University of Alberta

CLASS-BASED RATE DIFFERENTIATION IN WIRELESS SENSOR NETWORKS

by

Mansoureh Takaffoli

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2009
Edmonton, Alberta

Examining Committee

Ehab S. Elmallah, Computing Science

Walied Moussa, Mechanical Engineering

Chintha Tellambura, Electrical and Computer Engineering

Janelle Harms, Computing Science

Abstract

Many applications of wireless sensor networks (WSNs) require the sensor nodes of a network to belong to different priority classes where the nodes of a higher priority class enjoy higher data rates than nodes of a lower priority class. Practical design of such networks, however, faces challenges in satisfying the following basic design requirements:

a) the need to rely on the medium access control mechanisms provided by the IEEE 802.15.4 standard for low-rate wireless personal area networks,

b) the need to solve certain types of class size optimization problems to ensure adequate sensing coverage, and

c) the need to achieve good utilization of the available channels.

Unfortunately, the current version of the IEEE 802.15.4 does not provide adequate support for rate differentiation. Hence, many proposed solutions to the problem in the literature consider adding extensions to the standard.

In this thesis, we introduce some class size optimization problems as examples of coverage problems that may arise in designing a WSN. We then consider a method proposed in the literature for handling the rate differentiation problem. The method relies on modifying the CSMA-CA channel access mechanism of the IEEE standard. We use simulation to examine its performance and its applicability to solve some class size optimization problems. We next investigate the use of Time Division Multiple Access (TDMA) protocols in providing service differentiation among different classes of sensors. We show simple sufficient conditions for the existence of TDMA-based solutions to a class size feasibility problem.

Lastly, we consider the use of Guaranteed Time Slots (GTS) of the IEEE 802.15.4 standard in constructing TDMA schedules. We present a new algorithm that uses the GTS service to construct such schedules. The desired algorithm contains some optimization features. The obtained simulation results show the performance gain achieved by the algorithm.

Acknowledgements

I would like to gratefully acknowledge the supervision of my advisors, Prof. Ehab S. Elmallah and Prof. Walied Moussa, whom have been abundantly helpful and have assisted me in numerous ways. I specially thank them for their infinite patience.

I am grateful to all my friends, most importantly Ahmad, whom their love and support will be with me in whatever I pursue.

My final words go to my parents, without whom I would never have been able to achieve so much.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Symbols

APL	The Application Layer in the ZigBee Specification
BE	Backoff Exponent
BI	Beacon Interval
BO	MAC Beacon Order
CAP	Contention Access Period
CCA	Clear Channel Assessment
CFP	Contention Free Period
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
FFD	Full Function Device
GTS	Guaranteed Time Slot
LR-WPAN	Low Rate Personal Area Network
MAC	Medium Access Control
NB	Number of Backoff Periods
NWK	The Network Layer in the ZigBee Specification
PAN	Personal Area Network
PHY	The Physical Layer in the ZigBee Specification
RFD	Reduced Function Device
SD	Superframe Duration
SO	MAC Superframe Order
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

Chapter 1

Introduction

Many applications of wireless sensor networks (WSNs) require the design of a network where the sensor nodes belong to different priority classes. In such networks, nodes in higher priority classes are required to collect and send data more frequently than nodes in lower priority classes. Examples of such applications include health monitoring of human and mechanical systems. Supporting WSNs with different priority classes using the current IEEE 802.15.4 standard for low-rate wireless personal area networks (LR-WPANs) gives rise to some research problems.

In this chapter, we introduce some of such research problems. We also give an overview of the IEEE 802.15.4 standard, and discuss how such problems can be tackled.

1.1 Introduction

A wireless sensor network (WSN) consists of spatially distributed nodes where each node consists of one or more sensing devices, a radio transceiver, and a micro controller for data processing purposes. A sensor node is typically battery powered and often work in unsupervised environments. Typical applications of WSNs include environment and habitat monitoring, health care applications, home automation, and traffic control [30], to mention a few. In [6] the authors classify the applications of WSNs from the point of view of their communication requirements into two categories: a) event driven (e.g., fire monitoring applications) sensor network, and b) continuous monitoring sensor networks (e.g., ambient temperature monitoring). In event driven WSNs, sensor nodes do not send data until a certain event occurs.

In contrast, continuous monitoring networks sample and transmit data periodically.

Many continuous monitoring applications of WSNs require the design of a network where the sensor nodes belong to different priority classes. In such networks nodes with higher priority classes are required to collect and send data more frequently than nodes in lower priority classes. Examples of such applications include health monitoring of human and mechanical systems. For example, in [16] the authors discuss the use of wearable sensors to provide accurate measures of motor abilities after stroke. For instance, to determine the disability level of an arm, one needs to find the specific patterns of arm movement. The three movable joints of the hand, elbow, and shoulder need to be monitored. Since the amount of movement by these joints and their importance are different, three priority classes of sensing nodes may be needed (e.g., hand monitoring has the highest priority, and shoulder monitoring has the lowest priority).

To provide adequate sensing coverage in such class-based WSNs, the network designer may require each class to have some sufficient number of sensor nodes. Thus, depending on the exact class size requirement imposed by the designer, a particular class size optimization problem arises. One of the main objectives in this thesis is to explore how class-based data rate differentiation problems can be solved using different medium access control protocols. In section 1.2, we present some example optimization problems that we use throughout the thesis.

Providing rate differentiation among different classes of sensor nodes in the same WSN is a capability that can benefit from the services provided by the medium access control (MAC) layer. Currently, many researchers have proposed various MAC protocols for WSNs. Examples of such protocols appear in the surveys of [17] and [28].

The IEEE 802.15.4 work group has developed a standard for implementing low-rate wireless personal area networks (LR-WPANs). This standard has gained acceptance in industry for developing wireless sensor networks (WSNs). Today, many sensor nodes produced by different manufactures are compliant with the standard. In this thesis, we investigate extensions and methodologies that rely on the use of the IEEE standard (release 2006) in achieving class rate differentiation.

The rest of chapter is organized as follows. In section 1.2 we introduce some useful class size optimization problems. In section 1.3 we give an overview of some important aspects of the IEEE 802.15.4 standard. In section 1.4 we outline thesis contributions and organizations.

1.2 Examples of Class Size Optimization Problems

In this section, we present three class size optimization problems that may arise in designing WSNs with different priority classes. The formulation of our problems is driven by design requirements of networks used in health monitoring of human and mechanical equipment.

Throughout this section, we assume that the WSN under design uses a specified MAC protocol with a specified maximum data rate R bits per second (bps). All data collected from the sensor nodes are assumed to be routed to a sink node (also called the PAN coordinator). For simplicity of presenting the problems, we assume that the network under design is a star network where the sink node is at the center.

Problem P1 (class size feasibility problem). Given

- a specified number Q, $Q \geq 1$, of priority classes (class 1 is the highest priority),
- a specified data rate vector $\mathbf{r} = (r_1, r_2, ..., r_Q)$ where $r_1 \geq r_2 ... \geq r_Q$, and
- a specified class size distribution vector $\mathbf{n} = (n_1, n_2, ..., n_Q)$, where n_i is the number of class-i nodes, and $n = \sum_{i=1}^{Q} n_i$ is the total number of sensor nodes,

the feasibility problem asks whether the specified MAC protocol can support each class-i sensor node to transmit data at a rate $\geq r_i$ bps.

■

Problem P2 (class size rate-based maximization problem). Given

- a specified number Q, $Q \geq 1$, of priority classes,
- a specified data rate vector $\mathbf{r} = (r_1, r_2, ..., r_Q)$ where $r_1 \geq r_2 ... \geq r_Q$, and
- a specified ratio vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_Q)$, $\sum_{i=1}^{Q} \alpha_i = 1$, where the number of class-i nodes is $n_i = \alpha_i n$, and n is the total number of nodes in the network,

the maximization problem asks what is the maximum total number of nodes n that the network can support such that each class-i node can transmit data at rate $\geq r_i$.

■

Problem P3 (class size average-based maximization problem). Given

- a specified number Q, $Q \geq 1$, of priority classes,

- a specified ratio vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_Q)$, $\sum_{i=1}^{Q} \alpha_i = 1$, where the number of class-i nodes is $n_i = \alpha_i n$, and n is the total number of nodes in the network, and

- a specified target average data rate r_{target}, and target standard deviation σ_{target},

the maximization problem asks what is the maximum total number of nodes n that the network can support such that

a) there exists a data rate vector $\mathbf{r} = (r_1, r_2, ..., r_Q)$, $r_1 \geq r_2 ... \geq r_Q$, such that each class-i node can transmit at a data rate $\geq r_i$, and

b) the average received data rate over all sensor nodes $\overline{r} = \sum_{i=1}^{Q} \alpha_i r_i \geq r_{target}$ and the standard deviation $\sqrt{\sum_{i=1}^{Q} \alpha_i (\overline{r} - r_i)^2} \leq \sigma_{target}$.

■

We now draw the following remarks on the above problems.

1. Problem P1 is a core problem whose solution can be used as a subroutine to tackle problems P2 and P3 using search based algorithms, as discussed below.

2. To show that problem P2 reduces to problem P1 using a search based algorithm we consider the following method. We consider the space produced by setting $n_1 = 1, 2, 3, ..., n_{1,max}$ where $n_{1,max}$ is the maximum number of class-1 nodes that can be accommodated by the system. Assigning a value to n_1 implies values to n_2, ..., n_Q (since the ratio vector $\boldsymbol{\alpha}$ is given). Thus, for each possible setting of n_1, we can create an instance of the feasibility problem that can be handled using a subroutine that solves problem P1. We can then iterate over all points in the search space to find the maximum feasible number of nodes n.

3. Similar to the above remark, one may handle problem P3 by using a subroutine to solve problem P2 using a search based method. Here, the search space is made of some rate vectors where each rate vector $\mathbf{r} = (r_1, r_2, ..., r_Q)$ satisfies $r_1 \geq r_2 ... \geq r_Q$, and the constraints imposed by r_{target}, and σ_{target}. Each rate vector is a point in our search space for which we use a solver to problem P2 to find the maximum possible n. The algorithm iterates over all points in the search space and finds the largest accepted value n as a best-effort solution to the problem.

4. In Chapter 2, we investigate problem P1 when the IEEE 802.15.4 CSMA-CA protocol is used, and the topology of the given WSN is a star. The chapter presents examples of solving problem P1, and problem P2 by restricting the search space of the problems.

5. For networks where the star topology is not sufficient because of the distances between the nodes and the sink, we develop in Chapter 4 an algorithm for solving problem P1 in a multi-hop tree network. The algorithm uses a TDMA-based protocol. We assume, however, that the tree network is given as input.

1.3 An Overview of IEEE 802.15.4 Standard

In this section, we outline a number of basic aspects of the IEEE 802.15.4 standard (release 2006). The IEEE 802.15.4 is a standard for short-range (up to 100 meters) low-data-rate (up to 250 kbps) wireless personal area networks (LR-WPANs). In general, the standard defines the physical (PHY) layer and the medium access control (MAC) layers of the Internet protocol stack. The ZigBee specification [7], extends the IEEE 802.15.4 standard by specifying the network (NWK) layer and the application (APL) layer of the networking stack.

Devices in an IEEE 802.15.4 wireless network can either be *full function* devices (FFDs) or *reduced function* devices (RFDs). An FFD is capable of performing all the duties described in the standard. An RFD has limited capabilities. For example, an RFD can talk only with an FFD device. More detailed information is presented next.

1. **The PHY Layer.** The standard defines the physical layer in the following three frequency bands:

   - in the 868 MHz, one channel with 20 kbps rate is defined,
   - in the 915 MHz ISM band, 10 channels each with 40 kbps rate are defined,
   - in the 2.4 GHz ISM band, 16 channels each with 250 kbps are defined. Here, each symbol carries 4 bits.

   The PHY layer also defines a Clear Channel Assessment (CCA) procedure which enables a device to perform the required carrier sensing when it executes the CSMA-CA protocol. In addition, the PHY layer provides a link

quality indicator (LQI) for each received packet. The LQI has at least 8 levels and is reported to the MAC layer and is available to the NWK and the APL layers as well.

2. **Star Topologies.** The simplest type of network topologies supported by the standard is the star topology. Here, an FFD plays the role of a PAN coordinator to which other sensor nodes (the leaves in the star) associate. Among other functions, the PAN coordinator is responsible for choosing a PAN identifier, broadcasting beacon frames, accepting association requests and serving nodes with delay sensitive traffic. An example of star topology is depicted in Figure 1.1.

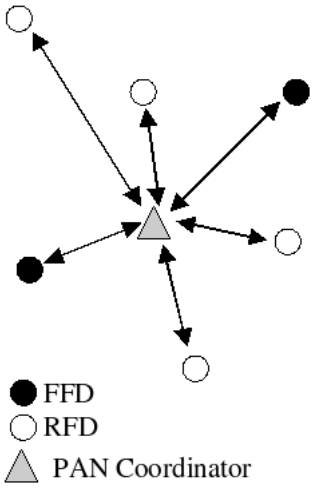

Figure 1.1: An example of star topology [19]

3. **The CSMA-CA Protocol.** The basic channel access method in the standard is the carrier sense multiple access with collision avoidance (CSMA-CA) method. An LR-WPAN can work in one of two modes: a *non-beacon-enabled* mode, or a *beacon-enabled* mode. In the non-beacon-enabled mode, the network uses unslotted CSMA-CA. In the beacon-enabled mode the network uses slotted CSMA-CA where the length of each slot equals to a *backoff period*. The length of a backoff period is 20 symbol duration (or 320 μs using the 250 kbps data rate). The beacon-enable mode offers a number of performance enhancements and, hence, it will be the only mode considered throughout the thesis.

4. **The Beacon-Enabled Mode.** When operating in a beacon-enabled mode,

the PAN coordinator can transmit beacon frames at regular intervals (ranging from 15 ms to 245 sec. [8]). The time between two beacon frames is called a *superframe* (or a *beacon interval*, denoted BI). Figure 1.2 illustrates a beacon interval. A beacon interval is composed of up to three types of periods: a Contention Access Period (CAP), a Contention Free Period (CFP), and an inactive period. The CAP and CFP constitute the active period of the superframe. The active period is divided into 16 equal time slots, independent of the duration of the superframe.

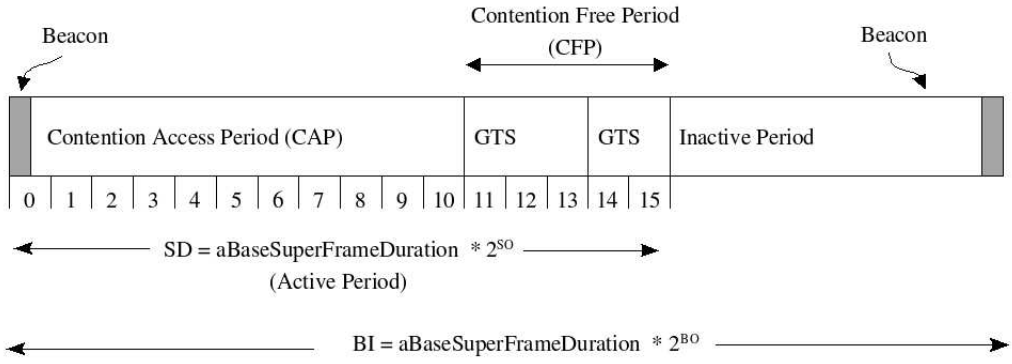

Figure 1.2: An example of the superframe structure [10]

Channel Access during the CAP is controlled by the slotted CSMA-CA protocol. Channel access during the CFP is contention-free, and is managed by the PAN coordinator through the allocation of Guaranteed Time Slots (GTS) to devices that require dedicated bandwidth or low latency transmissions. Nodes can enter low power (sleep) mode during the inactive periods. Beacon frames transmitted by the PAN coordinator contain critical information about the PAN including the beginning of the CFP, the duration of the beacon interval, and its active period. The following relations hold for a superframe.

a) The length of the superframe is determined by the parameter *macBeaconOrder* (BO):

$$BI = aBaseSuperFrameDuration \times 2^{BO}, 0 \leq BO \leq 14 \quad (1.1)$$

where $aBaseSuperframeDuration$ denotes the minimum duration of a superframe ($= 960$ symbols).

b) The length of the active part (or a *superframe duration*, denoted SD) is

determined by the *macSuperFrameOrder* (SO):

$$SD = aBaseSuperFrameDuration \times 2^{SO}, 0 \leq SO \leq BO \quad (1.2)$$

5. **Overview of the Slotted CSMA-CA Protocol.** The unit of time in the slotted CSMA-CA protocol is a backoff period that lasts 20 symbol duration (or 320 μs). The start of the first backoff period of all devices are aligned with the start of the PAN coordinator beacon's transmission. Each device maintains three variables: NB (number of backoff stages), BE (backoff exponent), and CW (contention window size expressed in terms of backoff slots). The algorithm works as follows. A device that has a packet ready for transmission performs the following steps:

   (a) The algorithms initializes $NB = 0$, $CW = 2$ backoff slots, and if the battery life extension feature is off, it sets $BE = macMinBE$ (= 3).

