree in a bottom-up manner. For each node v being visited, we will compute a number $n(v)$ for v as follows:
    (a) If v is a leaf node, let $n(v)$ be the smallest positive number that has not been used by any of v's one-hop and two-hop neighbors.
    (b) If v is an internal node. Let $max(v)$ be the maximum number used by v's child nodes. We set $n(v)$ to be a smallest positive integer $p > max(v)$, and p has not been used by any of v's one-hop and two-hop neighbors.
2. We then traverse nodes of the tree in a top-down manner. For each node v being visited, we try to increase the value of $n(v)$ such that $n(v)$ is less than the value used by v's parent and $n(v)$ has not been used by any of v's one-hop and two-hop neighbors.
3. Let $m_1 = \min_{\forall v}\{n(v)\}$ and $m_2 = \max_{\forall v}\{n(v)\}$. The length of the convergecast period will be $m_2 - m_1 + 1$ slots. Also, for each node v, we let $s(v) = n(v) - m_1 + 1$ be the convergecast slot of v.

An example is shown in Fig. 4.1. Fig. 4.1(a) shows the result of the bottom-up manner. Node V_0 is the root. The BFS sequence of the bottom-up manner is V_4, V_9, V_8, V_3, V_7, V_6, V_5, V_2, V_1. We follow the BFS sequence to traverse the tree nodes, and choose slot 1 for node V_4. Fig. 4.1(b) is the result of the top-down manner, node V_3 is assigned to a new slot 4 and V_4 is assigned to slot 5. We get that m_1 = 2, and m_2 = 9. We assign each node to a new slot by the rule of step 3). The final result is shown in Fig. 4.1(c), and the length of the convergecast period is 8 slots.

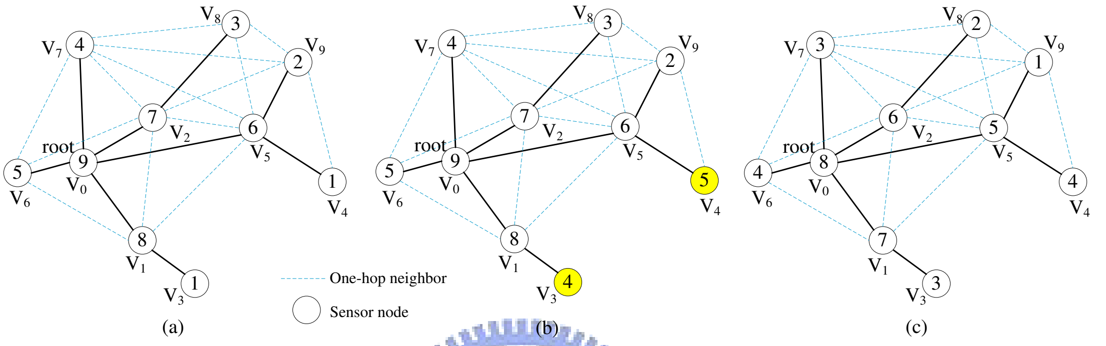


Figure 4.1: (a) The result of the bottom-up manner. (b) After the top-down manner, V_3 and V_4 are assigned to new slots. (c) The final result of the slot assignment scheme.

4.2 Event-reporting Scheme

This scheme consists of two parts. First, we will propose a distributed slot assignment algorithm to allocate an event-reporting slot to each node. Then we will show how nodes use these slots. The result will also determine the length of the event-reporting period.

The slot assignment algorithm works as follows. We allow a node to select a slot which is used by its one-hop neighbor, but not by its two-hop neighbors. Allowing a node to share the same slot with its one-hop neighbors is to reduce redundant reports of the same event. Not allowing a node to share the same slot with its two-hop neighbors is to avoid the hidden terminal problem. Since neighboring nodes may share the same slot, backoff is needed to access an event-reporting slot. Also, a loser may decide to delete its event-reporting packets if it overhears its neighbor's reports with high similarity.

The algorithm works as follows. We will use two control messages: *Request* and *Grant*. Each node x has to obtain a slot. To do so, x will broadcast a *Request* to its two-hop neighbors. On any y receiving x's *Request*, y will act as follows. If y does not own a slot and y.ID $<$ x.ID, y will reply a *Grant* to x with a null slot information.

Otherwise, y will do nothing. Once x receives *Grants* from all its two-hop neighbors, it will choose its slot as follows. It will first try to find the most used slot from the set of slots which are reserved by its one-hop neighbors but not reserved by any of its two-hop neighbors. If there is no such a slot, x will select the smallest slot which has not been used by its two-hop neighbors. After selecting its own slot, x will broadcast a *Grant* to all its two-hop neighbors carrying its slot information. At the end, each node will obtain an event-reporting slot and the maximum slot number used in the network will be the length of the event-reporting period.

Below, we will show how to use these event-reporting slots. Each node has two modes: *ES mode* (event-source) and *NES mode* (non-event-source). Initially, all nodes will stay in the NES mode. Once detecting an event, nodes that detect the event will enter the ES mode and compete on their event-reporting slots to report the event. A node in the ES mode overhearing an event-reporting packet will check its buffered packets and delete those packets with high similarity as the overheard packet. On the other hand, nodes in the NES mode will help to relay these event-reporting packets in any of the event-reporting slots. When a node in the ES mode does not have any event-reporting packet in its buffer, it will return to the NES mode in the next frame. Intuitively, in the event area, nodes will be partitioned into several sub-areas, each sub-area contains some neighboring nodes sharing the same event-reporting slot. From each sub-area, only few packets are expected to report the same event. Further, reports from different sub-areas are expected to leave the event area at different time slots (because two-hop neighbors should not share the same slot) and thus form "pipeline-like" flows leaving for the sink. An example is shown in Fig. 4.2.

