

# MAC Protocols for Low-latency and Energy-efficient WSN Applications

Pardeep Kumar      Mesut Güneş  
Computer Systems and Telematics  
Distributed Embedded Systems  
Institute of Computer Science  
Freie Universität Berlin, Germany  
 [{pkumar, guenes}@inf.fu-berlin.de](mailto:{pkumar, guenes}@inf.fu-berlin.de)

Technical Report  
TR-No. TR-B-08-18

December, 2008



---

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Introduction and Motivation . . . . .	3
1.2	MAC Characteristics for WSN . . . . .	4
1.3	Contribution . . . . .	5
1.4	Document Structure . . . . .	5
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	S-MAC . . . . .	7
2.2	T-MAC . . . . .	8
2.3	DSMAC . . . . .	8
2.4	B-MAC . . . . .	8
2.5	STEM . . . . .	8
2.6	TRAMA . . . . .	9
2.7	LEACH . . . . .	9
2.8	WiseMAC . . . . .	10
2.9	IEEE 802.15.4 . . . . .	10
2.10	Discussion of the Presented MAC Protocols . . . . .	11
<b>3</b>	<b>The IEEE 802.15.4 Standard</b>	<b>13</b>
3.1	PHY Layer . . . . .	14
3.2	MAC Layer . . . . .	14
3.2.1	MAC Superframe Structure . . . . .	15
3.3	IEEE 802.15.4 Limitations . . . . .	17
3.4	IEEE 802.15.4-based System Model . . . . .	18
3.4.1	A Proposed Solution . . . . .	18
3.5	Selection of Critical Parameters . . . . .	19
3.5.1	Beacon Interval & Superframe Duration . . . . .	20
3.5.2	Maximum CFP Slots available . . . . .	20
3.5.3	CFP Slots actually needed . . . . .	20
3.5.4	Superframe Order . . . . .	22
3.5.5	Bandwidth Under-utilization . . . . .	23
<b>4</b>	<b>AREA-MAC</b>	<b>25</b>
4.1	Basic Design Concepts . . . . .	25
4.1.1	Assumptions . . . . .	27
4.2	Network Model . . . . .	27
4.3	Time and Energy Saving for AREA-MAC . . . . .	28
4.4	Design Phases . . . . .	30

4.4.1	Network Setup Phase . . . . .	30
4.4.2	Periodic Duty Cycle Phase . . . . .	30
4.4.3	Adaptive Duty Cycle Phase . . . . .	30
4.4.4	Re-scheduling Phase . . . . .	30
4.5	Energy Model . . . . .	31
4.6	Delay Model . . . . .	34
4.7	LP Formulation . . . . .	35
<b>5</b>	<b>Conclusion and Future Work</b>	<b>37</b>

---

# List of Figures

1.1	An health care scenario . . . . .	4
3.1	IEEE 802.15.4 protocol architecture . . . . .	15
3.2	MAC superframe structure . . . . .	16
3.3	A system model . . . . .	18
3.4	GTS characteristics field . . . . .	19
3.5	BI-BO and SD-SO relation . . . . .	20
3.6	Maximum available CFP slots . . . . .	21
3.7	Data packets and required CFP slots (with acknowledgement) . . . . .	22
3.8	Data packets and required CFP slots (without acknowledgement) . . . . .	23
3.9	An example of bandwidth under-utilization . . . . .	23
4.1	The different roles of a node . . . . .	26
4.2	A grid-based WSN . . . . .	27
4.3	A time/energy gain comparison . . . . .	29
4.4	A process cycle for the real-time request . . . . .	34



---

# List of Tables

2.1	Comparison of different MAC protocols . . . . .	11
3.1	IEEE 802.15 WPAN standards . . . . .	14
3.2	Data packets and GTS slots . . . . .	22
4.1	Terms used for the network model . . . . .	29
4.2	Terms used for the energy model . . . . .	32





---

# Abstract

Most of medium access control (MAC) protocols proposed for wireless sensor networks (WSN) are targeted only for single main objective, the energy efficiency. Other critical parameters such as low-latency, adaptivity to traffic conditions, scalability, system fairness, and bandwidth utilization are mostly overlooked or dealt as secondary objectives. The demand to address those issues increases with the growing interest in cheap, low-power, low-distance, and embedded WSNs. In this report, along with other vital parameters, we discuss suitability and limitations of different WSN MAC protocols for time critical and energy-efficient applications. As an example, we discuss the working of IEEE 802.15.4 in detail, explore its limitations, and derive efficient application-specific network parameter settings for time, energy, and bandwidth critical applications. Eventually, a new WSN MAC protocol Asynchronous Real-time Energy-efficient and Adaptive MAC (AREA-MAC) is proposed, which is intended to deal efficiently with time critical applications, and at the same time, to provide a better trade-off between other vital parameters, such as energy-efficiency, system fairness, throughput, scalability, and adaptivity to traffic conditions. On the other hand, two different optimization problems have been formulated using application-based traffic generating scenario to minimize network latency and maximize its lifetime.



---

# CHAPTER 1

---

## Introduction

### 1.1 Introduction and Motivation

Wireless sensor networks (WSNs) [1, 2] are designed to play a huge role to our future ubiquitous world. The demands placed on such type of networks are expending exponentially with the increase in their dimensions. However, these networks are different from traditional networks and pose several challenges, such as harsh resources, low communication ranges, self-organization, error-prone conditions, ad hoc deployment, unattended operation, and dynamic environment conditions. Low communication ranges confirm the dense deployment of sensors and only an efficient medium access control (MAC) protocol can handle number of medium-sharing nodes in a better way and form an efficient infrastructure to establish communication links between nodes.

The research community has witnessed the intense research related to the WSN MAC protocols over the last years. Various MAC protocols proposed for WSNs [3, 4, 5, 6, 7, 8, 9, 10, 11] are mainly designed for single main objective, the energy efficiency. The energy consumption, no doubt, is a most critical parameter for WSN performance, but it should not be the *only* focal point. The growing interest in cheap, low-power, low-distance, and embedded WSNs attracts researchers to find out solutions for remaining problems so that they can be efficiently utilized on a large scale, especially for the environments where timeliness is vital. For medical urgency, surveillance, security, terrorist attacks, home automation, flood, fire, and seismic detection applications, the provision of real-time guarantees is as crucial as saving the energy. For example; a sensor node embedded in an e-textile worn by patients should automatically but timely alert doctors or emergency services when a patient suffers from severe disease. And, nodes must lively inform the security and emergency services about the persons, wounded by a terrorist bomb blast, rather than saving the energy at that critical time.

In clinical diagnostics, traditional paper based patient monitoring is difficult, complex, and expensive. The increase in the world population, diseases, unfortunate incidents, and with the inadequate number of doctors and clinicians available around, especially in third world countries, highlights the need of an automatic system. Such systems should efficiently but timely alert medical staff for any mishap or remind them for their scheduled checkups, regardless to the personal's physical location. It can also be used to store patient records for future research and automatic medication purposes. An example for such applications is shown in Figure 1.1, where the energy could be a factor to be efficiently handled, but the grandness of timeliness increases sharply.

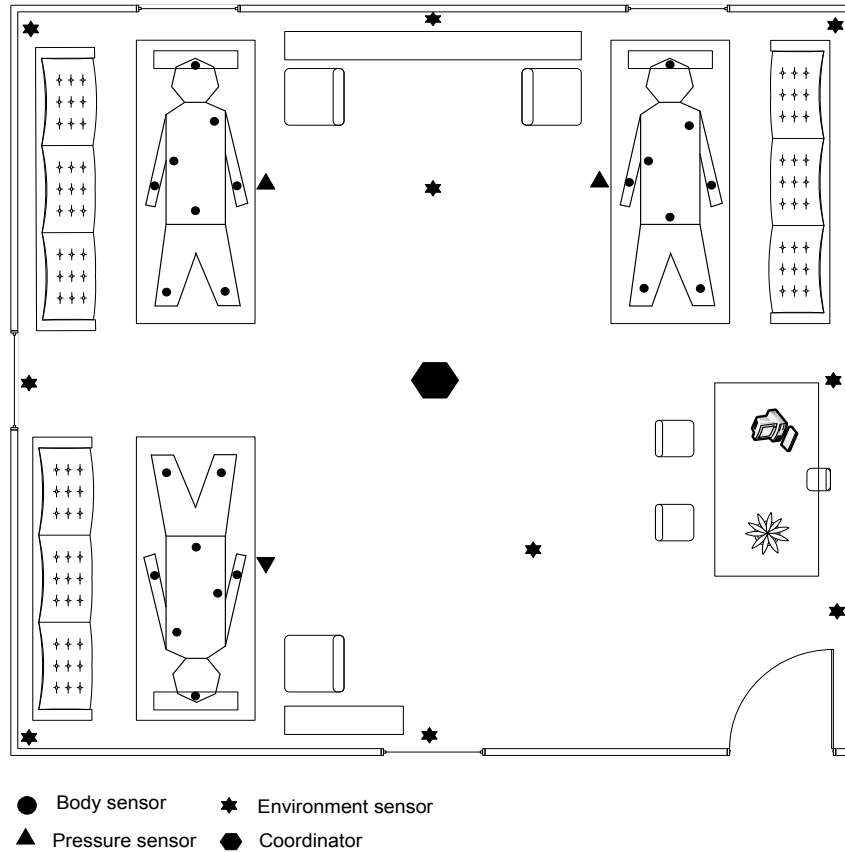


Figure 1.1: An health-care scenario. Some nodes are attached to the patient body to measure different biological parameters, such as heart and pulse rate, blood pressure, blood oxygen saturation, electrocardiogram (ECG), and electroencephalograph (EEG) values. Other nodes are deployed inside the room to measure the physical parameters, such as temperature, air pressure, humidity, and the light values. The pressure sensors, deployed under/near the patient bed, timely alert the medical staff for an unexpected downfall of the patient. The coordinator or the sink node is responsible to collect and forward the traffic generated by these nodes.

Unlike traditional distributed systems, the real-time guarantee for WSNs is more challenging. They interact directly with the real world, where the physical events occur in an unpredictable manner with different traffic and delay requirements. The factors like duty-cycling, i.e., putting the radio in sleep mode periodically and system fairness also restrict the design space we could trade off. A WSN MAC protocol specifies how nodes share the channel, avoid collision in correlated environment, response the inquirer timely, save the energy, and survive for longer period. Hence, the importance of designing novel solutions for WSN MAC protocol increases dramatically.

## 1.2 MAC Characteristics for WSN

In general, the fundamental task of any MAC protocol is to regulate the access of nodes to a shared medium in an efficient way. In case of WSN, the MAC protocols are responsible to deal with some additional requirements. The importance of timeliness for the WSN MAC protocol is already discussed. As energy is a scarce resource in WSN, therefore

every attempt should be made to minimize the energy wastage. The main sources of energy wastage are collision, overhearing, control packet overhead, idle listening, and over-emitting. The collision is a wasted effort when two frames collide and results in retransmission. An overhearing occurs when a node receives and processes a gratuitous packet not addressed for it. An unnecessarily increase in number and size of control packets results in more overhead and energy wastage. In idle listening, a node keeps its radio in ready-to-receive mode, which consumes almost as much energy as receive mode. An over-emitting occurs due to the transmission of a message when the destination node is not ready. Duty cycling is considered as one of the best solutions to overcome many energy wastage problems.

In addition, a MAC protocol should ensure high throughput, low overhead even with traffic fluctuations, both in time and space, low error rates, scalability, self-stabilization, and graceful adaption to topology changes.

### 1.3 Contribution

The main objective of this report is to design a novel MAC protocol for WSNs, which could efficiently be utilized for real-time and energy efficient applications. The proposed MAC protocol is intended to provide application-specific optimized performance in terms of timeliness and energy efficiency, while maintaining an acceptable system fairness and reasonable trade-off between different critical parameters. We aim to compare the proposed MAC protocol with state of the art protocols by using several vital metrics, such as end-to-end delay, energy consumption, through-put, and bandwidth efficiency.