   (b) The node backs off for a random number of backoff slots, chosen uniformly between 0 and $2^{BE} - 1$, before sensing the channel in step (d).

   (c) The node verifies that the remaining time of the CAP of the current superframe is long enough to accommodate the time required to sense the channel, transmit the packet, and optionally wait for an ACK. If the remaining time is insufficient, the node waits until the next superframe.

   (d) In the following the node sense the channel CW (backoff slots) times:

      i. The node sense the channel by performing the CCA procedure.

      ii. If the channel is idle, CW is decremented by 1. If CW reaches zero, then the node transmits the packet.

      iii. Else if the channel is busy, CW is reset to 2, and NB and BE are incremented by one. BE cannot exceed $macMaxBE$ (= 5) value (i.e., it is frozen at $macMaxBE$ value). If NB (the backoff stages) reaches a specified maximum value the algorithm terminates with channel access failure that is reported to the higher protocol layers. Else (if $NB <$ the maximum retry value), the algorithm goes to step (b).

   The algorithm of the CSMA-CA is depicted in Figure 1.3.

6. **Peer-to-Peer Topologies.** Star topologies are not suitable for constructing large networks, or networks that cover large areas. To enable the construction

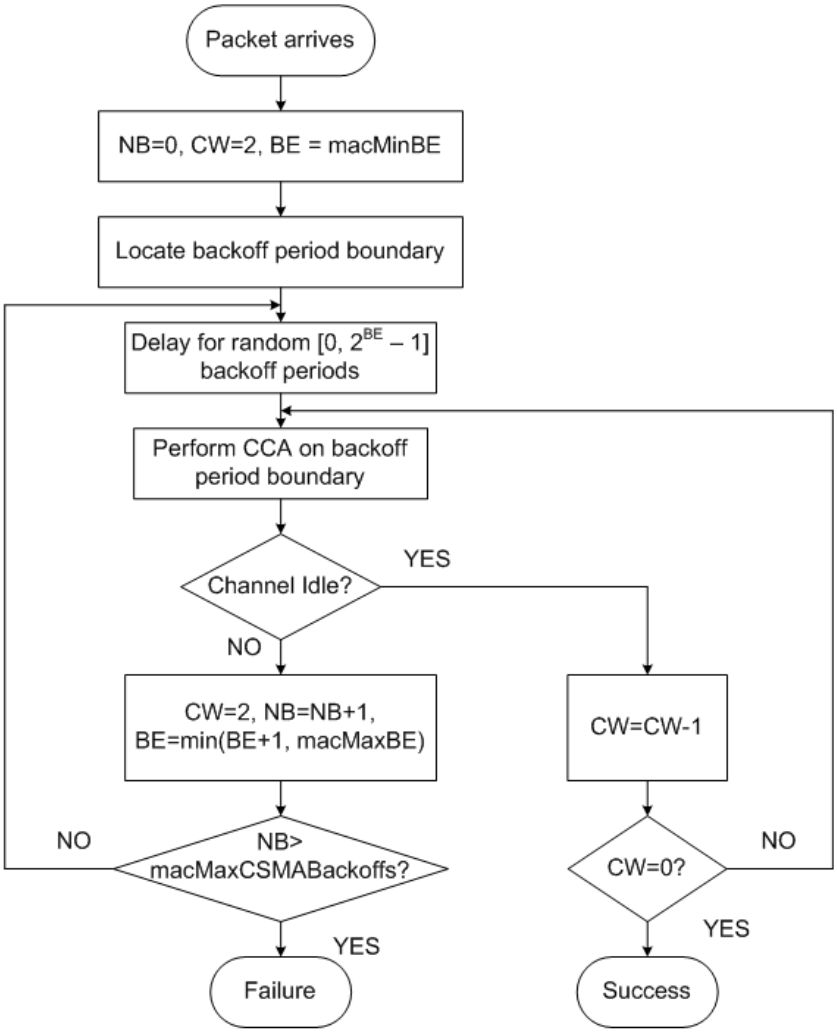

Figure 1.3: The slotted CSMA-CA algorithm [19]

of such networks, the standard supports peer-to-peer (p2p) topologies. A p2p network has a PAN coordinator and one or more coordinators. Each coordinator can have its superframe structure, subject to some length constraints (as will be discussed in Chapter 4). An example of p2p network is shown in Figure 1.4. Cluster-trees (see Figure 1.5) are a special type of p2p WPANs. In such networks, each cluster consists of a coordinator serving as a cluster head, and some leaf nodes.

We conclude with the following remarks:

1. The IEEE standard provides a limited support for implementing service differentiation among multiple classes of sensor nodes. This limitation has been

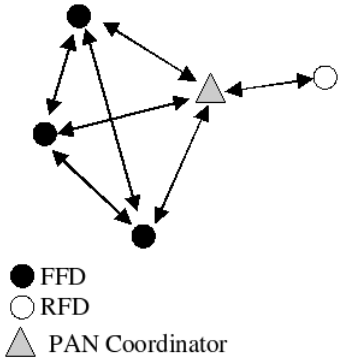

Figure 1.4: An example of peer-to-peer topology [19]

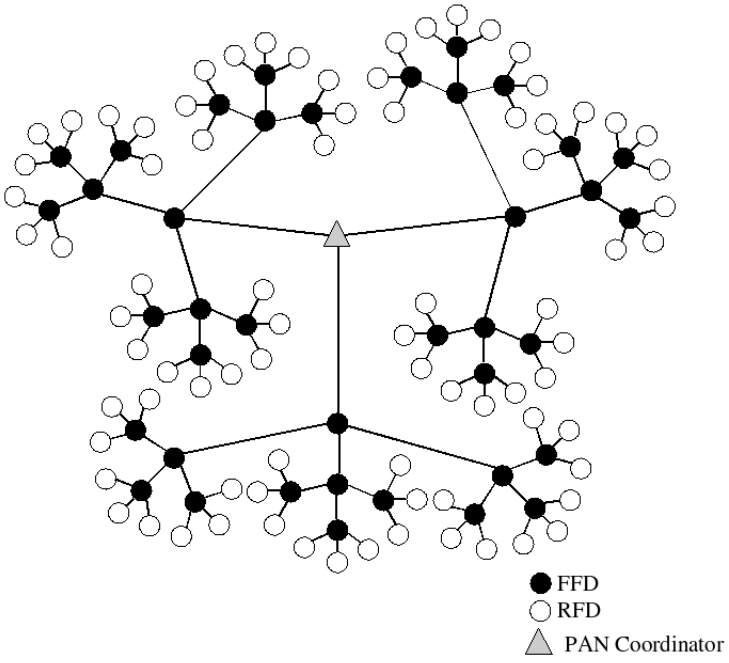

Figure 1.5: Cluster-tree example [25]

noted by many researchers, who proposed extensions to the standard. Examples of some proposed extensions are mentioned in the next chapter.

2. The Guaranteed Time Slot (GTS) features can be used to implement a time division multiple access (TDMA) protocol. Chapter 4 considers this aspect in more detail.

1.4 Thesis Contributions and Organization

The main contributions of the thesis are as follows.

1. In Chapter 2, we investigate the use of a Markov chain model [9] that aims at achieving service differentiation among classes of nodes by modifying the CSMA-CA protocol of the standard. The authors evaluate the Markov chain model using a Matlab simulator that they have developed. We modify the sensor library of the QualNet 4.0 simulator and perform simulation studies to assess the accuracy of the model. We also present examples of using the analysis in [9] to solve problem P1 and P2 of this chapter.

2. In chapter 3, we investigate the use of TDMA protocol in designing schedules that achieve rate differentiation among classes of sensor nodes. In section 3.4, we prove Theorem 3.1 that provides simple sufficient conditions for the existence of a solution to problem P1 (the class size feasibility problem).

3. In chapter 4, we investigate the use of the GTS feature of the IEEE standard to construct TDMA schedules for multi-level trees. We develop an algorithm for constructing TDMA schedules that satisfy the requirements imposed by the standard on GTSs. We also identify two methods for minimizing the cycle length of the constructed schedule. Finally, we use simulation to examine the performance gain of the algorithm.

The rest of the thesis is organized as follows. Chapter 2, investigates the use of the IEEE 802.15.4 standard's CSMA-CA protocol to provide class rate differentiation. Chapters 3 and 4 investigate the use of TDMA protocols to provide class rate differentiation. Finally, we outline some possible future research direction in Chapter 5.

1.5 Summary

In this chapter we presented motivation for investigating some class size optimization problems that involve multiple classes with different data transmission rates. Due to the fact that solving these problems mainly depends on the underlying MAC protocol, fundamental and technological aspects relevant to IEEE 802.15.4 are reviewed. Finally, we outline the thesis contributions and organization.

Chapter 2

Class-Based Rate Differentiation Using CSMA-CA

In this chapter we start by reviewing some research work on the performance of the IEEE 802.15.4 CSMA-CA protocol operating in the beacon-enabled mode. We identify a particular research work that proposes extensions to the standard to implement service differentiation among classes of sensor nodes as most relevant to our main objective in the thesis. We have conducted simulation experiments to assess the accuracy of the proposed method and to examine its use in solving two of the class size optimization problems introduced in Chapter 1. We present and discuss the obtained simulation results and draw some conclusions.

2.1 Literature Review

The IEEE 802.15.4 has attracted the attention of researchers in the field of WSNs since its introduction in 2003. A review of some published research work in this field indicates the following categories of research directions.

a) Work on the performance analysis of the protocols and procedures defined in the standard,

b) work on extending the standard to handle priority traffic, and

c) work on extending the standard to provide service differentiation among classes of nodes in a WSN.

Below, we mention some sample research work in each direction.

2.1.1 Performance of the Protocols Defined in the Standard

1. Zheng and Lee [20] developed an NS2 simulator and conducted several sets of experiments to study various features of the standard such as: a) Performance of beacon-enabled and non-beacon-enabled modes, b) association, tree formation, and network auto configuration, c) orphaning and coordinator relocation, direct, indirect and guaranteed time slot data transmission. The study concluded with noting the flexibility and smoothness of the protocol, and its efficiency compared with the IEEE 802.11 standard.

2. Similarly, the work of Lu *et al.* [12] uses NS2 simulation to study the throughput and energy efficiency. In their work, the effect of duty cycle, which is defined by $\frac{2^{SO}}{2^{BO}}$, on performance parameters is studied. They show that setting BO and SO in such a way to have low duty cycle will reduce the energy consumption while at the same time causes higher delay and lower bandwidth. Also the energy cost and delay performance of using GTS allocation mechanism is investigated.

3. In [26], [27], and [35] the authors develop Markov chain models to analyze the performance of the standard's slotted CSMA-CA under saturation conditions for star network. They obtain expressions for throughput and energy consumption and validate the results using simulation.

4. Ramachandran *et al.* [18] develop a Markov chain model to study performance under non-saturation conditions. Their work uses one Markov chain to model the node states and another Markov chain to model the channel states.

2.1.2 Work on Handling Priority Traffic

Work in this category considers serving high priority packets in event monitoring networks (e.g., forest fire detection networks). The objective is to provide bounded delay to high priority packets. The work in [34], [22], and [1] falls in this category. In general, the proposed priority based methods rely on extending the standard in one, or a combination, of the following ways:

- Allowing a node with high priority packets to transmit before performing the required two CCA steps. However, reducing the number of CCAs performed

before transmitting a high priority packet introduces the risk of collision between ACK packets and high priority packets. A scheme has been introduced in [22] to avoid such collisions.

- Restricting the range of the *backoff counter* to a constant value smaller than the range used for a normal packet in order to transmit higher priority packets faster than normal ones.

- Requiring the PAN coordinator to wake up some time before sending a beacon frame to receive notification of high priority packets. The PAN coordinator then uses its beacon frame to notify nodes that do not have high priority packets to defer their transmission.

- Applying priority queuing rather than FIFO queuing in the application layer.

2.1.3 Work on Supporting Multiple Priority Classes

In Kim *et al.* [9] the authors propose to extend the standard's CSMA-CA protocol by allowing each class of nodes to have its own backoff exponent (BE), and contention window (CW) values to achieve rate or delay differentiation among multiple priority classes. They constructed a three-dimensional Markov chain to model such systems and validate the model using MATLAB simulation.

2.1.4 Concluding Remarks

We identify the work of [9] as most useful to our goal of providing class-based rate differentiation in WSNs. In section 2.3, we give an overview of the Markov chain model in [9]. The Markov chain model, however, seems to be an extension of the Markov chain model in [35]. For convenience, we review the Markov chain of [35] in the next section.

2.2 The work of Park *et al.* [35]

In [35] the authors develop a Markov chain model for the slotted CSMA-CA protocol of the IEEE standard. The model considers a star WSN working under saturation condition. The model utilizes the probability of a device in the channel sensing states. Two random variables for a given device are defined in the model as follows.

- $s(t)$: The random process representing the backoff stage given by the counter NB; we recall that $s(t)$ can take one of the values in the range $[0, 1, ..., m]$, where $m = macMaxCSMABackoffs$. In the Markov chain of Figure 2.1, each row includes the possible sensing states that exist when the device is in the ith backoff stage, where $i = 0, 1, ..., m$. In the figure, backoff stage i appears as the first number of each state.

  We also recall that the ith backoff stage corresponds to a specific value of the backoff exponent BE. Specifically, we use

  $$BE = min(macMinBE + i, macMaxBE)$$

  where $macMinBE = 3$ and $macMaxBE = 5$.

  We use $BE_{min} = macMinBE$ and $BE_{max} = macMaxBE$, for short.

- $b(t)$: The random process representing the backoff counter; thus, if the device is at the ith backoff stage, the algorithm computes a random number uniformly chosen from the range $\left[0, 2^{BE} - 1\right]$. This random number becomes the current value of $b(t)$, that is decremented every time slot. Since $BE = min(BE_{min} + i, BE_{max})$ at the ith stage, i =0, 1, ..., m, one can associate with each stage i a variable denoted W_i, where $W_i = 2^{min(BE_{min}+i, BE_{max})}$. Thus, $W_0 = 2^{BE_{min}}$, $W_1 = 2W_0$, ..., and $W_i = W_0 2^{min(i, BE_{max}-BE_{min})}$, and if the device is at the ith backoff stage then the range of values of $b(t)$ is $[0, W_i - 1]$.

  In Figure 2.1, if the device is in the ith backoff stage (the ith row) then all possible values in the range $[0, W_i - 1]$ exist in the row (the value of $b(t)$ appears as the second number of each state). In addition, the Markov chain uses the special value $b(t) = -1$ to denote that the device is performing the second required CCA procedure.