In Fig. 4.2, node S is the sink. The number besides a node is its event-reporting slot. Nodes in the event area (A-G) turn to the ES mode. The event area is separated into three sub-areas. The nodes who have the same event-reporting slot will group into a sub-area. First, we illustrate the behavior of reporting an event. Node A and B have the same event-reporting slot, so they will contend slot 1 to transmit. Assume B is the winner, and A will suspend its transmission according to the overheard packet from B. The event-reporting packet will be routed through node H, I, J, and K to the sink at any event-reporting slot or the routing nodes' convergecast slots when there is no convergecast packet. Then, node B is the winner of sub-area 1, and the event-reporting packet detected by node B will be sent out of the event area at slot 1 in cycle i. Assume the winners of sub-area 2 and 3 are node D and G. D sends its event-reporting packet at slot 2 and G at slot 3 in cycle i, then the pipeline effect is formed. The collision which is caused by transmitting the event-reporting packets at any of the event-reporting slots in the non-event area will be

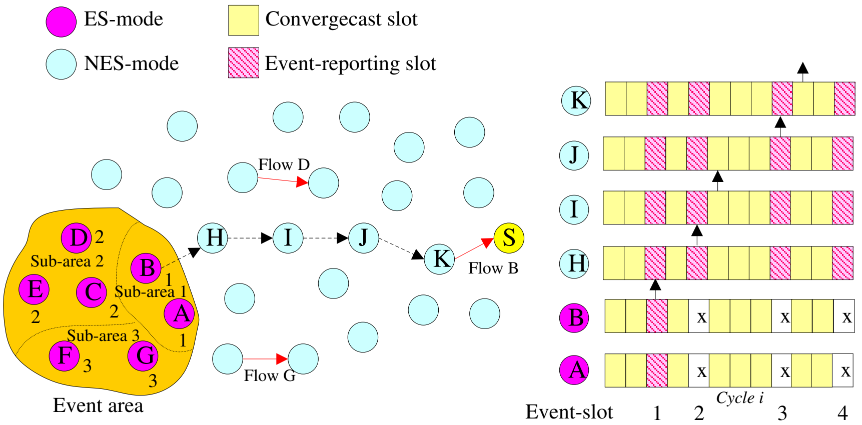


Figure 4.2: An example of an emergency event reporting and pipeline effect.

reduced through the separated flows by pipeline effect.

4.3 Opportunistic Slot-reuse Scheme

As we can see in Fig. 4.1(c), the convergecast slots which are used by V_8 and V_9 are available to be reused by V_3, because V_3 is more than two hops away from V_8 and V_9. Thus, the scheme is proposed to reuse the free convergecast slots to transmit on-demand traffics. This scheme consists of two parts. First, each node will be assigned original opportunistic slots. For a node, its opportunistic slots are fixed and thus inflexible. Thus, in the second part, we propose a *dynamic borrowing rule* to reuse the slots. Based on the traffic load, nodes can perform borrowing and returning processes to dynamically reuse the slots.

Now, we present how the first subscheme works. Motivated by [15], each node x will be assigned a priority value for each slot s in the convergecast period by a hash function $hash(s, x)$. Besides, we also define a Boolean function called $noC(s, x)$. If any of x's one-hop and two-hop neighbors is in its convergecast slot at slot s, then $noC(s, x)$ is FALSE; otherwise, $noC(s, x)$ is TRUE. For each slot s and a node x, if $noC(s, x)$ is TRUE and $hash(s, x)$ is the highest among x's one-hop and two-hop neighbors, then slot s is one of opportunistic slots of x.

Then, we describe the dynamic borrowing mechanism. First, we classify nodes into three states, the *satisfied* nodes (S-node), the *redundant* nodes (R-node), the *insufficient* nodes (I-node). During the CSMA period, each node will determine its state based on the number of packets stored in its buffer.

The dynamic borrowing mechanism works as follows. In the CSMA period, an S-node will do nothing. An R-node will announce their discarded opportunistic slots as early as possible by transmitting a *Discard* packet. (Note that an R-node can retrieve the discarded opportunistic slots by transmitting the *Retrieve* packet freely and directly.) When an I-node overhears a discard packet, it needs to pass an accessing qualification first. If it can pass the accessing qualification, then it will send a *Claim* packet using a random backoff mechanism to claim the discarded slots. (Note that according the rules for checking accessing qualification, nodes that can pass the accessing qualification will form a fully connected subnetwork.)

The rules for checking accessing qualification is as follows. An I-node i has to check whether it can contend a discarded slot d via the following rules.

1. d is not used by any i's one-hop neighbor yet.
2. $noC(d, i)$ is TRUE.
3. $hash(d, i)$ is the highest among i's two-hop neighbors. (Note that i does not need to check its one-hop neighbors.)

By this dynamic borrowing mechanisms, we believe that nodes can reuse the slots more flexibly when they transmit the on-demand traffics.

4.4 Energy-Efficiency

In WSNs, energy consumption is a significant issue because alternating the batteries of the sensor nodes frequently is inconvenience especially in an unfriendly area. A power-saving mechanism is proposed as follows.

A node needs to wake up to listen to the channel for a short period in the beginning of (1) the convergecast slots of its child nodes, (2) all the event-reporting slots, and (3) the opportunistic slots of its one-hop neighbors, otherwise, it does not need to wake up. During these slots, if there is no packet transmitted, the node can go to sleep. We guarantee that each node can receive the packets transmitted from it neighbors and idle listening will be reduced according to these rules.

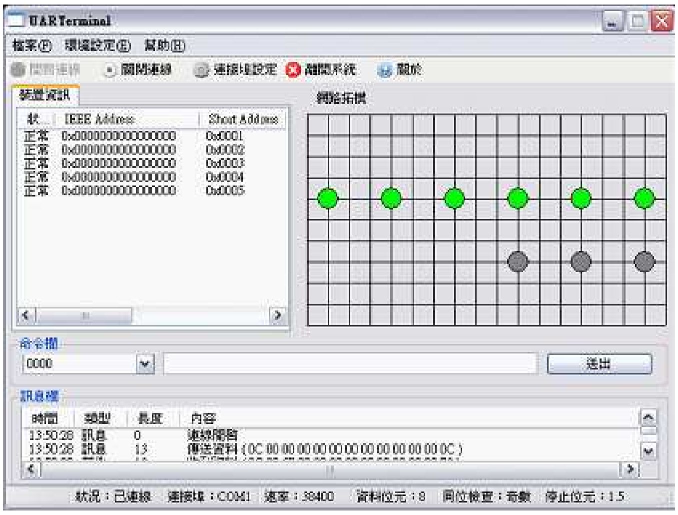
There is a trade-off between the transmission delay and energy consumption. Nodes should keep sensitive by waking up to listen to the channel frequently to achieve low transmission delay, such that a lot of energy will be consumed. In our protocol, to achieve low transmission delay is our first goal.

