Semi-Persistent Medium Access Control Protocols for Wireless Sensor Networks

By

Mouhcine Guennoun

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Ph.D. degree in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Wireless Sensor Networks (WSNs) are dense clusters of sensor nodes, made up of small, intelligent, resource-constrained wireless devices that are deployed to monitor a specific phenomenon in a certain field. The sensor nodes can be constrained by limited power supply, memory capacity and/or processing capabilities, which means that the design of WSNs requires all algorithms and protocols to be lightweight and efficient, and use as little power as possible. The Medium Access Control (MAC) protocol in WSNs, defined by the IEEE 802.15.4 standard, employs the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) algorithm to control the nodes contending for access to the communication medium. Though the performance of this protocol has been studied extensively, and several improvements to its backoff counter, superframe format and contention-free period (CFP) features have been proposed, very few studies have addressed improving the Clear Channel Assessment (CCA) feature. In this thesis, we study the impact of increasing the value of the contention window beyond the standard value of 2, on the performance of the MAC protocol. We propose a semi-persistent MAC protocol that is a hybrid form of 802.11 and 802.15.4, to achieve a favorable performance that can serve a broad range of applications over the IEEE 802.15.4-based WSNs. We build an analytical model of the proposed protocol based on Markov chain modelling and derive the analytical expressions of the performance metrics, which we then validate against the simulation result sets generated by our in-house built simulation framework. We prove analytically that the probability of collision of the semi-persistent MAC is lower than that of the standard protocol. Based on our theoretical and simulated models, we show that incorporating the semi-persistent feature into existing MAC protocols leads to significant improvement of the performance metrics, including the probability of collision, throughput, energy consumption, transmission delay and reliability, particularly for networks with a large number of sensor nodes.

Acknowledgements

All praise and thanks be to Almighty Allah for giving me the strength to complete this thesis.

I extend my sincerest appreciation to my supervisor, Prof. Hussein T. Mouftah, for his encouragement, support and constructive feedback. His sound advices had a positive impact on my research work. Thank you for all.

For their support and contributions, I would like to thank my colleagues Mounib Khanafer, Junaid Israr, Yamen Nasrallah and Sherif Awad.

Pour leurs encouragements et leur assistance, je remercie chaleureusement mon père, que Dieu l'ait en sa sainte miséricorde, ma mère, mes frères et sœurs, Zine El-Abidine, Loubna, Mohammed et Lamia, ainsi que tous les membres de ma famille.

Je remercie également mes amis Souhail, Yamina, Hassan, Hicham, Yassir et Khalil.

Dédicace

Je dédie ce travail à ma femme Maha et mon fils Hassan Abdallah. Votre amour est ma source d'inspiration.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABA	Adaptive Backoff Algorithm
ACK	Acknowledgement packet
ADAPT	ADaptive Access Parameters Tuning
ADW	Adaptive length Distribution Window
BCS	Backoff Counter Selection
BE	Backoff Exponent
BEB	Binary Exponent Backoff
BED	Backoff Exponent Differentiation
BI	Beacon Interval
BO	Beacon Order
BOAA	Order Adaptation Algorithm
BS	Base Station
CACCA	Coexistence Aware Clear Channel Assessment
CAP	Contention Access Period
CCA	Clear Channel Assessment
CDW	Constant length Distribution Window
CFP	Contention Free Period
CSMA-CA	Carrier Sense Multiple Access-Collision Avoidance
CTS	Clear-to-Send
CV-RMSD	Coefficient of variation of the root-mean-square deviation
CW	Contention Window
CWD	Contention Window Differentiation
DCLA	Duty Cycle Learning Algorithm
DW	Distribution Window
EBEM	Enhanced Beacon-Enabled Mode
ED	Energy Detection
FRT	Frame Tailoring
FSM	Finite State Machine
GSA	GTS Scheduling Algorithm
GTS	Guaranteed Time Slot

IFS	Inter Frame Space
LIB	Linear Increase Backoff
LIFS	Long Inter Frame Space
LPL	Low Power Listening
MAC	Medium Access Control
NB	Number of Backoff
PAN	Personal Area Network
PHY	Physical layer
PAN	Personal Area Network
PD	Preamble Detection
PDU	Protocol Data Unit
PJ	Priority Jamming
PRT	Priority Toning
PSD	Power Spectral Density
PW	Periodic Wakeup
QoS	Quality of Service
RTS	Request-to-Send
SCCA	Short Clear Channel Assessment
SD	Superframe Duration
SDS	Superframe Duration Scheduling
S-LPL	Synchronized Low Power Listening
SIFS	Short Inter Frame Space
SNR	Signal to Noise Ratio
SO	Superframe Order
SP	Semi Persistent
SP-MAC	Semi Persistent Medium Access Control
SP-BEB	Semi Persistent Binary Exponential Backoff
SP-ABA	Semi Persistent Adaptive Backoff Algorithm
TDBS	Time Division Beacon Scheduling
TDMA	Time Division Multiple Access
VPs	Virtual Preambles
Wi-Fi	Wireless Fidelity

WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

List of Symbols

$b_{i,j}$	Probability of being at state $S_{i,j}$
CCA_i	The i^{th} Clear Channel Assessment
D	Mean time transmit a frame (frame delay)
L	Frame length
n_{cca}	Maximum value of number of CCAs
E_c	Total energy consumed during collisions
E_{cca}	Total energy consumed during the CCA states
E_{idle}	Total energy consumed during the backoff states
E_{rx}	Total energy consumed during frame reception
E_{total}	Total energy consumed
E_{tx}	Total energy consumed during transmissions
m	Maximum number of backoffs, *macMaxCSMABackoffs*
n	Maximum number of retries, macMaxFrameRetries
n_{af}	Number of access failures
n_{BC}	Number of busy cycles
n_{CC}	Number of collision cycles
n_c	Number of collided frames
n_{cf}	Number of channel failures
n_s	Number of successful frames
n_{SC}	Number of successful cycles
$\eta(S_{i,j})$	Number of nodes, other than the current node, at state $S_{i,j}$
N	Total number of nodes of the network
p_c	Frame collision probability
Pr(X=Y)	Stationary probability of the random variable X being at state Y
R	Reliability
$S_{i,j}$	State (i,j)
T_{BC}	Total time consumed by busy cycles
T_{CC}	Total time consumed by collision cycles
T_{idle}	Channel idle time
T_{SC}	Total time consumed by successful cycles

T_{total}	Total time to transmit all frames
$\overline{T_{CC}}$	Mean time of a collision cycle
$\overline{T_{SC}}$	Mean time of a successful cycle
$\overline{T_{BC}}$	Mean time of a busy cycle
U	Channel utilization
W	Maximum number of random backoff time
α_i	Probability of finding the i^{th} CCA busy
π_{SC}	Probability of a successful cycle
π_{BC}	Probability of a busy cycle
π_{CC}	Probability of a collision cycle
τ	Channel sensing probability (Probability of being at state $S_{1,1}$)

Chapter 1

Introduction

1.1 Background

The need for Multiple Access (MA) schemes is necessary in situations where several nodes share the medium of communication. Collisions in these situations are inevitable and MA schemes play a crucial role in dealing with them such that network resources are utilized fairly and efficiently. MA schemes are further enhanced by adding the Carrier Sense (CS) capability; thus CSMA schemes have emerged. The basic idea of CSMA schemes is that any node, before commencing any frame transmission, is required to listen to the medium to ensure that no ongoing transmissions are occupying the communication medium. In case the medium is found busy, the node should refrain from sending its frame and should wait for a certain duration (dictated by the specific CSMA scheme deployed) before re-attempting to transmit. Collisions happen when at least two nodes send their frames at the same time. CSMA, as we explain shortly, provides means to prevent or recover from a collision. This CSMA mechanism is an essential constituent of the design of Medium Access Control (MAC) protocols.

CSMA has two main modifications depending on whether it operates in wired networks or wireless networks. In wired LANs, like Ethernet (IEEE 802.3), it is notable that the shared medium is characterized by low levels of thermal noise with little signal attenuation [RAM07]. This means, with broadcast communication, nodes can easily *detect* transmissions that belong to a certain single-hop neighbour node. Nodes can then detect collisions on the channel by measuring the level of power over the communication channel and comparing it a designated threshold. This paradigm is referred to as CSMA with Collision Detection or CSMA/CD. With CSMA/CD, transmissions are terminated as soon as a collision is detected. This way the node can avoid wasting time and energy in continuing its transmission.

On the other hand, in wireless networks, like Wi-Fi (IEEE 802.11) and ZigBee (IEEE 802.15.4), nodes are hindered by several challenges that make it difficult to adopt the CSMA/CD mechanism. First, signals transmitted among nodes suffer from severe attenuation, and therefore, not all nodes can be aware of the ongoing transmissions over the wireless medium. Moreover, the wireless medium is open and can be highly affected by interference and noise. Finally, wireless systems operate in a half-duplex manner and this obstructs collision detection. These facts necessitate the need for another version of CSMA that can achieve a more efficient use of the medium. Such a version has been designed and named as CSMA with Collision Avoidance or CSMA/CA. In essence, CSMA/CA does not support recovery from collisions. It allows nodes to sense the medium to check if it is busy or not before sending any frames. CSMA/CA adopts the concept of Clear Channel Assessment (CCA) to control access to the wireless medium. With CCA, a node, before transmitting any frame, is kept in the receive-mode for a certain period of time; the average received power is measured during this time. If the level of the received power is above a designated threshold, the node assumes the medium to be busy and it backs off. Otherwise, the node switches to the transmit-mode and starts sending the frame.

While both Wi-Fi and Zigbee implement the CCA concept, they differ in their operating parameters. In particular, Wi-Fi utilizes a bandwidth of 22 MHz, a transmission power of 20 dBm, Power Spectral Density (PSD) of 6.6 dBm/MHz, a CCA time less than 4 μs, a receive-to-transmit mode switch time less than 5 μs, a minimum frame duration of 28 μs, and a maximum frame duration of 12416 μs. For ZigBee, these parameters are set at 2 MHz, 0 dBm, -3 dBm/MHz, 128 μs, 192 μs, 320 μs, and 4256 μs, respectively. The differences in bandwidth and power settings affect the detection sensitivity of each technology.

The implementation of CCA, in both Wi-Fi and ZigBee networks, depends on whether we have narrowband systems or wideband systems. Narrowband systems are characterized by a signal bandwidth-to-center frequency ratio (fractional bandwidth) typically less than 0.1% [RAM07]. In these systems, the signal PSD is notably higher than the noise floor. This allows for detecting narrowband signal transmission reliably using non-coherent energy detection (ED) (that is, integrating the square of the received signal or signal envelope over a suitable period of time [RAM07]). However, in wideband systems, where the fractional bandwidth is 1-5%, the signal PSD is not above the noise floor sufficiently to be detectable by non-coherent ED. This situation worsens with ultra-wideband systems where the fractional bandwidth is around 20% with signal PSD below the noise floor. Therefore, coherent detection approaches are developed for wideband systems. In these approaches, the sensing node synchronizes with the ongoing transmission by reading the preamble transmitted in front of every frame (thus, this type of detection is called Preamble Detection (PD)). The preamble is formed of repetitions of a sequence of known symbols (the sequence is designed for a *near-ideal* autocorrelation property [RAM07]). The receiver correlates the known sequence with the incoming signal with different time offsets. The correlation will be high at the time offset corresponding to time synchronism; this is attributed to the processing gain that results from the repetition of the known symbols. The high correlation indicates that a signal is present.

Preamble Detection based CCA is supported by both Wi-Fi and ZigBee and can effectively improve the detection sensitivity. However, these technologies utilize different preambles, and this hinders cross-technology detection. The CCA duration in ZigBee is sufficiently long, which allows for a suitable level of sensing sensitivity. Therefore, PD is usually disabled in ZigBee. On the other hand, Wi-Fi uses a relatively short CCA, and therefore, PD is enabled by default in order to maximize the sensing sensitivity.

1.2 Motivation

In 802.11-based networks, a node has to persistently monitor the wireless medium to check if any ongoing transmission is in progress. If the medium is found idle for a Distributed Inter-Frame Space (DIFS) period (which is defined in the standard), the node proceeds and transmit the frame. If the medium is found busy, the node backs off for a randomly generated time drawn uniformly from the interval [0, CW], where CW is in the range [15, 1023] in units of aSlotTime (set at 50 μs)). The medium is sensed during each backoff slot. As long as the medium is sensed idle, the backoff timer keeps on decreasing in steps of aSlotTime. If the medium is sensed busy at any slot, the backoff timer is frozen and will not resume until the medium becomes idle again for a DIFS period. The transmission of the frame starts once the backoff timer reaches zero.

For 802.15.4-based nodes, after backing off similar to 802.11, two CCAs are conducted before any frame can be transmitted. The communication medium should be sensed free during the two CCAs for the frame to be sent out. It is important to note that all operations (including CCA) in 802.15.4 should only begin at the boundary of time slots [CAM11]. Furthermore, in 802.15.4, nodes do not conduct CCA unless the backoff counter is zero while 802.11-based nodes keep sensing the medium during backoff.

From these descriptions we can see that 802.11 implements a *persistent access mode* CSMA-CA while 802.15.4 implements a *non-persistent access mode* CSMA-CA.

Although a large body of research has tackled the weaknesses inherent in the CSMA-CA mechanism adopted by IEEE 802.15.4, a limited number of contributions have observed the importance of focusing on the CCA feature of this standard. We highlight this point because the total number of CCAs conducted by a node before managing to access the medium has a strong indication on the level of activities over the wireless medium. It also has a direct impact on the consumption of the node's power resources. Furthermore, there has been no effort that attempted to devise a MAC protocol, oriented to WSNs, that adopts the strengths of both 802.15.4 MAC and 802.11 MAC to best benefit the performance.