Based on the above notation, the Markov chain model for a given node is illustrated in Figure 2.1 where α and β denote the channel busy probability at the first and second CCAs respectively. The following is the one-step transition probabilities of the Markov chain:

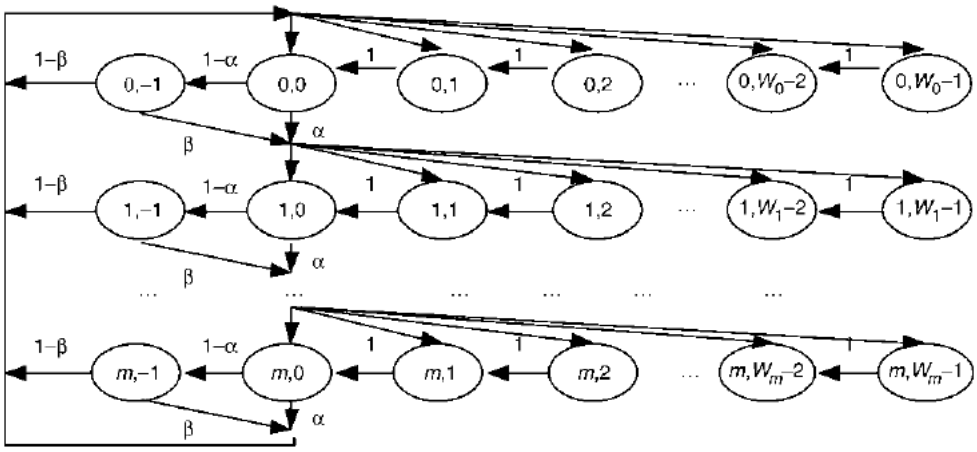

Figure 2.1: Markov chain model of IEEE 802.15.4 [35]

$$\begin{cases} P(b_{i,j}|b_{i,j+1}) = 1 & i \in (0,m) & j \in (0, W_i - 2) \\ P(b_{i,-1}|b_{i,0}) = 1 - \alpha & i \in (0,m) & \\ P(b_{i,j}|b_{i-1,0}) = \alpha/W_i & i \in (1,m) & j \in (0, W_i - 1) \\ P(b_{i,j}|b_{i-1,-1}) = \beta/W_i & i \in (1,m) & j \in (0, W_i - 1) \\ P(b_{0,j}|b_{i,-1}) = (1-\beta)/W_0 & i \in (0,m-1) & j \in (0, W_0 - 1) \\ P(b_{0,j}|b_{m,0}) = \alpha/W_0 & & j \in (0, W_0 - 1) \\ P(b_{0,j}|b_{m,-1}) = 1/W_0 & & j \in (0, W_0 - 1) \end{cases} \quad (2.1)$$

The stationary probabilities $b_{i,j} = lim_{t\to\infty} P\{s(t) = i, b(t) = j\}$ exist since all the states in the Markov chain are positive recurrence. In the following section, we present the Markov chain model of [9] which extends the structure of the above Markov chain.

2.3 The work of Kim *et al.* [9]

In [9], the authors investigate providing service differentiation among a given number $Q + 1$ of priority classes by utilizing the following mechanisms. Throughout the section, $q = 0$ is the index of the highest priority class, and $q = Q$ is the index of the lowest priority class.

- **Service Differentiation by CW Size.** This method sets the contention window CW of each node in class q to a specific value denoted $CW[q]$. To give better transmission opportunity for class q packets over class $q + 1$ packets, one may set the values as follows: $CW[0] < ... < CW[Q]$.

- **Service Differentiation by BE.** This method sets the backoff exponent BE of each node in class q to a specific value denoted $BE[q]$. To give better transmission opportunity for class q packets over class $q+1$ packets, one may set the values as follows: $BE[0] < ... < BE[Q]$. In addition, increasing the difference among the backoff periods of different classes is also expected to improve the service differentiation. Thus, for each class q and backoff stage i, if BE is the current value of the backoff exponent then the proposed mechanism uses the range $[2^{BE-1}, 2^{BE}-1]$, instead of the standard's range $[0, 2^{BE}-1]$, to set the backoff counter. Furthermore, the model does not impose a maximum value on the backoff exponent BE. So, the variable $macMaxBE$ is not used.

As done in [35], the analysis in [9] utilizes the probability of a device in the channel sensing states to model a slotted CSMA-CA protocol under saturation conditions. In the model, three variables are defined for each class q. The variables are denoted $n(t,q)$, $c(t,q)$, and $b(t,q)$. The analysis associates a Markov chain with the nodes of each class q. For abbreviation, the parameter q is dropped from the notation since it is assumed to be known for each Markov chain. Using this abbreviation, the variables $n(t)$, $c(t)$, and $b(t)$ are defined as follows.

- $n(t)$: The random process representing the backoff stage given by the counter NB; this variable is similar to variable $s(t)$ in the previous section. Thus, if $m = macMaxCSMABackoffs$ then $n(t)$ takes values in the range $[0, m]$. In the Markov chain of Figure 2.2, the possible sensing states that exist when the device is at the ith backoff stage is represented by two adjacent rows. In the figure, $n(t)$ appears as the first number of each state. For a given class q, the user sets a value to $macMinBE[q]$; this value is denoted $BE[q]$ for short. Since the model does not impose a maximum value on BE, the ith backoff stage corresponds to the backoff exponent value given by $BE = BE[q] + i$.

- $c(t)$: The random process representing the remaining slots in the contention window of a node in the specified priority class q; thus, $c(t)$ takes values in the range $[0, CW[q]]$. In Figure 2.2, $c(t)$ appears as the second number in each states.

  Note that $c(t) = 1$ means that a node has one more CCA step to perform, and $c(t) = 0$ means that a node has performed all required $CW[q]$ sensing steps and has successfully transmitted the current packet. In Figure 2.2, the path

(A,B,C) is a transition between a state (at point A) where the current packet has been successfully transmitted and one of the sensing states at backoff stage 0 (i.e., one of the states in the top row) in which the node contends for transmitting a new packet.

- $b(t)$: The random process representing the backoff counter; in Figure 2.2, $b(t)$ appears as the third number of each state. As mentioned above, since the proposed method does not impose a maximum value on BE, the ith backoff stage corresponds to $BE = BE[q] + i$, where $i \in [0, m]$. At the ith stage, the proposed algorithm sets a backoff counter to a random number uniformly chosen from the range $[2^{BE-1}, 2^{BE} - 1]$. For each stage i, let $W_i = 2^{BE[q]+i}$. Thus, $W_0 = 2^{BE[q]}$, and for $i > 0$, $W_i = 2^i W_0$. Using the W_i notation, the algorithm chooses a random integer in the range $[W_{i-1}, W_i - 1]$ when the algorithm is at backoff stage i.

  Example 2.1 In Figure 2.2, the path (I,J,K) indicates the following transition:

  - Point I corresponds to state (0,1,0) where the backoff stage $n(t) = 0$, $c(t) = 1$ (thus, the node is performing its last CCA step), and the channel is sensed busy.
  - Point K corresponds to state $(1, CW, W_0)$ where the backoff stage $n(t) = 1$, $c(t) = CW$, and $b(t) = W_0$.

  The (I,J,K) transition is an example transition where a node fails to transmit a packet at backoff stage $n(t) = 0$, and then advances to backoff stage $n(t) = 1$ and sets the backoff counter to the value $b(t) = W_0$. Here, $b(t)$ takes on a value in the range $[W_0, W_1 - 1]$, as mentioned above. ■

Based on the above notation, the Markov chain model for a given node in the priority q class is illustrated in Figure 2.2 where p_I is the idle probability of the channel at any CCA step. The following is the one-step transition probabilities of the Markov chain states [9]:

$$P(b_{0,CW,k}|b_{i,0,0}) = 1/W_0 \qquad i \in [0, m] \qquad k \in [0, W_0 - 1] \tag{2.2}$$

$$P(b_{0,CW,k}|b_{m+1,0,0}) = 1/W_0 \qquad k \in [0, W_0 - 1] \tag{2.3}$$

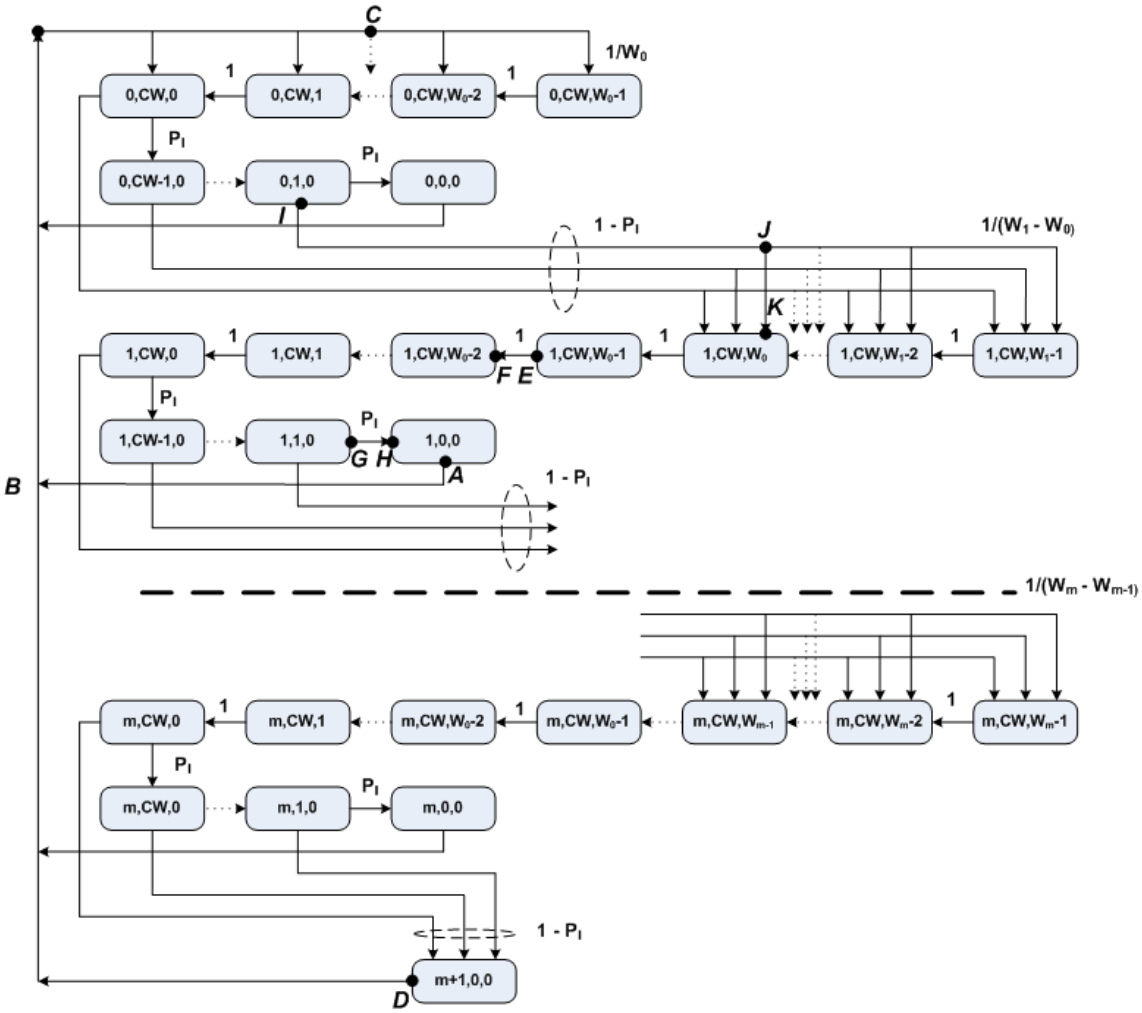

Figure 2.2: Markov chain model of priority-based IEEE 802.15.4 [9]

$$P(b_{i,CW,k-1}|b_{i,CW,k}) = 1 \qquad i \in [0,m] \qquad k \in [1, W_i - 1] \tag{2.4}$$

$$P(b_{i,j-1,0}|b_{i,j,0}) = p_I \qquad i \in [0,m] \qquad j \in [1, CW] \tag{2.5}$$

$$P(b_{i+1,CW,W_{i-1}+k}|b_{i,j,0}) = (1-p_I)/W_{i-1} \quad \begin{array}{l} i \in [0,m] \\ j \in [1,CW] \end{array} \qquad k \in [0, W_{i-1} - 1] \tag{2.6}$$

Example 2.2 Table 2.1 gives an example path for each of the above one-step transition probabilities. ■

Equation number	2.2	2.3	2.4	2.5	2.6
Example Path	A, B, C	D, B, C	E, F	G, H	I, J, K

Table 2.1: The examples of the transition probabilities

The stationary probabilities $b_{i,j,k} = lim_{t\to\infty} P\{n(t) = i, c(t) = j, b(t) = k\}$ exist since all the states in the Markov chain are positive recurrence. The stationary vector $\mathbf{b} = (b_{0,0,0}, b_{0,1,0}, ..., b_{0,q,W_0-1}, ..., b_{m,0,0}, ..., b_{m,q,W_{m-1}}, b_{m+1,0,0})$ can be solved by considering the following constraints:

$$\mathbf{b}P = b \quad \text{and} \quad \mathbf{b}e = 1 \tag{2.7}$$

where e is the column matrix containing only ones and P is the transition probability matrix. Applying the first constraint in Equation 2.7, the following relations between stationary probabilities are obtained in [9].

$$\begin{cases} b_{i,0,0} = b_{0,0,0}(1-p_I^{CW})^i & i \in [0,m] \\ b_{i,CW,k} = b_{0,0,0}\frac{(1-p_I^{CW})^{i+1}}{p_I^{CW}} & i \in [1,m] \quad k \in [0, W_{i-1}-1] \\ b_{i,CW,W_{i-1}+k} = b_{0,0,0}\frac{W_{i-1}-k}{W_{i-1}}\frac{(1-p_I^{CW})^{i+1}}{p_I^{CW}} & i \in [1,m] \quad k \in [0, W_{i-1}-1] \\ b_{0,CW,k} = b_{0,0,0}\frac{W_0-k}{W_0}\frac{1}{p_I^{CW}} & k \in [0, W_0-1] \\ b_{i,j,0} = b_{0,0,0}p_I^j(1-p_I^{CW})^i & i \in [0,m] \quad j \in [1, CW-1] \\ b_{m+1,0,0} = b_{0,0,0}\frac{(1-p_I^{CW})^{m+1}}{p_I^{CW}} \end{cases} \tag{2.8}$$

where $b_{0,0,0}$ is then determined by applying the second constraint of Equation 2.7 as follows:

$$\begin{aligned} b_{0,0,0} &= 2p_I^{CW} / \Bigg\{ 3 - 2(1-p_I^{CW})^{m+1} \\ &+ \frac{3W_0(1-p_I^{CW})^2(1-2^m(1-p_I^{CW})^m)}{2p_I^{CW}-1} \\ &+ \frac{2(1-p_I^{CW})^{m+1}(p_I^{CW}-2p_I^{CW}+1)+2p_I^{CW}(1-p_I^{CW-1})}{1-p_I} \Bigg\}. \end{aligned} \tag{2.9}$$

2.3.1 Performance Measures

Here, we assume a star network with n leaf nodes of which n_q nodes belong to class $q \in [0, Q]$. In [9], the authors obtain the following expressions.

- The probability that a device in priority class q transmits a packet at the

boundary of any backoff period is

$$\begin{aligned}\tau_q &= \sum_{i=0}^{m} b_{i,0,0} \\ &= b_{0,0,0} \sum_{i=0}^{m} (1-p_I^{CW})^i \\ &= b_{0,0,0} \frac{1-(1-p_I^{CW})^{m+1}}{p_I^{CW}} \ .\end{aligned} \tag{2.10}$$

- The idle probability of the channel at any backoff period is

$$p_I = \prod_{l=0}^{Q} (1-\tau_l)^{n_l}, \qquad n = \sum_{l=0}^{Q} n_l \ . \tag{2.11}$$

- The probability that a successful transmission by any node in priority class q occurs is

$$p_{S,q} = n_q \tau_q (1-\tau_q)^{n_q-1} \prod_{l=0, l\neq q}^{Q} (1-\tau_l)^{n_l} = \frac{n_q \tau_q}{1-\tau_q} p_I \ . \tag{2.12}$$

- The probability that a successful transmission by any node occur is

$$p_S = \sum_{l=0}^{Q} p_{S,l} = \sum_{l=0}^{Q} \frac{n_l \tau_l}{1-\tau_l} p_I \ . \tag{2.13}$$

- The busy probability of the channel at any backoff period is

$$p_B = 1 - p_I = 1 - \prod_{l=0}^{Q} (1-\tau_l)^{n_l} \ . \tag{2.14}$$

- To derive an expression of the normalized throughput of a device in priority class q, we need the following values:
  - δ: the length of a backoff slot,
  - L: a random variable representing packet length,
  - T_C: the average time the channel is sensed busy because of a collision; this value takes into account the average time required to send a packet, any inter-frame space periods (SIFS) required in the standard, and
  - T_S: the average time the channel is sensed busy because of a successful transmission; this value is of length T_C plus the optional time required to receive an ACK packet.