Chapter 5

Implementation

We implement a simple TDMA-based link protocol on Jennic JN5121 [1] sensor board to ensure that a TDMA protocol can be implemented. The MAC layer of Jennic platform is IEEE 802.15.4/ZigBee protocol. We develop our simple version protocol based on the 802.15.4 stack API. We define the packet size to 128 bytes. Slot size is set to 1 second in our implementation.

The network is under a chain topology. Nodes (end device) periodically gather data from the environment and send to the coordinator (sink). The coordinator is connected to the PC by RS-232 serial port and UART. Through UART, the coordinator can report the data to PC, and the user can use the user interface (UI) to send command to the coordinator, then the coordinator and end devices communicate with RF. Fig. 5.1 shows the user interface, and figure 5.2 and 5.3 show the environment of demonstration.



狀	IEEE Address	Short Address
正常	0x0000000000000000	0x0001
正常	0x0000000000000000	0x0002
正常	0x0000000000000000	0x0003
正常	0x0000000000000000	0x0004
正常	0x0000000000000000	0x0005

命令欄: 0000

時間	類型	長度	內容
13:50:28	訊息	0	連線開啟
13:50:28	訊息	13	傳送資料 (0C 00 00 00 00 00 00 00 00 00 00 00 0C)

狀況：已連線 連接埠：COM1 速率：38400 資料位元：8 同位檢查：奇數 停止位元：1.5

Figure 5.1: User interface.

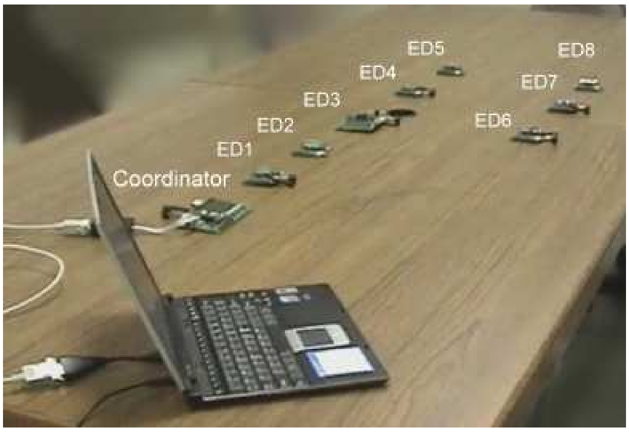


Figure 5.2: The environment of demonstration.

Figure 5.3: The environment of demonstration.

We assign time slots to the nodes according to an order way to reduce transmission delay. A node will use the slot which is different from that of its interference neighbors (We assume the interference range is 2-hops). A node should turn on the radio at the slots which is used by its child node and itself, and the synchronization slot, otherwise, it can turn off the radio to achieve power saving. The coordinator will send a synchronization packet to synchronize the network nodes at the synchronization slot. Fig. 5.4 is the remaining energy (Volt) of power-saving mode and non-power saving mode. The testing node sent packet per 2.5 seconds. When the energy of the battery is lower than 2.1 V, the node can not work in normal status (the node has no ability to work).

We measure the variation of energy (Am) in a time interval to observe the radio on or off. Fig. 5.5 shows the energy consumption of node 3 whose slot is 4 and its child node's slot is 3. The synchronization slot which is owned by the coordinator is set to slot 1. Fig. 5.6 is the network topology and slot allocation. The length of a TDMA cycle is 5 slots (seconds). We can observe that the node turns on the radio at slot 1, 3, 4, and turn off at slot 2 and 5 according to the Am readings of Fig. 5.5.

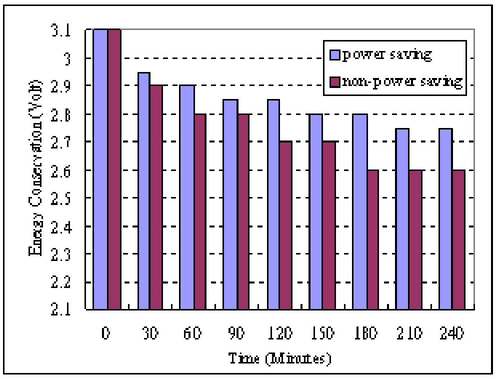


Figure 5.4: The remaining energy (Volt) of power-saving mode and non-power saving mode.

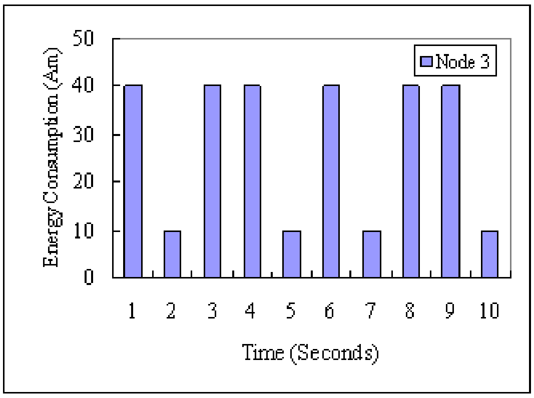


Figure 5.5: The energy conservation (Am) of node 3.

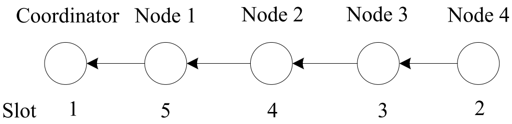


Figure 5.6: The network topology and slot allocation of Fig. 5.5.