### 1.4 Document Structure

The remainder of this report is organized as follows. In Chapter 2, we elaborate the state of the art, where we discuss some of the well-known MAC protocols proposed for WSNs. Afterwards in Chapter 3, we talk about the IEEE 802.15.4 standard in detail and rectify several limitations of the standard. We also propose a solution to overcome those limitations, and outline the application-specific optimal parameter setting for real-time, energy, and bandwidth critical IEEE 802.15.4 based applications. Subsequently, in Chapter 4, we propose a new MAC protocol and formalize two optimization problems for delay-bound and energy-efficient WSNs scenarios. In the last Chapter 5, we conclude our work.



---

## CHAPTER 2

---

### Related Work

Most of the WSN MAC protocols proposed so far are designed with the goal to conserve the energy. Other goals like latency, throughput, adaptation to traffic conditions, and scalability are often traded-off for energy conservation. There is not any generic *best* MAC protocol; the design choice mainly depends on the nature of the application. Broadly, these protocols can be classified into two categories: *contention-based* and *schedule-based* protocols.

In contention-based MAC protocols nodes compete to acquire the channel. These protocols are designed for minimum delay and maximum throughput and require transceivers to monitor the channel at all times. In unlucky cases, for example, due to hidden-node problem, a collision might occur, resulting in energy wastage and possible retransmission of packets. In scheduling-based protocols, a schedule regulates which participant may use which resource at what time. The schedule can be fixed or computed on demand (or a mixed). Though collisions, overhearing, and idle listening are not the issues here, but the overhead caused by time synchronization and the latency are the major concerns. The well-known MAC protocols from both categories are discussed below.

#### 2.1 S-MAC

The Sensor-MAC (SMAC) protocol [3] circumvent idle listening, collisions, and overhearing by using periodic and fixed-length wake-up and sleep periods according to its *schedule*. S-MAC attempts to coordinate the schedules of neighboring nodes for their listen and sleep periods. The listen period consists of SYNCH, RTS, and CTS phases. In the synchronization (SYNCH) period, a node accepts SYNCH packets from its neighbors and stores in its schedule table. In the RTS (request-to-send) phase, a node listens for RTS packets from its neighbors and in the CTS (clear-to-send) phase, a node transmits a CTS packet if a RTS packet was received in the previous phase. S-MAC allows neighboring nodes to agree on the same schedule and to create schedule-based virtual clusters. S-MAC also includes the concept of message passing, in which long messages are divided into frames and sent in a burst. With this technique, one may achieve energy savings by minimizing communication overhead at the expense of unfairness in medium access. S-MAC can significantly reduce idle listening, but it is rigid and optimized for a predefined set of workloads; as it is hard to adapt the length of the wakeup and sleep periods to changing load situations. Synchronization and longer sleep periods pay the price in terms of latency. Another drawback is the possibility of following two different schedules, which

results in more energy consumption via idle listening and overhearing.

## 2.2 T-MAC

Time-out MAC (TMAC) [9] protocol is similar to S-MAC but adaptively shortens the listen period. The listen period ends when no activation event has occurred for a time threshold. It reduces idle listening by transmitting all messages in bursts of variable length, and sleeping between bursts. Although T-MAC gives better results under variable loads (in variable workloads, it uses one fifth of the power of S-MAC), the synchronization of the listen periods within virtual clusters is broken. This is one of the reasons for the early sleeping problem (node goes to sleep when a neighbor still has messages for it). T-MAC saves power at a cost of reduced throughput and additional latency and suffers with complexity and scaling problems.

## 2.3 DSMAC

Dynamic Sensor-MAC (DSMAC) [8] improves the latency over SMAC by dynamically adjusting duty-cycle. All nodes start with the same duty cycle and share their one-hop latency values in SYNC period. When a receiver node notices that the average one-hop latency value is high, it decides to shorten its sleep time. Accordingly, after a sender node receives this sleep-period decrement signal, it checks its queue for packets destined to that receiver node. If there is one, it decides to double its duty cycle when its battery level is above a specified threshold. The latency observed with DSMAC is better than that observed with S-MAC, but it uses only 40% of the frame duration, and thus, achieves less throughput for high traffic.

## 2.4 B-MAC

The University of California, at Berkeley, has developed a CSMA-based B-MAC [11] protocol, which uses low power listening (LPL) with an extended preamble to reduce duty cycle and minimize idle listening. B-MAC supports on-the-fly reconfiguration and provides bidirectional interfaces for system services to optimize performance. Nodes have an awake and a sleep period, and each node can have an independent schedule. While transmitting, a node precedes the data packet with a preamble that is slightly longer than the sleep period of the receiver. During the awake period, a node samples the medium and if a preamble is detected it remains awake to receive the data. With the extended preamble, a sender is assured that at some point during the preamble the receiver will wake up, detect the preamble, and remain awake in order to receive the data. While the authors claim better performance over other protocols, B-MAC suffers from the overhearing problem and the long preamble dominates the energy usage.

## 2.5 STEM

Sparse Topology and Energy Management (STEM) [7] uses two different channels, the wakeup channel and the data channel, and requires two transceivers in each node. On



the wakeup channel, if incoming signals are found, only then transmitter and receiver communicates on data channel, otherwise the data channel is always in sleep mode. The network has a monitor state, where the nodes remain idle and a transfer state, where nodes sense and communicate. To have the attention of the receiver, two different variants are used. In STEM-B, the transmitter wakes up the receiver by sending a beacon on the wakeup channel (no RTS/CTS), which indicates source and destination addresses. In STEM-T, the transmitter sends busy tone signal on the wakeup channel, but it contains no destination address and in result, all the neighbors will sense the busy tone and switch on their data channel. In STEM-B, more than one transmitter might send their beacon simultaneously, resulting in beacon collisions, whereas, in STEM-T, all neighbors (including destined node) receive busy tone and switch on their receiver unnecessarily.

## 2.6 TRAMA

TRAffic-Adaptive Medium Access (TRAMA) protocol [5] is mostly a TDMA-based protocol and creates the schedules in distribute and on-demand basis. It assumes that all nodes are time synchronized and divides the time cycle into random access and scheduled-access periods. The random-access period is used to establish two hop topology information by broadcasting neighborhood information and uses contention-based access. Nodes broadcast their schedule information containing an update list of receivers for the packet to their neighbors and execute a distributed scheduling algorithm to determine receiving, transmitting and sleeping nodes. TRAMA consists of three components: the Neighbour Protocol (NP), the Schedule Exchange Protocol (SEP) and the Adaptive Election Algorithm (AEA). NP works during the random access period and gets the two-hop topology information. During SEP, node transmits its current transmission schedule and also picks up it neighbor's schedules. To compute its schedule, a node computes its own priority and the priority of all its two-hop neighbors for each time slot. AEA selects transmitters and receivers to achieve collision free transmission and uses traffic information to improve the channel utilization. Though, TRAMA achieves higher percentage of sleep time and less collision probability as compared to CSMA-based protocols, but all nodes are defined to be either in receive or transmit states during the random-access period for schedule exchanges and for each time slot, every node calculates each of its own and two-hop neighbor's priorities. It results in significant computation and memory in dense sensor network since the two-hop neighborhood can be large enough. Therefore, TRAMA is a feasible for networks having sufficient resources.

## 2.7 LEACH

Low-Energy Adaptive Clustering Hierarchy (LEACH) [4] divides the dense and homogeneous sensor networks into clusters supervised by the clusterheads. Each clusterhead is responsible for creating and maintaining a TDMA schedule and is always switched on and therefore, the chances of clusterhead to die sooner are bright. To avoid the situation of "headless", LEACH rotates the selection of clusterhead. Each node independently can decide to become clusterhead, considering the last time, when it was clusterhead. The member nodes select their clusterhead on the basis of received signal strength. LEACH works in rounds, and each round is divided into setup and steady-state phases. Cluster formation occurs in setup phase, where each clusterhead node broadcasts an advertise-

ment message (ADV) using CSMA protocol. Each non-clusterhead node transmits a join-request message (REQ) using CSMA MAC Protocol by determining minimum communication energy and largest signal strength. Each clusterhead node sets up a TDMA schedule, picks a random CDMA code, and broadcasts this information; which ensures that there is no collision in data messages. The radio components are turned off at all times except during transmit time. During the steady-state phase, nodes transmit their data in the corresponding slots. Although, LEACH guarantees that each member node belongs to at most one cluster, but due to ADV collision, it does not guarantee that each member node belongs to a cluster. In that case, this protocol considers that all nodes are within the range of the sink node and hence limits the network scalability. It also considers that nodes always have data to send in the allotted time. Perfect correlation is assumed, which might not be true always.

## 2.8 WiseMAC

WiseMAC [6] is the first protocol working on non-persistent CSMA (np-CSMA) with preamble sampling technique to decrease the idle listening. The idea is to start transmitting a packet just before the intended receiver wakes up to sample the channel. Nodes do not need to be explicitly synchronized. All nodes in the network sample the medium with a common basic cycle duration, but their wake-up patterns are independent and left unsynchronized. If a node finds the medium busy after it wakes up and samples the medium, it continues to listen until it receives a data packet or the medium becomes idle again. WiseMAC uses short preambles for regular traffic and switches to longer preambles for infrequent communication. Overemitting can occur, if the receiver is not ready at the end of the preamble, due to factors such as interference and it can be increased further with the length of the preamble and the data packet, since no handshake is done with the intended receiver. Its decentralized sleep-listen scheduling can result in different sleep and wake-up times for each neighbor of a node. This is an important problem especially for broadcast-type communication, since broadcasted packets will be buffered for neighbors in sleep mode and delivered many times as each neighbor wakes up. However, this redundant transmission will lead in higher latency and power consumption. In addition, the hidden terminal problem can spring up with this protocol. This problem will result in collisions when one node starts to transmit the preamble to a node that is already receiving another node's transmission where the preamble sender is not within range.

## 2.9 IEEE 802.15.4

The IEEE 802.15.4 standard [12] is emerging as an important building block for WSN communication stack. It provides several attractive features to support unique WSN characteristics. Chapter 3 deals with the detailed working of IEEE 802.15.4. Here we outline some of the research conducted on IEEE 802.15.4 with respect to low-latency, energy, and bandwidth critical WSN applications.

Most of the IEEE 802.15.4 related research has been subjected to the CSMA-CA and general performance evaluation. Only a small amount of the literature is available for IEEE 802.15.4-based timeliness and real-time related applications. An implicit GTS allocation scheme (i-GAME) for time-sensitive WSN is proposed in [13]. The coordinator uses admission control algorithm to decide whether to accept new GTS request or not,

based on the traffic specification of the flows, their delay requirements and bandwidth resources. They show that their proposal improves the bandwidth utilization compared to the explicit allocation used in the IEEE 802.15.4 protocol standard. In [14], two accurate models for service curve for the GTS allocation are proposed. By using network calculus formulation, they derive the delay bound guaranty and present an expression of the duty cycle as a function of the delay. Based on the results, the impact of Beacon Order (BO) and Superframe Order (SO) on the maximum throughput and delay bound is analyzed. [15] proposes an adaptive GTS allocation (AGA) scheme for IEEE 802.15.4 by considering low-latency and fairness. There are two phases for the proposed scheme. In the classification phase, devices are assigned priorities in a dynamic fashion based on recent GTS usage feedbacks. Devices that need more attention from the coordinator are given higher priorities. In the GTS scheduling phase, GTSs are given to devices in a non-decreasing order of their priorities. A starvation-avoidance mechanism is presented to regain service attention for lower-priority devices that need more GTSs for data transmissions. The simulation and analytical models are developed to investigate the performance of AGA scheme. A multi-beacon superframe (MBS) structure with multiple sub-beacon intervals for different slot sizes in a superframe is proposed in [16]. It also proposes a greedy GTS allocation (GGA) algorithm, where device determines the most appropriate slot sizes based on their traffic characteristics. They claim to have significant improvement in bandwidth utilization at the expense of only a very small increase in the device active periods. A distance-based, real-time offline periodic message scheduling algorithm is proposed in [17], which generates BO, SO and GTS information to schedule the given message set.

## 2.10 Discussion of the Presented MAC Protocols

Table 2.1 compares all the discussed MAC protocols with respect to their support for timeliness, energy-efficiency, synchronization, adaptivity, and scalability factors.