Thus, the normalized throughput of a device in priority class q is

$$S_q = \frac{p_{S,q}E(L)}{p_I\delta + p_S T_S + (p_B - p_S)T_C} \ . \tag{2.15}$$

We note the following remarks. For a given class size distribution $\mathbf{n} = (n_0, ..., n_Q)$, **CW** vector $\mathbf{CW} = (CW_0, ..., CW_Q)$, and **BE** vector $\mathbf{BE} = (BE_0, ..., BE_Q)$ one can solve the nonlinear equations in 2.10 and 2.11 to obtain the channel idle probability p_I. Once p_I has been computed, one can use equation 2.12 through 2.15 to compute the performance of interest.

2.4 Performance Evaluation

In this section, we present simulation results that we have done to

a) assess the accuracy of the multi-priority Markov chain model of [9], and

b) provide numerical examples of the use of the search based methods described in Section 1.2 to tackle two of the class size optimization problems presented in Chapter 1.

Below we describe the simulation parameters used in our investigations, and then present the obtained numerical results. Simulation of the IEEE 802.15.4 CSMA-CA utilizes the industry-strength QualNet 4.0 simulator [36] and the sensor add-on library. The sensor library of QualNet has been modified so that the CSMA-CA protocol implements the modifications required by the method proposed in [9], and the network input configuration files allows each node to use its own BE and CW values and pass these values throughout the protocol stack to the MAC layer. Also, solving nonlinear equations that arise in handling various class size optimization problems uses the $fsolve$ function in Maple.

2.4.1 Simulation Parameters

The following is a list of important simulation settings that are common to the three types of performance studies presented in this section.

- We use star WSNs with n sensor nodes organized in a circle of radius $d = 10$ meters. The n sensors are partitioned among $Q = 3$ priority classes: high ($q = 1$), medium ($q = 2$), and low ($q = 3$). More information on the different values of n are mentioned in the following sections. Moreover, the

transmission radius is set to $R_T = 50$ meters so that the network becomes fully connected (i.e., the network forms one collision domain).

- Application traffic (packets transmitted by sensor nodes) lasts for 2000 seconds. Sensor nodes associate with the PAN coordinator over 1000 seconds at the beginning of the simulation.

- Traffic from the sensor nodes to the PAN coordinator is made of fixed length packets each of length $L = 1376$ bits (120 bytes of application data plus 52 bytes header). Each sensor node generates Poisson traffic and the traffic saturates the network.

- The IEEE 802.15.4 operates at an over air data rate of 250 kbps (in the 2.4 GHz ISM band).

- The length of each slot in the superfarme (in symbols) is calculated by

$$aBaseSlotDuration \times 2^{SO} ,$$

where $aBaseSlotDuration$ consists of 60 symbols. Since each symbol carries 4 bits, it follows that each slot accommodates $4 \times 60 \times 2^{SO}$ bits. In order to accommodate one packet ($L = 1376$ bits) per slot, SO is set to 3. The beacon order BO is also set to 3.

2.4.2 Assessing the Accuracy of the Model in [9]

In this section we compare the throughout predicted in [9] for each of the three priority classes with the values obtained by simulation. Our simulation work uses the QualNet 4.0 simulator with the following additional settings:

- The total number of nodes in the star network is varied over the values $n = 6$, 12, 18, 24, 30, 36, and 42 (seven network sizes), where the number of class-i nodes is obtained by $n_i = \alpha_i n$, and the ratio vector is set to $\boldsymbol{\alpha} = (\alpha_1 = \frac{1}{2}, \alpha_2 = \frac{1}{3}, \alpha_3 = \frac{1}{6})$.

- The **BE** vector is set to **BE** $= (BE_1 = 3, BE_2 = 4, BE_3 = 5)$.

- The **CW** vector is set to **CW** $= (CW_1 = 2, CW_2 = 3, CW_3 = 4)$.

- For each value of n, we use function $fsolve$ in Maple to solve the nonlinear equations 2.10 and 2.11 to find the per-class throughput.

Figure 2.3 (a) presents the throughput predicted by the analytical model, and Figure 2.3 (b) presents the throughout found by simulation. We remark that each point on a class-i curve is obtained by averaging the achieved throughput of the n_i sensor nodes in that class. We observe the following:

- For each network size, the predicted average throughput of class-1 nodes (the highest priority class) is higher than the average throughput obtained by simulation. For example, for $n = 12$ nodes, where $(n_1 = 6, n_2 = 4, n_3 = 2)$, Figure 2.3 (a) predicts that class-1 nodes will enjoy on the average 17 kbps where Figure 2.3 (b) reports only 12 kbps.

- Contrary to the above remark, the predicted average throughput of class-2 and class-3 are lower than the observed average throughput.

Since the highest priority class is expected to consume a significant portion of the available channel bandwidth, we conclude with the following recommendation: For a given class size distribution $\mathbf{n} = (n_1, n_2, n_3)$, if the analysis predicts that the highest priority class will not enjoy a specified average throughput then the distribution should be viewed as being infeasible. That is, the simulation results are likely to show that the system will not support the particular data rate specified for the highest priority class.

2.4.3 Example: Solving a Class Size Feasibility Problem

In this section, we present a numerical example for solving a given class size feasibility problem. In the problem we ask whether it is feasible to have a star network with a class size distribution $\mathbf{n} = (n_1 = 9, n_2 = 6, n_3 = 3)$, where the required data rate vector $\mathbf{r} = (r_1 = 10 \text{ kbps}, r_2 = 3 \text{ kbps}, r_3 = 1 \text{ kbps})$, when each sensor node generates Poisson traffic and the traffic saturates the network.

In this problem, the space of possible BE and CW values that can be assigned to each priority class needs to be searched to determine if the problem has a feasible solution. In our example, we restrict the search space as follows:

- We set $BE_1 = 3$ and $CW_1 = 2$.

- For classes 2 and 3, we set $BE_2 = BE_1 + i$, and $BE_3 = BE_1 + 2i$, where i is an integer that may take one of the following values: $i = 0, 1, ..., \frac{aMaxBE-3}{3}$ where $aMaxBE = 9$ (resulting in 3 possible values for variable i).

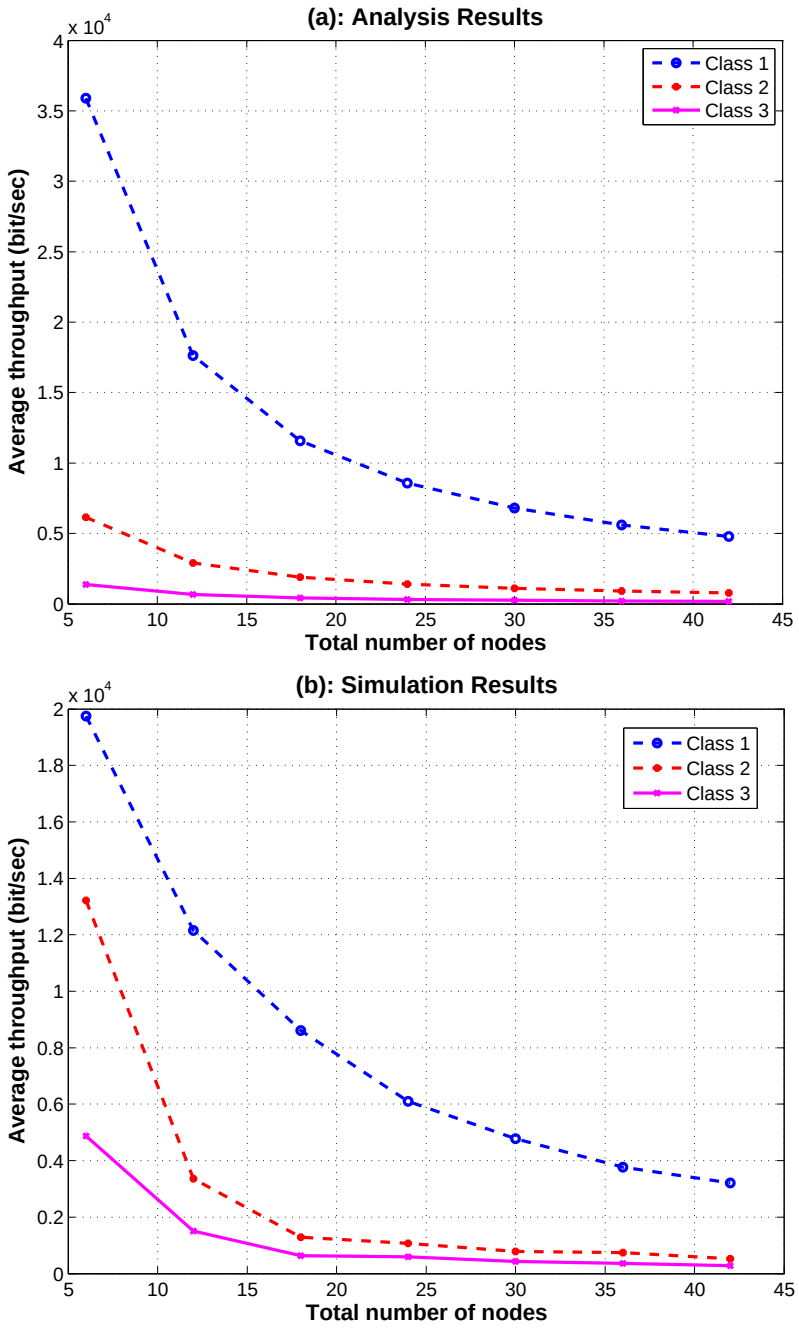

Figure 2.3: Predicted average throughput versus observed throughput for $Q = 3$ priority classes.

- For classes 2 and 3, we set $CW_2 = CW_1 + i$, and $CW_3 = CW_1 + 2i$, where i is an integer that may take one of the following values: $i = 0, 1, ..., \frac{aMaxCW-2}{3}$ where $aMaxCW = 5$ (resulting in 2 possible values for variable i).

This restricted space obtained by using the above rules has only $3 \times 2 = 6$

tuples. On the x-axis of Figure 2.4, we use the notation $(BE + i_1,\ CW + i_2)$ to denote the combination defined by using the specified values of i_1 and i_2. For example, $(BE + 0, CW + 1)$ corresponds to the following vectors:

BE $= (BE_1 = 3, BE_2 = 3, BE_3 = 3)$ and **CW** $= (CW_1 = 2, CW_2 = 3, CW_3 = 4)$.

As done in the previous section, for each choice of the **BE** and **CW** vectors, we solve nonlinear equations to find the per-class average throughput.

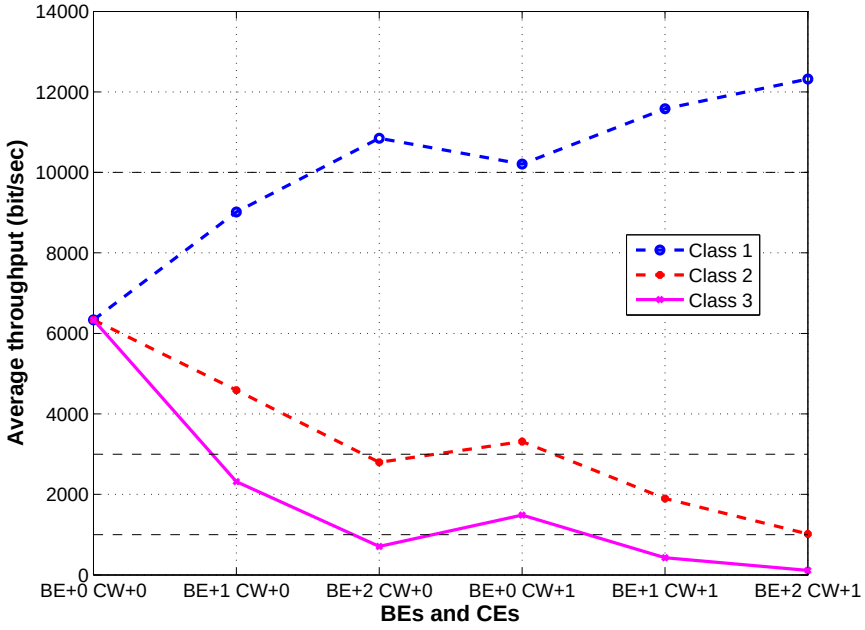

Figure 2.4: Achieved average throughput using different BE and CW values.

Figure 2.4, presents the throughput predicted by analysis for each of the six choices of values in the **BE** and **CW** vectors. The obtained values indicate that the setting corresponding to the combination $(BE + 0, CW + 1)$ is the only setting that is predicted to satisfy all the data rates specified by the input vector $\mathbf{r} = (r_1, r_2, r_3)$.

In light of the recommendation derived in the previous section, we conclude that it is possible that a network with the given class size distribution and the derived **BE** and **CW** settings, deliver the required per-class throughput when simulated. That is, the analysis does not preclude that the given problem instance is feasible.

2.4.4 Example: Solving a Class Size Maximization Problem

In this section, we present a numerical example for solving a class size maximization problem. In the problem, we ask what is the maximum total number of nodes n in a star network that can be accommodated if

a) the nodes are partitioned into $Q = 3$ priority classes where class-i has $n_i = \alpha_i n$ nodes and the given $\boldsymbol{\alpha}$ is $\boldsymbol{\alpha} = (\alpha_1 = \frac{1}{2}, \alpha_2 = \frac{1}{3}, \alpha_3 = \frac{1}{6})$, and

b) the achieved average throughput of each class is defined by a specific rate vector **r** where $\mathbf{r} = (r_1 = 5 \text{ kbps}, r_2 = 2 \text{ kbps}, r_3 = 1 \text{ kbps})$.

Our strategy is based on reducing the maximization problem to a feasibility problem. Since the feasibility problem takes as input a specified distribution vector $\mathbf{n} = (n_1, n_2, n_3)$, our search method generates a number of instances of the feasibility problem, each instance having some possible distribution vector **n**. Since the vector of node ratio $\boldsymbol{\alpha}$ is given as input, assigning a value to n_1 (or n_2, or n_3) uniquely determines a distribution vector **n**. Hence, we include in our search space the possibilities that n_1 can take values from 1 to the maximum number of nodes allowed by the channel data rate (250 kbps in our example). For simplicity of the example, we only consider seven possible values of n_1: $n_1 =$ 3, 6, 9, 12, 15, 18, and 21.

Figure 2.5 shows the throughput predicted at each point in the search space. Similar to the previous section, the space has 6 tuples, where each tuple corresponds to the values in vectors **BE** and **CW**, and some network size. The results predicts that many of the constructed instances of the feasibility problems are feasible, and that the maximum number of nodes is achieved by setting $\mathbf{BE} = (BE_1 = 3, BE_2 = 4, BE_3 = 5)$, and $\mathbf{CW} = (CW_1 = 2, CW_2 = 2, CW_3 = 2)$ when $\mathbf{n} = (n_1 = 15, n_2 = 10, n_3 = 5)$. As mentioned above, the predictions need further verification by simulation.

2.5 Summary

In this chapter, we give an overview of some existing work on modeling and performance evaluation of IEEE 802.15.4 CSMA-CA protocol. We have identified the work presented in [9] as most relevant to solving the problem of providing rate differentiation among multiple classes of sensor nodes in WSN. The approach in [9] proposes some modification to the IEEE 802.15.4 standard. We have implemented the required modifications in the QualNet network simulator. Using the modified sensor library of the QualNet, we investigate the gap between the throughput predicted by the Markov chain model of [9] and the simulation results. We also presented examples of using the analysis in the [9] to solve the class size feasibility and maximization problems.