1.3 Objective

The main objective of this thesis is to explore the potential of building a *hybrid* MAC protocol for beacon-enabled 802.15.4-based WSNs that incorporates the persistent mode of 802.11 MAC into the operation of 802.15.4 MAC. We study the impact of increasing the number of Clear Channel Assessments (CCAs) on the performance, and control this increase adaptively to enhance the efficiency of the network. Our main goal is to model, analyze and design a MAC protocol that can respond to changes in the network, in terms of network size, traffic intensity or traffic urgency.

1.4 Contributions

The main contributions of our research in the area of IEEE 802.15.4-based WSNs are to:

1. Introduce a novel study that demonstrates how the CCA feature of 802.15.4 and 802.11 standards can be combined to design an efficient, hybrid MAC protocol for WSNs.

2. Model the proposed protocol using Markov chains and analytically derive performance metrics, including the probability of collision, throughput, idle time, collision time, delay, reliability, energy consumption and others.
3. Develop a general purpose simulator to model the operations of the CSMA/CA algorithm of the 802.15.4 WSN MAC protocol and measure the performance metrics using the simulation platform. The data collected from the simulation environment is the basis for the evaluating the performance of the proposed protocol.
4. Validate the theoretical and simulation models, using the covariance square root mean deviation metric to measure the deviation between the analytical and simulation models.
5. Develop a Semi-Persistent IEEE 802.15.4 MAC protocol to support energy efficient, reliable and timely communications, by tuning the CW parameter of CSMA/CA algorithm.
6. Prove analytically that the proposed protocol achieves better performance than the standard protocol.
7. Study the effect of incorporating the semi-persistent feature of the proposed MAC protocol with existing MAC protocols for wireless sensor networks.

1.5 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 reviews related work intended to improve the 802.15.4 MAC in WSNs by exploiting the CCA feature. In Chapter 3 we develop a mathematical model based on Markov chain that describes the functionality of the Variable CCA MAC protocol, and in Chapter 4 we conduct extensive simulations to validate the model. Chapter 5 analyzes the performance of the Variable CCA protocol, and proposes a semi-persistent hybrid MAC protocol that takes advantage of the strength of both the 802.11 and

802.15.4 protocols. Finally, Chapter 6 concludes the thesis and discusses potential future research directions.

1.6 List of Publications

In preparing this thesis, the following publications have been contributed in the literature:

1. Mouhcine Guennoun and Hussein T. Mouftah, “Semi-Persistent CSMA/CA for Efficient and Reliable Communication in Wireless Sensor Networks”, Proc. of 2014 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE2014), pp CN-3.4.1-CN-3.4.6, Toronto, Canada, May 4-May 7, 2014.
2. Mounib Khanafer, Mouhcine Guennoun and Hussein T. Mouftah, “A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks”, IEEE Communication Surveys and Tutorials, Volume 16, Issue 2, pp 856-876, December 2013.
3. Mouhcine Guennoun and Hussein T. Mouftah, “Modeling of Variable Clear Channel Assessment MAC Protocol for Wireless Sensor Networks”, Elsevier Computer Communications. (under review)
4. Mouhcine Guennoun and Hussein T. Mouftah, “Model Validation and Analysis of Variable Clear Channel Assessment MAC”, IEEE Access journal. (under review)

Chapter 2

Survey of Related Work

2.1 Introduction

In this chapter we pay a closer look to the state-of-the-art in the area of enhancing the IEEE 802.15.4 MAC protocol. Our aim is to identify the directions followed by the researchers to improve this protocol and unveil the aspects that have gained limited attention from them, yet can be exploited for further improvements.

2.2 Wireless Sensor Networks

The advent of small-sized, cheap, and intelligent wireless devices that are able to self-organize themselves and collaborate in collecting data about certain physical phenomena (like sound, temperature, humidity, vibration…etc) and events paved the way for the realization of a new set of applications. These devices, or sensor nodes, are densely deployed in a certain field to form a Wireless Sensor Network (WSN). The applications supported by such a network span civil, industrial, and military-based domains (see [AKY02], [BAR07], and [XIA11]). Each of these sensor nodes is characterized by being battery-powered and having a processor, a sensor, and a radio transceiver. As data is collected by these nodes, they get conveyed to a resourceful Base Station (BS) that processes it and perform appropriate actions whenever needed. Sensor nodes are distinguished by being highly limited in their processing capabilities, memory~~,~~ and power resources. This requires special protocols and algorithms that should be lightweight, yet efficient. WSNs are usually highly populated with sensors. Furthermore, these sensors are usually deployed in hostile environments, like battlefields or forests, with minimal human supervision.

These facts make it unattainable to replace or recharge the depleted sensor batteries. Therefore, the primary design requirement in any WSN is to be power-conservative. This is a unique feature of WSNs that distinguishes it from other types of wireless networks (like wireless ad hoc networks).

The IEEE 802.15.4 standard defines the specifications of the PHY layer and MAC sub-layer for Low-Rate Wireless Personal Area Networks (LR-WPAN) [ZIG06]. This is the de facto standard for WSNs and provides the appropriate specifications that conform to their requirements. It supports both star and peer-to-peer network topologies. The star topology requires the existence of a PAN coordinator (or simply, the *coordinator*) that conveys the messages between any communicating pairs of nodes. In the peer-to-peer topology, however, this coordinator is not required. The IEEE 802.15.4 standard defines a superframe structure to coordinate the communications over the wireless medium. This superframe is managed and sent by the coordinator. To synchronize the nodes, the coordinator uses beacons that are sent bounding the superframe. The superframe is composed of 16 equal slots as shown in Figure 2-1. As shown in this figure, the superframe is generally composed of active and inactive periods. The active period is a mandatory portion of the superframe. The inactive period is an optional part of the superframe. If available, the inactive period provides a chance for the coordinator and the communicating nodes to go into a sleep mode and conserve power. The active period itself is divided into a contention access period (CAP) and an optional contention-free period (CFP). Nodes contend to access the wireless medium during the CAP. Basically, they utilize the slotted CSMA-CA mechanism during this period. The CFP is designed to support QoS parameters. It is used by time-sensitive applications that require bandwidth guarantees. The CFP is divided into guaranteed time slots (GTSs). Each GTS is granted by the coordinator to the nodes upon request.

A GTS, which is composed of several time slots, is dedicated to a certain node upon request. Once the coordinator assigns a node a certain GTS, the node can access the medium during that GTS without contention from other nodes.

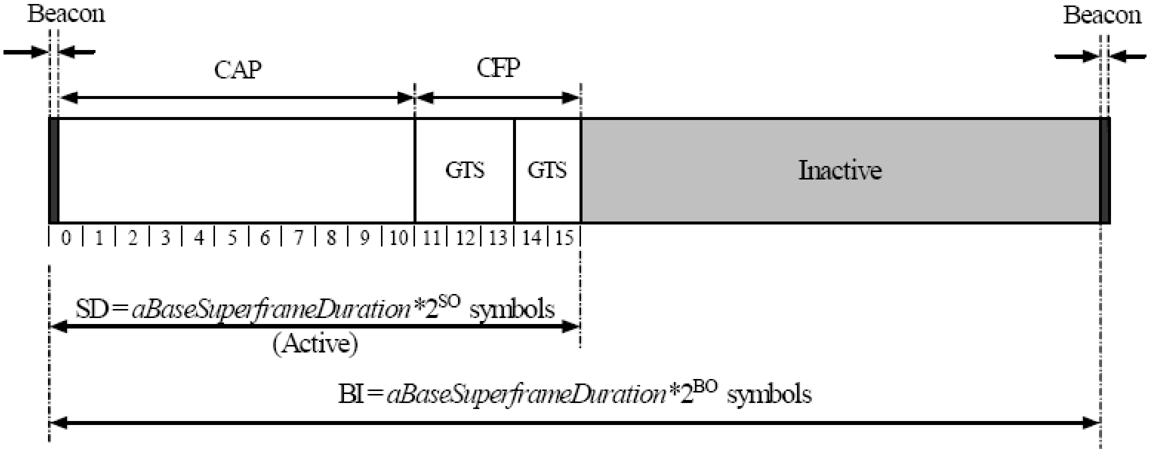


Figure 2-1: Superframe Structure (taken from [ZIG06]).

Figure 2-1 shows that the structure of the superframe is specified by the *macBeaconOrder* (BO) and *macSuperframeOrder* (SO) attributes. The BO attribute specifies when the coordinator can send out the beacon frames. The SO attribute specifies the duration of the active part of the superframe.

PANs that employ the superframe are referred to as beacon-enabled PANs. These PANs depend on the slotted CSMA-CA mechanism to coordinate the access to the communication medium. On the other hand, PANs that do not use the superframe structure are referred to as nonbeacon-enabled PANs. With these PANs the unslotted CSMA-CA mechanism is used to control how nodes access the communication medium. In this thesis, our focus is on beacon-enabled PANs.

The operation of the slotted CSMA-CA mechanism is described as follows. Before a node attempts to send any frame it initializes three parameters, namely, the number of backoff stages (NB), the contention window (CW), and the backoff exponent (BE). The initialization values are well-described in [ZIG06]. Then, the node backs off for a duration that is chosen randomly from the range [0, 2^{BE}-1] time units. Once the backoff counter expires, the node conducts an initial clear channel assessment (CCA1). This is intended to check whether the wireless medium is free of any ongoing transmissions or not. CCA1 is followed by another CCA (CCA2). The node starts transmitting its frame only if finds the medium idle during both CCAs. In contrast, if any of the CCAs reveals that the medium is busy, the node increases the values of NB and BE by one. The standard specifies that the maximum values of NB and BE are *macMaxCSMABackoffs* (with default value of 4) and *macMaxBE* (with default value of 5), respectively (check [ZIG06] for more details). If NB exceeds *macMaxCSMABackoffs*, the node will discard the frame. If BE reaches *macMaxBE*, the node will keep BE at this value until it is reset (due to frame transmission or dismissal). Whenever the node fails to access the medium, it dismisses the frame and re-starts the CSMA-CA backoff process again by generating a new number of complete backoff periods. On the other hand, when the node manages to access the medium and transmit its frame, it waits for the acknowledgement (ACK) to be sent back. If the node receives no ACK, it will re-transmit its frame several times up to *macMaxFrameRetries* attempts (with default value of 3). With every transmission retry, the process of CSMA-CA backoff, which we have explained above, is performed. If the *macMaxFrameRetries* limit is exceeded, the node will dismiss the frame. This whole process of backing off to ensure that the wireless medium is free to commence a transmission is referred to as the Binary Exponential Backoff (BEB) algorithm.

2.3 State-of-the-art IEEE 802.15.4 MAC Protocols

2.3.1 CCA-Based Approaches

Lee et al. in [LEE10] propose a new algorithm, called the Additional Carrier Sensing (ACS) algorithm, to improve the performance of CSMA-CA in WSNs. ACS operates under acknowledged traffic conditions. The design of ACS notices that a node finds CCA2 busy in two situations. The first situation is when this node conducts its CCA1 while another node is conducting CCA2; the latter node will start its frame transmission during the CCA2 of the former node. The second situation is when a node starts its CCA1 while another node has already finished sending its frame and waiting for the ACK frame to be sent back. This means that the latter node will be receiving the ACK frame during the first node's CCA2. The ACS algorithm tackles the second situation by letting the node that sensed a busy CCA2 to further conduct a third CCA (i.e., CCA3) after a delay of one timeslot. This gives the node the opportunity to send its frame directly after the ACK frame transmission is finished, provided that the medium is sensed idle during CCA3. The advantage of this approach is that it saves the node the time that it will waste, after the busy CCA2, in backing off and then re-conducting two CCAs. The authors have shown through simulations that ACS can outperform the standard CSMA-CA in terms of throughput and delay. In fact, ACS has even lessened the number of CCAs encountered by a node to successfully send its frame. The direct benefit of this behavior is achieving more savings in power consumption during CCAs.

Kim in [KIM12] argues that always performing two CCAs before accessing the medium leads to inefficient performance as it burdens nodes with unneeded delay and energy consumption. Therefore, the author proposes to improve the efficiency of the CSMA-CA algorithm by employing only a single CCA; the resulting scheme is referred to as the Short Clear Channel Assessment (SCCA) algorithm. SCCA aims at learning the status of the communication channel by utilizing a single CCA with minimal effort. The author firstly highlights that the employment of two CCAs in the standard CSMA-CA is to prevent any collision with the ACK frame. Basically, the time waited by a node to receive the ACK frame is defined to be between

12 and 32 symbols. After a frame has been sent, the time remaining to hit the boundary of a backoff slot (denoted as t_r) is checked. If it is greater than or equal to 12 symbols, the ACK frame will be sent during the next backoff slot, which makes any node conducting CCA1 at that slot see the medium busy. However, if t_r is less than 12 symbols, the next backoff slot will be idle and a node conducting its CCA1 at that slot will see the medium free. In order to avoid colliding with the ACK frame, another CCA is enforced. Thus, the latter node will find the medium busy during CCA2 because the ACK frame will be in transmission. With SCCA, the behavior of the system when t_r is greater than or equal to 12 symbols will not be changed. However, when t_r is less than 12 symbols, SCCA keeps the sending node in the transmission mode and lets it send a 1-byte *busy tone* packet at the beginning of the next backoff slot. This way, the rest of the nodes that are conducting their CCA1 will sense the medium busy. In other words, SCCA manages to eliminate the need for a second CCA and nodes need always to conduct a single CCA to check the status of the communication channel. Simulation results show the ability of SCCA to outperform the standard CSMA-CA in terms of average CCA delay and average energy consumption.