Chapter 6

Simulation Results

We develop a C-based simulator to evaluate our protocol. We compare our protocol against Carrier Sense Multiple Access (CSMA) and Time Division Multiple Access (TDMA) schemes through simulations. In the TDMA scheme, a random slot assignment is used to assign slots to the nodes so that a node will own a slot different from its one-hop and two-hop neighbors.

6.1 Protocol parameters

N (up to 500) nodes are uniformly distributed over 256×256 units sensing area in all experiments. The transmission range of each node is 30 units. We construct a BFS tree for convergecast and event-reporting traffics. The sink is set to node 0. The simulation parameters are shown in Table 6.1.

Table 6.1: Simulation parameters

Transmission range	30 units
Packet length	64 bytes
Bit rate	250 Kbps
Slot size	7.6 ms
k (number of cycles per frame)	16
Simulation time	3600 seconds

6.2 Traffic generation models

We consider three traffic types in this work. All the traffics are triggered at the beginning of the simulation. In our protocol, each node maintains three different buffers for convergecast, emergency event, and on-demand traffics. In the CSMA and TDMA schemes, each node only maintain an FIFO buffer. The traffic generation models are described as follows.

- Convergecast generation model: The convergecast event is periodically generated by the simulator through a parameter *CONVERGECAST_INTEVAL*. Each node senses the data, and encapsulates the data into packet and put in the buffer until its child nodes report. Each node should report the sensing data which is aggregated with its child nodes. However, the collision and buffer overflow problem will cause packet lost. In order not to wait for the child packets forever, we propose dropping rules for each protocol. In the CSMA scheme, a node should wait for a time interval based on the number of its descendant nodes. In our protocol and the TDMA scheme, a node should wait for a half of the convergecast generation interval. After timeout, the nodes will aggregate its data with the received packets which are sent by its child nodes, and then send to its parent node.
- Emergent event generation model: We give a parameter *MAX_EVENT_INTERVAL* to determine the maximum time interval between two emergency events. We randomly pick a point to be the event center of an event area at every time the emergency event triggered. Sensors in the event area should report the event to the sink.
- On-demand traffic: We use two models to generate on-demand traffics. One is broadcast generation model and the other is unicast generation model.
  - Broadcast generation model: We used a parameter *MAX_BROADCAST_INTERVAL* to determine the broadcast generation interval. The interval of two broadcast events is determined randomly with uniform distribution. 20% nodes will be randomly selected to be the source nodes to generate a broadcast event independently. The source nodes generate the broadcast packets at the same time, and flood the packets to the whole network. To avoid flooding problem, each node broadcast the same event only once, and old broadcast packet will be ignored after a node receiving or generating a newer event.
  - Unicast generation model: A parameter *MAX_UNICAST_INTERVAL* is used to determine the unicast generation interval. 10 pairs of nodes are randomly

Table 6.2: Simulation Cases

	Case 1	Case 2	Case 3	Case 4
CONVERGECAST_INTEVAL	10 s	10 s	180 s	180 s
MAX_EVENT_INTERVAL	20 s	20 s	350 s	350 s
MAX_BROADCAST_INTERVAL	84 s	630 s	84 s	600 s
MAX_UNICAST_INTERVAL	23 s	170 s	23 s	480 s

selected to be the source nodes and destination nodes every time. The unicast packets are routed by shortest paths.

In this simulation, we use four cases of different traffic rates to evaluate the performances. The parameters used in these four cases are shown in Table 6.2.

6.3 Performance metrics

We used five performance metrics to evaluate the performance. In the experiments, four traffic types will be run concurrently but statistics will be taken individually. These five metrics are as follows.

- **Success Rate**: It is the ratio of the number of packets received by the intended receiver to the number of packets transmitted by the sender per pair.

- **Average Delay**: For convergecast traffics, delay is calculated from the sensing time to the time at which the packet are received by the sink. For event-reporting traffics, delay is calculated from the detection time to the time at which reports are received by the sink. For broadcast, delay is calculated from the time at which a broadcast packet is generated to that at which the packet is received by whole network nodes. For unicast, delay is calculated from the time at which an unicast packet is generated to that at which the packet is received by the destination.

- **Convergecast Coverage**: It is defined as the ratio of the number of nodes whose data has been reported to the sink to the number of nodes generating the convergecast packets.

- **Event Coverage**: It is defined as the ratio of the area which is covered by the reported packets received by the sink to the original event area.

Table 6.3: The ranking of the importance of metrics.(more stars = more important)

	Success Rate (per pair)	Delay	Converge-cast Coverage	Event Coverage	Broadcast Coverage
Convergecast	★★	★★	★★★		
Event-reporting	★	★★★		★★★	
On-demand (Broadcast)		★			★★
On-demand (Unicast)	★	★			

- **Broadcast Coverage**: It is defined as the ratio of the number of broadcast packets actually received by the whole network nodes to the number of packets that should be received.

For each type of traffics, the importance levels of metrics are different. Table 6.3 shows the ranking of the importance of metrics for different traffic types. This table can be used to evaluate whether a protocol can fit all requirements of different traffic types.

6.4 Success Rate

Fig. 6.1(a), 6.2(a), 6.3(a) and 6.4(a) show the success rates of convergecast, event and unicast under our protocol, CSMA and TDMA schemes. (Our protocol is denoted by the CEO scheme.) CSMA has the worst performance in terms of success rates, because it is contention-based. The success rates of event-reporting in our protocol only achieves 90% because nodes in the NES mode will contend any of the event-reporting slots to send. However, for event-reporting traffics, we may tolerate some loss, because some reports may be redundant. The success rates of convergecast and unicast traffics for our protocol are almost 100% as TDMA scheme which is for collision-free transmission.

6.5 Delay

CSMA scheme has the best performance in terms of delay because when the medium is idle, each node who has packet in the buffer can contend the medium immediately. In our protocol and TDMA scheme, nodes that intend to transmit packets have to wait until their own slots arrive.