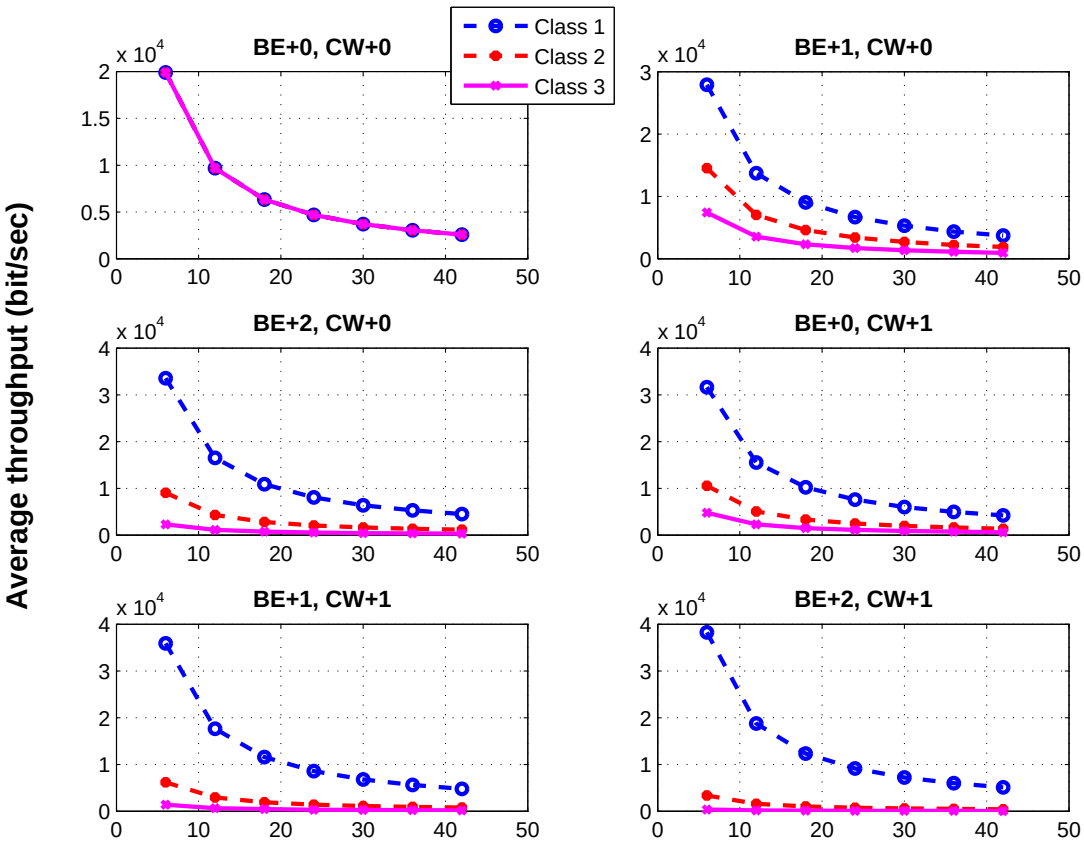

Figure 2.5: Achieved average throughput using different BE and CW values when the total number of nodes n (the x axis) varies among seven values $n = 6$, 12, 18, 24, 30, 36, and 42.

Chapter 3

Class-Based Rate Differentiation Using TDMA

TDMA protocols are used in many wireless networks where a base station exists, or can be elected, since they are capable of satisfying many quality of service (QOS) constraints. In this chapter we introduce some basic definitions and methodologies that are used in constructing TDMA protocols. We conclude the chapter by showing some basic sufficient conditions for solving the class size feasibility problem using TDMA schedules.

3.1 Introduction

In this section we present the motivation for the work on TDMA schedules for WSNs and introduce some basic definitions. Our work in this thesis is largely motivated by the need to design *continuous monitoring* [6] sensor networks where data is sampled and transmitted at regular intervals. The required rates of sampling and transmission can be high so as to result in channel saturation. This type of WSNs is in contrast with *event driven* sensor networks where sensor nodes stay idle (or sleep) until certain events occur [6]. Hence, *scheduling-based* MAC schemes are suitable for continuous monitoring networks since they are able to efficiently support multiple traffic classes, provide different levels of service quality through bandwidth allocation, and maximize energy savings [6].

The simplest type of scheduling-based MACs is TDMA MACs where time is divided into identical frames. Each frame is divided into fixed length slots. The schedule specifies for each slot the set of links that are allowed to transmit during the slot. So, each link occupies cyclically repeating time slots. Bandwidth can be

supplied on demand to different links by allocating different number of time slots per frame to different links.

TDMA systems are classified as TDD (time division duplexing) systems or FFD (frequency division duplexing) system. In TDMA/TDD one channel is used for both uplink and downlink communications. In TDMA/FFD one channel is used for each direction. Examples of wireless networks that use TDMA include: GSM 900 (Global system for Mobile Communication) cellular networks for serving voice traffic, WiMax for providing broadband access, and bluetooth for implementing short-range, high-rate WPANs. Examples of proposed protocols that use TDMA include the Energy-Conserving MAC for wireless ATM networks [32], and the Self-Organizing MAC for sensor networks [33].

Currently there is a vast number of published work on scheduling and routing using TDMA wireless networks, see for example [23], [13], [29], [31], [24], [2], [21]. The obtained results include a few exact algorithms, approximation algorithms, and many heuristic algorithms for solving different scheduling and routing problems.

Of particular interest to our work here is the work in [21]. We note that the work in [21] implies that problem P1 (the class size feasibility problem) can be solved in polynomial time in the restricted case when the network is a tree on n nodes and all interference relations can be deduced from its tree diagram. In section 3.4, we give simple expressions that serve as sufficient conditions for solving the feasibility problem, also under the assumption that all interference relations can be deduced from the tree diagram. To explain the derived conditions we present more basic concepts in the next two sections.

3.2 Definitions and Notations

In this section we introduce some basic definitions, notations, and assumptions that are commonly used in developing TDMA scheduling algorithms.

Network Graph. We denote the transmission range of node x by $R_T(x)$. Thus, x can transit reliably to any node y where the distance $d(x, y) \leq R_T(x)$. For the purpose of constructing TDMA schedules, many results in the literature assume that all nodes in a network have the same transmission range, denoted R_T. In such cases, the network can be represented by a graph $G = (V, E)$ on the set V of nodes and the set E of bidirectional communication links. The graph is called the *network*

or *connectivity* graph. Figure 3.1 illustrates an example of a network graph.

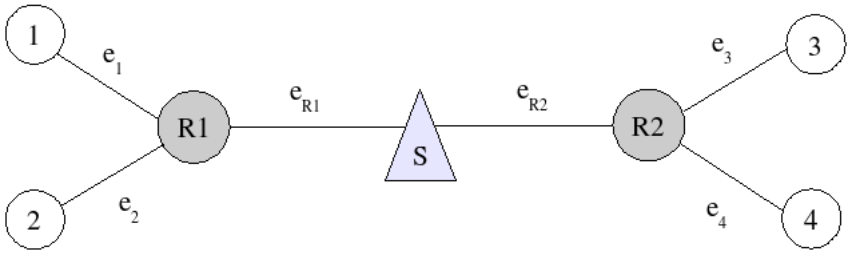

Figure 3.1: An example of network.

Interference Model. We denote the interference range of node x by $R_I(x)$, $R_I(x) \geq R_T(x)$. Again, many results on constructing TDMA schedules in the literature assume that all nodes have the same interference range denoted R_I. Typical values of R_I in the literature are $R_I = R_T$ or $R_I = 2R_T$. As done in many research work, we adopt the *protocol model* of interference, defined in [15]. According to the protocol model, the transmission from node x to node y is considered successful if and only if the receiver y lies outside the interference range of any other transmitting node w.

Primary and Secondary Interference. Again, in constructing TDMA schedules, many results assume that transmission on any link is bidirectional. Bidirectional communication over all links are specially required when the nodes exchange RTS/CTS/DATA/ACK packets, as is the case with the IEEE 802.11 family of protocols. Given the protocol model of interference, and the need of bidirectional communication over any link, it is then convenient to classify the interference relation between links into two types: *primary* interference and *secondary* interference as follows:

- Two links have a primary interference if they share a node.
- Two links have a secondary interference if any node on the first link is within the interference range of any node on the second link.

In Figure 3.1, for example, links e_1 and e_{R2} have only secondary interference. The justification of separating the above two types of interference is that secondary interference can be mitigated by using different methods such as using directional antennas.

Conflict Graph and Node Colouring. A valid TDMA schedule must not assign two interfering links (either having primary or secondary interference) to the same

time slot. Such two links are viewed as being conflicting. A common approach to construct valid TDMA schedules is to consider the *conflict* graph, denoted $G_C = (V_C, E_C)$, of a network. In a conflict graph, every vertex in the set V_C corresponds to a link in the network. Two vertices in V_C are adjacent if their corresponding links in the network interfere with each other. Figure 3.2 illustrates the conflict graph of our example network in Figure 3.1 assuming that $R_T = R_I$.

If each link in a subset $E' \subseteq E$ of links needs to have exactly one transmission slot in every frame in a TDMA schedule then the problem of finding the minimum number of slots per frame that satisfies all transmission requirements is equivalent to finding the minimum number of colours required to properly colour the subgraph of G_C defined by the nodes in G_C that correspond to the subset E' of links. Thus, the schedule construction problem is a special case of the well known graph colouring problem [14].

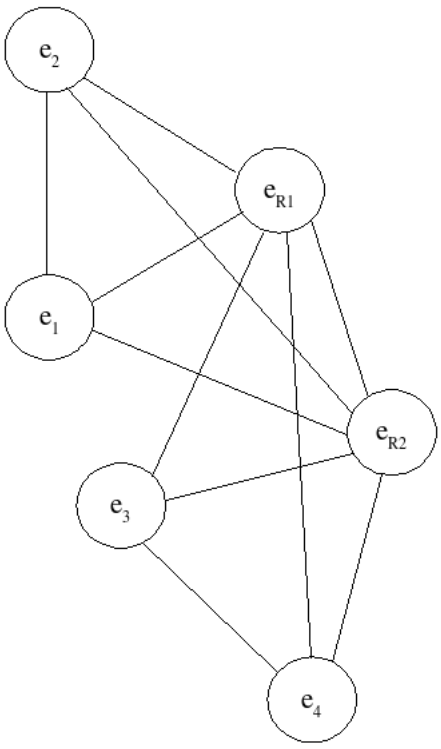

Figure 3.2: The conflict graph of the example network.

Chromatic Numbers and Greedy Colouring. In graph theory, the minimum number of colours required to properly colour the vertices of a graph G is called the *chromatic number* of the graph, denoted $\chi(G)$. Finding $\chi(G)$ of any arbitrary graph

is known to be NP-complete [14]. However, if the maximum degree of G is $\Delta(G)$ then it is easy to verify that $\chi(G) \leq \Delta(G) + 1$. In Figure 3.2, for example, the conflict graph G_C has maximum degree $\Delta(G_C) = 5$ (the degree of vertices e_{R1} and e_{R2}). Thus, we know that $\chi(G_C) \leq 5 + 1 = 6$ colours (in fact, $\chi(G_C) = 4$).

A simple algorithm to colour the vertices of the graph with at most $\Delta(G) + 1$ colours is the *greedy* colouring algorithm. In the algorithm, we consider the vertices of the graph in some arbitrary chosen order. At each step, the algorithm assigns a valid colour to a node. The algorithm uses a new colour only if a new colour is required.

3.3 Expanded Conflict Graphs

In this section, we describe the concept of expanded conflict graphs, used, for example, in [24]. Expanded conflict graphs are useful in using node colouring algorithms to construct transmission schedules for networks where each link e has an associated number $w(e)$ of packets that need to be transmitted in each frame. For our purpose of designing WSNs, we restrict our attention to tree networks. In particular, we consider a tree T with the following structure:

a) Each leaf in T corresponds to a sensor node. In each frame, the sensor node needs to transmit to its parent over the link e that joins the node to the parent an associated number of packets, denoted $w(e)$.

b) Each internal node, excluding the sink node, is a relay that does not generate packets of its own. Each relay needs to transmit to its parent over the link e a number of packets $w(e)$ in every frame. Here, $w(e)$ is the sum of all incoming packets to the relay from its children in each frame.

Example 3.1 In Figure 3.1, for example, we may assume that $w(e_1) = w(e_3) = 2$, and $w(e_2) = w(e_4) = 1$ packets per frame (pkt/frame). Thus, $w(e_{R1}) = w(e_{R2}) = 3$ pkt/frame. ■

The expanded conflict graph, denoted $G_e = (V_e, E_e)$ has the following structure:

a) For each link e in the tree T, G_e has a cluster of $w(e)$ vertices. All vertices in a cluster are pairwise adjacent in G_e.

b) For each pair of interfering links e_1 and e_2 in T, G_e has $w(e_1) \times w(e_2)$ edges between each vertex in e_1's cluster and each vertex in e_2's cluster.

One may verify that the minimum number of slots in a frame required to satisfy all transmissions equals the chromatic number $\chi(G_e)$.

Example 3.2 Figure 3.3 illustrates G_e for the tree in Figure 3.1 when $w(e_1) = w(e_3) = 2$ pkt/frame, and $w(e_2) = w(e_4) = 1$ pkt/frame. A proper colouring of G_e that uses 9 colours is shown in Figure 3.4 (in each vertex the first number is the degree of the node, the second number is the assigned colour). Table 3.1 gives the resulting transmission schedule with 9 slots. ■

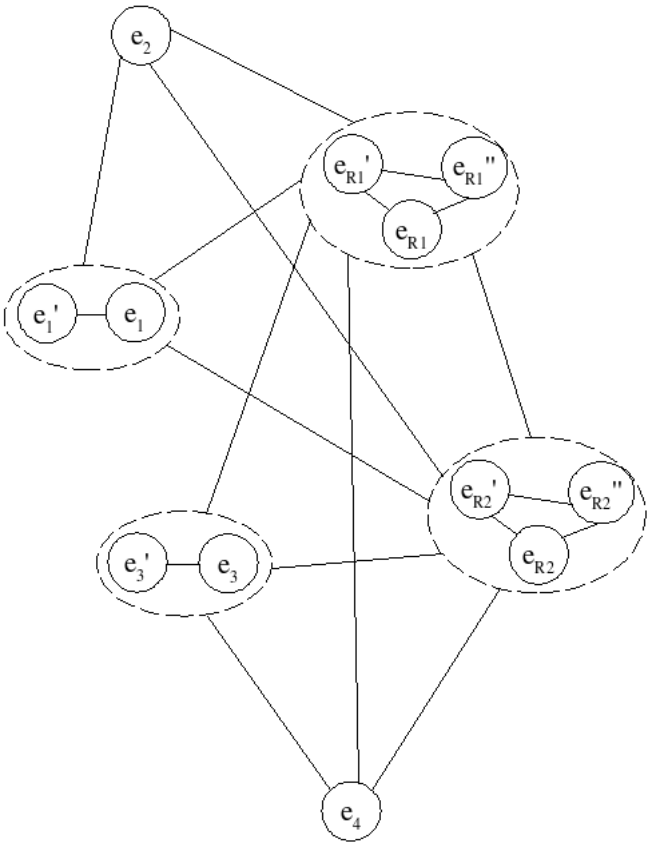

Figure 3.3: The expanded conflict graph of the example network.

Slot No.	1	2	3	4	5	6	7	8	9
Active Links	e_{R1}	e_{R1}	e_{R1}	e_{R2}	e_{R2}	e_{R2}	e_1, e_3	e_1, e_3	e_2, e_4
Transmitting node	R1	R1	R1	R2	R2	R2	1, 3	1, 3	2, 4

Table 3.1: A valid schedule for the example network

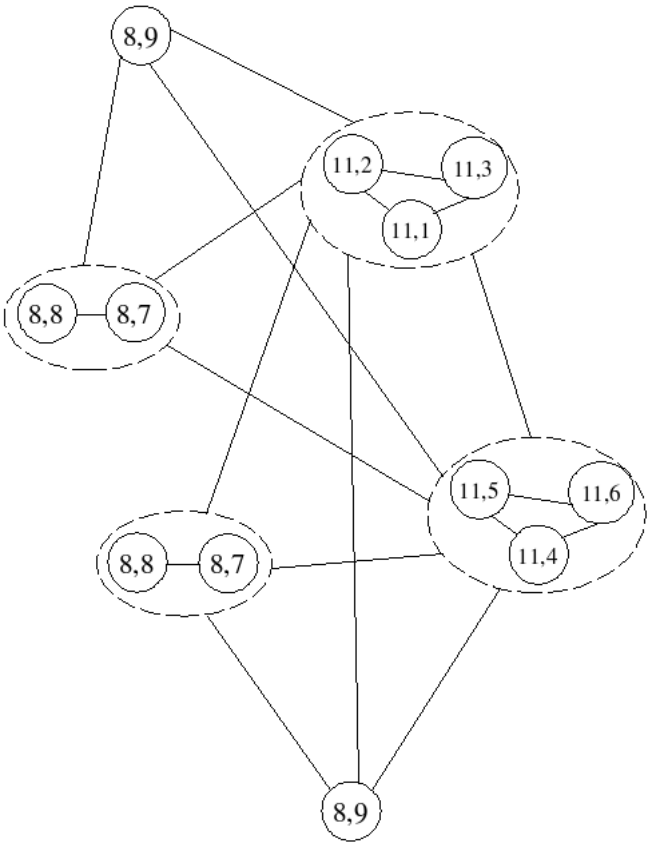

Figure 3.4: The expanded conflict graph and its colouring (in each circle, the first number is the degree of the node, and the second number is the assigned colour)

3.4 Sufficient Conditions for the Class Size Feasibility Problem

In this section, we use the concept of expanded conflict graphs to derive simple sufficient conditions for the existence of a TDMA schedule for solving problem P1 (the class size feasibility problem) on arbitrary trees. We note that exact solutions to the problem exist in the literature (see, for example [21]). However, the exact algorithms are more complicated than the results obtained here.