Shin et al. propose the Cascaded-CCA approach in [SHI07] to enhance the MAC protocol performance in both 802.11 and 802.15.4. The Cascaded-CCA integrates the advantages of Energy Detection (ED) and Preamble Detection (PD), both of which we explain shortly, mechanisms to develop a more flexible CCA approach. ED has been used traditionally to implement narrowband CCA. With ED, the detection of a signal is based on measuring the signal energy around the carrier frequency. In PD, on the other hand, the sensing node synchronizes with the ongoing transmission by reading the preamble transmitted in front of every frame. The preamble is formed of repetitions of a sequence of known symbols. The receiver correlates the known sequence with the incoming signal with different time offsets. A high correlation indicates that a signal is present. Although ED is conservative in its power requirements, it is unreliable in detecting wideband signals. In contrast, PD operates reliably with expensive power requirements. Cascaded-CCA works on combining the strengths of ED and PD in one scheme as follows. The ED module is kept running at all times and it integrates the received signal over several symbol durations (say, *n* symbols) and produces an output at symbol rate [SHI07]. If the integrated output crosses the predefined ED threshold, the PD module starts up. Then, the node directly performs a correlation between the received signal and the known sequence of symbols as we have explained above. If the correlation output crosses the threshold of the PD, the Cascaded-CCA concludes that the signal is present. Otherwise, the node returns to the ED module and continues observing the communication channel. The benefit of keeping ED running is to reduce the expenditure of power. This strategy proves usefulness in situations where the traffic is of a sporadic nature. Furthermore, by varying the threshold of ED, Cascaded-CCA offers an interesting ability to control the trade-off between power consumption and reliability. In particular, as ED generates more false alarms, PD will be triggered more frequently unnecessarily, but the detection of signals will rise and a better throughput is achieved. On the other hand, if the ED is configured to generate a low false alarm rate, the power efficiency will improve, but the signal detection probability will get reduced. Simulations results show that the Cascaded-CCA scheme achieves an intermediate performance, in terms of power consumption and throughput, when compared to pure ED and PD schemes. In other words, Cascaded-CCA provides an opportunity to balance power consumption and throughput for the best MAC performance.

Another approach that exploits the CCA feature of 802.15.4 is proposed by Yuan et al. in [YUA10b]. This is a decentralized approach that aims at resolving the problem of interference that occurs when 802.15.4 nodes operate in the vicinity of 802.11b/g networks. The idea is to adjust the CCA thresholds, in the presence of severe interference, in an adaptive and distributive manner. As 802.15.4 and 802.11b/g networks coexist, interference leads to high channel access failures and/or frame collisions. As nodes suffer from excessive channel access failures, they will have to conduct CCAs repeatedly to transmit a single frame, and this costs nodes more power. Therefore, to reduce the level of interference, resulting from the neighboring 802.11b/g networks, the authors propose to control the ED thresholds that the CCA depends on to check the channel status. In the presence of heavy interference, 802.15.4 nodes will increase their ED thresholds such that the number of channel access failures is reduced. However, as the level of interference reduces, nodes set the ED thresholds back to their default values in order to give nodes a fair channel access privilege. The conducted simulations show that this approach improves the throughput of the 802.15.4 nodes in the presence of 802.11b/g networks.

2.3.2 Cross Layer-Based Approaches

Ramachandran and Roy in [RAM06] focus on the cross-layer interaction between the PHY and MAC layers and its direct impact on CCA. They highlight that CCA is basically implemented at the PHY layer, but its functionality directly impacts the operation of MAC. Consequently, performance parameters like throughput and energy efficiency are highly dependent on how CCA works. The authors evaluate the cross-layer dependency of CCA and devise a set of heuristics that can adjust CCA parameters, based on traffic and channel conditions, to better improve MAC performance. The three methods of CCA that we have discussed earlier are evaluated based on their ability to detect signals, generate false alarms, and reduce power consumption. The evaluation has revealed that the following heuristics are recommended for a better performance. In one heuristic, it is recommended to use ED for very sparse traffic, PD for near-saturation traffic, and decorrelation-based CCA (which uses a coherent signal detection without utilizing the preamble) when the network is moderately loaded [RAM06]. In another heuristic, we may adjust CCA to generate lower false alarms at low traffic rates and higher detection ability at higher traffic rates. A third heuristic may be a SNR-dependent one that initiates ED at large SNRs and PD otherwise.

Kim and Choi in [KIM06] tackle the problem of inherent frame delays in 802.15.4 slotted CSMA-CA. The focus of this study is on event monitoring networks. The authors propose a priority-based scheme constituted by Frame Tailoring (FRT) and Priority Toning (PRT). The core idea is to create a schedule according to which different groups of nodes are permitted access to the medium. This schedule is solely based on the priority of frames. This approach can be better viewed as if we are scheduling the medium access according to a "pseudo" time division multiple access (TDMA) [KIM06]. The direct benefit of this scheduling is a reduced level of contention among the nodes to access the medium. In consequence, nodes experience less delay in their communications. Furthermore, the use of a priority-based scheduling draws an upper limit on the delay that a high priority frame encounters. FRT utilizes a one-time CCA to support frame prioritization (reducing the CCA overhead by half is anticipated to boost the performance). The use of a one-time CCA without having collisions between the ACK frame and regular frames is explained as follows. Firstly, the authors define the *frame tail* to be the length of the remainder after the total frame length is divided by the backoff slot length (i.e., 20 symbols) [KIM06]. Then they explain the behaviour of the standard in dealing with different lengths of the frame tail. If the frame tail is of length less than 8 symbols, the ACK frame is sent back in the next timeslot after the transmitted frame. However, if the frame tail is 9 to 19 symbols long, the ACK is delayed one timeslot after the frame has been sent. In the latter case, we can see the effectiveness of using a two-CCA strategy in order to avoid having collisions with the ACK frame. FRT, however, adjusts the length of the data frame such that it keeps t_{ACK} at a 12-symbol length. This way, a one-time CCA will always be sensed busy in the time period between a frame and its ACK. As a result, using a one-time CCA can be highly effective in supporting the notion of high prioritization. Based on that, a node that has a frame to transmit will firstly check if tailoring is needed or not. If tailoring is needed, the node will pad at the end of its frame as many zeros as needed. Of course, the length field in the physical layer PDU will record the correct length of the original frame. This way, the receiving node can easily identify the useful part of the frame. The attached zeros will be useful in making the rest of the other nodes, other than the receiving one, see the medium busy and abandon the need for a second CCA. Besides FRT, PRT dedicates a portion of the active period of the superframe for the transmission of high-priority frames. With PRT, nodes with high-priority frames send a tone signal in the timeslot right before the coordinator sends the beacon. The coordinator is required to listen to the medium during that timeslot in order to learn about any important frames, if any. Upon detecting a tone signal, the coordinator includes this information in the beacon so that all nodes with normal-priority frames defer their transmissions by a certain period of time. This behavior plays an important role in reducing the level of contention among the nodes. The authors finally assess the performance of this new priority-based scheme through simulations. The simulation results show that this scheme, compared to the standard, is able to reduce frames delays and also relax the probability of deferring the transmission of a frame.

The CCA concept is also employed by Tytgat et al. in [TYT12] to tackle the coexistence problem through the Coexistence Aware CCA (CACCA) concept. The authors firstly stress that all CSMA-CA-based technologies employ the CCA concept to check the status of the communication channel. However, each technology tailors the CCA to its specific needs. Therefore, precautions should be taken when different CSMA-CA-based networks are collocated because the CCA feature may make a network less aware of the existence of other networks in its vicinity. This will eventually lead to unwanted cross-technology collisions. The CACCA concept can mitigate the latter situation by enabling the coexisting nodes to backoff such that the level of interference is reduced. This can be achieved by the use of a fast and accurate device, called the sensing engine; that measures spectral power density across a wide bandwidth [TYT12]. The sensing engine can detect the existence of other networks quickly and reliably due to its ability to analyze a limited bandwidth within a very short time period. For example, the sensing engine can reliably detect a ZigBee network within the Wi-Fi CCA time. In such a scenario, the sensing engine will be able to reduce the ZigBee CCA time from 128μs to only 4μs, which is equal to the Wi-Fi CCA time. The authors further study the impact of deploying the sensing engine on the ZigBEE side only, the Wi-Fi side only, or both sides. Their results show that implementing the sensing engine on both ZigBee and Wi-Fi nodes helps in lowering the level of frame loss and improves the reliability in the ZigBee network.

2.3.3 Priority-Based Approaches

The lack of support for QoS in CSMA-CA is addressed by Shin in [SHI13]. The author proposes a new priority scheme, called the Priority Jamming (PJ), to support service ***differentiation*** in 802.15.4. PJ favors the transmission of high priority frames over low priority ones. The CCA feature is modified for PJ to achieve its set objective. Basically, a node with a high priority frame to transmit will perform CCA for duration of 8 symbols as required by the standard. However, the node will dedicate some slot time to send a jamming signal that signifies the availability of high priority data to be transmitted. The jamming signal can be realized with any signal with duration less than 8 symbols. For normal priority frames, the CCA will span over 20 symbols. This arrangement allows nodes with normal priority frames to sense the jamming signal sent by nodes with high priority frames, and therefore, the former nodes will deem the communication channel as busy and defer their frame transmissions. This way, PJ manages to reduce the likelihood of collision among nodes with high and normal priority frames. It is interesting to notice that in case all of the frames are of the same priority (normal or high), PJ will behave similar to the standard CSMA-CA. Exploiting the CCA feature of CSMA-CA in the way we have described allows PJ to be applied in both beacon-enabled and non-beacon enabled 802.15.4 networks. The performance of PJ has been assessed through simulations. The collected results have shown that PJ outperforms the standard CSMA-CA in terms of delay and throughput. However, the use of the jamming signal (high priority frames) and the extended CCA duration (normal priority frames) costs PJ a marginal increase in its energy consumption compared to CSMA-CA.

An algorithm for efficient scheduling of beacons has been devised by Yen et al. in [YEN08]. The algorithm is probabilistic and risk-aware and can form collision-free beacon schedules. The authors discuss the traditional methodology of letting nodes schedule their beacons to avoid the reuse of beacon slots and argue that this rule is too restrictive. Instead, the authors see slot reuse can be benefited from when the level of collisions is relatively low. The target of this approach is to reduce frame delivery delays. Basically, a classification scheme is used to identify pairs of nodes that are separated by two hops as a maximum. The classification is helpful in learning the risks associated with reusing a slot by a pair of nodes. If the anticipated risks are high, no slot reuse is performed. Otherwise, slot reuse is accepted. The conducted simulations reveal that the new algorithm can achieve a substantial reduction in the frame delivery delay.

Various proposals worked on exploiting the GTS feature of 802.15.4 to enhance the support for QoS parameters. Koubâa et al. in [KOU06] introduce the Implicit GTS Allocation Mechanism (i-GAME) to tackle the drawbacks of the GTS allocation mechanism. While in 802.15.4 a GTS is allocated for a node (upon request), i-GAME allows multiple nodes to share the same GTS. This approach is conditioned on the ability of the PAN coordinator to prepare a schedule that can accommodate the needs of the sharing nodes. The operation of i-GAME is directly dependent on traffic specification, delay requirements, and the available GTS resources. With i-GAME, nodes communicate their traffic and delay requirements to the PAN coordinator; in the standard, nodes request a fixed number of GTSs. The coordinator runs an admission control algorithm that processes the information received from the nodes, and assesses the available GTSs to see whether it can prepare a schedule that reflects the nodes' requirements. Finally, the authors model i-GAME mathematically and study its performance through simulations. Compared to the standard, i-GAME shows promising results in terms of bandwidth utilization efficiency.

2.3.4 GTS-Based Approaches

Shrestha et al. in [SHR10] also deals with enhancing the GTS allocation scheme. The target of this work is to improve the performance in Wireless Body Area Sensor Networks in terms of reliability and bandwidth utilization. The proposed scheme formulates optimization problem that aims at minimizing bandwidth requirements. The authors argue that the strategy of first-come-first-serve followed by the standard to allocate GTSs wastes the bandwidth; the asymmetric traffic incoming from the different nodes may not be accommodated properly. To deal with this challenge, the authors introduce a priority measure that depends on the traffic generation rates of the nodes. Each node is required to examine its buffer and see whether the number of frames in it has crossed a certain threshold. The threshold enables nodes to learn their priorities, and the nodes specify these priorities in the GTS allocation requests they send. The coordinator receives the requests during the CAP and uses them to solve a fractional knapsack optimization problem that tries to assign GTSs based on nodes' priorities. The main advantage of this algorithm is that it imposes no modifications on 802.15.4. Mathematical modeling and simulations are provided to study the performance of this algorithm. The results demonstrate that the new algorithm, compared to the standard GTS allocation scheme, can achieve a favorable performance in terms of the average frame delivery ratio, delay, and frame discard rate.