<b>Protocol</b>	<b>Real-time support</b>	<b>Energy efficiency</b>	<b>Asynchronous</b>	<b>Adaptivity</b>	<b>Scalability</b>
S-MAC	no	partially yes	no	no	no
T-MAC	no	yes	no	partially yes	no
DSMAC	trade-offs with energy	trade-offs with latency	no	yes	no
B-MAC	no	yes	yes	yes	yes
STEM	trade-offs with energy	trade-offs with latency	yes, but needs two radios	no	no
TRAMA	no	yes	no	yes	no
LEACH	no	partially yes	no	no	no
WiseMAC	no	partially yes	partially yes	partially yes	no
802.15.4	partially yes	yes	depends on the topology	no	depends on the topology
AREA-MAC Chapter4	yes	yes	yes	yes	yes

Table 2.1: Comparison of different MAC protocols for WSN. This table clearly propels the need for a MAC protocol suitable for both real-time and energy-efficient WSN applications, and at the same time, it could also deal with other critical parameters.



---

## CHAPTER 3

---

# The IEEE 802.15.4 Standard

A recent trend towards achieving short-range communications has laid down the foundation of wireless personal area networks (WPAN). IEEE 802.15 [18] deals with different competing wireless standards to provide that capability, as it is unlikely that any single WPAN technology will meet all the ever-increasing consumer demands in terms of bandwidth, data rates, and QoS. Table 3.1 briefly compares the specifications, applications, and capabilities of well-known IEEE 802.15 WPAN standards.

A low-rate wireless personal area network (LR-WPAN) is a simple, low cost, low power, low QoS, and low data-rate communication network that works within the limited range of around 10 meters. The main objectives of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol. IEEE 802.15.4 [12] defines the PHY and MAC layers for such networks. This standard is not basically designed for WSN, but the following appealing features have made it a front runner for several WSN applications.

- Over-the-air data rates of 250 kb/s, 100kb/s, 40 kb/s, and 20 kb/s
- Personal operating space (POS) of 10 meters
- Star or peer-to-peer operation with fully and reduced function devices (FFDs and RFDs)
- Allocated 16-bit short or 64-bit extended addresses
- Optional allocation of guaranteed time slots (GTSs) for low latency applications
- Carrier sense multiple access with collision avoidance (CSMA-CA) channel access
- Fully acknowledged protocol for reliable data transfer
- Low power consumption
- Energy detection (ED)
- Link quality indication (LQI)
- 16 channels in the 2450 MHz band, 30 channels in the 915 MHz band, and 3 channels in the 868 MHz band

Standard	Name	Data rate	Applications	QoS
802.15.1	Bluetooth	1 Mbps	cell phones, laptops, PDAs, printers, bar code readers, sensors, microphones	suitable for voice applications
802.15.2	Coexistence of bluetooth and 802.11b	N/A	N/A	N/A
802.15.3	High-rate WPAN	>20 Mbps	low-power and low-cost solutions for digital imaging and multimedia applications	very high QoS
802.15.4 [12]	Low-rate WPAN	<0.25 Mbps	industrial, agricultural, medical, surveillance, sensors, actuators with very low-power, low-cost	relaxed QoS and data rates
802.15.5	Mesh Networks	N/A	Coverage extension without increasing the TX power and RX sensitivity, route redundancy, easier network configuration, better battery life	low to high (application dependent)

Table 3.1: IEEE 802.15 WPAN standards. This table depicts comparison of specifications, data rates, applications, and capabilities for different IEEE 802.15 WPAN standards. Although each one has targeted slightly different applications, these standards are aimed at reliable connectivity between portable devices in close propinquity for better interoperability.

The IEEE 802.15.4 architecture is centered on two bottom layers, PHY layer and MAC layer, as shown in Figure 3.1. The specification and working of the upper layers is defined by ZigBee standard [19].

### 3.1 PHY Layer

The PHY layer provides two services: the PHY data service and the PHY management service interfacing to the physical layer management entity (PLME) service access point (SAP), known as the PLME-SAP. The PHY data service enables the transmission and reception of PHY protocol data units (PPDUs) across the physical radio channel. The PLME-SAP allows the transport of management commands between the MLME and the PLME. The features of the PHY are activation and deactivation of the radio transceiver, ED, LQI, channel selection, clear channel assessment (CCA), and transmitting as well as receiving packets across the physical medium. The radio operates at three different unlicensed bands of 868 MHz, 902 MHz and 2400 MHz.

### 3.2 MAC Layer

The MAC layer provides two services: the MAC data service and the MAC management service interfacing to the MAC sublayer management entity (MLME) service access point (SAP), known as MLME-SAP. The MAC data service enables the transmission

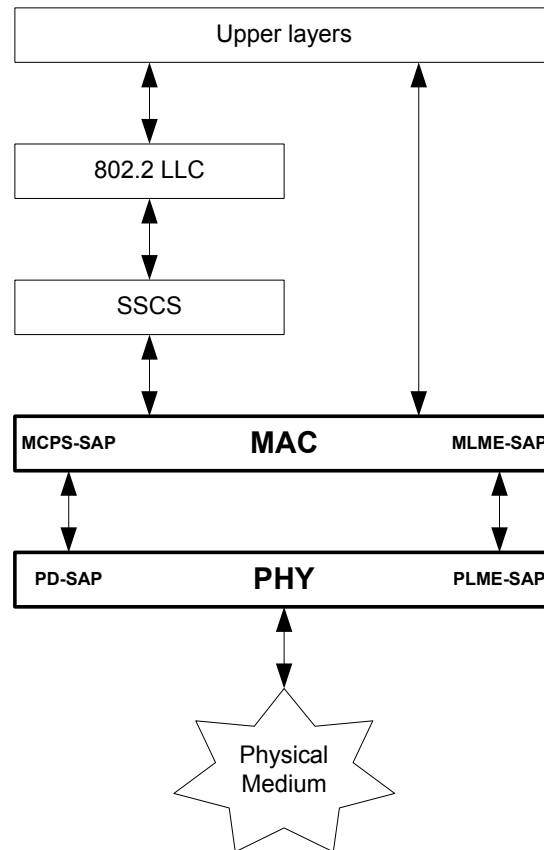


Figure 3.1: IEEE 802.15.4 protocol architecture. Upper layers consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of device. An IEEE 802.2 Type 1 logical link control (LLC) can access the MAC sublayer through the service-specific convergence sublayer (SSCS).

and reception of MAC protocol data units (MPDUs) across the PHY data service. The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. The features of the MAC sublayer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation. In addition, the MAC sublayer provides hooks for implementing application-appropriate security mechanisms. The MAC protocol supports two operational modes selected by the coordinator, the non beacon-enabled mode, in which MAC is simply ruled by non-slotted CSMA/CA and the beacon-enabled mode, in which beacons are periodically sent by the coordinator to synchronize associated nodes.

### 3.2.1 MAC Superframe Structure

The format of the (optional) MAC superframe is defined by the PAN coordinator (FFD) and is shown in Figure 3.2. The superframe is bounded by network beacons sent by the coordinator and is divided into 16 equally sized slots. Optionally, the superframe can have an active and an inactive portion. During the inactive portion, the coordinator may enter a low-power mode. The beacon frame is transmitted in the first slot of each superframe. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes. Any device wishing to communicate during the

contention access period (CAP) between two beacons competes with other devices using a slotted CSMA-CA mechanism. All transactions are completed by the time of the next network beacon.

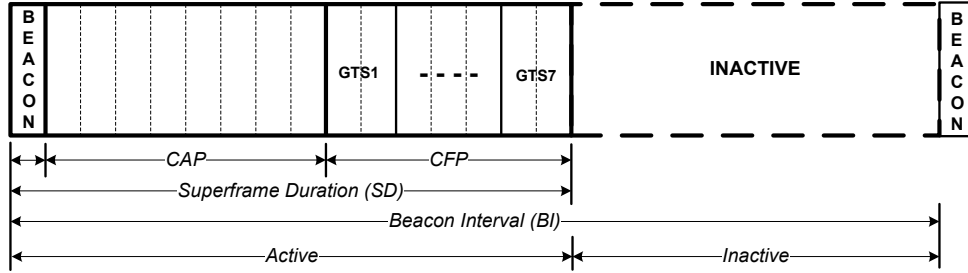


Figure 3.2: MAC superframe structure. It is divided into 16 equally sized slots and is bounded by network beacons. The superframe can have an active and an inactive portion. The CFP portion of the active part is used for low-latency applications.

For low-latency applications or applications requiring specific data bandwidth, the PAN coordinator may dedicate portions of the active superframe to that application. These portions are called guaranteed time slots (GTSs). The GTSs form the contention-free period (CFP), which always appears at the end of the active superframe starting at a slot boundary immediately following the CAP. The PAN coordinator may allocate up to seven of these GTSs, and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP remains for contention-based access of other networked devices or new devices wishing to join the network. All contention-based transactions are completed before the CFP begins. Also each device transmitting in a GTS ensures that its transaction is complete before the time of the next GTS or the end of the CFP.

The structure of this superframe is described by the values of BO and SO. *BeaconOrder*, BO, describes the interval at which the coordinator shall transmit its beacon frames, i.e., Beacon Interval (BI). The value of BO and BI are related as follows:

$$BI = aBaseSuperFrameDuration * 2^{BO} \quad 0 \leq BO \leq 14 \quad (3.1)$$

If BO = 15, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command. *SuperframeOrder*, SO, describes the length of the active portion of the superframe, which includes the beacon frame, i.e., Superframe Duration (SD). The values of SO and SD are related as follows:

$$SD = aBaseSuperFrameDuration * 2^{SO} \quad 0 \leq SO \leq BO \leq 14 \quad (3.2)$$

If SO = 15, the superframe shall not remain active after the beacon. The parameter *aBaseSuperframeDuration* depends on the frequency range of operation. For 2.4 GHz frequency band, the value of *aBaseSuperframeDuration* lies between 15.36ms and 251.6s. SD and BI allows conclusion about the duration of inactive period, in which the device can turn to sleep mode. Therefore, SO and BO are key parameters for potential energy savings.

The device requests GTS slots by using GTS characteristics field (1 byte) shown in figure 3.4(a). GTS Length defines the number of GTS slots requested, GTS Direction defines receive/transmit GTS, and Characteristics Type shows allocation/deallocation request.



### 3.3 IEEE 802.15.4 Limitations

Though an IEEE 802.15.4-based network fulfills many of the WSN challenges, it still endures different limitations, especially for timeliness, energy, and bandwidth critical applications and holds a room to be improved. In order to take full benefit of services provided by this protocol, following limitations need to be addressed:

- The first and foremost problem with the current GTS allocation mechanism provided by IEEE 802.15.4 is the bandwidth under-utilization. Unfortunately, the standard only supports the values of BO and SO by the power of two and the slot length must be 1/16 of the SD. Most of the time, device uses only a small portion of the allocated GTS slots, major portion may remain unused. It creates an empty hole in the CFP, like the memory fragmentation problem for operating systems.
- The protocol only supports explicit GTS allocation and hence a maximum of seven GTS descriptors can be allocated in each superframe.
- The protocol only supports first come first serve (FCFS) based GTS allocation and does not take into account the traffic specification, delay requirements, and the energy resources.
- The device uses GTS length bits of its GTS Characteristics field, shown in Figure 3.4(a), for the number of GTS slots it wants. The device can request for all seven GTS slots, even if it is not really needed. Such unbalanced slot distribution can block other needful devices to take advantage of the guaranteed services.
- The protocol uses GTS expiration on the basis of some constant factors. For a transmit GTS, the coordinator shall assume that a device is no longer using its GTS if a data frame is not received from the device in the GTS at least every  $2 \times n$  superframes and for receive GTS, the coordinator shall assume that a device is no longer using its GTS if an acknowledgement frame is not received from the device at least every  $2 \times n$  superframes, where the value of  $n$  in both cases is given by Equation (3.3). Moreover, the assigned GTS slots are broadcasted for the  $aGTSDescPersistenceTime$  (a constant having value of 4) number of superframes. Such restrictions cause unnecessary energy consumption and CFP slots blockage for the longer time.

$$n = \begin{cases} 2^{8-\text{macBeaconOrder}} & 0 \leq BO \leq 8 \\ n = 1 & 9 \leq BO \leq 14 \end{cases} \quad (3.3)$$

- Even if the CFP is not present in the superframe, beacons transmitted by the coordinator always use unnecessarily one byte for CFP, resulting in energy wastage.
- The current superframe structure must contain at least  $aMinCAPLength$  (a constant) size CAP. For strict real-time applications, we may need flexible size CAP rather than the fixed one.
- The GTS slots assigned by the coordinator will be applicable to the devices only in the upcoming beacon. It means that devices have to wait for the next beacon to use the guaranteed service. If the value of BI is large enough and the beacon arrives after longer time, then there is not any real advantage of such real-time service.