To start, we recall that an instance of problem P1 is specified by the following parameters:

- R: the maximum allowable data rate of the system in bps.
- Q: the number of priority classes,
- $\mathbf{n} = (n_1, n_2, ..., n_Q)$: the class size distribution vector, where $n = \sum_{i=1}^{Q} n_i$,

- $\mathbf{r} = (r_1, r_2, ..., r_Q)$: the data rate vector, and
- T: a tree with n leaves, where each leaf is a sensor node that belongs to one of the Q priority classes.

To describe our approach, we first introduce the following variables:

- t_{Slot}: the duration of one time slot in the constructed schedule.
- $l_{slot} = t_{slot} \times R$: the number of bits that can be transmitted in one slot which is equal to one packet length.
- s_i: the number of slots assigned to each class-i node (or, more accurately, assigned to the link between the node and its parent) in each frame. Thus, each class-i node transmits $s_i \times l_{slot}$ bits per frame.
- $w(e)$: for each link $e = (x, y)$ between a node x and its parent y in the tree, the weight $w(e)$ is the number of slots in each frame that is assigned to the edge. If x is a class-i leaf node then $w(e) = S_i$, as mentioned above. Else, if x is a relay node then $w(e)$ is the sum of the weight of links between x and its children.

  Example 3.3 For the tree T in Figure 3.1, assume that class-1 nodes are 1, and 3, and class-2 nodes are 2, and 4. Then the weights assigned to the links are as follows:

  - $w(e_1) = w(e_3) = s_1$
  - $w(e_2) = w(e_4) = s_2$
  - $w(e_{R1}) = w(e_{R2}) = s_1 + s_2$

  ■

- G_e: the expanded conflict graph of T.

Using the above notation, one can then state that a schedule of $\chi(G_e)$ slots allows each class-i node to transmit at the following rate:

$$\frac{s_i \times l_{slot}}{\chi(G_e) \times t_{slot}} = \frac{s_i \times R}{\chi(G_e)} \text{ bps.} \tag{3.1}$$

Thus, to satisfy the requirements of each class in T, we need:

$$\frac{s_i \times R}{\chi(G_e)} \geq r_i \text{ for each class } i. \tag{3.2}$$

The results shown in the next theorem, relies on approximating $\chi(G_e)$ for trees with different depth values.

Theorem 3.1 *Let R, Q, **n**, **r**, and T be an input instance of the class size feasibility problem. Assume that*

- *T has depth d,*
- *each leaf of T is a node that generates traffic,*
- *each internal node of T (excluding the root) is a relay node, and*
- *the secondary interference relations between links in T are determined by the existing communication links in T only (and no other additional interference relations).*

Then the following are sufficient conditions for the unknown variables $\{s_i : i = 1, ..., Q\}$ to give a TDMA schedule that satisfies the data rate requirements:

1. *if $d = 1$ then it suffices to have:*

$$\frac{s_i \times R}{\sum_{i=1}^{Q} n_i \times s_i} \geq r_i \text{ for each class } i \tag{3.3}$$

2. *if $d = 2$ then it suffices to have:*

$$\frac{s_i \times R}{2\sum_{i=1}^{Q} n_i \times s_i} \geq r_i \text{ for each class } i \tag{3.4}$$

3. *if $d > 2$ then it suffices to have:*

$$\frac{s_i \times R}{5\sum_{i=1}^{Q} n_i \times s_i} \geq r_i \text{ for each class } i \tag{3.5}$$

Proof. For each link e in the tree T, let $E_{int}(e)$ be the set of links that interferes with e. In addition, let $w_{int}(e) = w(e) + \sum_{e' \in E_{int}(e)} w(e')$. By definition of the expanded conflict graph G_e of T, the link e corresponds to a cluster of $w(e)$ vertices in G_e. The degree of each vertex of the cluster is $w_{int}(e) - 1$. If this degree is the largest degree in G_e then $\chi(G_e) \leq w_{int}(e)$. In each of the following cases we identify a link e with the largest possible $w_{int}(e)$:

1. **Case 1:** $d = 1$**.** In this case T is a star. Each link e incident to the sink node has $w_{int}(e) = \sum_{i=1}^{Q} n_i \times s_i$. The result then follows from Equation 3.2 by noting that $\chi(G_e) = w_{int}(e)$.

2. **Case 2:** $d = 2$**.** Each link e incident to the sink node has $w_{int}(e) \leq 2 \sum_{i=1}^{Q} n_i \times s_i$.

3. **Case 3:** $d > 2$**.** In the worst case, link e interferes with all links in its level in T, all links in the two levels above its level, and all links in the two levels below its level. Thus, link e has $w_{int}(e) \leq 5 \sum_{i=1}^{Q} n_i \times s_i$. ■

In the above theorem, we remark that the use of a tree of depth 2 (case 2) results in a tree that can span a larger geographic area than a star tree at the expense of achieving reduced data rate. This can be explained in light of the increased interference on the links of a depth 2 tree.

3.5 Summary

In this chapter we introduced some basic definitions used in constructing TDMA schedules and presented simple expressions that provide sufficient conditions for solving the class size feasibility problem. The derived expressions apply to trees where interference relations can be deduced from their diagrams.

Chapter 4

Scheduled Access Using the IEEE 802.15.4 Guaranteed Time Slots

In this chapter we investigate the use of the IEEE 802.15.4 standard in providing time-division multiple access (TDMA) to the nodes of a wireless sensor network. The ability to implement a TDMA-based schedule using the facilities provided by the IEEE standard enables the network designer to achieve the goal of providing rate differentiation among classes of nodes in a wireless sensor network (WSN). In this chapter, we develop a method for constructing TDMA-based schedules using the Guaranteed Time Slots (GTSs) facility of the IEEE standard.

The chapter starts by reviewing the use of the GTSs of the IEEE standard with the goal of identifying the restrictions that must be considered in our development. Next, we define a class of schedules, called *flow balanced* schedules, that are efficient in terms of the delay incurred by packets transmitted from any node in a tree network to the sink node, and the number of packets that should be queued in each node. Based on the structure of flow balanced schedules, we identify useful optimization aspects that help in constructing schedules with short cycle length. We then introduce an algorithm, called GTS-TDMA, that integrates some heuristic algorithms to utilize the identified optimization aspects. We then present simulation results that show the performance gains of the devised GTS-TDMA algorithm over the IEEE 802.15.4 CSMA protocol.

4.1 Introduction

Time division multiple access (TDMA) protocols provide scheduled access to broadcast channels. Hence, they are more efficient than CSMA protocols since they avoid transmission collisions. In addition, TDMA protocols allow the network designer to construct schedules that allocate bandwidth to the nodes of a network in a deterministic way. Consequently, the constructed schedules allow the designer to obtain tight bounds on both the throughput achieved by each node, and the worst case packet delay over any given path in the network. The IEEE 802.15.4, however, does not provide an explicit support for TDMA protocols. Rather, the IEEE standard provides the Guaranteed Time Slots (GTSs) mechanism to serve delay sensitive traffic in a wireless sensor network (WSN).

In this chapter, we investigate the use of the GTS facility of the standard to construct TDMA-based schedules for managing the available channel bandwidth. To the best of our knowledge, prior work on utilizing the GTSs on managing priority traffic (e.g., [3], [4], [11], [37], and [5]) do not consider the construction of TDMA-based schedules, as done in this chapter. Thus, the work presented in this chapter is one of the thesis contributions.

The rest of the chapter is organized as follows. Section 2 gives an overview of the main constraints regulating the use of GTSs in an IEEE 802.15.4 compliant WSN. Sections 3, 4, and 5 develop the concept of flow balanced schedules through the use of concrete examples. Section 6 identifies two optimization aspects that can be used to optimize the length of the constructed schedule cycle. Section 7 presents the main algorithm of the chapter. Section 8 presents the results of our simulation based performance evaluation study, and section 9 gives some concluding remarks.

4.2 Overview of the IEEE 802.15.4 GTSs

For our purpose, we summarize the main aspects and restrictions pertaining to the use of GTSs in the IEEE 802.15.4 standard [10]. All nodes in the WSN considered are assumed to be compliant with the standard.

1. In a beacon-enabled operation of a WSN, both the PAN coordinator and other relay nodes (acting as coordinators) can transmit beacon frames. Each coordinator can use a superframe structure composed of up to three types of periods: Contention Access Period (CAP), Contention Free Period (CFP), and inactive period.

2. The IEEE standard requires that the active period of each superframe, known as the superframe duration (SD), of all coordinators in the network to have equal length.

3. In contrast to the above constraint on the length of the SD intervals, the length of the beacon intervals (i.e., the lengths of the superframes) belonging to different coordinators need not be equal.

4. The active period of each superframe is divided into a fixed number of equal length slots, denoted $n_{SD\_slots\_max}$. In the IEEE standard, $n_{SD\_slots\_max} = 16$ slots. The beacon frame always starts at the beginning of the first time slot.

5. In each superframe, there can be up to a fixed number, denoted n_{GTS_max} of GTSs. In the IEEE standard, $n_{GTS\_max} = 7$. Each GTS can occupy one or more SD slots. So, for example, the 16 slots of an SD interval can be divided into 2 GTSs serving 2 different nodes. The first GTS may occupy 15 slots, and the second GTS may occupy the remaining slot.

6. The beacon frame issued at the beginning of an SD interval determines the number of GTSs in the SD interval (at most $n_{SD\_slots\_max}$). In addition, the beacon frame carries the following information: (a) the beneficiary node of each GTS, (b) the start slot of each GTS, (c) the length of each GTS, and (d) the data transfer direction of all GTSs in the SD (the direction is either uplink to the coordinator issuing the beacon frame, or downlink to the associated nodes).

7. The standard specifies mechanisms for a node to request the allocation of GTSs from its serving coordinator, and mechanisms for expiring allocated GTSs, if the associated node does not use an allocated slot for a certain number of consecutive superframes.

From the above, we conclude that it is possible to use the GTSs to construct a TDMA schedule. In a constructed schedule, each active period in each beacon interval contains the minimum allowed CAP, defined by the $aMinCAPLength$ period ($= 440$ symbols) in the standard, and the rest of the active period contains a CFP only. Moreover, the active periods of all superframes should respect the equality constraint mentioned in point 2 above. A constructed schedule is specified by describing its *cycle* structure (figure 4.2 illustrates one cycle of a schedule) that is repeated over the time.

4.3 An Example Network

To illustrate the construction of a schedule that takes into account the constraints mentioned above, we use the following example network throughout the chapter.

Example 4.1 Figure 4.1 illustrates a weighted tree with the following properties:

- The tree has 6 coordinators, labelled c_0 to c_5, where c_0 is the PAN coordinator, and c_1 through c_5 are relay nodes that don't generate packets of their own. The leaf nodes s_1 to s_{10} are sensor nodes that generate traffic.
- With each leaf node in the diagram there is an associated number of packets. We interpret that a leaf node periodically generates this number of packets every schedule cycle. For example, sensor node s_1 generates 3 packets every schedule cycle.
- Similar to the above property, with each non-leaf node in the diagram there is an associated number of packets that the node should periodically collect from its children. For example, relay c_3 should periodically collect 4 packets from relays c_4 and c_5.
- The solid lines are communication links, and the dashed lines are interference edges. Not shown in the figure are the primary and secondary interference relations implied by the links of the tree. For example, the secondary interference relation between links (c_0, c_3) and (c_3, s_4) is not shown. Additional interference edges that are assumed to exist appears as dashed lines.

■

The weighted tree of the above example constitutes an instance of the schedule construction problem tackled in this chapter. A valid solution to the problem is a schedule that satisfies the constraints identified in the previous section, and serves each node x by transmitting its associated number $w(x)$ of packets in each cycle. If the cycle length of the constructed schedule is τ seconds then the data rate enjoyed by each node under the constructed schedule is $w(x)/\tau$ packets/second. Thus, maximizing the data rate enjoyed by each node amounts to minimizing the cycle length of the constructed schedule. We discuss methods for optimizing the cycle length in a subsequent section.

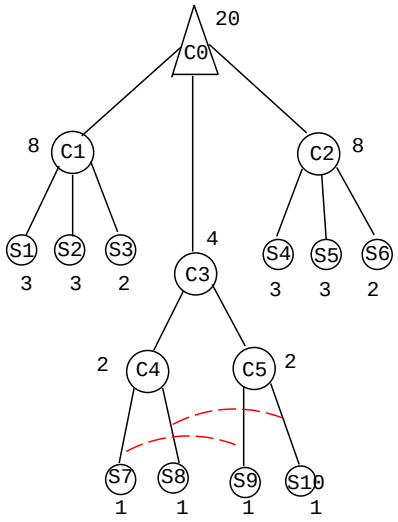

Figure 4.1: An example WSN. Primary and secondary interference relations are not shown. Dashed lines indicate additional interference relations.

4.4 Flow Balanced Schedules

In this section, we introduce the concept of *flow balanced* schedules as a restricted class of TDMA-based schedules that can be used to control packet transmissions in tree networks such as the network of Example 4.1. Flow balanced schedules have a special structure that is intended to simplify the task of minimizing the length of a typical schedule cycle. A concrete example using the introduced concepts is presented in the next section.

Definition 4.1 Given a problem instance where each internal node in the given network is associated with a given number of packets that should be collected from its children periodically in each schedule cycle, a schedule that satisfies the IEEE 802.15.4 requirements on GTSs is called flow balanced if it satisfies the following properties:

1. In each cycle, each internal node collects from its children exactly the specified number of packets associated with the node.

2. Transmissions that are scheduled in one time slot are interference-free.

■

The structure of flow balanced schedules allows us to prove the following result.

Theorem 4.1 *Let S be a flow balanced schedule for a tree network T. Denote the length of the beacon interval of the sink node by BI_{sink}. Then*

1. *If x is a node at distance d hops away from the sink node in the tree then the time required for all packets transmitted by x during one cycle to reach the sink is bounded by $d.BI_{sink}$.*

2. *If node x is scheduled to transmit $w(x)$ packets in each cycle then x needs to buffer at most $2.w(x)$ packets at any instant.*

Proof. Part 1 follows since each internal node collects from its children all packets that the children need to transmit in one cycle. Thus, the schedule guarantees that all packets transmitted by a node during one cycle advance one hop towards the sink at the end of each cycle.

Part 2 follows since at any instant, each node is required to buffer the incoming packets (if the node is a relay) and the outgoing packets. Thus, no more than $2.w(x)$ packets need to be buffered at any instant. ■

Finally, we remark that a flow balanced schedule can leave many slots unused (since it is only required to serve exactly the specified number of packets associated with each node). Nevertheless, by part 2 of the above theorem, the schedule ensures that no node becomes congested at any point of time.

4.5 An Example Flow Balanced Schedule

In this section we present a flow balanced schedule for the network in Example 4.1. The constructed schedule incorporates a number of optimization aspects, as mentioned below.

Example 4.2 Consider the weighted tree T of Example 4.1. For the purpose of illustrating the key features of the constructed schedule, we assume (for simplicity) that $n_{SD\_slots\_max} = 6$, $n_{GTS\_max} = 2$, and each slot accommodates one packet. Figure 4.2 illustrates a valid flow balanced schedule for the given problem instance. In particular, one may verify that (a) each SD has exactly $n_{SD\_slots\_max} = 6$ slots, (b) each SD is partitioned into at most $n_{GTS\_max} = 2$ GTSs (the nodes served by an SD are written below the SD in the diagram), (c) each internal node (a coordinator) collects exactly its associated number of packets in each cycle, and (d) transmissions that occur in one time slot are interference free.