The GTS Scheduling Algorithm (GSA) is introduced by Na et al. in [NA08]. GSA is an optimal GTS scheduling algorithm that addresses the delay requirements of time-sensitive applications (like wireless video surveillance). GSA is designed for implementation in star WSNs. In this algorithm, any node that intends to perform a time-sensitive transaction (*T*) is required to define this transaction in terms of the time-constraint and the total payload. This information will be included in the GTS request before sending it to the coordinator. The coordinator processes the received information and works on granting GTSs based on it. Basically, the coordinator follows three steps to respond to the nodes' requests. In the first step, it checks the possibility of adding a new *T* to the schedule without changing the available scheduled transactions. Secondly, if the first step has been successful, the coordinator estimates the time needed to serve *T* and the impact of that time on the number of GTSs assigned to *T* in each beacon interval. All of these analyses allow the coordinator, in the final step, to allocate the minimum number of GTSs to *T* in each beacon interval. The operation of GSA guarantees that the available GTS resources are assigned in optimal and adaptive manner that can respond to any changes that may occur in the received payloads. GSA can intelligently spread out the GTSs over several beacon intervals to have a smooth traffic flow in the network. The authors conduct simulations to study the performance of GSA under bursty, periodic, and aperiodic traffic conditions. The simulations have shown that GSA can achieve improvements over the standard GTS allocation scheme in terms of several performance metrics, like transaction abort ratio, and GTS utilization.

2.3.5 Duty Cycle-Based Approaches

Several research contributions focused on modifying the duty cycle of each node, as defined by the IEEE 802.15.4 standard, in order to improve the overall performance. The Duty Cycle Learning Algorithm (DCLA) has been introduced by De Paz Alberola and Pesch in [DEP11]. This algorithm aims at configuring each node properly such that optimal network performance is achieved under varying traffic conditions. With DCLA the need for human intervention to adapt nodes' duty cycles is eliminated. The algorithm enables each node to self-adapt its duty cycle such that its power consumption is minimized while data delivery rates are boosted. DCLA has been designed to run on coordinator nodes. In the beginning, the algorithm learns about the intensity of the traffic by collecting statistics from the nodes. After that, and using the collected statistics, the algorithm utilizes the Reinforcement Learning (RL) framework (see [SUT98]) to determine the duty cycles to be used. The RL framework interacts with the nodes frequently in order to update their duty cycles. The interactions aim at selecting the optimal duty cycles that allow the nodes to operate with the best performance. In other words, the DCLA algorithm enables each node to modify its parameters depending on the traffic conditions over the wireless medium. The functionality of DCLA allows for the support of various applications. That is, no need for human intervention to change the setting of the nodes to conform to certain application requirements. Instead, nodes will be able to self-adapt to the traffic conditions and serve different applications. The benefit of such a system is achieving conservations in times and costs of installation, operation and management; DCLA is implemented as a software that runs on the coordinator nodes with low requirements of memory and processing. The authors have examined the performance of DCLA through simulations. The collected data prove that this algorithm can achieve improvements in energy efficiency, packet delay, and success of packet delivery.

In another proposal, Li et al. in [LI11] introduce some changes to the functionality of the beacon-enabled 802.15.4 MAC in order to boost the performance. The new protocol is called the Enhanced Beacon-Enabled Mode (EBEM). This protocol targets low data rate applications. To realize this protocol, two functions are devised. The first function is the Synchronized Low Power Listening (S-LPL). This function can effectively reduce the burden of synchronization with applications of low data rates. The second function is the Periodic Wakeup (PW); activated during the inactive portion of the superframe. PW reduces end-to-end delays as well as packet loss rates. S-LPL and PW can either operate individually or in tandem. S-LPL aims mainly at conserving more power in the network. As the 802.15.4 standard uses a BO of value of 0-14, it is noticeable that this is a small range that hinders the possibility of conserving power in applications with low duty cycle. The reason behind having such small values for the BO is that large values make nodes suffer from synchronization overhead; a node will need extra time to join a PAN coordinator's cluster, which increases the node's power expenditure. Furthermore, using large values for the BO may result in synchronization problems due to the clock drift between the different nodes. These facts necessitate the use of a new approach (S-LPL) that allows the use of extended synchronization periods only for a portion of the nodes. This portion of nodes is configured to receive the beacon less frequently compared to the remaining nodes. In other words, these nodes will listen to the wireless medium for longer durations before getting the next beacon. To further avoid extra power consumptions due to these longer durations, the coordinator is configured to send Virtual Preambles (VPs) to those designated nodes. Under S-LPL, these nodes activate the Low Power Listening (LPL) mode that allows them to detect the VPs to synchronize with the coordinator. The nodes listen to the medium for duration of two VPs and sleep for a duration of one VP. This way, S-LPL helps the nodes in reducing their power usage, which reduces the consequences of using long synchronization periods. The introduction of PW, on the other hand, is to deal with delays imposed on the packets that are generated during the superframe's inactive period. PW allows nodes, with packets ready to be sent, to transmit their packets to the coordinator during the inactive period. This way, the packet delays are reduced. The coordinator itself needs not be active during the inactive period. Instead, periodic sleeps can be effective in reducing its power consumption without affecting the expected functionality. When operating both S-LPL and PW in tandem, the experienced packet delays can be comparable to the delays encountered with 802.15.4. Simulations reveal that the Enhanced Beacon-Enabled Mode is capable of outperforming 802.15.4 in terms of power loss, packet loss, and the mean duty cycle.

Gilani et al. in [GIL11] also tackle the duty cycle modification approach to improve the performance of 802.15.4 MAC. The authors have proposed an adaptive and MAC protocol that adopts a hybrid CSMA/TDMA mechanism. Their main target is to boost the performance in terms of throughput and energy consumption. The motivation behind this protocol is that CSMA-CA performs poorly under heavy traffic load conditions. To mitigate this drawback, it has been proposed to involve the TDMA concept in the CAP of the superframe. The coordinator node is responsible for dividing the slots of the CAP into CSMA-CA slots and TDMA slots. This dividing is done in a dynamic manner; it depends on both the state of the queue of each node and the rate of collisions. Learning about the state of a node's queue can be achieved by letting each packet carry reserved bits that describe that queue. The coordinator can handle the task of assigning the TDMA slots easily using the beacon frames. Furthermore, to resolve the problem of underutilizing the communication channel, that is typical in TDMA networks, the authors use a greedy algorithm to assign the TDMA slots. This is facilitated by the fact that intense traffic loads are assumed in the WSNs under consideration in [GIL11]. The benefit of having TDMA slots in the CAP is that it limits the number of nodes that contend to access the communication medium. Based on that, it is anticipated that the level of collisions will drop, and therefore, the throughput improves. Furthermore, by reducing the number of nodes that contend to access the medium during the CAP, a portion of the nodes will be in a sleep mode, and this leads to more power savings. The authors confirm the accuracy of these discussions through simulations. Of course, these gains come at the expense of increased end-to-end delays; long superframes force TDMA nodes to wait for a longer period of time before transmitting their packets. This can be mitigated by a proper configuration of the superframe duration.

2.3.6 Parameter Tuning-Based Approaches

Researchers also considered the option of intelligently setting the parameters of the 802.15.4 MAC without introducing any changes to its basic functionality. In [NEU05], Neugebauer et al. have described an algorithm called the Beacon Order Adaptation Algorithm (BOAA) that aims at setting the value of the BO based on the frequency of communications. The design of BOAA is oriented to star-based WSNs. The algorithm itself is run on the coordinator node. This node carefully monitors the communication activities of the nodes in its star network. This monitoring enables the coordinator to tune the BO properly, which adjusts the duty cycle in a power-conserving manner. BOAA works as follows. In the beginning, the coordinator uses a low duty cycle in order to hear all the incoming messages from the nodes. This means that BO is set to zero. As the coordinator monitors the communications of the nodes, it constructs a buffer matrix to store information about how frequent each node is involved in communications. The rows of this matrix correspond to superframe steps while the columns refer to the sensor nodes. By simply tracking the changes in any row, the coordinator can learn about the frequency of communications of each node. BOAA is designed to operate in cycles. The number of these cycles is set equal to the number of the buffer matrix rows (that is, the superframe steps). Since the pattern of each row resembles the nodes' duty cycles (due to the BI), the time between each two adjacent rows (steps) will depend on the BO. Based on that, the coordinator keeps modifying the value of the BO and broadcasts it to all of the nodes. Simulations of BOAA show that there is a trade-off between power conservation and transmission delays.

Another algorithm that adopts the approach of properly adjusting the parameters of the 802.15.4 standard has been proposed by Francesco et al. in [FRA11]. The authors have introduced the ADaptive Access Parameters Tuning (ADAPT) algorithm that follows a cross-layer and distributed design. The target of ADAPT is to support reliability and energy efficiency in sensor networks. The authors are motivated by the fact that reliability requirements differ from an application to another. Also, the consumption of energy is dictated by the operating conditions of a network, which are changing continuously. These facts necessitate that nodes adapt their parameters in response to the changes over the wireless medium. ADAPT uses an *adaptation module* that has the task of gathering information from the different layers of the ZigBee stack.

The adaptation module is implemented as a vertical component that has a direct access to each layer of the ZigBee stack. This design enables the adaptation module to better optimize the operation of the node. Based on that, as the application layer determines a certain reliability to be guaranteed, the adaptation module conveys this information to the MAC layer. In response, the latter adjusts its parameters (macMinBE, macMaxCSMABackoffs, and macMaxFrameRetries) to achieve that targeted reliability. ADAPT also employs control schemes that handle the factors that degrade the reliability (contention and channel errors). These control schemes adjust the parameters of the MAC protocol to maintain the reliability in a certain designated range. The performance of ADAPT has been evaluated through simulations. The collected results have shown that ADAPT achieves promising results, compared to 802.15.4, in terms of reliability of power conservation.

In [KOU08] Koubâa et al. have worked on designing the Time Division Beacon Scheduling (TDBS) mechanism that aims at resolving the problem of beacon frame collisions in cluster-tree WSNs. Beacon frames may collide when two (or more) coordinators happen to operate within the transmission range of each other, or when their transmission ranges overlap. Apparently, to avoid these beacon collisions, node synchronization should be tackled carefully. TDBS tackles the synchronization problem through the Superframe Duration Scheduling (SDS) algorithm. This algorithm accommodates at least one SD in each BI. Moreover, SDS ensures that each two consecutive SDs are distant by BI. These settings eliminate any overlapping among beacon transmissions. In addition to these settings, SDS operates along with an ingenious duty cycle management scheme that guarantees the fair distribution of bandwidth resources. The latter scheme utilizes an optimization formulation that controls the allocation of bandwidth resources to the coordinators. The allocation of resources takes into consideration the traffic needs of each coordinator. The constraints of the optimization formulation describe the relation between the duty cycles of the adjacent coordinators. The formulation is made flexible such that its objective function can be modified easily to reflect the interest in optimizing different metrics. The performance of the TDBS is evaluated through experimentation on a test-bed.

MeshMAC is another mechanism, proposed by Muthukumaran et al. in [MUT09] that aims at intelligently scheduling the beacon transmissions in peer-to-peer WSNs. The central assumption in MeshMAC is that the entire network operates using the same standard defined BI and SD. The schedule assigned to each node requires that its 2-hop neighbors have the sum of their duty cycles a maximum of one. MeshMAC further requires that one active SD be reserved for broadcasts. In networks operating MeshMAC each node should be aware of the time slots being used by all of its 2-hop neighbors. This awareness allows a node to occupy the first idle time slot it finds. Once the node selects that time slot, it should broadcast this selection to its neighbors. Beside broadcast transmissions, MeshMAC also supports unicast transmissions. This type of transmissions takes place during the active period of a targeted destination node. MeshMAC is of a distributed nature; nodes schedule their transmissions based on information, about neighboring nodes, that is collected locally. That is, no collaboration from a coordinator node is needed for the sensor nodes to operate properly. Performance evaluation has shown that MeshMAC can outperform 802.15.4 MAC in terms of energy efficiency and scalability.

Changes to the superframe CAP has been introduced by Wang et al. in [WAN11]. The authors conduct a thorough study of the superframe CAP in the 802.15.4 MAC protocol. The study has observed that the standard MAC has a poor performance in terms of the throughput. The authors have reasoned that to the standard length of the backoff slot (aUnitBackoffPeriod); the length of the slot is big and limits the number of instants at which nodes are allowed to compete for medium access. This fact leads to increases in the probability of collisions. Based on this discussion, the authors have proposed to divide the standard backoff slot evenly into sub-slots. The target of this dividing is to increase the number of the instants at which nodes can compete to access the wireless medium. This reflects in improving the nodes' chances in accessing the medium. It is worth mentioning that, by proposing this new dividing of the backoff slot, no modifications are imposed on the original CSMA-CA algorithm. The conducted simulations have shown that the throughput can improve significantly with the new changes to the backoff slot. To further improve the overall performance, the authors have devised a novel backoff algorithm that uses a new time unit as well as a relatively wider interval from which a backoff period is selected. The new time unit benefits from the aforementioned findings when the backoff off has been divided into sub-slots. The bigger interval, on the other hand, is used to reduce the probability of having a large number of nodes using the same backoff period (which is the main reason behind increasing the probability of collisions). Simulations have shown that the new backoff algorithm can outperform the standard CSMA-CA in terms of the throughput, delay, and packet delivery rate.