### 3.4 IEEE 802.15.4-based System Model

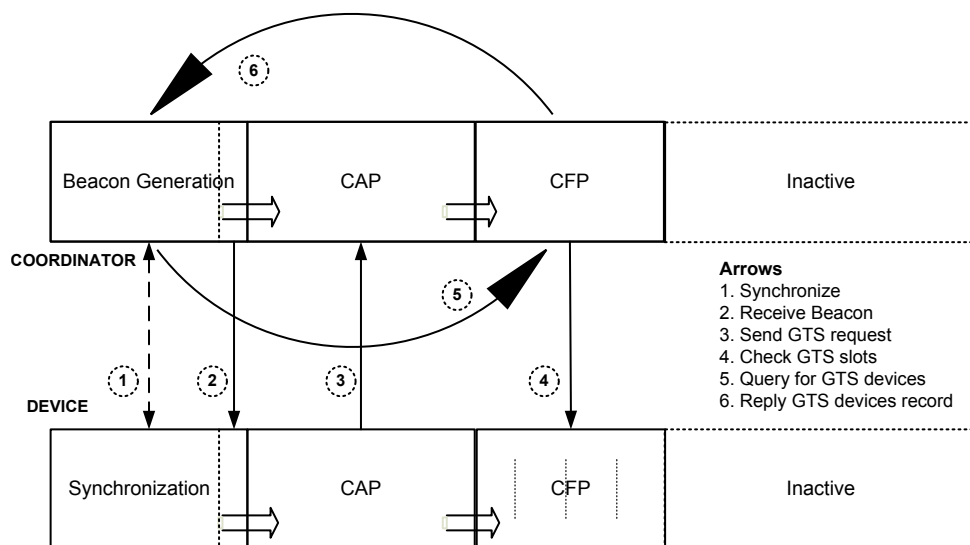


Figure 3.3: A system model. The device synchronizes itself with the coordinator and receives a beacon. It can also send a GTS request in its CAP and checks its turn in the CFP. The coordinator collects information about the devices needing GTS slots from its CFP.

We consider an IEEE 802.15.4-based WSN, which works in star topology with beacon-enable mode and uses 2450 MHz frequency band with O-QPSK modulation, as this configuration provides the most efficient network parameters setting [20]. We aim to improve the performance of IEEE 802.15.4, by keeping most of the protocol formatting intact. Our system model is shown in Figure 3.3, where the coordinator broadcasts beacons periodically depending on the value of BO. All devices in the communication range can easily synchronize with the coordinator and receive the beacon. After transmitting a beacon, CAP starts which follows the optional CFP. At the beacon generation stage, the coordinator collects data from its CFP regarding the devices, that need GTS slots, assembles the beacon, and transmits it. After that CAP starts which follows the CFP.

On the other hand, the device first synchronizes itself with the coordinator, makes its receiver on just before the beacon arrival time and receives the beacon. After CAP, CFP starts, where the device first checks for its GTS slots. If it finds its GTS entry in the current beacon, it can utilize it by sending/receiving data. After using its GTS slots, the device makes its radio off again. The device can send new GTS request in its CAP, which is connected to the CAP of the coordinator and if the request is accepted, its address and GTS characteristics are stored in CFP of the coordinator. That record is then included to the upcoming beacon. Each associated node generates the data packets with some delay and energy constraints. The node requests for GTS slot(s) by sending a request to the coordinator and specifying its data and delay conditions. The coordinator, who runs an admission control (AC), decides the fate of the request.

#### 3.4.1 A Proposed Solution

In order to address almost all aforementioned IEEE 802.15.4 limitations, a new GTS characteristic field is proposed, where the device, rather than sending fixed slot length,

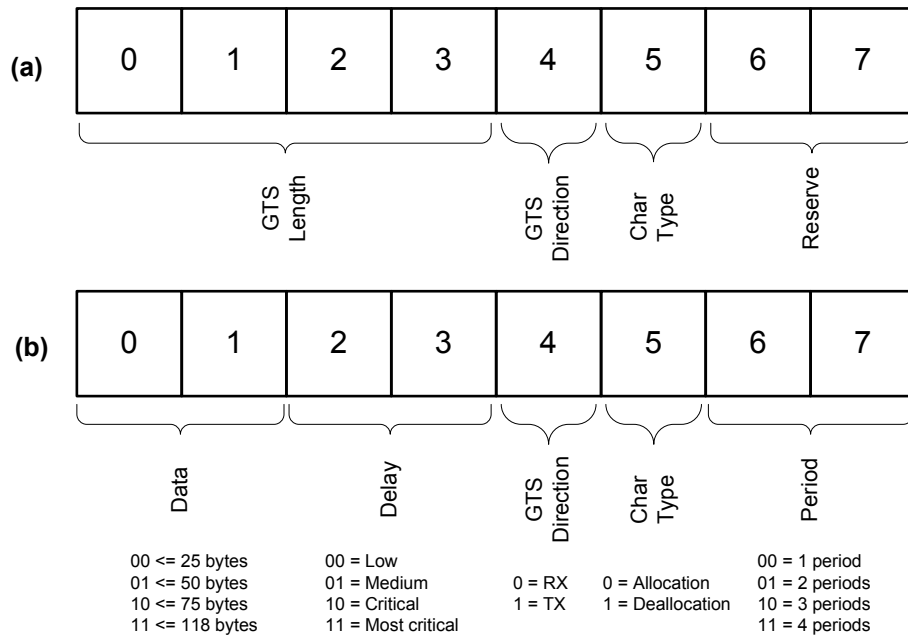


Figure 3.4: GTS characteristics field. The original GTS characteristics field (a) consists of 4-bit long GTS length for the number of GTS slots a device needs, GTS direction bit to show whether it is a transmit or receive GTS request, and the characteristics type bit to show whether it is an allocation or deallocation request. The revised GTS characteristics field (b) is proposed, where rather than sending the GTS length directly, the device sends its data, delay, and period-cycle information to the coordinator.

sends its data and delay specification to the coordinator. The original GTS characteristics field is shown in Figure 3.4(a), whereas, the proposed one is shown in Figure 3.4(b). The coordinator, who runs an AC, decides the slot length for the device. AC works on the basis of total cluster load, remaining CAP size, device’s data and delay specifications, and device’s recent CFP usage. In this way, bandwidth under-utilization and the restriction of entertaining at most seven GTS requests are avoided. To annul the constant GTS expiration, the coordinator uses the period bits of the GTS characteristics field and performs GTS expiration dynamically. Rather than using one whole byte in the superframe to indicate the presence of the CFP part, we use one available reserve bit in the superframe. With our approach, the coordinator could allocate GTS slots on round-robin or some time factor basis (for example, at every 5 seconds) to the device, who once declares to have periodic applications.

### 3.5 Selection of Critical Parameters

We have implemented the IEEE 802.15.4 protocol on the Tinyos 2.x [21] platform, where the coordinator works in star topology and periodically broadcasts the beacons. The devices scan the channels, synchronize with the coordinator, receive the beacon, and request for association. If the coordinator accepts the association request, the device can request for the GTS allocation. On successful GTS allocation by the coordinator, the device can use its GTS slots to send/receive the data. Moteiv Tmote [22] nodes with Chipcon CC2420 transceivers [23] are used for the implementation. Now we will discuss our analysis for the application-specific optimal parameters setting.

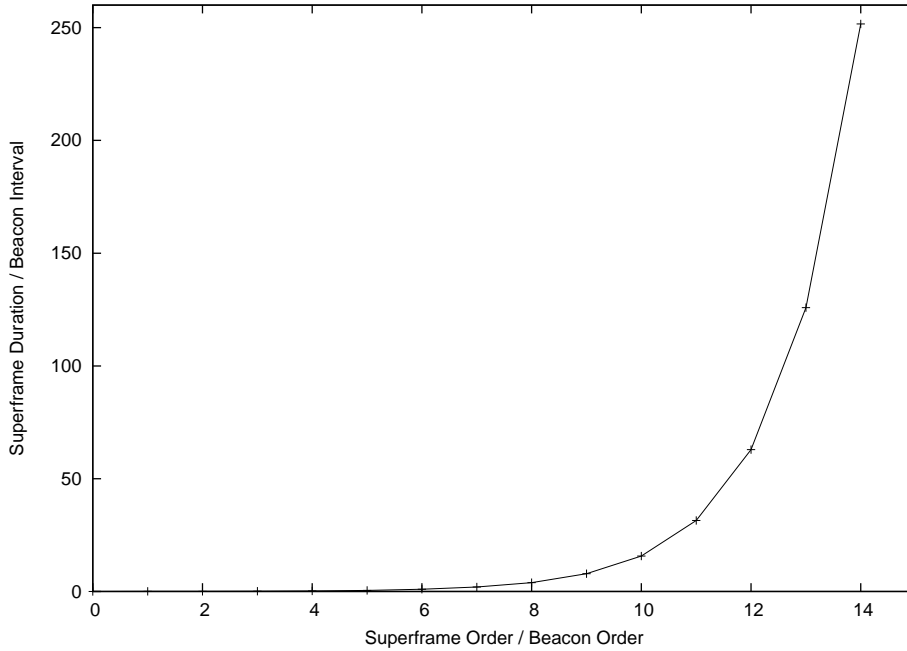


Figure 3.5: BI-BO and SD-SO relation. The values of beacon interval and superframe duration lie between 15.36 ms and 251.6 s, whereas the values of beacon order and superframe order range between 0 and 14.

### 3.5.1 Beacon Interval & Superframe Duration

Figure 3.5 shows the relationship between SO-SD and BO-BI, based on Equations (3.1) and (3.2). The values of BO and SO vary between 0 and 14, whereas, the values of BI and SD lie between 15.36 ms and 251.6 s. We can decide for the application-specific value of BI. For the periodic application, where nodes send their data at an interval of 60 seconds, we can choose 12 as the value of BI.