Fig. 6.1(b), 6.2(b), 6.3(b) and 6.4(b) show the average delay of convergecast and event-reporting. The convergecast delay for our protocol is almost as low as CSMA scheme because the convergecast algorithm allocates slots in a specific order where the slot assigned to the parent node will not be far from those used by its child nodes. The event-reporting delay for our protocol is longer than the convergecast delay because the event-reporting slots are inserted into the convergecast slots. Because the slot assignment algorithm used in TDMA scheme does not take traffic types into consideration. TDMA has the worst performance. For example, the convergecast packet may be blocked by other packets generated early

Fig. 6.1(c), 6.2(c), 6.3(c) and 6.4(c) show the average delay of on-demand traffics. The broadcast delay and unicast delay of our protocol is longer than that of TDMA scheme, because the on-demand traffics have lower priority than the convergecast and event-reporting traffics have in our protocol.

6.6 Convergecast, Event and Broadcast Coverage

Fig. 6.1(d), 6.2(d), 6.3(d) and 6.4(d) show the convergecast coverage for the different traffic rates. Fig. 6.1(d) shows the results of case 1. When the network density increases to 500 nodes, the convergecast coverage for our protocol still keeps 100% because the convergecast packets will be sent in low latency according to the convergecast algorithm, such that the convergecast packets will not be blocked in the buffer. In TDMA scheme, we can see that the convergecast coverage decreases obviously when the network density increases, because the convergecast packets might be blocked by other types of packets in the FIFO buffer, such that the convergecast packets will be dropped when timeout triggers. Fig. 6.2(d) shows the same results, since the convergecast reporting rate is high. In CSMA scheme, the collision problem causes poor convergecast coverage. Our protocol achieves the best convergecast coverage than other protocols.

Fig. 6.1(e), 6.2(e), 6.3(e) and 6.4(e) show the event coverage. We can see that TDMA scheme achieves the best results because of collision-free transmission, all the event-reporting packets can be received by the sink. However, we should consider the data redundancy of the event-reporting packets. We do not allow all the sensor nodes to report the same event to the sink, so the nodes in our protocol will be allowed to contend the event-reporting slots to reduce packets. The results of the event coverage for our protocol almost achieve 90%.

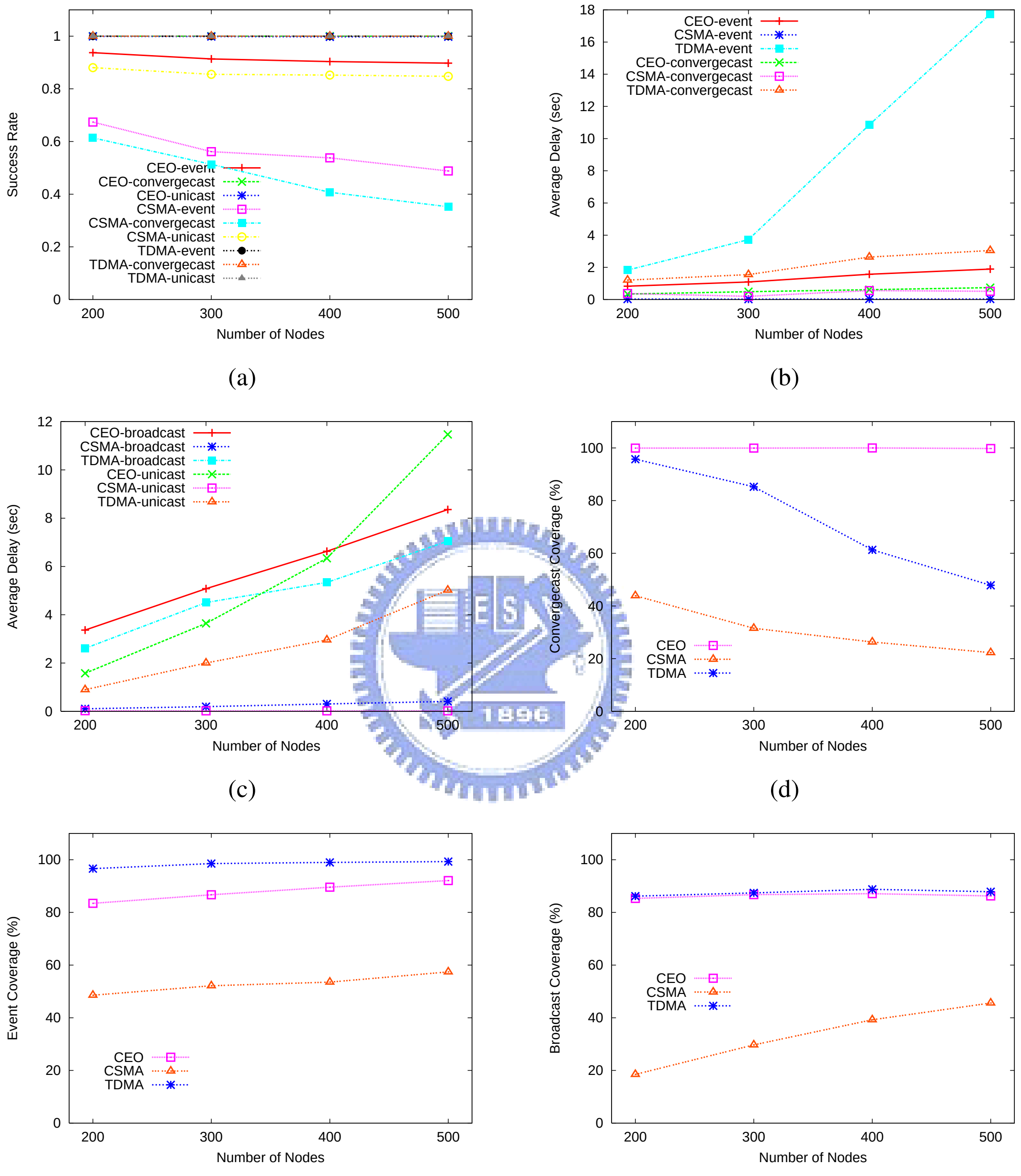
Fig. 6.1(f), 6.2(f), 6.3(f) and 6.4(f) show the broadcast coverage. The broadcast coverage for CSMA scheme increases when the network density increases. The reason is that the number of a node's one-hop neighbors will increase in high density network, such that fewer rebroadcast packets are needed to cover the nodes which need to receive the