Mori et al. in [MOR11] focus on cluster-based WSNs and study the transmission performance of the standard MAC in these networks. It has been observed that postponing packet transmissions to the next CAP, when the time remaining in the current CAP is not enough to complete the transmissions, leads to channel access congestion. This congestion causes collisions to increase, at the beginning of the next CAP, and this affects the transmission performance in the network. To deal with this situation, the authors in [MOR11] have proposed that nodes, during the next CAP, use backoff periods that differ in their starting points within the CAP. The value of BE is maintained at macMinBE. To guarantee having different starting points, the concept of the Distribution Window (DW) is introduced. The DW can be of either a constant length (CDW) or an adaptive length (ADW). The length of the ADW depends on the intensity of the traffic load; light traffic loads make the ADW narrower while heavy traffic loads expand it. Information about the intensity of the traffic load over the communication medium is provided by the cluster coordinator. Simulations have been conducted to test the efficiency of the new backoff algorithm. The collected data has shown a direct relation between the throughput and the lengths of the DW. In other words, by properly setting the length of DW one can achieve throughput values that are comparable to the standard MAC. The simulations have also shown that the new algorithm, with the proper setting of DW, can significantly reduce packet delays in the presence of heavy traffic loads.

2.3.7 Backoff-Based Approaches

In [ZHU11a] Zhu et al. describe the Linear Increase Backoff (LIB) mechanism to support time-sensitive applications. The main target of LIB is to boost the performance in terms of end-to-end delay. LIB introduces changes into the standard CSMA-CA mechanism. In this new mechanism, the backoff counters no more increase exponentially after finding any of the CCAs busy. Instead, the increase is proposed to be of a linear nature. The problem with the exponential approach is that it may cause some nodes to wait for unnecessary periods of time before initiating their CCAs. The negative side of that is that nodes with short backoff periods will have stronger chances in accessing the communications medium. By following the linear approach in increasing the backoff counter, it is anticipated that the nodes will share the medium in a fairer manner. LIB also imposes that nodes that cannot complete the transmission of their packets within the current superframe should drop their packets (not postpone the transmissions to the next superframe). Furthermore, LIB requires that nodes switch to the *sleep* mode (instead of the *receiving-idle* mode) during backoff, after a transmission, or after discarding a packet due to transmission failure or medium access failure. Besides, LIB eliminates the use of ACK packets as the huge number of sensor nodes is sufficient to guarantee the delivery of data. The performance of LIB has been assessed through simulations. The results have shown that LIB has a strong capability of reducing the end-to-end delay. Furthermore, LIB has managed to improve the throughput and energy conservation in large networks that operate under heavy traffic loads.

An adaptive backoff algorithm has been introduced by Jing et al. in [JIN11] to improve the performance in terms the throughput in 802.15.4-based WSNs. Nonlinear programming (NLP) has been used to search for the optimal value of the throughput. The new adaptive mechanism dictates that if the node finds the wireless medium busy during a CCA, it selects its next backoff period in a probabilistic manner that takes the network size into consideration. As a result, backoff periods increase (decrease) as the size of the network increases (decreases), which controls the level of contention to access the medium properly. Simulation results have shown that the new adaptive algorithm has the ability to outperform the standard backoff algorithm in terms of the throughput and the rate of successful packet transmissions.

2.3.8 QoS-Based Approaches

Some researchers enhance the support of QoS in 802.15.4 by focusing on the problem of traffic differentiation and prioritization. In [SHI11] Shi et al. focus on the issue of monitoring real-time events and the importance of identifying alarm signals from normal ones. The authors have introduced improvements to the CSMA-CA mechanism based on Weighted-Fair-Queue (FQ-CSMA/CA) algorithm. The latter algorithm enhances the standard CSMA-CA by enabling it to balance the transmission quality of the different signals according to their criticality. In particular, FQ-CSMA/CA works on reducing transmission delays for alarm (important) signals while preserving the transmission quality of normal signals. To achieve that, FQ-CSMA/CA relies on the concept of weighted-fair-queuing classify the packets based on their priority. This concept enables each node to assign each packet in its queues a certain weight. A packet weight is defined as the percentage of bandwidth allocated for a certain queue, and it is clearly specified in the priority-label field of the packet. The weight is highly useful for a node to better organize its queues' packets based on their priority (importance). Once the classification of the packets is done, a packet scheduler is used to pick the high-priority packets and service them first. With FQ-CSMA/CA, the data is classified into five categories. These categories are, sensor collected data, control command, ACK packet, system setting, and alarm signals. The alarm signals are assumed to be of the highest priority (largest weight), while the lowest priority is given to ACK packets. The other categories of data are assumed to be of equal priorities. The goal behind classifying data in this manner is to ensure that all data, except for ACK packets, are treated equally and fairly, provided that the alarm signals are transmitted first once they occur. The conduced simulations can clearly show that the FQ-CSMA/CA algorithm achieves a promising performance, when compared to the standard CSMA-CA algorithm, in terms of the average queue delay and the frame success probability.

In [SEV10] Severino et al. tackle the problem of traffic differentiation during the CAP. The authors propose an approach that offers differentiated services to time-sensitive packets. The approach works by properly adjusting the 802.5.4 parameters, namely, macMinBE, macMaxBE, and the contention window, to better process high-priority packets. The approach deals with command frames (like GTS requests and alarm reports) as high-priority messages while ordinary data frames are viewed as low-priority messages. This means that each node will be locally using its own configuration of the parameters based on the type of traffic it communicates. The local configurations take into consideration that the backoff periods when sending high-priority packets are made shorter. Furthermore, priority queuing is adopted to guarantee that high-priority packets are placed in the queue such that they are chosen first to be sent out. Experimental evaluation has been conducted to evaluate the performance of the new approach in [SEV10]. The collected data has shown that this approach has high capability of properly queuing high-priority packets.

Kim and Kang in [KIM10] deal with service differentiation in WSNs that communicate non-saturated packet traffics. Basically, the authors have introduced two approaches to achieve their goals: the Contention Window Differentiation (CWD) mechanism and the Backoff Exponent Differentiation (BED) mechanism. Both of these mechanisms classify nodes into variant priority classes. Priory of a packet is dictated by the importance of that packet. Based on that, nodes that are hungry for bandwidth and active in sending emergency messages should be assigned high priority. CWD achieves service differentiation by properly controlling the size of the contention window. Basically, CWD guarantees that nodes with high-priority packets use shorter contention windows. That is, the nodes that have low-priority packets are assigned longer waiting times in order to treat the more important packets urgently. On the other hand, BED achieves service differentiation through controlling the binary exponent for the different nodes. BED guarantees that nodes with different priorities are using the appropriate binary exponents that give priority of service for high-priority packets. Both BED and CWD depend in their functionality on the Backoff Counter Selection (BCS) scheme. With BCS, whenever the medium is found busy during a CCA, the next backoff period is selected from a range that is smaller than the default range of the 802.15.4 standard. This smaller range increases the likelihood of selecting distinct backoff periods for the different nodes, which decreases the probability of collisions and helps in preserving nodes’ priorities. The importance of using BCS is that it can effectively accelerate the differentiation of services among nodes. The performance of both CWD and BED has been evaluated through simulations. Simulations have shown that both mechanisms are effective in properly prioritizing the packets based on their importance. The two mechanisms, however, show different behaviours depending on the performance parameters under consideration. As a result, the authors provide a set of recommendations on when to use either of the mechanisms once the performance goals are defined.

Another approach to recognize the diverse service needs of the sensor nodes is by defining a secondary beacon to be used by the coordinator node. In [JAR10] Jardosh et al. describe an explicit priority scheme that classifies the nodes into two classes, namely, critical nodes and normal nodes. The former nodes are distinguished by the need to transmit highly important messages (high-priority nodes) while the latter nodes need to communicate routine messages (low-priority nodes). The critical nodes inform the coordinator about the importance of their packets through a secondary beacon frame. Once the coordinator is notified, it permits only the critical nodes to contend for the wireless medium during the CAP. The coordinator enforces this permission through priority information that it includes in the periodic beacon that it already sends to all nodes. The authors have studied the performance of their proposal through simulations that concentrated on three parameters, namely, packet delivery ratio, prioritized packet delivery and per bit energy consumption. Compared to the standard mechanism, the new mechanism has shown superior performance.

Ndih et al. in [NDI09] criticize the fact that 802.15.4 specifies no prioritization among the nodes; all nodes of a network use the same standard-defined parameters when attempting to access the medium. To improve this design, the authors present a Markov-based mathematical model for the CAP that assumes that nodes with diverse priorities are allowed to adopt distinct sets of access parameters. Similar to the previous studies, priorities are categorized into class 1 (high-priority) and class 2 (low-priority). The authors have provided a node-state Markov-chain model for each priority class. Also, a channel-state Markov-chain model has been developed. Service differentiation among the two classes of nodes is accomplished through allocating high-priority (low-priority) nodes a contention window of 1 (2). This configuration guarantees that high-priority nodes can access the medium with better chances (provided that the standard backoff parameters are kept at their default values). The overall performance has been studied through simulations. The collected data has shown promising results in terms of the probability of channel access, throughput, and latency.

2.4 Conclusion

In this chapter we have provided an overview of the available proposals that target improving the MAC protocol of the IEEE 802.15.4 standard. We have seen that this is an active area of research and various contributions have been made to achieve better performance in a diverse set of applications. An important observation we make is that a limited attention has been given to the exploitation of the number of periods of the Clear Channel Assessment (CCA) feature. This is evident by the small number of proposals we have found that target enhancing IEEE 802.15.4 MAC through introducing changes to the way CCA is conducted. This is a driving force for us to explore new opportunities to strengthen the functionality of IEEE 802.15.4 MAC.

Chapter 3

Analytical Approach for Modeling Variable CCA MAC

3.1 Introduction

The survey in Chapter 2 provides insight into research directions that could address the problem of enhancing the 802.15.4 MAC protocol using the CCA feature. To devise a new, effective MAC protocol that overcomes the shortcomings of the standard MAC, we propose exploiting the CCA feature of the standard MAC more efficiently. In this chapter, we study the impact of increasing the number of CCAs performed by a node beyond the standard value of 2, and based on this we develop a mathematical model that describes the functionality of the outcome MAC protocol. This chapter focuses on providing detailed derivations of the mathematical expressions that better describe the system. We derive the theoretical expressions of the probability of collision, throughput, channel idle time, channel collision time, energy consumption, transmission delay and reliability.

3.2 Variable CCA MAC Protocol

In Figure 3-1 we show a flowchart that describes the functionality of the MAC protocol with variable CCAs.

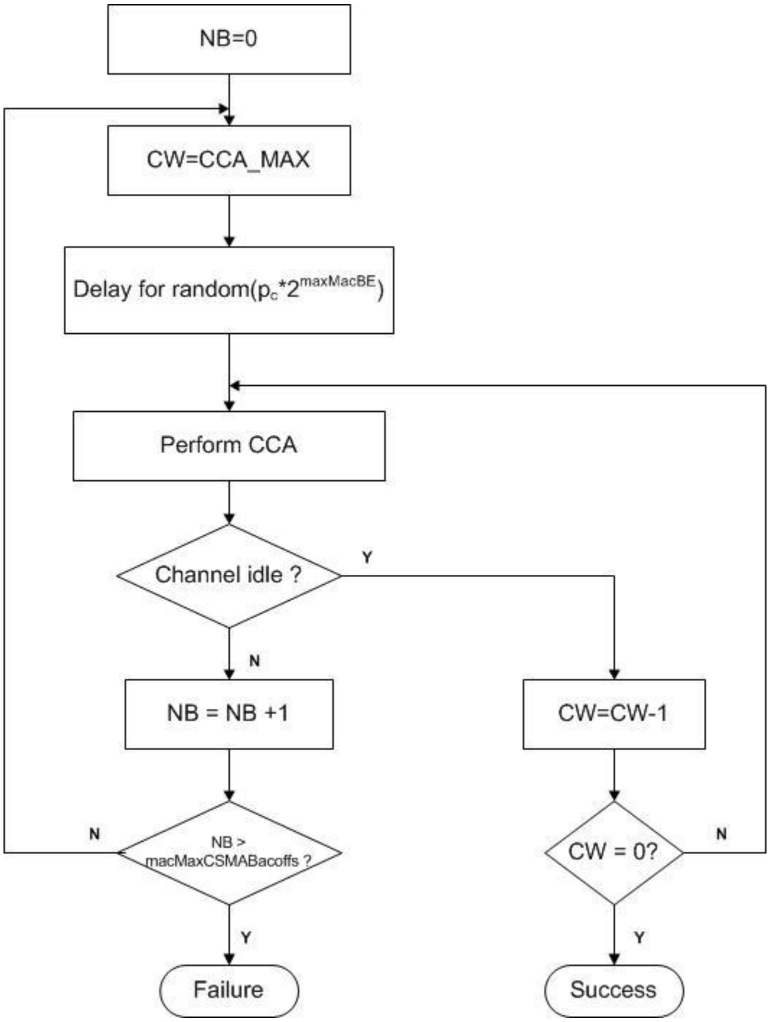


Figure 3-1**:** Algorithm of the Variable CCA MAC Protocol.

The main differences with the standard CSMA-CA are the duration of the contention window and the value of the backoff window. The contention window is initialized to a maximum value (CCA_MAX) that is greater or equal to 2. Similar to the standard, the transmitting node initializes the number of backoff stages (NB) to zero, and the contention window (CW) to fixed value to a chosen value CCA_MAX. Then, the node backs off for a duration that is chosen randomly from the range [0, p_c*$2^{macMaxBE}$] time units, where p_c is the locally computed probability of collision, and macMaxBE is a parameter defined as the maximum value taken by the variable backoff exponent BE. Once the backoff counter expires, the node conducts a number of clear channel assessments. This is intended to check whether the wireless medium is free of any ongoing transmissions or not. Each successful CCA, which finds the medium idle, is followed by another CCA. The node starts transmitting the frame only if finds the medium idle during all CCAs. In contrast, if any of the CCAs reveals that the medium is busy, the node increases the values of NB by one. The standard specifies that the maximum value of NB is *macMaxCSMABackoffs* (with default value of 4). If NB exceeds *macMaxCSMABackoffs*, the node will discard the frame. Whenever the node fails to access the medium, it dismisses the frame and re-starts the CSMA-CA backoff process again by generating a new number of complete backoff periods.