In the figure, each cycle is composed of 4 strides. The width of each stride equals the length of the beacon interval of the PAN coordinator. ■

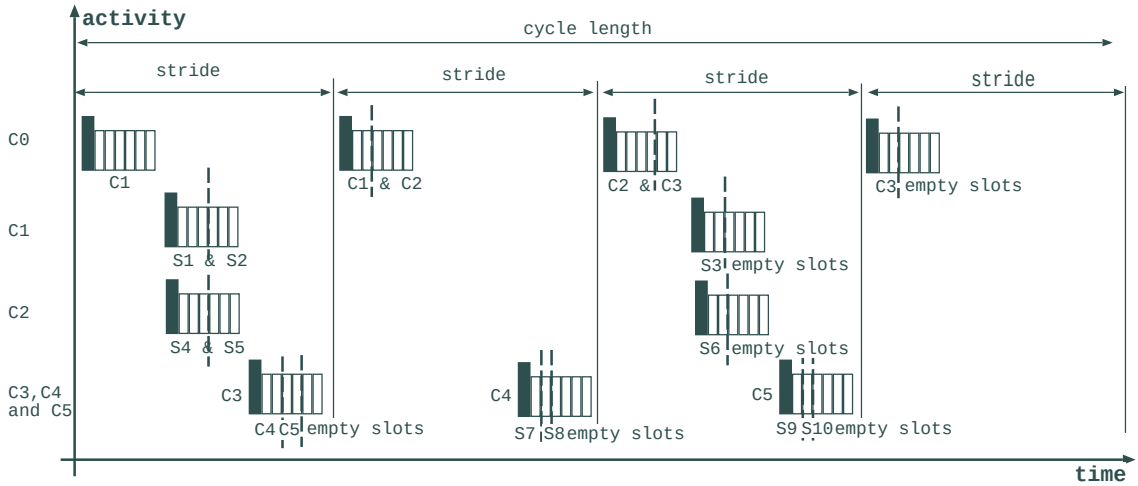

Figure 4.2: An example flow balanced schedule

We now draw some remarks on the above schedule. In the remarks, we use BI_x to denote the length of the beacon interval (equivalently, the superframe length) of node x in the constructed schedule.

1. The SDs of coordinators c_1 and c_2 are scheduled in *parallel*. That is, in each stride where the SDs of c_1 and c_2 occur simultaneously, they start form the same offset relative to the stride they occur in. In the example, $BI_1 = BI_2 = 2BI_0$.

2. The SDs of coordinators c_3, c_4, and c_5 are scheduled in *alternation*. That is, in each stride, at most one SD of at most one of the three coordinators occurs in the stride, and all SDs start at the same offset relative to their respective strides. In the example, $BI_3 = BI_4 = BI_5 = 4BI_0$.

3. The above schedule assumes that each SD slot can accommodate only one packet. In general, SD slots can be set to accommodate multiple packets. However, a careful look at the overall cycle length of the constructed schedule reveals that the use of short slots that accommodates the fewest number of packets have a good potential of minimizing the schedule length. However, since the use of short slots is not guaranteed to provide optimal solutions, our proposed algorithm, presented in the next section, constructs a schedule for each possible value of the SO parameter and chooses the schedule with the shortest cycle length.

4. In the above example, the PAN coordinator requires the largest number of

SDs in each cycle; this largest number determines the cycle length of the constructed schedule. In general, however, it is possible that the node requiring the largest number of SDs is not the PAN coordinator. For example, consider a network where the PAN coordinator c_0 collects packets form relay node c_1, and c_1 collects packets from $n = 6$ sensor nodes. Furthermore, assume that each sensor node generates one packet every cycle, $n_{SD\_slots\_max} = 6$, $n_{GTS\_max} = 2$, and each slot accommodates one packet. In this case, relay c_1 needs $n/n_{GTS\_max} = 6/2 = 3$ SDs, where as the PAN coordinator c_0 requires only one SD every cycle.

In our work here, we assume that the PAN coordinator can afford to send as many beacon frames in one cycle as any other coordinator. This assumption allows the width of each stride in a typical cycle of the constructed schedule to be equal to the length of the beacon interval of the PAN coordinator, as in Figure 4.2.

4.6 Construction of an Optimized Flow Balanced Schedule

The previous section presents the concepts of scheduling SDs in parallel, and scheduling SDs in alternation by example. In this section we give formal definitions of the concepts, and present methods for recognizing situations where each concept can be applied.

To introduce the definitions, we use the following notation. We consider a problem instance of the schedule construction problem specified by a given weighted tree T. We denote by S a valid schedule serving the packets in T. A typical cycle of S may have one, or more, strides. We also let C denote a given set of two, or more, coordinators.

4.6.1 Scheduling SDs in Parallel

Given the above notation, we define the concept of scheduling SDs in parallel, as follows.

Definition 4.2 The set of SDs of the coordinators in C are scheduled in parallel if the following conditions hold:

1. some strides of the schedule S contains two, or more, SDs belonging to two (or more) coordinators of C,

2. all SDs have the same offset relative to the beginning of their respective strides, and

3. each cycle contains all SDs required by the k coordinators.

■

We now present a sufficient condition for the SDs of the coordinators in C to be schedulable in parallel. To state the condition, we first define conflict graphs between coordinators as follows.

Definition 4.3 For each coordinator c_x in C, denote by E_x the set of communication links between x and its children in the tree T. The conflict graph of the coordinators in T, denoted $G_{cc}(T)$, is a graph on the set of coordinators of T where two coordinators c_x and c_y are adjacent in the conflict graph if at least one link in E_x interferes with at least one link in E_y. ■

We now state the following observation, and omit the proof since it is straightforward.

Theorem 4.2 *The set of SDs of the coordinators in C can be scheduled in parallel if no two coordinators in C are adjacent to each other in the conflict graph $G_{cc}(T)$ (that is, the set C forms an independent set in $G_{cc}(T)$).*

We draw the following remarks.

1. The theorem presents a sufficient condition for the SDs of the coordinators in C to be schedulable in parallel. The condition, however, is not necessary. That is, it may be possible to schedule the SDs in parallel even in the presence of an interference between two communication links in the two sets E_x and E_y. Achieving this parallelism can be done by carefully assigning slots in each SD to the relevant communication links. Accommodating such additional optimization features, however, is left as a future work.

2. The problem of partitioning the set of coordinators in a given tree T so that

   (a) the coordinators in each partition satisfy the above theorem, and

   (b) we obtain the fewest possible number of partitions,

arises as a subproblem if we want to construct a schedule with the minimum possible cycle length. Such partitioning problem can be modeled as node coloring problem of the conflict graph $G_{cc}(T)$. Coordinator conflict graphs, however, are not general graphs with arbitrary structure. Hence, the computational complexity of the above minimum partitioning problem may not be NP-complete as is the case with the general graph coloring problem [14].

4.6.2 Scheduling SDs in Alternation

The concept of scheduling SDs in alternation can be defined as follows.

Definition 4.4 The set of SDs of the coordinators in C are scheduled in alternation if the following conditions hold:

1. each stride of the cycle has at most one SD of at most one of the k coordinators,
2. All SDs have the same offset relative to the beginning of their respective strides, and
3. each cycle contains all SDs required by the k coordinators.

■

We now present an algorithm for deciding whether a given set of coordinators can be scheduled in alternation. The algorithm uses the following input parameters:

- $n_{strides}$: the number of strides in a typical cycle of the schedule under construction.
- $(n_i : i = 1, 2, \cdots, k)$: n_i is the number of SDs required by the ith coordinator.
- $offset$: the offset at which each of the alternating SDs is scheduled to start relative to the beginning of its respective stride.

The algorithm produces a scheduling of the SDs in alternation if possible. Else, the algorithm returns with failure. The main insights in designing the algorithm are summarized below.

1. If the ith coordinator considered by the algorithm requires n_i SDs in each cycle to satisfy its requests, then the algorithm must allocate at least $n'_i \geq n_i$ to serve that particular coordinator.

2. The allocation of these n'_i SDs must be evenly spaced within a cycle so that the SDs appear cyclically in the schedule. That is, given that the cycle has $n_{strides}$ strides, the distance $\ell_i = n_{strides}/n'_i$ must be an integer.

3. The use of the distance ℓ_i makes the length of the beacon interval of the ith coordinator, denoted BI_i, equals to $\ell_i.BI_0$, where BI_0 is the length of the beacon interval of the PAN coordinator. We also recall that the restrictions imposed by the IEEE standard on the length of each beacon interval implies that ℓ_i must be a power of 2.

Putting the above observations together, we obtain Algorithm Check-Alternation, summarized in figure 4.3.

Algorithm Check-Alternation

1. Initialize the set of available strides: $available = \{1, 2, \cdots, n_{strides}\}$.

2. Process the requested number of strides $(n_i : \ i = 1, 2, \cdots, k)$ sequentially.
   for $i = 1, 2, \cdots, k$ {
   - 2.1 Compute the smallest integer $n'_i \geq n_i$ such that the distance $\ell_i = n_{strides}/n'_i$ is an integer, and ℓ_i is a power of 2. If no such n'_i exists then return with failure.
   - 2.2 If the set $available$ contains n'_i strides that are evenly spaced by the distance ℓ_i (e.g., $(1, 1 + \ell_i, 1 + 2\ell_i, \cdots)$) then allocate the n'_i strides to the SDs of the ith coordinator. Delete the allocated strides from the set $available$. In the final schedule, the SDs of the ith coordinator uses the offset value specified by the input $offset$ parameter.
   - 2.3 Else, if no such set of ℓ_i-spaced strides exists in the set $available$ then return with failure.

   }

Figure 4.3: Pseudo-code of Algorithm Check-Alternation

4.7 The Main Algorithm

Figure 4.4 presents the structure of the overall algorithm. The main aspect of the algorithm are summarized below.

1. The main loop of the algorithm (step 1) iterates over all possible values of the SO parameters in the IEEE standard (form 0 to 14). Each iteration tries to construct a schedule with a short cycle length.

2. At the end of each iteration, step 1.5 keeps the best valid schedule found thus far.

3. For a given SO value (and hence a given SD slot length) steps 1.1 through 1.2 compute the largest number of SDs required by each node, and set the number of strides in the constructed schedule ($n_{strides}$) to this value.

4. Step 1.3 initializes the schedule by assigning the SDs of the PAN coordinator to the beginning of each stride (i.e., $offset = 0$).

5. The loop in step 1.4 iterates over all non-sink coordinators sequentially level by level from top to bottom. Each coordinator is checked to see if it can be scheduled in parallel, or alternation along with some group of coordinators that the algorithm has already processed. Checking for parallel scheduling uses the precomputed coordinator conflict graph $G_{cc}(T)$.

Running Time. If n is the number of nodes in the tree T then the overall running time of the algorithm is determined by the execution of steps 1.4.1 and 1.4.2 which is $O(n^2)$.

4.8 Performance Evaluation

In this section we investigate the performance of three versions of the GTS-TDMA algorithm: (a) a baseline version that does not use either scheduling SDs in parallel nor scheduling SDs in alternation, (b) a baseline version that uses scheduling SDs in parallel only, and (c) a baseline version that uses scheduling SDs in alternation only. Our objective is to gain insight into the impact of using each optimization method on the overall performance of the algorithm. In particular, our work investigates the following aspects:

1. The ability of the baseline GTS-TDMA algorithm to deliver higher throughput over the throughput delivered by the CSMA-CA protocol of the IEEE 802.15.4 standard.

2. The possible performance gains obtained by scheduling SDs in parallel.

Algorithm GTS-TDMA

Input: A weighted tree network T as in Example 4.1, and its coordinator conflict graph $G_{cc}(T)$

Output: A flow balanced schedule S satisfying the traffic requirements, if the algorithm succeed.

1. For each possible SO value of the IEEE 802.15.4 construct a schedule where the length of each SD in the schedule is determined by the selected SO value.
   for $SO = 0, 2, \cdots, 14$ {
   - 1.1 Traverse the tree T level by level from bottom to top. Use the current SO value to compute for each node x its required number of SDs.
   - 1.2 Set $n_{strides}$ of the constructed schedule to the largest number of SDs required by any node in the tree.
   - 1.3 Allocate $n_{strides}$ SDs to the PAN coordinator, each SD starts at $offset = 0$ in its respective stride.
   - 1.4 Let $c_1, c_2, \cdots$ be the ordering of the non-sink coordinators in the tree T when the tree is traversed level by level from top to bottom. Process the coordinators sequentially in the order $c_1, c_2, \cdots$.
     for $i = 1, 2, \cdots$ {
     - 1.4.1 Determine if c_i can be scheduled in parallel along with a subset of coordinators that have been already processed. If such a subset exists, schedule c_i accordingly.
     - 1.4.2 Repeat the above step with respect to scheduling in alternation.
     - 1.4.3 If neither parallelism nor alternation is possible then increment the offset value and allocate SDs to the coordinator so as to maximize the length of its beacon interval.

     }
   - 1.5 If the length of the beacon interval of the PAN coordinator is valid (i.e., the corresponding $BO \leq 14$) then store the schedule if it has the sheerest cycle length encountered thus far.

   }

Figure 4.4: Pseudo-code of Algorithm GTS-TDMA

3. The possible performance gains obtained by scheduling SDs in alternation.

Below we describe the simulation parameters used in our investigations, and then present the obtained numerical results. Simulation of the IEEE 802.15.4 is made possible by the availability of an educational license to the QualNet 4.0 network simulator and the sensor network add-on library. Performance of the various versions of the GTS-TDMA algorithm is studied by developing a MATLAB implementation.

4.8.1 Simulation Parameters

The simulation study uses the following networks and parameters.

Network Topologies. Two parametrized classes of tree networks, denoted $T(n : d)$ and $T(n_1 : d_1, n_2 : d_2)$ are used in the study. They are defined as follows.

- $T(n : d)$ is a star network with n sensor nodes connected to the PAN coordinator. Each sensor node is placed d meters away from the PAN coordinator. The n sensor nodes form a circle around the coordinator.

- $T(n_1 : d_1, n_2 : d_2)$ is a tree network of depth 2. The PAN coordinator (at level 0) is connected to n_1 relay nodes (at level 1 of the tree). Each relay node is placed d_1 meters away from the PAN coordinator. Each relay is connected to n_2 sensor nodes at level 2. Each sensor node is placed d_2 meters away from its serving relay. In the special case where the network has $n_1 = 2$ relays, the two relays and the PAN coordinator are placed on a straight line where the two relays lie on two opposite sides of the PAN coordinator. In addition, the n_2 sensor nodes connected to each relay node form a half circle whose center is the relay node.

Other Parameters. The following is a list of the important traffic, MAC layer, physical layer parameters.

1. Traffic from the sensor nodes to their serving relay nodes is made of fixed length packets of $L = 1376$ bits each (120 bytes of application data plus 52 bytes header).

2. Simulation duration of the CSMA-CA protocol in our work is 3000 seconds; the application traffic runs for 2000 seconds; the first 1000 seconds allows the nodes to associate with the PAN coordinator.

3. Simulation of the CSMA-CA protocol sets the length of the active part of each beacon interval using a superframe order $SO = 3$ so that each slot of the active period can accommodate one packet of $L = 1376$ bits.

4. The IEEE 802.15.4 is assumed to operate at an over air data rate of 250 kbps (the standard channel rate when operating in the 2.4 GHz ISM band).

5. For each node, the transmission radius R_T is controlled by the node's transmission power P_T. QualNet uses a two-ray path loss model (path loss exponent= 4).

4.8.2 Baseline GTS-TDMA versus CSMA-CA

To compare the achieved throughput of the baseline GTS-TDMA algorithm with the throughput achieved by the CSMA-CA algorithm of the standard, we use the following settings.

1. The WSN is the star network $T(n = 40, d = 10 \text{ meters})$. The transmission radius of of each node is set to $R_T = 10$ meters.

2. The aggregate traffic generated by the 40 sensor nodes and destined to the sink node (the PAN coordinator) is Poisson with average λ in the interval $[40, 180]$ packets/second.

Figure 4.5 illustrates the achieved throughput by each method. We note that the GTS-TDMA can achieve twice the throughput value achieved by the CSMA-CA algorithm. In the context of solving the class-based rate differentiation feasibility problem (Problem 1 introduced in chapter one), the obtained results indicate that many instances of the problem will be declared infeasible when the network runs under the control of the CSMA-CA protocol, yet the instances can be declared feasible when the network uses the devised baseline GTS-TDMA algorithm.