### 3.5.2 Maximum CFP Slots available

Before choosing the number of CFP slots needed to the devices, we first calculate the maximum number of CFP slots, which are available for different values of SD. MAC Superframe is divided into 16 equal slots (0-15). The beacon is always transmitted in the first slot and CAP should be at least of the size of  $aMinCAPLength$ , which is equal to 440 symbols (1 slot = 60 symbols). The maximum number of available CFP slots can be calculated by Equation (3.4) and is shown in Figure 3.6.

$$MaxCFP_{avail} = \frac{15 - aMinCAPLength/60}{aBaseSlotDuration * 2^{SO}} \quad (3.4)$$

### 3.5.3 CFP Slots actually needed

With our approach, the coordinator allocates GTS slots to the device depending on its data specification. In order to calculate the number of slots needed for the device, we first calculate the total length of data packets, which are generated by the device. The upper layers of the device generate the data frame, which is passed to the MAC as a MSDU (MAC service data unit). IEEE 802.15.4 supports 118 bytes as the maximum

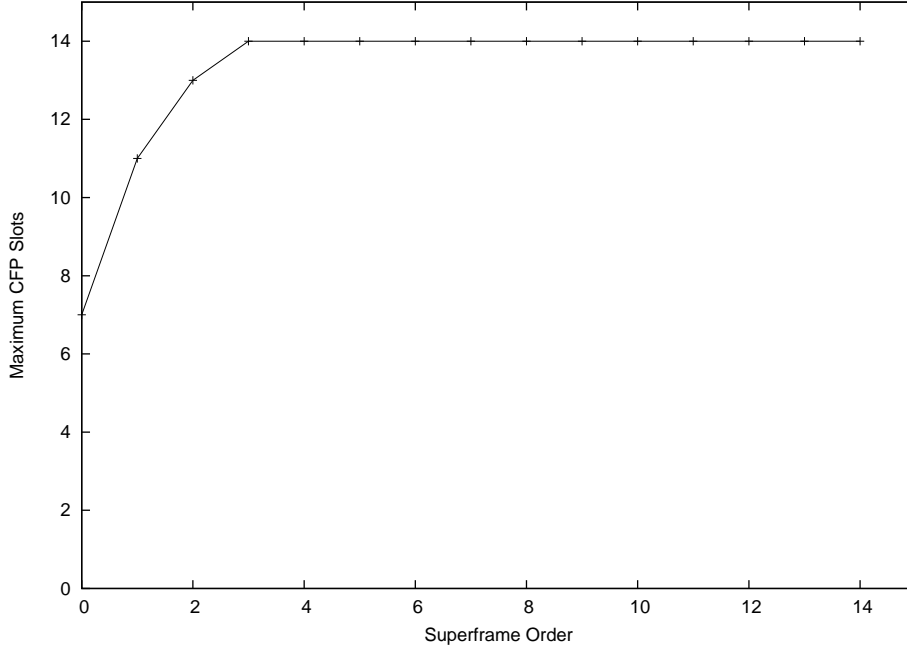


Figure 3.6: Maximum available CFP slots. For the 0 value of superframe order, we have 7 CFP slots available, but for the superframe value of 4 to 14, the number of CFP slots available is 14.

length of MSDU. MAC layer adds its header (MHR) of 9 bytes and changes MSDU into MPDU (MAC protocol data frame). Then this MPDU is passed to the PHY layer as the PSDU (PHY service data unit). The PHY layer supports PSDU of at most 127 bytes in length (MSDU + MHR). The PHY layer also adds its header (PHR) of 6 bytes and converts the PSDU into the PPDU (PHY protocol data unit). Further 11 bytes are required for an (optional) acknowledgement request. To receive acknowledgement, device needs  $aTurnaroundTime$  (12 symbols) to change its radio from the TX to RX mode (or vice-versa).

Additionally, we have to consider the interframe spacing (IFS), which separates two successive frames sent by the device. Its length is dependent on the size of the frame that has just been transmitted. Frames (i.e., MPDUs) of up to  $aMaxSIFSFrameSize$  octets (18 bytes) in length shall be followed by a SIFS (short interframe spacing) period of duration of at least  $macMinSIFSPeriod$  (12 symbols). Frames with lengths greater than  $aMaxSIFSFrameSize$  octets shall be followed by a LIFS (long interframe spacing) period of duration of at least  $macMinLIFSPeriod$  (40 symbols); as shown in Equation (3.5), whereas, the number of CFP slots required for the device is shown in Equation (3.6).

For an application, where the devices generate data frames having length of 10, 25, 50, 100 or 118 bytes, we calculate the required CFP slots with the acknowledgement transmission in Figure 3.7. We calculate same for the unacknowledgement transmission in Figure 3.8. In both figures, we consider the value of SO as 0, 1, and 2. It is clear that, even for the maximum size of data packet, we need only one GTS slot, if we use SO of 2.

$$IFS = \begin{cases} 12 & MPDU \leq 18 \text{ Bytes} \\ 40 & MPDU > 18 \text{ Bytes} \end{cases} \quad (3.5)$$

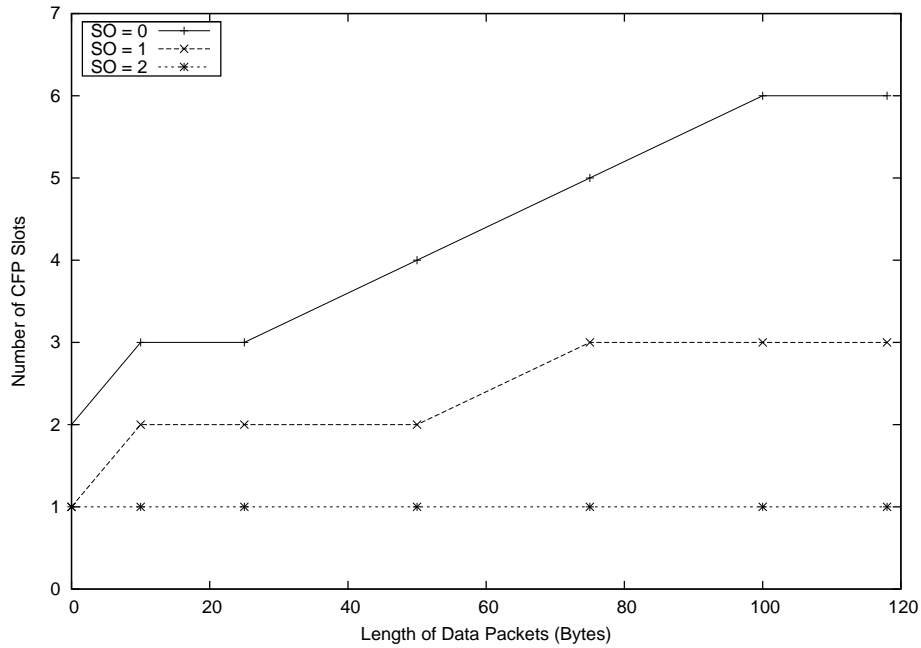


Figure 3.7: Data packets and required CFP slots. It shows the number of CFP slots needed for the transmission of given length of data packets *with acknowledgement* request, while keeping the superframe order to 0, 1, and 2.

$$CFP_{req} = \frac{2 * Data + IFS + aTurnaroundTime}{aBaseSlotDuration * 2^{SO}} \quad (3.6)$$

where  $Data = MSDU + MHR + PHR + ACK$

### 3.5.4 Superframe Order

SO defines the length of the active part of the superframe. Therefore, it is an important variable for the energy saving. From Figures 3.7 and 3.8, it is clear that, if we consider 2 as a value of SO, then only one GTS slot is needed, even for the maximum size of data packets. For SO of 1, we need at most three GTS slots.

Device	Packet length [bytes]	GTS slots requested	GTS slots needed
1	100	3	1
2	75	2	1
3	50	1	1
4	100	2	1
5	25	2	1

Table 3.2: Data packets and GTS slots. The number of the CFP slots requested and actually needed by five devices for the given length of data packets.

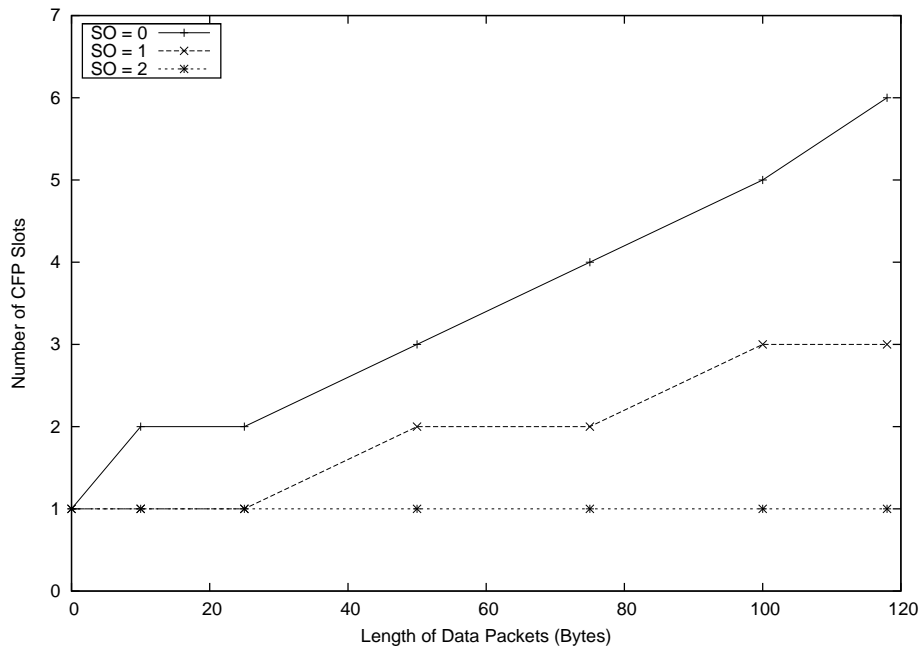


Figure 3.8: Data packets and required CFP slots. It shows the number of CFP slots needed for the transmission of given length of data packets *without acknowledgement* request, while keeping the superframe order to 0, 1, and 2.

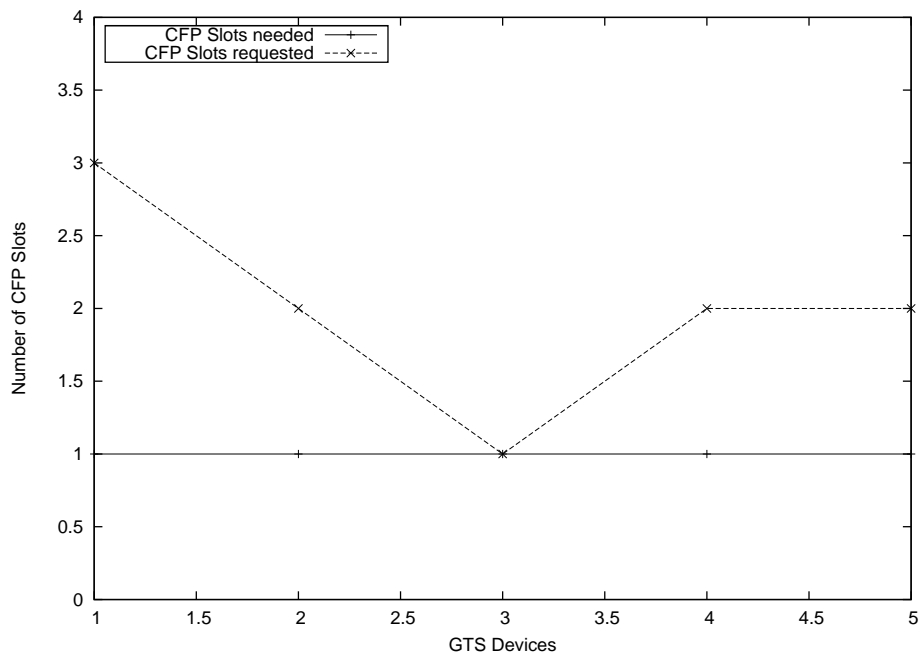


Figure 3.9: An example of bandwidth under-utilization. Five different devices request and acquire the different number of CFP slots for the given length of data packets. The default GTS allocation of the standard results in bandwidth under-utilization, as devices mostly need less than the requested slots.

### 3.5.5 Bandwidth Under-utilization

Bandwidth under-utilization is the major problem with the current IEEE 802.15.4 protocol. The improvement, given by our approach, can be validated by an example shown in table 3.2., where five devices are generating data packets, each of 100, 75, 50, 100 and 25

bytes and request for 3, 2, 1, 2 and 2 GTS slots respectively. Figure 3.9 confirms that our technique outperforms the original protocol in terms of bandwidth utilization and thus more devices can take benefit of the guaranteed services.