Table 6.4: Performances for the three protocols

	Our Protocol	CSMA	TDMA
Success Rate (Convergecast)	★★★	★	★★★
Success Rate (Event-reporting)	★★	★	★★★
Success Rate (Unicast)	★★★	★	★★★
Delay (Convergecast)	★★	★★★	★
Delay (Event-reporting)	★★	★★★	★
Delay (Broadcast)	★	★★★	★
Delay (Unicast)	★	★★★	★★
Convergecast Coverage	★★★	★	★★
Event Coverage	★★	★	★★★
Broadcast Coverage	★★	★	★★

broadcast information. The broadcast coverage of our protocol is similar to that of TDMA scheme, and it does not achieve 100% because some of the older broadcast packets will be dropped by a node when the node receives the newer packet.

Finally, the comparisons of performances for our protocol, CSMA, and TDMA schemes are shown in Table 6.4. Our protocol combines the advantages of CSMA and TDMA for low transmission delay and high success rate. It is suitable for the applications in which both of the transmission delay and the completeness of the reporting data are important. Our protocol achieves better performance in high density networks. CSMA scheme achieves low transmission delay but the collision problem is significant. TDMA scheme achieves collision-free transmission but the delay is significantly high in high density networks.



(a) (b)

(c) (d)

(e) (f)

Figure 6.1: The results of case 1. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage

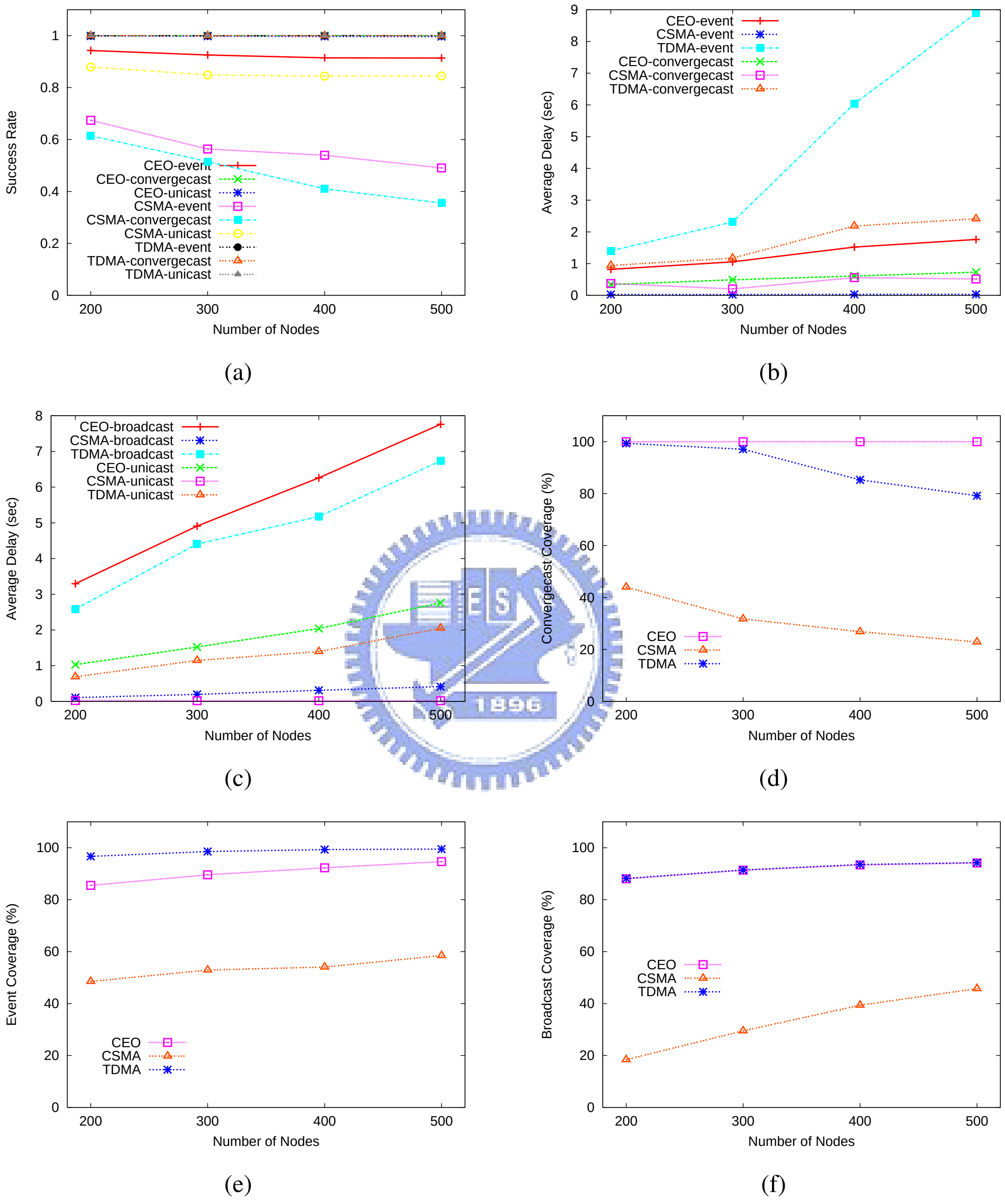
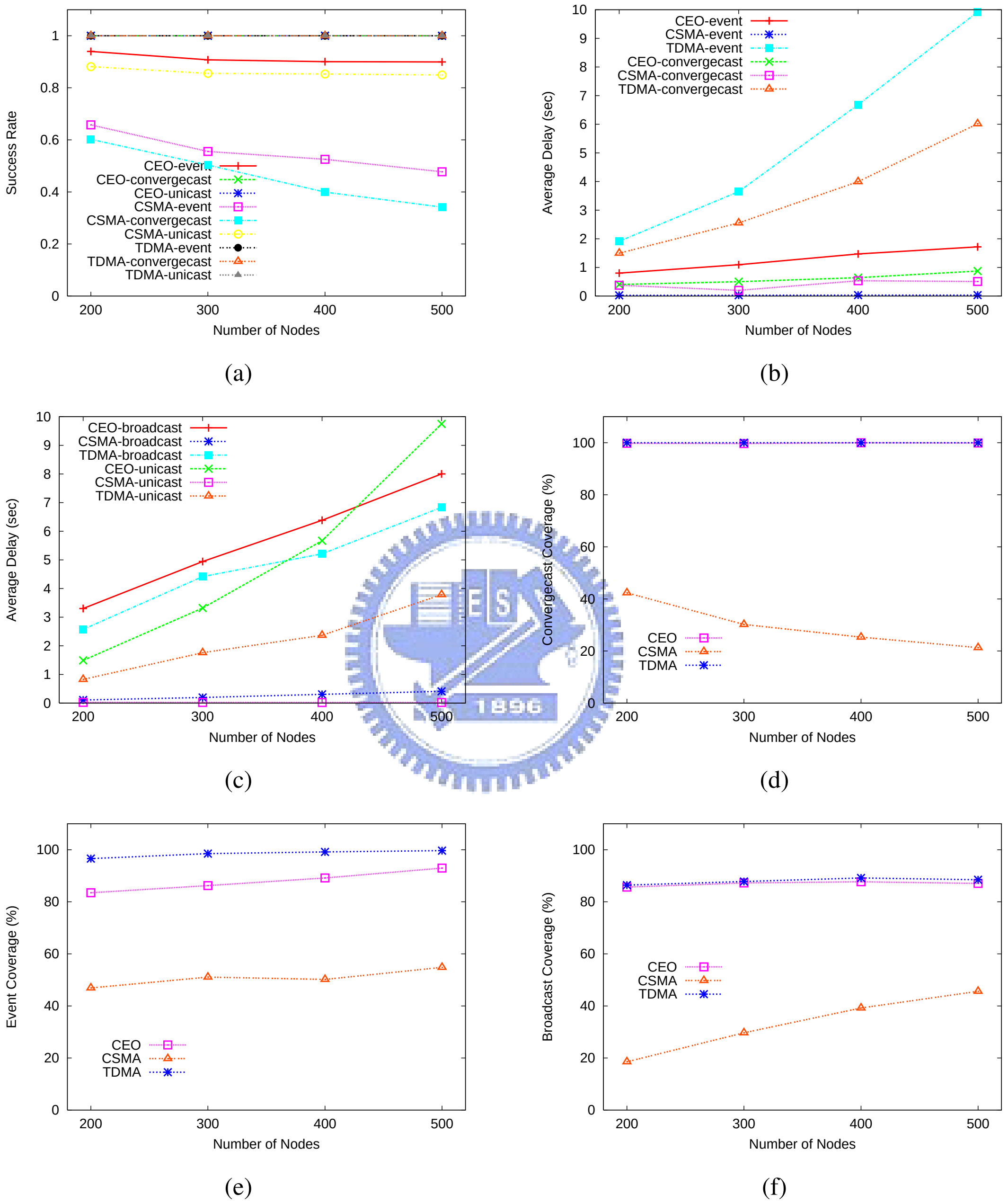