In Figure 3-2 we show the Markov chain that describes the new hybrid MAC protocol. Each state in this chain is described by a pair *(i, j)*. The *i* index takes the values *0* for backoff states, *1* for CCA states, *2* for successful transmission states, or 3 for collision states. On the other hand, the *j* index for the states *(0, j)* refers to the backoff duration and takes values from $[1, W-1]$. For the CCA states, *j* is 1 for CCA1, 2 for CCA2…etc. For the states *(2, j)*, $j \in [0, L\text{-}1]$ and it corresponds to the successful transmission states of a frame. Finally, $j \in [0, L\text{-}1]$ for the states *(3, j)* and it refers to the frame collision states. The probability of finding the medium busy at CCA_i is α_i.

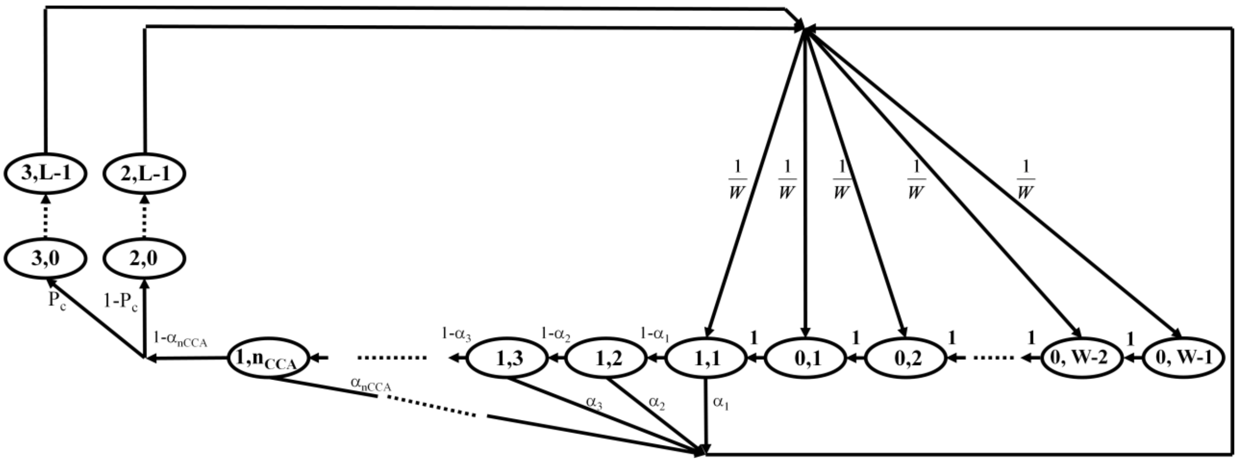


Figure 3-2: Markov chain for the Variable CCA MAC protocol.

Each state in the model represents a timeslot. The packet length is fixed for all frames. We note that the Markov model is represented using a one backoff stage only. This is due to the fact that we use a backoff window that is equal to $p_c * 2^{macMaxB\ E}$. The probability of collision can be estimated by each sensor node by the following expression:

$$p_c = \frac{n_c}{n_s + n_c}$$

Where n_c and n_s are the number of collided and successful frames experienced by each node, respectively.

Assuming the fairness of the proposed protocol, which will be proved in Chapter 4, the local probability of collision should be the same for all nodes. Under saturated traffic, all nodes will reach a steady state where the probability of collision reaches a constant and equal value for all sensor nodes.

The state transition probabilities of the Markov chain are listed with their explanation in the following equations:

$$P((0, j-1)|(0, j)) = 1 \qquad for \quad 1 < j \leq W-1 \tag{3.1}$$

$$P((1,j)|(1,1)) = (1-\alpha_1)\dots(1-\alpha_{j-1}) \tag{3.2}$$

$$P((2,j)|(1,1)) = (1-\alpha_1)(1-\alpha_2)\dots\left(1-\alpha_{n_{CCA}}\right)(1-P_c) \quad for \quad 0 \le j \le L-1 \tag{3.3}$$

$$P((3,j)|(1,1)) = (1-\alpha_1)(1-\alpha_2)\dots\left(1-\alpha_{n_{CCA}}\right)P_c \qquad for \quad 0 \le j \le L-1 \tag{3.4}$$

Equation (3.1) reflects how the backoff counter decrements before starting the CCAs. Equation (3.2) is the probability of conducting j^{th} CCA given that the medium was sensed idle during the previous CCA periods. Equation (3.3) describes the probability of successfully transmitting a frame while equation (3.4) is the probability of experiencing a frame collision. In what follows, we derive the mathematical expressions for the stationary distribution.

The theoretical probability of collision can be calculated as follows:

The complement of the probability of collision $(1-p_c)$ is equal to the probability that no node (other than the transmitting node) is in state $S_{1,1}$. This can be written as:

$$1 - p_c = P\left(\eta\left(S_{1,1}\right) = 0\right)$$

Where $P\left(\eta\left(S_{1,1}\right) = 0\right)$ is the probability that the number of nodes, other than the current node, at state $S_{1,1}$ is equal to zero.

We can easily see that $P\left(\eta\left(S_{1,1}\right) = 0\right)$ *is equal to* $(1-\tau)^{N-1}$ since all remaining nodes are in a state other than $S_{1,1}$.

We refer by τ to the probability of being at state $S_{1,1}$ which is equal to $b_{1,1}$.

Therefore, the theoretical expression of the probability of collision is given by equation (3.5):

$$p_c = 1 - (1-\tau)^{N-1} \tag{3.5}$$

- **Calculation of α_1:**

Assume that *A* refers to the event that the channel has been sensed busy for the *first* time (CCA$_1$). Since the two events of being at states $S_{2,*}$and states $S_{3,*}$are mutually exclusive, then we have:

$$\alpha_1 = P(A) = P\big(\eta(S_{2,*}) > 0\big) + P\big(\eta(S_{3,*}) > 0\big)$$

This can be expressed as the probability of having at least one node in a transmission state (which can lead to either a successful frame delivery or a collision).

Since the probability of being at state $S_{2,i}$ is equal for every i; and the probability of being at state $S_{3,j}$ is equal for every j, we can write:

$$P\big(\eta(S_{2,*}) > 0\big) = L * P\big(\eta(S_{2,1}) > 0\big)$$

Similarly,

$$P\big(\eta(S_{3,*}) > 0\big) = L * P\big(\eta(S_{3,1}) > 0\big)$$

Therefore,

$$\alpha_1 = L * [P\big(\eta(S_{2,1}) > 0\big) + P\big(\eta(S_{3,1}) > 0\big)]$$

$$P\big(\eta(S_{2,1}) > 0\big) = (1 - p_c) * P\big(\eta(S_{1,1}) > 0\big) * (1 - \alpha_1)(1 - \alpha_2) \ldots (1 - \alpha_{n_{CCA}})$$

$$P\big(\eta(S_{3,1}) > 0\big) = p_c * P\big(\eta(S_{1,1}) > 0\big) * (1 - \alpha_1) * (1 - \alpha_2) \ldots (1 - \alpha_{n_{CCA}})$$

$$\alpha_1 = L * [(1 - p_c) + p_c] * P\big(\eta(S_{1,1}) > 0\big) * \prod_{i=1}^{n_{CCA}} (1 - \alpha_i)$$

Knowing that: $P\big(\eta(S_{1,1}) > 0\big) = 1 - P\big(\eta(S_{1,1}) = 0\big)$

And $P(\eta(S_{1,1}) = 0) = (1-\tau)^{N-1}$; since all remaining nodes are in a state other than $S_{1,1}$.

Therefore, we can write:

$$\therefore\ \boldsymbol{\alpha_1 = \left(1-(1-\tau)^{N-1}\right) L \prod_{i=1}^{n_{CCA}} (1-\alpha_i)} \tag{3.6}$$

- **Calculation of $\boldsymbol{\alpha_2}$:**

Assume that B refers to the event that the channel has been sensed for the *second* time (CCA$_2$) and found busy; given that (CCA$_1$) was found idle. Then we have:

$$\alpha_2 = P(B) = P(CCA_2 = busy \mid CCA_1 = idle) = \frac{P(CCA_2 = busy \cap CCA_1 = idle)}{P(CCA_1 = idle)}$$

$$= \frac{P(\eta(S_{\{2,3\},1}) > 0)}{1-\alpha_1}$$

$$= \frac{P(\eta(S_{1,1}) > 0) * \prod_{i=1}^{n_{CCA}} (1-\alpha_i)}{1-\alpha_1}$$

$$= \frac{(1-(1-\tau)^{N-1}) * \prod_{i=1}^{n_{CCA}} (1-\alpha_i)}{1-\alpha_1}$$

$$\therefore\ \boldsymbol{\alpha_2 = \left(1-(1-\tau)^{N-1}\right) \prod_{i=2}^{n_{CCA}} (1-\alpha_i)} \tag{3.7}$$

Calculation of $\boldsymbol{\alpha_3}$:

Assume that C refers to the event that the channel has been sensed for the *third* time (CCA3) and found busy; given that CCA2 was found idle (which implicitly means that CCA1 was idle too). Then we have:

$$\alpha_3 = P(C) = P(CCA_3\ is\ busy \mid CCA_2 = idle\ and\ CCA_1 = idle)$$

$$= \frac{P(CCA_3 = busy \cap (CCA_2 = idle\ and\ CCA_1 = idle))}{P(CCA_2 = idle\ and\ CCA_1 = idle)}$$

$$= \frac{P\big(\eta\big(S_{\{2,3\},1}\big) > 0\big)}{(1-\alpha_1)(1-\alpha_2)}$$

$$= \frac{P\big(\eta\big(S_{1,1}\big) > 0\big) * \prod_{i=1}^{n_{CCA}}(1-\alpha_i)}{(1-\alpha_1)(1-\alpha_2)}$$

$$= \frac{\big(1-(1-\tau)^{N-1}\big) * \prod_{i=1}^{n_{CCA}}(1-\alpha_i)}{(1-\alpha_1)*(1-\alpha_2)}$$

$$\therefore\ \boldsymbol{\alpha_3 = \big(1-(1-\tau)^{N-1}\big) * \prod_{i=3}^{n_{CCA}}(1-\alpha_i)} \tag{3.8}$$

- **Calculation of $\boldsymbol{\alpha_i}$:**

Based on the analysis above, we can express the probability of having the channel sensed busy at any CCA_i, $i \in [2, n_{CCA}]$, as follows:

$$\alpha_i = P\left(CCA_i = busy \mid \bigcap_{j=1}^{i-1} CCA_j = idle\right)$$

$$\alpha_i = \frac{P\big(CCA_i = busy \cap (\bigcap_{j<i} CCA_j = idle)\big)}{P(\bigcap_{j<i} CCA_j = idle)}$$

$$\alpha_i = \frac{P\big(\eta\big(S_{\{2,3\},1}\big) > 0\big)}{\prod_{j=1}^{i-1}(1-\alpha_j)}$$

$$\alpha_j = \frac{P\big(\eta\big(S_{1,1}\big) > 0\big) * \prod_{i=1}^{n_{CCA}}(1-\alpha_i)}{\prod_{j=1}^{i-1}(1-\alpha_j)}$$

$$\therefore\ \boldsymbol{\alpha_i} = \boldsymbol{\big(1-(1-\tau)^{N-1}\big) \prod_{j=i}^{n_{CCA}}\big(1-\alpha_j\big)} \qquad \boldsymbol{i} > 1 \tag{3.9}$$

Note that in case we have only two CCA states (i.e., the standard case), $\alpha_1$and $\alpha_2$reduce to the following expressions:

$$\boldsymbol{\alpha_1 = L\big(1-(1-\tau)^{N-1}\big)(1-\alpha_1)(1-\alpha_2)}$$

$$\boldsymbol{\alpha_2 = (1 - \alpha_2)(1 - (1 - \tau)^{N-1})}$$

Which are the same equations derived for the standard IEEE 802.15.4 MAC (see [POL08], [PAR10], and [ZHU11b]).