---

## CHAPTER 4

---

# AREA-MAC

### 4.1 Basic Design Concepts

As discussed earlier, the MAC protocol for WSN has to deal with some additional concerns than those designed for traditional wired or wireless networks. The major concerns are energy wastage and higher latency. The prevalent sources of the energy waste are packet collision, idle listening, overhearing, and over-emitting [3]. The direct interaction with the real world, where the physical events occur in an unpredictable manner and duty cycling of radio result in higher latency for WSN. In this report, we propose a MAC protocol called AREA-MAC (Asynchronous, Real-time, Energy-efficient, and Adaptive MAC), which is intended to deal efficiently with time critical and energy-efficient applications, and at the same time, to provide a better trade-off between vital parameters, such as, system fairness, throughput, scalability, and adaptivity to traffic conditions. The main characteristics of AREA-MAC are:

- **Asynchronous:** The system-wide synchronization results in overhead and scaling problems. In our scheme, we do not consider any type of synchronization. All the nodes are independent to sleep and wake-up schedules of other nodes. However, for better routing and link cost measurements, which is discussed later, nodes save wakeup-sleep schedules and energy information of their 1-hop neighbors.
- **Energy-efficiency:** The nodes in AREA-MAC use the LPL technique and wake up very shortly to check the channel activity without actually receiving the data. They go back to sleep mode if the channel is idle, otherwise they receive the data. The previously proposed MAC protocols, like [6] and [11], use LPL with long preambles, where, whenever a node has data to send, it transmits an extended preamble proceeding to the data packet. On wake up, all the other nodes sample the medium and if a preamble is detected, they remain awake for the remainder of the long preamble. After receiving the full preamble, if the node is not a target node, as shown in Figure 4.1, it goes back to sleep mode. However, nodes in AREA-MAC use short and adaptive preamble with destination address and acknowledgement combination, also suggested by [24]. It solves many of the problems which arise with long preambles, such as energy consumption both at receiver and sender, overhearing at non-target receivers, and excess latency at each hop [10]. The neighboring nodes wake up for a small period of time and check the destination address. The target node acknowledges the source node immediately, which causes the source node to stop sending further preambles and

to start transmitting data packets. All the other non-target nodes go back to sleep mode immediately. It minimizes the possibility of a collision, idle listening, and overhearing.

- **Real-time support:** For real-time data, the source node requests/forces the suitable neighbor to wake up regardless to its normal schedule and the intended target node responses the source node immediately, which almost eliminates the possibility of over-emitting. The suitable neighbor is selected on the basis of a cost metric, which is calculated by the delay between source and target node, energy level of the target node, the last time the target node was in wake-up mode, closeness of the target node to the destination node, and the number of neighbors around the target node. If the target node is not the destination node, then it forwards the data to its up-level neighbors.
- **Adaptivity:** The nodes adapt their duty cycle with respect to the real-time request received from their neighbors. On reception of such a request, the target node responses and treats it most urgently by adapting its duty cycle accordingly. One more advantage of AREA-MAC is its robustness to topology changes. Unlike cluster-based approaches, where nodes only communicate via cluster heads, nodes in AREA-MAC communicate directly with peers and exchange their wakeup-sleep schedules and energy information.

As the main design objective of AREA-MAC is to provide a suitable solution for real time applications, therefore, the interesting questions which arise here are, how this scheme effects on the energy consumption and fairness of the system and how to achieve optimal trade-off between the energy efficiency and the timeliness.

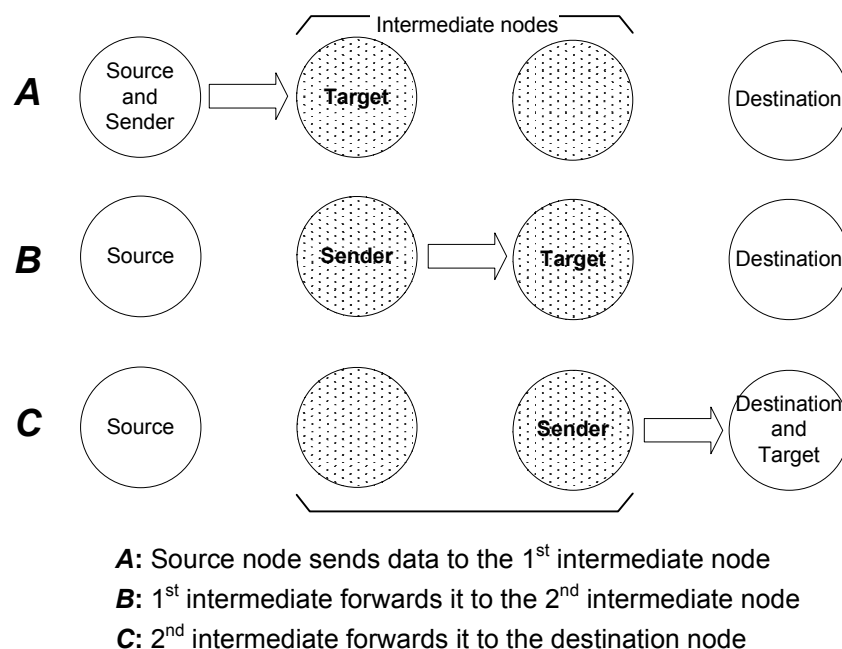


Figure 4.1: The different roles of a node depending on when it generates, forwards, and receives a data packet. Source node always generates a data packet, sender sends or forwards it, target node receives it, and destination node is the final destination for that packet.

### 4.1.1 Assumptions

We consider a grid-based WSN shown in Figure 4.2, consisting of several nodes and terminating at the sink node. The function of the sink node is to receive data from all nodes and forward it to the terminating point. All other nodes are normal nodes performing the functions of sensing, receiving, and transmitting data packets without having any aggregating or in-network processing capabilities. We assume that all nodes are fixed and know their locations with respect to few reference nodes. The selection and working of reference nodes is out of our scope. We also assume that the density of nodes is high enough, that a node can directly communicate with its multiple neighbors. All nodes carry unique node IDs and are deployed in an ascending order with the sink node having the highest deployment level. The normal nodes forward data only to the up-level direction, i.e., towards the sink node. The up-level neighbors of a node are the nodes having higher deployment level, i.e., less ID number than its own. We also assume that the sink node does not have any energy problem, as it is AC-powered, whereas other nodes have limited and non-replicable energy resources.

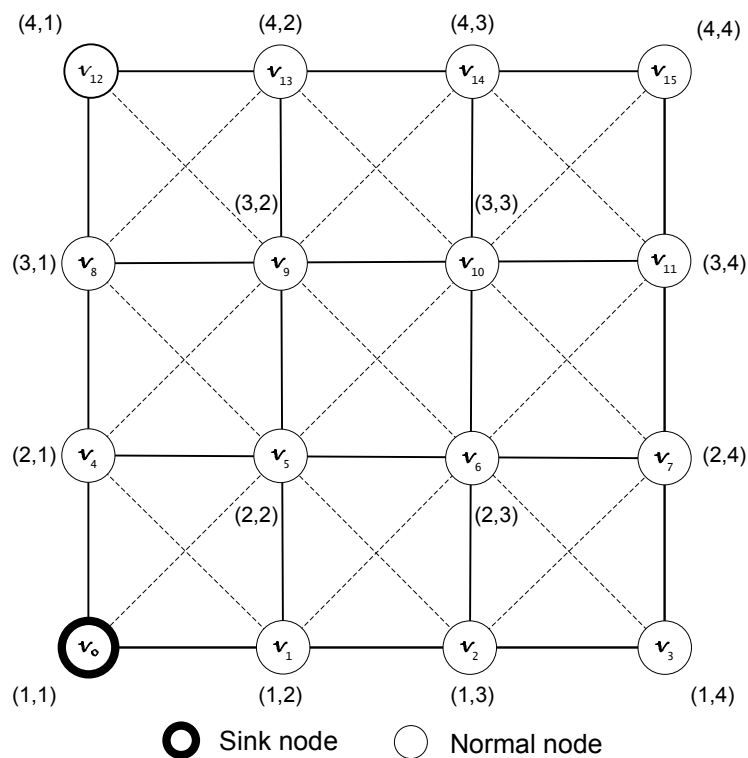


Figure 4.2: A portion of grid-based WSN. All nodes are deployed in an ascending order having unique node IDs and know their locations (x,y). The deployment level decreases as the order or ID number increases. They also have information about their all 1-level and 2-level neighbors and send data only towards the sink node.

## 4.2 Network Model

A WSN can be represented by an undirected graph  $G(V, E)$ , called a connectivity graph, where  $V = \{v_0, v_1, \dots, v_{N-1}\}$  is the set of  $N$  sensor nodes' IDs and  $E$  is the set of edges connecting those nodes. Such a graph can be described as a grid topology of  $m \times n$  order

with  $m$  rows and  $n$  columns. The nodes are placed at the location  $(x, y)$ , where  $1 \leq x \leq m$  and  $1 \leq y \leq n$ . A small portion of such a grid is shown in Figure 4.2. If the grid location is given, then the node ID is determined by:

$$ID(x, y) = (x - 1) \times n + (y - 1) \quad \text{where} \quad 1 \leq x \leq m; 1 \leq y \leq n \quad (4.1)$$

Alternatively, given the node ID, its location can be calculated as follows:

$$y = (ID(x, y) \bmod n) + 1 \quad \text{and} \quad x = (ID(x, y) - y + 1)/n + 1 \quad (4.2)$$

The node  $v_0$  represents the sink node, whereas nodes from  $v_1$  to  $v_{N-1}$  represent the normal sensor nodes. Table 4.1 shows all the terms used for the network model. An up-level neighbor of a node  $v_i$  is called an 1-level neighbor for  $v_i$ , if its location parameters  $(x, y)$  satisfy one of the following conditions:

1. If its  $x$  value is equal to the  $x$  value of  $v_i$ , then its  $y$  value should be one less than  $y$  value of  $v_i$ .
2. If its  $y$  value is equal to the  $y$  value of  $v_i$ , then its  $x$  value should be one less than  $x$  value of  $v_i$ .
3. Both  $x, y$  values are one less than the  $x, y$  values of  $v_i$ .

Similarly, an up-level neighbor of a node  $v_i$  is called a 2-level neighbor for  $v_i$ , if its location parameters  $(x, y)$  satisfy one of the following conditions:

1. Its  $x$  value is two less than the  $x$  value of  $v_i$ .
2. Its  $y$  value is two less than the  $y$  value of  $v_i$ .
3. Both  $x, y$  values are two less than the  $x, y$  values of  $v_i$ .

The set  $N_{v_i}^1$  contains all the 1-level neighbors and  $N_{v_i}^2$  contains all the 2-level neighbors for the node  $v_i$ . For example, in Figure 4.2,  $N_{v_{10}}^1 = \{v_5, v_6, v_9\}$  and  $N_{v_{10}}^2 = \{v_0, v_1, v_2, v_4, v_8\}$ . Each node  $v_i \in V$  has a limited circular transmission range. As we assume a sufficient density of nodes, hence a node  $v_i$  can easily communicate with all of its  $N_{v_i}^1$  and  $N_{v_i}^2$  neighbors. The degree of the node  $v_i$ , denoted by  $\delta_{v_i}$ , represents the total number of up-level neighbors for the node and is equal to  $N_{v_i}^1 + N_{v_i}^2$ , hence  $\delta_{v_{10}} = 8$  for the node  $v_{10}$  in the above example. A bidirectional wireless link exists between  $v_i$  and its neighbor  $v_j \in \delta_{v_i}$  and is represented by an edge  $(v_i, v_j) \in E$ . The euclidean distance between the node  $v_i$  and its neighbor  $v_j$  is given by  $d_{v_i, v_j}$  and their connectivity is shown by the binary variable  $C_{v_i, v_j}$ .

### 4.3 Time and Energy Saving for AREA-MAC

Figure 4.3 depicts an obvious gain of AREA-MAC in terms of energy and time over the traditional long preamble technique. The gain is not limited to the sender and receiver, non-target receivers also benefit from it. Specially, for real-time traffic, where the sender directly requests target node to wake-up, gain is 100% for non-target nodes. Moreover, for real-time traffic, the node sends data to its 2-level neighbor, which minimizes the packet latency almost by half of the time which is required for the periodic traffic. At the same

Symbol	Used for
$N$	Total number of sensor nodes
$N_{v_i}^1$	1-level neighbors for $v_i$
$N_{v_i}^2$	2-level neighbors for $v_i$
$\delta_{v_i}$	Total number of up-level neighbors for $v_i$
$\partial_{v_i,v_j}$	Euclidean distance between $v_i$ and $v_j$
$C_{v_i,v_j}$	Connectivity between $v_i$ and $v_j$
$F_{v_i,v_j}$	Data flow from $v_i$ to $v_j$
$R_{v_i}$	Residual energy of $v_i$
$L_{v_i}$	Last wake-up time of $v_i$
$TH_1$	Threshold for link cost
$TH_2$	Threshold for energy consumption

Table 4.1: Terms used for the network model

time, it paves way for 1-level neighbor to remain in sleep mode. In order to forward a packet to the farthest, i.e., 2-level neighbor, a node needs more transmission energy. But, that is much less than the energy consumption at each 1-level neighbor to wake-up and process data, if it would have been requested to do so.