Figure 6.2: The results of case 2. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage



(a)

(b)

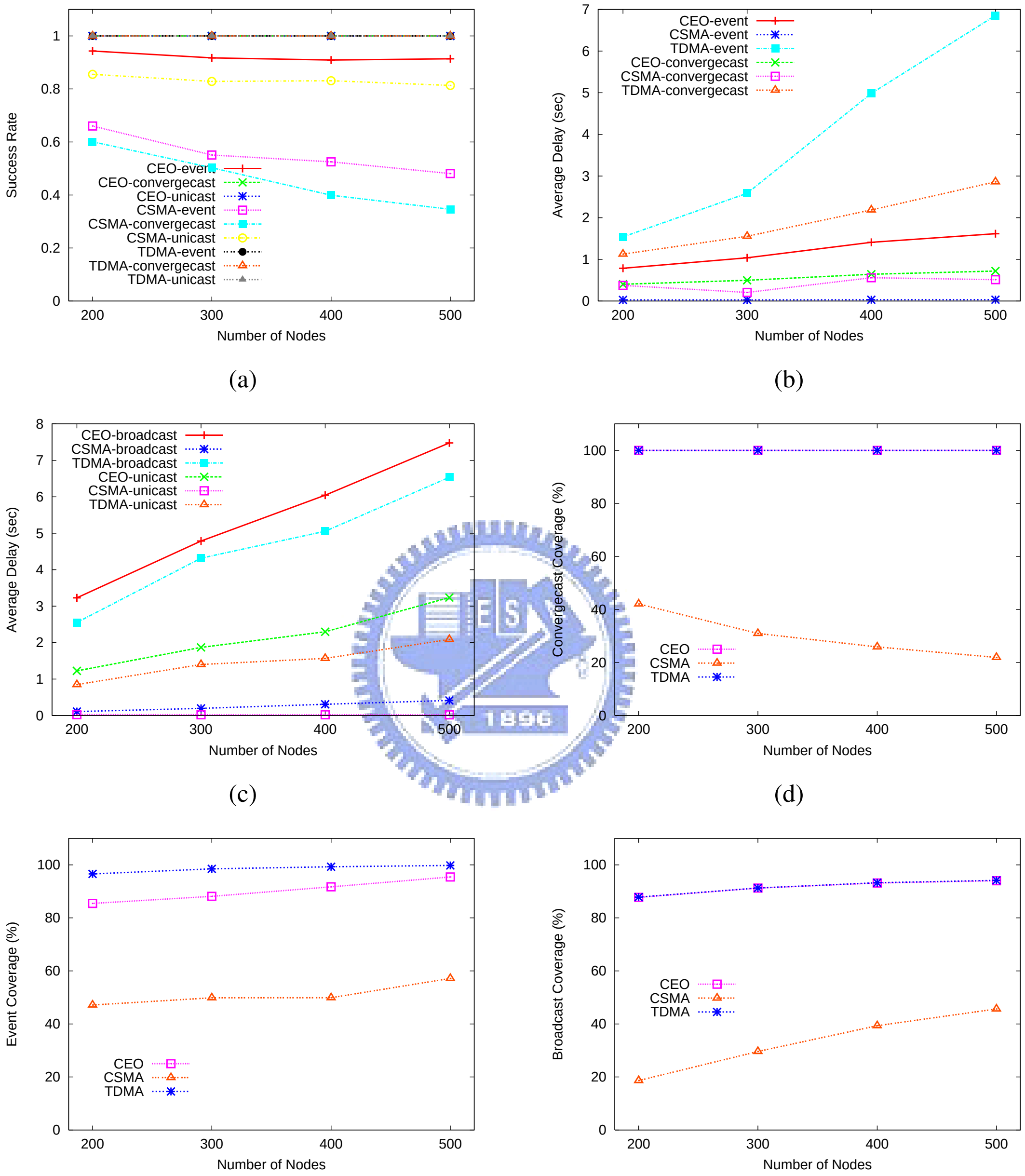
(c)