- **Calculation of Stationary Probabilities:**

By the normalization condition, we have the following formula:

$$b_{1,1} + \sum_{j=2}^{n_{CCA}} b_{1,j} + \sum_{j=1}^{W-1} b_{0,j} + \sum_{j=1}^{L} b_{2,j} + \sum_{j=1}^{L} b_{3,j} = 1 \tag{3.10}$$

The stationary probability of being at any backoff state, $j \epsilon [1, W - 1]$, is expressed as follows:

$$b_{0,j-1} = b_{0,j} + \frac{b_{1,1}}{W}$$

Which is equivalent to:

$$b_{0,j} = b_{0,j-1} - \frac{b_{1,1}}{W}$$

Since the stationary probability is in the form of an arithmetic progression, the general term can be written as:

$$b_{0,j} = \frac{W-j}{W} * b_{1,1} \tag{3.11}$$

Therefore,

$$\sum_{j=1}^{W-1} b_{0,j} = \sum_{j=1}^{W-1} \frac{W-j}{W} * b_{1,1} = \frac{W-1}{2} * b_{1,1}$$

Next, the probability of being in frame transmission is:

$$\sum_{j=1}^{L} b_{2,j} = L * b_{2,1}$$

Since $b_{2,1} = b_{1,1} * (1 - p_c) * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right)$, we can write:

$$\sum_{j=1}^{L} b_{2,j} = L * b_{1,1} * (1 - p_c) * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right)$$

Similarly, we can write

$$\sum_{j=1}^{L} b_{3,j} = L * b_{1,1} * p_c * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right)$$

Last, the probability of being at any of the channel sensing states $CCA_j, j \epsilon [2, n_{CCA}]$, is :

$$b_{1,j} = b_{1,1} * \prod_{k=1}^{j-1} (1 - \alpha_k)$$

$$\sum_{j=2}^{n_{CCA}} b_{1,j} = b_{1,1} * \sum_{j=2}^{n_{CCA}} \prod_{k=1}^{j-1} (1 - \alpha_k)$$

Then, based on equation (3.10), we can write the following expression:

$$b_{1,1} * \left[1 + \frac{W-1}{2} + L * (1 - p_c) * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right) + L * p_c * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right) + \sum_{j=2}^{n_{CCA}} \prod_{k=1}^{j-1} (1 - \alpha_k)\right] = 1$$

We can then express $b_{1,1}$ as:

$$b_{1,1} = \tau = \frac{2}{W + 1 + 2 * L * \prod_{j=1}^{n_{CCA}} \left(1 - \alpha_j\right) + 2 * \sum_{j=2}^{n_{CCA}} \prod_{k=1}^{j-1} (1 - \alpha_k)} \qquad (3.12)$$

Equations (3.6), (3.9), and (3.12), which form a system of $(n_{cca} + 1)$ nonlinear equations, can be solved to find the operating point of the network.

3.3 Performance Metrics

In this section we derive the mathematical expressions of the performance metrics used to assess the efficiency of the protocol. These metrics are the channel utilization, channel idle time, channel collision time, power consumption including the energy wasted during collisions, reliability, and transmission delay.

3.3.1 Channel Utilization

The Channel Utilization (*U*) metric is used to measure how efficient the MAC with variable CCAs is in utilizing the wireless channel in communicating useful data. In other words, *U* informs us of how the channel is being used in delivering meaningful data percentage-wise. By referring to the Markov chain in Figure 3-2, we can see that the probability that a certain node achieves a successful transmission of a frame of length L is $L * \tau * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i)$. By taking into consideration that we have *N* nodes in the network, we can write the following mathematical expression for *U*:

$$U = N * L * \tau * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) \tag{3.13}$$

3.3.2 Channel Idle Time

The Channel Idle Time (T_{idle}) metric refers to the percentage of time the wireless channel is free of any activity (transmissions or collisions). T_{idle} is a measure of the percentage of time all nodes are in backoff or CCA states. This means that T_{idle} should be kept low as it indicates that the wireless channel is idle. Previously, we have defined α_1 to be the probability of having the wireless channel busy during *CCA1*. From this definition we can see that T_{idle} is the complement of α_1. Therefore, we can express T_{idle} mathematically as follows:

$$T_{idle} = 1 - \alpha_1 \tag{3.14}$$

3.3.3 Channel Collision Time

The Channel Collision Time ($T_{collision}$) parameter is a measure of the proportion of time the wireless channel is busy with collisions. Similar to T_{idle}, $T_{collision}$ should also be kept low because it refers to the percentage of time the communication channel is wasted with useless activities. From the definitions of U and T_{idle}, we can easily conclude that $T_{collision}$ can be expressed as follows:

$$T_{collision} = 1 - U - T_{idle}$$

Therefore,

$$T_{collision} = \alpha_1 - N * L * \tau * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) \quad (3.15)$$

3.3.4 Power Consumption

It is highly important to assess the performance of any new protocol oriented for WSNs in terms of its power requirements. Power consumption should be minimized in these networks in order to prolong their lifetime. The total power consumed in the network can be expressed as follows:

$$E_{total} = E_{idle} + E_{CCA} + E_{tx} + E_{rx}$$

E_{idle} is the total power consumed during the backoff states:

$$E_{idle} = P_{idle} \sum_{i=1}^{W-1} b_{0,i}$$

According to equation (3.11):

$$b_{0,i} = \frac{W-i}{W} * b_{1,1}$$

Therefore,

$$E_{idle} = P_{idle} \frac{W-1}{2} b_{1,1} \tag{3.16}$$

E_{CCA} is the total power consumed during the CCA states:

$$E_{CCA} = P_{CCA}\left(b_{1,1} + \sum_{i=2}^{n_{CCA}} b_{1,i}\right) = P_{CCA}\left(1 + \sum_{i=2}^{n_{CCA}} \prod_{j=1}^{i-1}\left(1-\alpha_j\right)\right) b_{1,1} \tag{3.17}$$

E_{tx} is the total power consumed during successful and collision frame transmission:

$$E_{tx} = P_{tx}\left(\sum_{i=1}^{L} b_{2,i} + \sum_{i=1}^{L} b_{3,i}\right) = P_{tx} * L * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1} \tag{3.18}$$

E_{rx} is the total power consumed during frame reception:

$$E_{rx} = P_{rx} \sum_{i=1}^{L} b_{2,i} = P_{rx} * L * (1-p_c) * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1}$$

Since $1 - p_c = (1-\tau)^{N-1}$, the expression of the energy consumed during frame reception can be written as:

$$E_{rx} = P_{rx} * L * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1} \tag{3.19}$$

Therefore, the total energy consumption can be expressed as:

$$E_{total} = P_{idle} \frac{W-1}{2} b_{1,1} + P_{CCA}\left(1 + \sum_{i=2}^{n_{CCA}} \prod_{j=1}^{i-1}\left(1-\alpha_j\right)\right) b_{1,1} + P_{tx} * L * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1} + P_{rx} * L * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1} \tag{3.20}$$

Finally, E_c is the total power consumed during frame collisions:

$$E_c = P_{tx} \sum_{i=1}^{L} b_{3,i} = P_{tx} * L * p_c * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1}$$

$$E_c = P_{tx} * L * \left(1-(1-\tau)^{N-1}\right) * \prod_{i=1}^{n_{CCA}} (1-\alpha_i) * b_{1,1} \tag{3.21}$$

where, P_{idle}, P_{CCA}, P_{tx}, and P_{rx} refer to the average power consumed during backoff, channel sensing (CCA), frame transmission, and frame reception, respectively.

3.3.5 Reliability

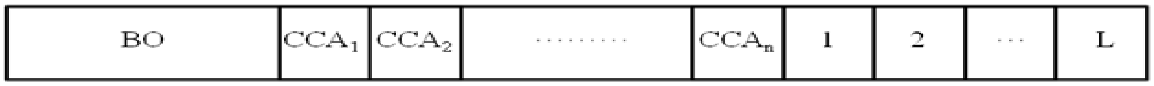
We define *reliability* (R) as the probability of receiving a frame successfully. The reliability can be then expressed as:

$$R = \frac{n_s}{n_s + n_{af} + n_{cf}} \tag{3.22}$$

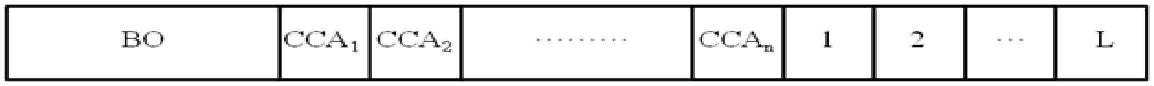
Where n_s, n_{af}, n_{cf} are the number of successfully transmitted frames, the number of access failures, and the number of collision failures, respectively. In the following subsection we detail our approach to find the values of n_s, n_{af}, and n_{cf}.

3.3.5.1 Definition of a transmission cycle

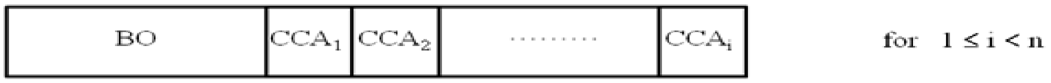
A transmission cycle is the duration of an attempt to transmit a frame. Upon completion, a cycle can have one of the following three states: Success, Collision, or Busy. The Success state refers to the case where a frame is successfully transmitted. The Collision state occurs when the frame collides with one or more frames. Finally, the Busy state refers to the condition when the channel is sensed busy in one of the CCAs states.



Success Cycle



Collision Cycle



Busy Cycle

Figure 3-3: Illustration of the Success cycle, the Collision cycle, and the Busy cycle.

These states are illustrated in Figure 3-3. In Figure 3-4 we show a finite-state machine (FSM) that illustrates the three states of a node as it attempts to transmit a frame.

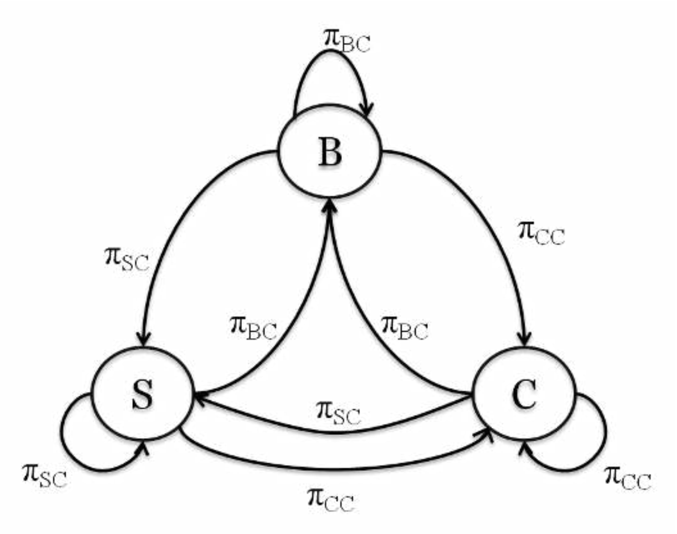


Figure 3-4: Node's states during a complete cycle.

Let's denote by X the random variable of being at states S, B, or C. Then, according to Figure 3-4, we can easily see that the stationary probability of being at the state S is equal to π_{SC}, the stationary probability of being at state B is equal to π_{BC}, and the stationary probability of being at state C is equal to π_{CC} . Therefore, we can write:

$$\Pr(X = S) = \pi_{SC}$$

$$\Pr(X = B) = \pi_{BC}$$

$$\Pr(X = C) = \pi_{CC}$$

The Busy and Collision states are broken down in Figure 3-5 and Figure 3-6, respectively. In Figure 3-5, we illustrate the fact that a node may encounter a maximum of macMaxCSMABackoffs (*m* in the figure) backoff stages. On the other hand, in Figure 3-6 we depict the fact that the node can retry the transmission of a frame for a maximum macMaxFrameRetries (*n* in the figure) times before discarding a frame (note that π_{CC_i} denotes the probabilities of experiencing a collision after finding the channel busy in the previous state while being at the i^{th} retry stage).

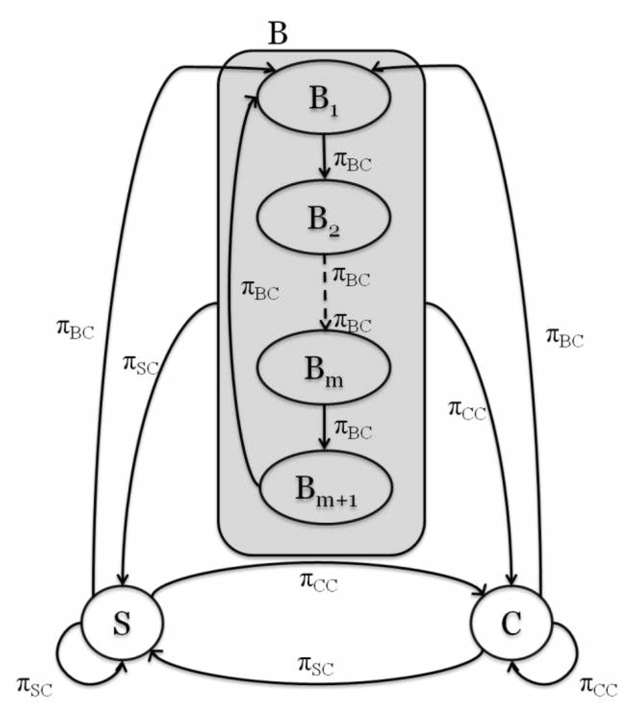


Figure 3-5: Breakdown of the busy state into multiple backoff states.

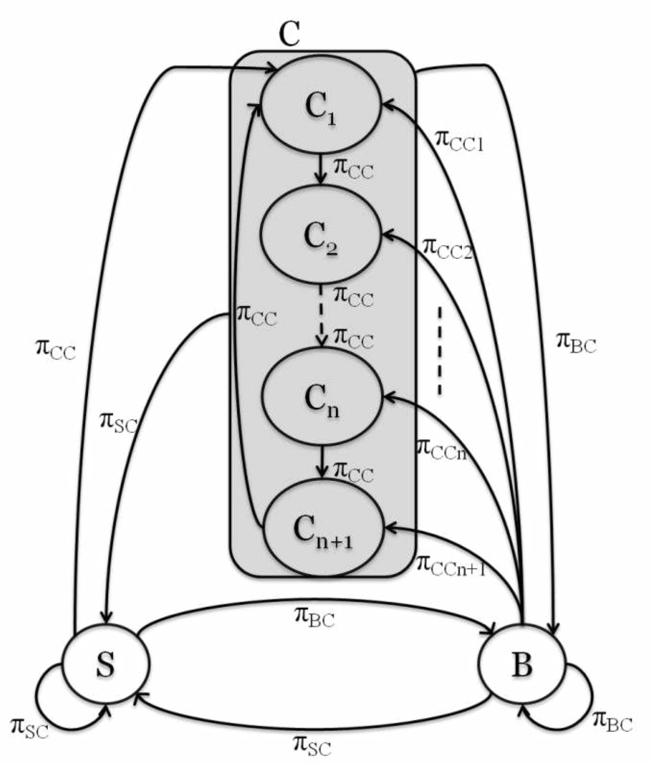


Figure 3-6: Breakdown of the collision state into multiple retries states.