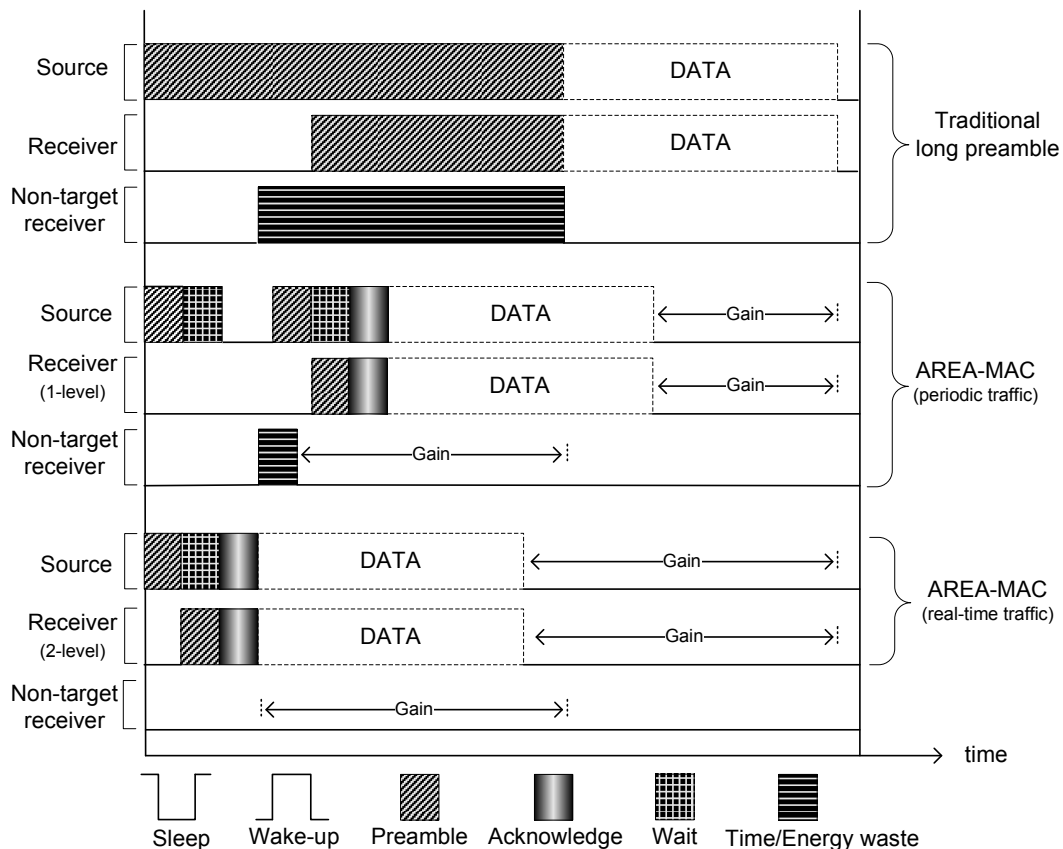


Figure 4.3: AREA-MAC gain in terms of time and energy over the traditional long preamble technique.

## 4.4 Design Phases

The overall design of AREA-MAC can be categorized into the following phases. These phases are also outlined in Algorithms 1-4.

### 4.4.1 Network Setup Phase

At startup, each node scans the channel for the  $T_{setup}$  time and receives hello packets from its neighbors. It saves information contained in the hello packet for all of its 2-level neighbors. After that, it configures its own hello packet and broadcasts. Each hello packet contains node-id, sleep and wake-up schedule, and the residual energy of the node. Each node initially selects a random sleep and wake-up schedule and broadcasts it. We assume that  $T_{setup}$  is large enough to collect the information about all the 2-level neighbors. As network setup phase occurs once, therefore, this consideration does not have any significant effect on overall network performance.

### 4.4.2 Periodic Duty Cycle Phase

To save the energy, nodes use LPL and most of the time remain in sleep mode. For the periodic traffic, they wake up at every  $I_{wakeup}$  and listen the channel. If the node finds a preamble on the channel, it tries to decode the preamble and checks for the destination address written inside the preamble. On successfully matching the destination address with its own address, the node sends an acknowledgement to the source node and receives the data. After forwarding data to its nearest up-level neighbor, the node goes back to sleep mode.

### 4.4.3 Adaptive Duty Cycle Phase

For aperiodic or real-time data, the node wakes up in response to the on-demand request sent by the neighbor and receives the data. It calculates the link cost with the up-level neighbors and forwards data to the neighbor with the highest link cost. Afterwards, the node prepares itself for the new wakeup-sleep schedule, i.e., for a new hello packet.

### 4.4.4 Re-scheduling Phase

In case of real-time data, a node wakes up regardless to their normal schedule in response to the real-time request received from its low-level neighbor. After receiving this request, the node acknowledges the source node, receives the data, and processes it. Then, the node reconfigures its sleep and wake-up schedule, broadcasts a new hello packet with updated sleep and wake-up schedule, and goes back to sleep mode. All the neighbors including the source node save this updated information.

<b>Algorithm 1:</b> netSetup
Scan the channel; Receive hello packets from neighbors; Save hello packets of all 1-level and 2-level neighbors; Configure own hello packet and broadcast it;

**Algorithm 2:** forPeriodic

```

Wake up at every  $I_{wakeup}$  interval;
Perform carrier sense;
if found preamble then
  if address matches then
    Acknowledge the source node;
    Receive the data packet;
    Forward it to the nearest up-level neighbor;
    Go to sleep;
  end
else
  Go to sleep immediately
end

```

**Algorithm 3:** forRealTime

```

if found an RT request then
  Success = FALSE;
  Reject = FALSE;
  Acknowledge the source node;
  Receive the data packet;
  Estimate link cost with up-level neighbors;
  repeat
    Select (next) best up-level neighbor;
    if link cost < threshold then
      Request the neighbor;
      if Success then
        Forward the data;
      end
    else
      Say sorry to the source node and Reject = TRUE
    end
  until Success or Reject ;
  Prepare to update sleep-wakeup schedule;
end

```

**Algorithm 4:** reSchedule

```

if got an RT request recently then
  Reschedule hello packet;
  Broadcast updated hello packet;
end

```

## 4.5 Energy Model

We consider two types of WSN traffic generating scenarios, periodic, where nodes generate data with fix periodic interval and aperiodic, where nodes generate data at random intervals. The later is used for real-time applications, where data is generated in an

Symbol	Used for	Unit
$T_{lpl}, P_{lpl}$	Time and power in LPL	ms, mW
$T_{carrier}, P_{carrier}$	Time and power in carrier sense	ms, mW
$T_{sense}, P_{sense}$	Time and power in sensing	ms, mW
$T_{sleep}, P_{sleep}$	Time and power in sleep	ms, mW
$T_{rx}, P_{rx}$	Time and power in reception	ms, mW
$T_{tx}, P_{tx}$	Time and power in transmission	ms, mW
$T_{alpl}$	Average time in LPL	ms
$T_{acARRIER}$	Average time in carrier sense	ms
$T_{asense}$	Average time in sensing	ms
$T_{rxpre}$	Time for preamble reception	ms
$T_{txpre}$	Time for preamble transmission	ms
$T_{txhello}$	Time for hello packet transmission	ms
$T_{rxhello}$	Time for hello packet reception	ms
$T_{ack}$	Time for acknowledgement	ms
$T_{switch}$	Time for radio switching	ms
$T_{txdata}$	Time for data packets transmission	ms
$T_{rxdata}$	Time for data packets reception	ms
$I_{wakeUp}$	Wake-up interval	ms
$I_{hello}$	Hello interval	ms

Table 4.2: Terms used for the energy model

unpredictable manner. As the sink node does not have any energy or storage restriction, the energy model is applicable only to normal nodes. First we calculate the energy consumption for a node for periodic traffic.

In order to save energy, nodes use the LPL technique and most of the time are in sleep mode. We divide the system time  $T$  in small discrete time intervals,  $t_0, t_1, \dots, t_n$ . For simplicity, all the time intervals are normalized to one time unit. A node wakes up at every wake-up interval,  $I_{wakeUp}$ , to perform LPL and broadcasts a hello packet at every hello interval,  $I_{hello}$ . To transmit a preamble, a node first performs carrier sense. The node also senses the environment and sends and receives the data packets. The total energy consumption of a node  $v_i$  per unit of time,  $E_{v_i}$ , is given by its energy consumption in LPL, carrier sense, environment sense, reception, transmission, and sleep states, respectively.

$$E_{v_i} = E_{lpl} + E_{carrier} + E_{sense} + E_{rx} + E_{tx} + E_{sleep} \quad (4.3)$$

$$= P_{lpl}T_{lpl} + P_{carrier}T_{carrier} + P_{sense}T_{sense} + P_{rx}T_{rx} + P_{tx}T_{tx} + P_{sleep}T_{sleep} \quad (4.4)$$

Equation (4.4) shows the power consumption and the time spent by a node in the respective state for the respective time interval. Table 4.2 shows all the terms used for the energy model. Now we calculate the time spent by a node in each of the state. At every wake-up interval, a node performs LPL.

$$T_{lpl} = \frac{T_{alpl}}{I_{wakeUp}} \quad (4.5)$$

Before sending a preamble, a node performs carrier sense. The time required to sense the carrier is given by:



$$T_{carrier} = T_{acARRIER}R_{data} \quad (4.6)$$

Where  $R_{data}$  is the rate at which a node is sending and receiving data packets. The node also senses the environment to measure some application specific physical values such as temperature, humidity, air velocity, and light. The time required to sense the environment is given by:

$$T_{sense} = T_{asense}R_{sense} \quad (4.7)$$

Where  $R_{sense}$  is the rate at which a node is sensing the environment. The transmission time of a node is the sum of the times required to send data packets, preambles, hello packets, and acknowledgement. At every hello interval, a node sends a hello packet. Whenever, it has data to send, it sends a preamble and immediately changes its radio to listen mode in order to receive the acknowledgement from the target node. This process is continuous until it receives an acknowledgement from the target node. In order to support low-latency applications and to save the energy, a node attempts this process for at most  $P$  times.

$$T_{tx} = T_{txdata} + Q(T_{txpre} + T_{switch}) + T_{ack} + T_{txhello} \quad (4.8)$$

Where  $Q$  is the number of attempts a node sends a preamble and changes its radio to receive an acknowledgement, such that  $0 \leq Q \leq P$ .

$$T_{txdata} = L_{data}R_{data}T_{byte} \quad (4.9)$$

$$T_{txhello} = \frac{L_{hello}}{I_{hello}}T_{byte} \quad (4.10)$$

Where  $L_{data}$  and  $L_{hello}$  is the total length of data and hello packet in bytes respectively. The reception time of a node is given by:

$$T_{rx} = T_{rxdata} + \sum T_{rxpre} + 2 \times T_{switch} + T_{ack} + T_{rxhello} \quad (4.11)$$

A node may receive multiple preambles during a time period. But when it becomes a target node for an specific preamble, it immediately changes its radio from receive to transmit mode and sends an acknowledgement to the sender node. In order to receive data from the sender, the node again changes its radio to the receive mode.

$$T_{rxdata} = L_{data}R_{data}T_{byte} \quad (4.12)$$

$$T_{rxhello} = \frac{L_{hello}}{I_{hello}}T_{byte} \quad (4.13)$$

A node is supposed to be in the sleep mode, if it is not doing anything else.

$$T_{sleep} = 1 - (T_{lpl} + T_{carrier} + T_{sense} + T_{rx} + T_{tx}) \quad (4.14)$$

For aperiodic traffic, the important aspect is to decide for the wake-up interval, i.e., the time when a node wakes up in order to perform the LPL and carrier sense and if necessary, the data processing. We calculate this wake-up interval on the basis of the Poisson distribution and calculate the expected number of real-time events, which occur

in the respective interval. If the rate of occurrences in an interval is  $\lambda$ , then the probability that there are exactly  $k$  occurrences is given by:

$$f(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad k \geq 0 \quad (4.15)$$

For every occurrence of the real-time event,  $k$ , the node wakes up and performs a complete process cycle discussed in Figure 4.4.