(d)

(e)

(f)

Figure 6.3: The results of case 3. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage



(a) (b)

(c) (d)

(e) (f)

Figure 6.4: The results of case 4. (a) Success rate (b) Average delay of convergecast and event-reporting (c) Average delay of broadcast and unicast (d) Convergecast coverage (e) Event coverage (f) Broadcast coverage

Chapter 7

Conclusions

In this paper, we classify the traffic types of WSNs into periodical traffics and sporadic traffics, and propose a TDMA-over-CSMA link layer protocol which can support multi-type traffics simultaneously. The proposed protocol takes the characteristics of different traffic types into consideration. A scheduled slot assignment approach is adopted to deal with the convergecast traffics. To support the emergent event reporting, a distributed slot assignment algorithm is proposed to reduce the reports of an event. Besides, an opportunistic slot-reuse scheme is proposed to exploit spatial reuse dynamically. Simulation results also demonstrated the efficiency of our protocol. As far as we know, this is the first MAC protocol that can be used to support multi-type traffics simultaneously.

Bibliography

[1] Jennic, JN5121. http://www.jennic.com/.

[2] G.-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo. Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks. In *Proceedings of the 4th international conference on embedded networked sensor systems (SenSys)*, pages 293–306, Boulder, Colorado, USA, 2006.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *IEEE Computer*, 38(4):393–422, 2002.

[4] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[5] Z. Chen and A. Khokhar. Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks. In *Proceedings of IEEE SECON*, pages 335–341, 2004.

[6] B. Hohlt, L. Doherty, and E. Brewer. Flexible power scheduling for sensor networks. In *Proceedings of the 3rd international symposium on information processing in sensor networks (IPSN)*, pages 205–214, Berkeley, California, USA, 2004.

[7] K. Jamieson, H. Balakrishnan, and Y. Tay. Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks. In *Proceedings of the 3rd European Workshop on Wireless Sensor Networks*, pages 260–275, Zurich, Switzerland, Feb. 2006.

[8] K. Kredo, II, and P. Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961–994, 2007.

[9] S. S. Kulkarni and M. Arumugam. TDMA services for Sensor Networks. In *Proceedings of 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, pages 604–609, 2004.

[10] H.-C. Le, H. Guyennet, V. Felea, and N. Zerhouni. A Low Latency MAC Scheme for Event-Driven Wireless Sensor Networks. *Mobile Ad-Hoc and Sensor Networks (MSN)*, 4864 (LNCS):291–301, 2007.

[11] W. L. Lee, A. Datta, and R. Cardell-Oliver. FlexiTP: A Flexible-Schedule-Based TDMA Protocol for Fault-Tolerant and Energy-Efficient Wireless Sensor Networks. *IEEE Ttansactions on parallel and distributed systems*, 19(4), 2008.

[12] G. Lu, B. Krishnamachari, and C. S. Raghavendra. An adaptive energy-efficient and low-latency mac for tree-based data gathering in sensor networks: Research articles. *Wireless Communication and Mobile Computing*, 7(7):863–875, 2007.

[13] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd international conference on embedded networked sensor systems (SenSys)*, pages 95–107, Baltimore, MD, Nov. 2004.

[14] V. Rajendran, J. Garcia-Luna-Aceves, and K. Obraczka. Energy-Efficient, Application-Aware Medium Access for Sensor Networks. In *Proceedings of IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Washington, DC, Nov. 2005.

[15] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, Feb. 2006.

[16] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-MAC: a hybrid MAC for wireless sensor networks. In *Proceedings of the 3rd international conference on embedded networked sensor systems (SenSys)*, pages 90–101, New York, NY, USA, 2005.

[17] Y.-C. Tseng and M.-S. Pan. Quick convergecast in ZigBee/IEEE 802.15.4 tree-based wireless sensor networks. In *MobiWac '06: Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 60–66, Terromolinos, Spain, 2006.

[18] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC Protocol for wireless sensor networks. In *Proceedings of the 1st international conference on embedded networked sensor systems (SenSys)*, pages 171–180, Los Angeles, CA, Nov. 2003.

[19] L. van Hoesel and P. Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks. In *Proceedings of the International Conference on Networked Sensing Systems (INSS)*, Tokyo, Japan, June 2004.

[20] M. C. Vuran and I. F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(2):316–329, 2006.

[21] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. of IEEE Infocom*, pages 1567–1576, June 2002.

[22] M. Zhao, Z. Chen, and Z. Ge. QS-Sift: QoS and spatial correlation-based medium access control in wireless sensor networks. *International Journal of Sensor Networks*, 2(3/4):228–234.

[23] M. Zhao, Z. Chen, L. Zhang, and Z. Ge. HS-Sift: hybrid spatial correlation-based medium access control for event-driven sensor networks. *IET Communications*, 1(6):1126–1132, Dec. 2007.