4.8.3 Effect of Scheduling SDs in Parallel

To evaluate the possible throughput improvements gained by scheduling SDs in parallel (whenever possible), we use the following settings.

1. The WSN is the tree $T(n_1 = 2 : d_1 = 15 \text{ meters}, n_2 = 7 : d_2 = 10 \text{ meters})$. All three coordinators (the PAN coordinator and its two children) are placed

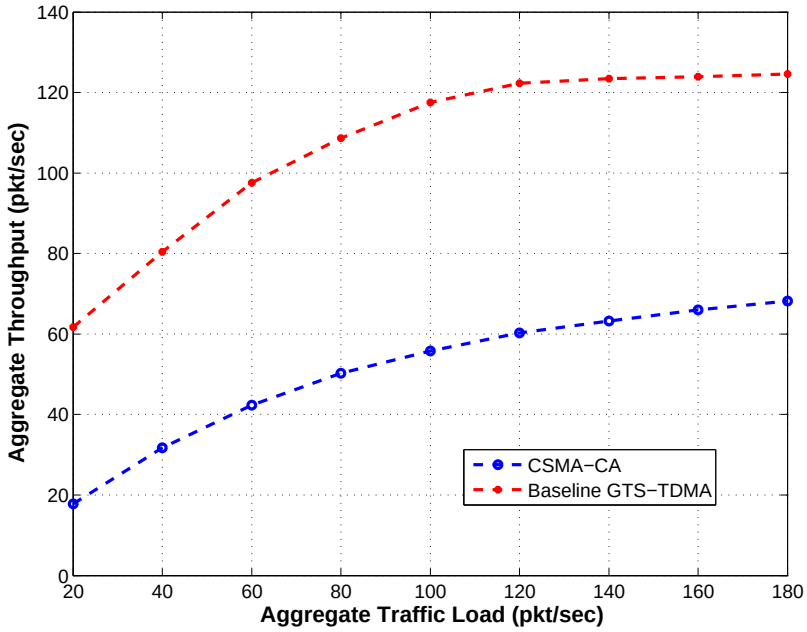

Figure 4.5: Throughput of baseline GTS-TDMA algorithm

on a straight line with the PAN coordinator in the center. The 7 nodes connected to each of the two coordinators lie on a half circle. The transmission radius of of each node is set to $R_T = 15$ meters.

2. The aggregate traffic generated by the 14 sensor nodes and destined to the sink node (the PAN coordinator) is Poisson with average λ in the interval $[10, 90]$ packets/second.

Figure 4.6 illustrates the achieved throughput by the baseline GTS-TDMA algorithm and the enhanced algorithm with the parallel optimization feature enabled. The results show a notable increase in the achieved throughput by the optimized algorithm. This increase is attributed to the ability of the algorithm to detect that the SDs of the two relays at level 1 of the tree can be scheduled in parallel; thus, reducing the schedule cycle length.

4.8.4 Effect of Scheduling SDs in Alternation

Similar to the simulation setup used in the previous section, to evaluate the possible throughput improvements gained by scheduling SDs in alternation (whenever possible), we use the following settings.

1. The WSN is the same tree network used in the previous section. However,

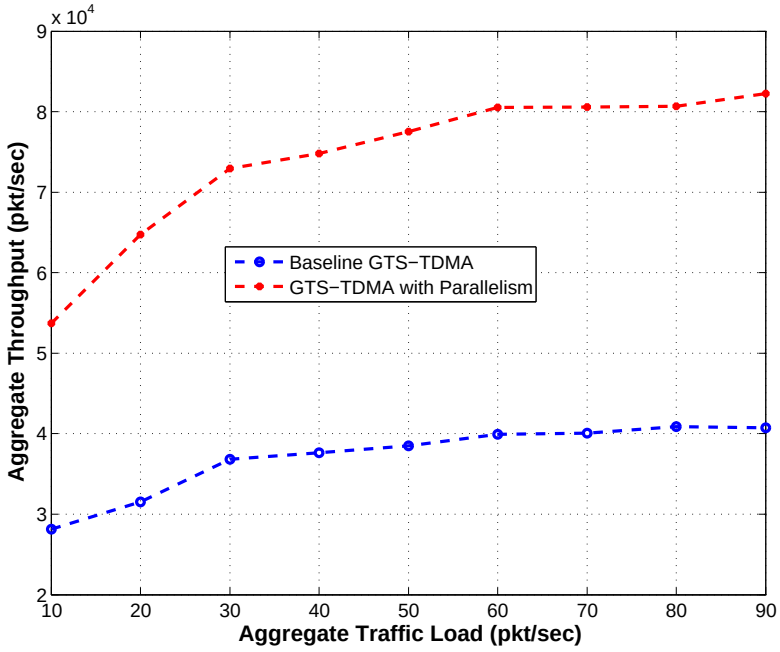

Figure 4.6: Effect of scheduling SDs in parallel

we set the transmission radius of each node to $R_T = 50$ meters. Hence, scheduling SDs in parallel is not possible for the tree.

2. The aggregate traffic generated by the 14 sensor nodes and destined to the sink node (the PAN coordinator) is Poisson with average λ in the interval $[10, 90]$ packets/second.

Figure 4.7 illustrates the achieved throughput by the baseline GTS-TDMA algorithm and the enhanced algorithm with the alternation optimization feature enabled. Again, the results show a notable increase in the achieved throughput by the optimized algorithm. This increase is attributed to the ability of the algorithm to construct a schedule where in each cycle, the SDs of the two relays at level 1 of the tree to be scheduled in alternation. Here, each relay is required to collect half the packets destined to the sink. As a result, the length of the beacon interval of each of the two relays is twice that of the PAN coordinator.

4.9 Summary

In this chapter we have identified relevant constraints imposed by the IEEE 802.15.4 standard on the use of GTSs in a WSN with multiple coordinators. We then developed a framework for constructing TDMA schedules under such constraints. In

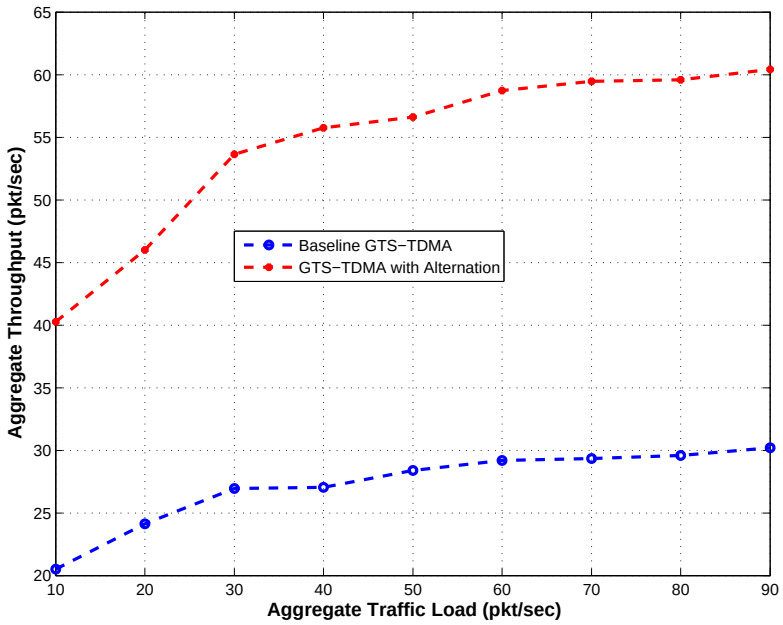

Figure 4.7: Effect of scheduling SDs in alternation

the developed framework each schedule cycle is composed of a number of strides where the length of each stride is equal to the length of the beacon interval of the PAN coordinator. Working with this framework has enabled us to construct a baseline algorithm for constructing TDMA schedules.

We have also identified two optimization methods that can be used to shorten the length of a schedule cycle. The two methods are referred to as scheduling SDs in parallel, and scheduling SDs in alternation. The obtained simulation results indicate that the throughput achieved by the developed algorithm improves on the throughput achieved by the CSMA algorithm of the IEEE 802.15.4 standard. In addition, the results illustrate the improvements in throughput made possible by utilizing the devised optimization methods in conjunction with the devised baseline algorithm.

Chapter 5

Future Research

Below, we briefly discuss some possible future research directions.

1. The problem formulated in Chapter 2 assumes that the routing tree of a WSN is given as part of the problem specification. In practice, only exact or approximate locations of the sensor nodes are given. By adjusting the transmission power of each node, different routing trees can be realized. The use of different routing trees, however, may result is different answers to any given instance of any class size optimization problem considered in the thesis. Moreover, different routing trees can potentially results in different energy consumption levels for each node. Thus, many interesting problems arise in computing routing trees that satisfy various possible rate differentiation and energy consumption requirements. Formalizing such problems and investigating possible solutions using different types of MACs appears to be a research direction worth investigating.

2. The Markov chain model discussed in Chapter 2 analyzes a modified IEEE 802.15.4 CSMA-CA protocol under saturation conditions (i.e., each node is assumed to be backlogged with packets to send). If the nodes are not saturated, then it is expected that more nodes can be accommodated while still satisfying the required data rate for each class. We propose developing a model for the modified protocol under non-saturation conditions.

3. The Markov chain model discussed in Chapter 2 assumes a star network under the control of a modified IEEE 802.15.4 CSMA-CA protocol. If the sensor nodes are spread over a large geographic area, the use of a multi-level network with relays becomes necessary. Assuming all nodes use the modified CSMA-CA protocol, it becomes interesting to investigate the achieved

throughput for each class when such a multi-level tree is used.

4. In Chapter 4, we developed a method for scheduling the SDs of two coordinator x and y under the strict condition that no interference exists between a link in E_x and a link in E_y. We noted that scheduling two SDs in parallel may be possible even in the existence of some interferences between links in E_x and E_y. In this case, a third optimization method can be developed to schedule these SDs in parallel while preventing simultaneous transmission of conflicting links by assigning GTS slots to the links wisely.

5. In this thesis, we consider class size optimization problems in single-channel networks. In some of the applications of WSNs multiple sinks are placed for gathering data from the sensor nodes. In such networks, one may use a separate channel for each sink. A problem arises on how to assign nodes to the available channels so as to conserve energy consumption and satisfy the required data rates. An efficient solution to this problem provides a tool to handle the class size optimization problems of interest in multi-channel networks.

Bibliography

[1] A. Koubaa and M. Alves and B. Nefzi and Y. Q. Song. Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks. *Workshop on Real Time Networks (RTN)*, July 2006.

[2] M. Alicherry, R. Bhatia, and L. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 58–72, New York, NY, USA, 2005. ACM.

[3] Anis Koubaa and Mário Alves and Eduardo Tovar. GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks. In *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, 2006.

[4] Anis Koubaa and Mário Alves and Eduardo Tovar. i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks. In *ECRTS '06: Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 183–192, Washington, DC, USA, 2006. IEEE Computer Society.

[5] Chewoo Na and Yaling Yang and Amitabh Mishra. An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks. *Computer Networks*, 52(13):2543 – 2557, 2008.

[6] Duminda Dewasurendra and Amitabh Mishra. Design challenges in energy-efficient medium access control for wireless sensor networks. In Mohammad Ilyas and Imad Mahgoub, editors, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, chapter 28. CRC Press LLC., 2004.

[7] ZigBee Alliance Document. Zigbee specification, version 1.0. *http://www.zigbee.org/*, 2005.

[8] E. Callaway and P. Gorday and L. Hester and J.A. Gutierrez and M. Naeve and B. Heile and V. Bahl. Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine*, 40(8):70–77, August 2002.

[9] E. Kim and M. Kim and S. Youm and S. Choi and C. Kang. Priority-Based Service Differentiation Scheme for IEEE 802.15.4 Sensor Networks. *International Journal of Electronics and Communications(AEU)*, 61(2):69–81, February 2007.

[10] Shahin Farahani. *ZigBee Wireless Networks and Transceivers*. Elsevier Inc, Oxford, UK, 2008.

[11] Feng Chen. Improving IEEE 802.15.4 for Low-latency Energy-efficient Industrial Applications. In *Proceedings of Echtzeit 2008 (Fachtagung der GI-Fachgruppe Echtzeitsysteme Boppard)*, pages 61–70, Germany, 2008.

[12] G. Lu and B. Krishnamachari and C. S. Raghavendra. Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks. *IEEE International Performance Computing and Communications Conference (IPCCC)*, April 2004.

[13] S. Gandham, M. Dawande, and R. Prakash. Link scheduling in sensor networks: distributed edge coloring revisited. In *INFOCOM 2005.*, volume 4, pages 2492–2501 vol. 4, March 2005.

[14] Michael R. Garey and David S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.

[15] P. Gupta and P. R. P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[16] T. Hester, R. Hughes, D. M. Sherrill, B. Knorr, M. Akay, J. Stein, and P. Bonato. Using wearable sensors to measure motor abilities following stroke. In *BSN '06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pages 5–8, Washington, DC, USA, 2006. IEEE Computer Society.

[17] I. Demirkol and C. Ersoy and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.

[18] I. Ramachandran and A. K. Das and S. Roy. Analysis of the Contention Access Period of IEEE 802.15.4 MAC. *ACM Transactions on Sensor Networks (TOSN)*, 3(1), March 2007.

[19] IEEE Computer Society. PART 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE P802.15.4a/D5*, 2006.

[20] J. Zheng and M. J. Lee. A Comprehensive Performance Study of IEEE 802.15.4. *IEEE Press Book*, 2004.

[21] Abdul Kader Kabbani, Theodoros Salonidis, and Edward W. Knightly. Distributed low-complexity maximum-throughput scheduling for wireless backhaul networks. In *INFOCOM 2007*, pages 2063–2071, 2007.

[22] T. Kim and S. Choi. Priority-based delay mitigation for event-monitoring ieee 802.15.4 lr-wpans. *IEEE Communications Letters*, 10(3):213–215, 2006.

[23] Murali Kodialam and Thyaga Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 42–54, New York, NY, USA, 2003. ACM.

[24] N.L. Lai, C.T. King, and C.H. Lin. On maximizing the throughput of convergecast in wireless sensor networks. *In Proceedings of International Conference on Grid and Pervasive Computing (GPC)*, May 2008.

[25] M. Kohvakka and M. Kuorilehto and M. Hnnikinen and T. D. Hmlinen. Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications. *In Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (PE-WASUN '06)*, page 4857, August 2006.

[26] Jelena Misic and Vojislav B. Misic. Access delay for nodes with finite buffers in IEEE 802.15.4 beacon enabled PAN with uplink transmissions. *Computer Communications*, 28(10):1152 – 1166, 2005.

[27] Jelena V. Misic, Shairmina Shafi, and Vojislav B. Misic. Performance limitations of the MAC layer in 802.15.4 low rate WPAN. *Computer Communications*, 29(13-14):2534–2541, 2006.

[28] Quanhong Wang and Hossam S. Hassanein. A Comparative Study of Energy-Efficient (E2) Protocols for Wireless Sensor Networks. In Mohammad Ilyas and Imad Mahgoub, editors, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, chapter 18. CRC Press LLC., 2004.

[29] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking*, 1(2):166–177, 1993.

[30] K. Romer, F. Mattern, and E. Zurich. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.

[31] S. C. Ergen and P. Varaiya. TDMA Scheduling Algorithms for Sensor Networks. *Technical Report, Department of Electrical Engineering and Computer Sciences University of California*, July 2005.

[32] K. M. Sivalingam, J. C. Chen, P. Agrawal, and M. B. Srivastava. Design and analysis of low-power access protocols for wireless and mobile atm networks. *Wireless Networking*, 6(1):73–87, 2000.

[33] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7:16–27, 2000.

[34] T. Kim and D. Lee and J. Ahn and S. Choi. Priority toning strategy for fast emergency notification in IEEE 802.15.4 LR-WPAN. *in Proceedings of the 15th Joint Conference on Communications & Information (JCCI)*, April 2005.

[35] Tae Rim Park and Tae Hyun Kim and Jae Young Choi and Sunghyun Choi and Wook Hyun Kwon. Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA. *IEEE Electronics Letters*, 41:1017 – 1019, 2005.

[36] Scalable N. Technologies. Qualnet. http://www.scalable-networks.com/.

[37] Yu-Kai Huang and Ai-Chun Pang and Hui-Nien Hung. An Adaptive GTS Allocation Scheme for IEEE 802.15.4. *IEEE Transactions on Parallel and Distributed Systems*, 19(5):641–651, 2008.