**System Lifetime:** It is difficult to have a precise criterion to define the WSN lifetime. There exist many lifetime definitions, such as the time when the first node dies, the time some fraction of the nodes die, or the time that the network breaks in two or more segments. We use a more general definition of the lifetime, namely as the time the network is able to provide application-specific services. Therefore, in order to maximize the system lifetime, we simply minimize the energy consumption at each node  $v_i$  except the sink node.

$$\min_{v_i \in V} E_{v_i} \quad i > 0 \quad (4.16)$$

## 4.6 Delay Model

The long sleep duration of LPL results in higher latency for WSN. In AREA-MAC, a node who has real-time data can request/force up-level neighboring nodes to wake up regardless to their normal schedule. The process cycle from sleep-to-sleep mode for the real-time request at a node  $v_i$  is shown in Figure 4.4.

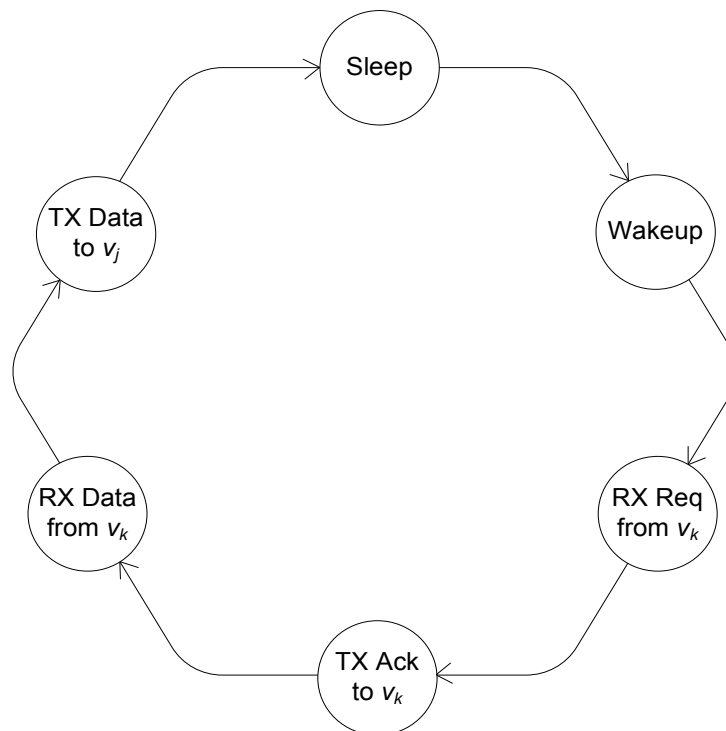


Figure 4.4: A process cycle for the real-time processing at the node  $v_i$ , when it receives a real-time request from its low-level neighbor  $v_k$  and forwards to its up-level neighbor  $v_j$ .

In order to decrease delay and energy consumption, nodes request/force the neighbor nodes with an optimal link cost. First we calculate the total time required to transfer a

data packet from the source node  $v_i$  to the sink node  $v_0$ , where  $i > 0$ . It can be further divided into three steps, delay at the source node, delay at all the intermediate nodes, and delay at the destination node. The delay at the source node  $v_i$  is given by:

$$D_{v_i} = T_{lpl} + T_{sense} + T_{carrier} + T_{tx} + T_{switch} + T_{process} \quad (4.17)$$

Where  $T_{process}$  is the delay to process the packet before forwarding it to the next hop. It depends on network data processing algorithms. In case of multi-hop (in our case, more than 2-level) communication, the forwarding delay,  $D_{forward}$ , is given by the sum of delays at all the forwarding nodes. Let  $F$  be the set containing all the forwarding nodes each denoted by  $v_f$ , such that  $v_f \in F \subset V$ , then  $D_{forward}$  is given by:

$$D_{forward} = \sum_{v_f \in F} (T_{lpl} + T_{carrier} + T_{rx} + T_{tx} + T_{switch} + T_{process} + T_{queue}) \quad (4.18)$$

Where  $T_{queue}$  is the queuing delay, which depends on the traffic load on the node. The delay at the destination node  $v_j$  is given by:

$$D_{v_j} = T_{lpl} + T_{carrier} + T_{rx} + T_{switch} + T_{process} + T_{queue} \quad (4.19)$$

Therefore, the total delay from the node  $v_i$  to the node  $v_j$  is the sum of all three delays.

$$D_{v_i, v_j} = D_{v_i} + D_{forward} + D_{v_j} \quad (4.20)$$

The link cost for sending a real-time packet from node  $v_i$  to  $v_j$ ,  $LC_{v_i, v_j}$ , is a function of the total (expected) delay between  $v_i$  and  $v_j$ , the distance between  $v_j$  and the sink node  $v_0$ , the residual energy at  $v_j$ ,  $R_{v_j}$ , the last time it was in wake-up mode,  $L_{v_j}$ , and the number of neighbors of  $v_j$ ,  $\delta_{v_j}$ . The parameter related to the last wake-up time ensures that all nodes get a fair time to be in the sleep mode. It brighten the chances of all nodes, i.e., the network to die at almost the same time. All the link cost parameters have decreasing priority from left to right.

$$LC_{v_i, v_j} = \{D_{v_i, v_j}, \partial_{v_j, v_0}, R_{v_j}, L_{v_j}, \delta_{v_j}\} \quad (4.21)$$

## 4.7 LP Formulation

The main objective of AREA-MAC is to provide application-specific optimized performance in terms of timeliness and energy efficiency, while maintaining an acceptable system fairness and reasonable trade-off between different critical parameters. In order to achieve the target, two separate optimization problems are discussed in this section.

Our linear program formulation corresponds to two different types of data gathering scenarios, i.e., periodic and aperiodic. Periodic traffic contains the routine data and is not usually critical in terms of delays. Therefore, our objective in such scenarios is to save energy (LP2). This optimization is applicable for all nodes, but the sink node,  $v_0$ . However, aperiodic traffic is generated on the basis of some unexpected events which occur in the sensing area and is usually very critical and needs strict timeliness requirements. In such scenarios, our objective function is to minimize the overall network delay (LP1). For the periodic traffic, nodes send data packets to the sink node along the minimum-energy shortest path (mostly via nearest neighbor) to save the energy and for aperiodic traffic,

nodes send data packets to the sink node along the minimum-delay longest path (mostly via farthest neighbor) to minimize the network delay.

### LP1

$$\begin{aligned}
& \min D_{v_i, v_j} && v_i, v_j \in V; i > j \\
& \text{subjected to:} \\
& \sum C_{v_i, v_j} > 0 && v_i, v_j \in V; i > j && (4.22a) \\
& \sum F_{v_i, v_j} \geq \sum F_{v_k, v_i} && v_i, v_j, v_k \in V; k > i > j && (4.22b) \\
& LC_{v_i, v_j} \leq TH_1 && v_i, v_j \in V && (4.22c) \\
& C_{v_i, v_j} \in \{0, 1\} && && (4.22d) \\
& F_{v_k, v_i}, LC_{v_i, v_j}, TH_1 \geq 0 && && (4.22e)
\end{aligned}$$

### LP2

$$\begin{aligned}
& \min E_{v_i} && v_i \in V; i > 0 \\
& \text{subjected to:} \\
& \sum C_{v_i, v_j} > 0 && v_i, v_j \in V; i > j && (4.23a) \\
& \sum F_{v_i, v_j} \geq \sum F_{v_k, v_i} && v_i, v_j, v_k \in V; k > i > j && (4.23b) \\
& E_{v_i} \leq TH_2 \leq R_{v_i} && && (4.23c) \\
& C_{v_i, v_j} \in \{0, 1\} && && (4.23d) \\
& F_{v_k, v_i}, E_{v_i}, TH_2, R_{v_i} \geq 0 && && (4.23e)
\end{aligned}$$

The constraints (4.22a) and (4.23a) show that each node must be connected to at least one up-level neighbor that can be the sink node. The constraints (4.22b) and (4.23b) assure that the data flow sent by the node should be at least equal to what it has received from the low-level nodes. The constraint (4.22c) restricts the delay conditions and (4.23c) restricts the total energy consumption. Both of these values should not exceed their respective thresholds. The binary variables in (4.22d) and (4.23d) represent the connectivity between  $v_i$  and its neighbors, whereas, (4.22e) and (4.23e) show the non-negativity constraints.

---

## CHAPTER 5

---

# Conclusion and Future Work

In this report, we have elaborated the suitability of different WSN MAC protocols for the time and energy critical applications. As an example, we have discussed detailed working of IEEE 802.15.4, and found several limitations of the protocol for time and energy critical scenarios. We have proposed a scheme to improve those limitations. Application-specific optimal parameters setting is also discussed, where we have found that the GTS slots length requested by devices mostly causes bandwidth under-utilization problem. The coordinator should assign slots depending on the traffic generated by the device. Finally, we have proposed a new MAC protocol for WSN, called as AREA-MAC, which is intended to deal with time and energy critical WSN applications. We have derived the two different linear programs to minimize network delay and to maximize its lifetime, while maintaining reasonable system fairness. Our future plan is to implement the proposed protocol with OMNet++ simulator and to compare it with state of the art protocols. We will use several vital metrics such as end-to-end delay, energy consumption, throughput, and bandwidth efficiency.



---

# Bibliography

- [1] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, 22(2):20–23, April 2003.
- [2] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley and Sons Ltd, 2006.
- [3] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Net.*, vol. 12, no. 3, pp. 493–506, June 2004.
- [4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, October 2002.
- [5] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *ACM SenSys, Los Angeles, CA*, Nov, 2003.
- [6] A. El-Hoiydi and J. Decotignie, "WiseMAC: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," *Ninth IEEE Symposium on Computers and Communication, ISCC04*, pp. 244–251, June 2004.
- [7] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.
- [8] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," *IEEE WCNC*, vol. 3, pp. 1534–1539, March 2004.
- [9] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, pp. 171–180, Nov. 2003.
- [10] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-mac: A short preamble mac protocol for duty-cycled wireless networks," *4th ACM Conf. on Embedded Networked Sensor Systems (SenSys), Boulder, CO*, pp. 307–320, Nov. 2006.
- [11] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," *2nd ACM Conf. on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD*, pp. 95–107, Nov. 2004.
- [12] "IEEE 802.15.4-2006: MAC and PHY specifications for LR-WPANs," <http://ieee802.org/15/pub/TG4.html>.

- 
- [13] A. Koubaa, M. Alves, and E. Tovar, “i-game: An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks,” in *Proc. of 18th Euromicro Conference on Real-Time Systems*.
- [14] —, “GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks,” in *Proc. of the 14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, Apr. 2006.
- [15] Y.-K. Huang, A.-C. Pang, and H.-N. Hung, “An adaptive GTS allocation scheme for IEEE 802.15.4,” *IEEE transactions on parallel and distributed systems*, August 2007.
- [16] L.-C. Ko and Z.-T. Chou, “A novel multi-beacon superframe structure with greedy GTS allocation for IEEE 802.15.4 wireless pans,” *IEEE Communication Society WCNC*, 2007.
- [17] S. Yoo, D. Kim, M. Phan, Y. Doh, E. Choi, and J. Huh, “Scheduling support for guaranteed time services in IEEE 802.15.4 low rate WPAN,” *11th international conference on Embedded and Real-Time Computing Systems and Applications IEEE*, 2005.
- [18] “IEEE 802.15 working group for WPAN,” <http://www.ieee802.org/15/index.html>.
- [19] “Zigbee,” <http://www.zigbee.org>.
- [20] N. Salles, N. Krommenacker, and V. Lecuire, “Performance analysis of IEEE 802.15.4 contention free period through real-time industrial maintenance applications.”
- [21] “Tinyos 2.x,” <http://www.tinyos.net/tinyos-2.x>.
- [22] “Moteiv Tmote,” <http://www.sentilla.com/pdf/eol/tmote-sky-brochure.pdf>.
- [23] “Chipcon CC2420,” <http://focus.ti.com/docs/prod/folders/print/cc2420.html>.
- [24] I. Joe, “A patterned preamble mac protocol for wireless sensor networks,” *IEICE Trans. Communication*, vol. E91-B, no. 2, Feb. 2008.