By dividing the numerator and denominator of the reliability equation (3.22) by the total number of cycles, the equation of the reliability can be rewritten as follows:

$$R = \frac{\pi_{sc}}{\pi_{sc} + Pr(X = B_{m+1}) + Pr(X = C_{n+1})}$$

Where $\Pr(X = B_{m+1})$ is the stationary probability of being at backoff stage m+1, and $\Pr(X = C_{n+1})$ is the stationary probability of being at retrial stage n+1. These two probabilities can be found using Figure 3-5 and Figure 3-6, respectively.

Therefore, the reliability is expressed as:

$$R = \frac{\pi_{sc}}{\pi_{sc} + \pi_{BC_{m+1}} + \pi_{CC_{n+1}}}$$

Finding the final expression of the reliability R requires that we find the probability that a node backs off m+1 times or suffers from collisions for n+1 times. Let's denote by π_1 the stationary distribution $\begin{bmatrix}\pi_S & \pi_{B_1} & \pi_{B_2} & \cdots & \pi_{B_{m-1}} & \pi_{B_m} & \pi_{B_{m+1}} & \pi_C\end{bmatrix}$, where, π_{B_i} is the probability of being at the *i*th backoff stage. Moreover, denote by π_2 the stationary distribution $\begin{bmatrix}\pi_S & \pi_{C_1} & \pi_{C_2} & \cdots & \pi_{C_n} & \pi_{C_{n+1}} & \pi_B\end{bmatrix}$, where π_{C_i} is the probability of suffering from the *i*th collision. Based on that, we can write:

$$P_1 \times \pi_1 = \pi_1$$

and

$$P_2 \times \pi_2 = \pi_2$$

where, P_1 and P_2 are the transition matrices of the FSMs in Figure 3-5 and Figure 3-6, respectively.

By expanding the last two equations, we find:

$$\begin{array}{c|cccccccccc|}
 & S & B_1 & B_2 & . & . & . & B_m & B_{m+1} & C \\
\hline
S & \pi_{SC} & \pi_{SC} & \pi_{SC} & . & . & . & \pi_{SC} & \pi_{SC} & \pi_{SC} \\
B_1 & \pi_{BC} & 0 & 0 & . & . & . & 0 & \pi_{BC} & \pi_{BC} \\
B_2 & 0 & \pi_{BC} & 0 & . & . & . & 0 & 0 & 0 \\
. & . & . & \pi_{BC} & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . \\
B_m & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\
B_{m+1} & 0 & 0 & 0 & . & . & . & \pi_{BC} & 0 & 0 \\
C & \pi_{CC} & \pi_{CC} & \pi_{CC} & . & . & . & \pi_{CC} & \pi_{CC} & \pi_{CC}
\end{array}
\times \begin{pmatrix} \pi_{SC} \\ \pi_{BC1} \\ \pi_{BC2} \\ \vdots \\ \pi_{BCm} \\ \pi_{BCm+1} \\ \pi_{CC} \end{pmatrix} = \begin{pmatrix} \pi_{SC} \\ \pi_{BC1} \\ \pi_{BC2} \\ \vdots \\ \pi_{BCm} \\ \pi_{BCm+1} \\ \pi_{CC} \end{pmatrix}$$

(3.23)

$$\begin{array}{c|ccccccccc|} & S & C_1 & C_2 & . & . & . & C_n & C_{n+1} & B \\ \hline S & \pi_{SC} & \pi_{SC} & \pi_{SC} & . & . & . & \pi_{SC} & \pi_{SC} & \pi_{SC} \\ C_1 & \pi_{CC} & 0 & 0 & . & . & . & 0 & \pi_{CC} & \pi_{CC1} \\ C_2 & 0 & \pi_{CC} & 0 & . & . & . & 0 & 0 & \pi_{CC2} \\ . & . & . & \pi_{CC} & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ C_n & 0 & 0 & 0 & . & . & . & 0 & 0 & \pi_{CCn} \\ C_{n+1} & 0 & 0 & 0 & . & . & . & \pi_{CC} & 0 & \pi_{CCn+1} \\ B & \pi_{BC} & \pi_{BC} & \pi_{BC} & . & . & . & \pi_{BC} & \pi_{BC} & \pi_{BC} \end{array} \times \begin{pmatrix} \pi_{SC} \\ \pi_{CC1} \\ \pi_{CC2} \\ \vdots \\ \pi_{CCn} \\ \pi_{CCn+1} \\ \pi_{BC} \end{pmatrix} = \begin{pmatrix} \pi_{SC} \\ \pi_{CC1} \\ \pi_{CC2} \\ \vdots \\ \pi_{CCn} \\ \pi_{CCn+1} \\ \pi_{BC} \end{pmatrix}$$

(3.24)

From equation (3.23) we can write:

$$\pi_{BC_{i+1}} = \pi_{BC} * \pi_{BC_i}$$

Therefore,

$$\pi_{BC_{m+1}} = (\pi_{BC})^m \pi_{BC_1} \tag{3.25}$$

From equation (3.23), we can express the probability π_{BC_1} as

$$\pi_{BC_1} = \pi_{BC}\pi_{SC} + \pi_{BC}\pi_{BC_{m+1}} + \pi_{BC}\pi_{CC} \tag{3.26}$$

By solving both equations (3.25) and (3.26), we end up with the following expression:

$$\pi_{BC_{m+1}} = \frac{(\pi_{BC})^{m+1}(1-\pi_{BC})}{(1-\pi_{BC})^{m+1}} \tag{3.27}$$

Based on equation (3.24), we can write the following two equations:

$$\pi_{CC}\pi_{SC} + \pi_{CC}\pi_{CC_{n+1}} + \pi_{CC_1}\pi_{BC} = \pi_{CC_1} \tag{3.28}$$

$$\pi_{CC}\pi_{CC_n} + \pi_{CC_{n+1}}\pi_{BC} = \pi_{CC_{n+1}} \tag{3.29}$$

From Equation (3.29), we can easily see that $\pi_{CC_{n+1}}$can be expressed in terms of π_{C_1} as follows:

$$\pi_{CC_{n+1}} = \left(\frac{\pi_{CC}}{1-\pi_{BC}}\right)^n \pi_{CC_1} \tag{3.30}$$

By solving both Equations (3.28) and (3.30) for $\pi_{CC_{n+1}}$, we end up with the following expression:

$$\pi_{CC_{n+1}} = \pi_{SC} \frac{\pi_{CC}{}^{n+1}}{(1-\pi_{BC})^{n+1} - \pi_{CC}{}^{n+1}} \tag{3.31}$$

Finally, the reliability *R* can be expressed as follows:

$$R = \frac{1}{1+\frac{(1-\pi_{BC})\pi_{BC}{}^{m+1}}{(1-\pi_{BC}{}^{m+1})\pi_{SC}} + \frac{\pi_{CC}{}^{n+1}}{(1-\pi_{BC})^{n+1} - \pi_{CC}{}^{n+1}}} \tag{3.32}$$

With

$$\pi_{BC} = \sum_{i=1}^{n_{CCA}} \alpha_i \prod_{j=1}^{i-1} (1-\alpha_j)$$

$$\pi_{CC} = p_c \prod_{i=1}^{n_{CCA}} (1-\alpha_i)$$

$$\pi_{SC} = (1-p_{c)} \prod_{i=1}^{n_{CCA}} (1-\alpha_i)$$

3.3.6 Delay

The delay is the mean time needed to successfully transmit a frame. It's important to note that sensor data must reach the base station within an acceptable deadline. Time delay is a very important QoS measurement since it influences performance and stability of the control system.

If we consider n_{SC} as the number of successfully transmitted frames over a period of time T_{total} (n_{SC} is equivalent to the number of successful cycles), we can write:

$$T_{total} = n_{SC} D$$

Where D is the mean time to transmit a frame.

We can notice that the total time is composed of three times: time consumed by successful cycles T_{SC}, time consumed by collision cycles T_{CC}, and the time consumed by busy cycles T_{BC}. Therefore,

$$T_{total} = T_{SC} + T_{CC} + T_{BC}$$

If we define $\overline{T}_{SC}, \overline{T}_{CC}, \overline{T}_{BC}$ as the mean time of a successful, collision, and busy cycles, respectively, we can write:

$$T_{SC} = n_{SC} * \overline{T}_{SC}$$

$$T_{CC} = n_{CC} * \overline{T}_{CC}$$

$$T_{BC} = n_{BC} * \overline{T}_{BC}$$

Where, n_{SC}, n_{CC}, n_{BC} are the number of successful, collision, busy cycles, respectively. Therefore,

$$n_{SC} * D = n_{SC} * \overline{T}_{SC} + n_{CC} * \overline{T}_{CC} + n_{BC} * \overline{T}_{BC}$$

$$D = \overline{T}_{SC} + \frac{n_{CC}}{n_{SC}} * \overline{T}_{CC} + \frac{n_{BC}}{n_{SC}} * \overline{T}_{BC}$$

$$D = \overline{T}_{SC} + \frac{\pi_{CC}}{\pi_{SC}} * \overline{T}_{CC} + \frac{\pi_{BC}}{\pi_{SC}} * \overline{T}_{BC}$$

From Figure 3-3, we can see that:

$$\overline{T}_{SC} = \overline{T}_{BO} + n_{cca} + L$$

$$\overline{T}_{CC} = \overline{T}_{BO} + n_{cca} + L$$

$$\overline{T}_{BC} = \overline{T}_{BO} + \overline{T}_{CCA}$$

- **Calculation of $\overline{T}_{CCA}$**

$$\overline{T}_{CCA} = \sum_{i=1}^{n_{CCA}} i * P(T_{CCA} = i)$$

$$\overline{T}_{CCA} = \sum_{i=1}^{n_{CCA}} i * \alpha_i^{'} * \prod_{j=1}^{i-1} (1 - \alpha_j^{'})$$

Where $\alpha_i^{'}$ is the probability of having CCAi busy knowing that the cycle state is busy and that the channel state in the previous CCAs is idle. Therefore, we can write:

$$\alpha_i^{'} = P(CCA_i = Busy \mid (Cycle = Busy \cap \bigcap_{j=1}^{i-1} (CCA_j = Idle)))$$

The event (Cycle = Busy) is the union of events where the channel is sensed busy. Therefore:

$$(Cycle = Busy) = (\bigcup_{j=1}^{n_{CCA}} (CCA_j = Busy))$$

Therefore,

$$\alpha_i^{'} = P(CCA_i = Busy \mid (\bigcup_{j=1}^{n_{CCA}} (CCA_j = Busy) \cap \bigcap_{j=1}^{i-1} (CCA_j = Idle)))$$

By expanding the conditional probability, we can express $\alpha_i^{'}$ as:

$$\alpha_i^{'} = \frac{P(CCA_i = Busy \cap \left(\bigcup_{j=1}^{n_{CCA}} CCA_j = Busy\right) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}{P(\bigcup_{j=1}^{n_{CCA}} CCA_j = Busy \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}$$

Which can be simplified to:

$$\alpha_i^{'} = \frac{P((CCA_i = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}{P(\bigcup_{j=1}^{n_{CCA}} (CCA_j = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}$$

Since the probabilities of finding CCA busy are independent, we can write:

$$\alpha_i^{'} = \frac{P((CCA_i = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}{\sum_{k=1}^{n_{CCA}} P(CCA_k = Busy \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}$$

Since it’s given that $\bigcap_{j=1}^{i-1}(CCA_j = Idle)$, then

$P((CCA_k = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle)) = 0$ for k<i

Therefore,

$$\alpha_i^{'} = \frac{P((CCA_i = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}{\sum_{k=i}^{n_{CCA}} P((CCA_k = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))}$$

We can see that:

$$P((CCA_k = Busy) \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle)) = \alpha_k \prod_{r=i}^{k-1}(1 - \alpha_r)$$

Also, knowing that $\alpha_i = P(CCA_i = Busy \mid \bigcap_{j=1}^{i-1}(CCA_j = Idle))$, the expression of $\alpha_i^{'}$ can be written as:

$$\alpha_i^{'} = \frac{\alpha_i}{\sum_{k=i}^{n_{CCA}} \alpha_k \prod_{r=i}^{k-1}(1-\alpha_r)}$$

Finally, we can express $\overline{T}_{CCA}$ as:

$$\overline{T}_{CCA} = \sum_{i=1}^{n_{CCA}} i * \frac{\alpha_i}{\sum_{k=i}^{n_{CCA}} \alpha_k \prod_{r=i}^{k-1}(1-\alpha_r)} * \prod_{j=1}^{i-1}(1 - \frac{\alpha_j}{\sum_{k=j}^{n_{CCA}} \alpha_k \prod_{r=j}^{k-1}(1-\alpha_r)})$$

- **Backoff Time $\overline{T}_{BO}$**

The mean time $\overline{T}_{BO}$ spent in a backoff period is expressed as:

$$\overline{T}_{BO} = p_c * W_{max} = (1-(1-\tau)^{N-1}) * W_{max}$$

Last, we can express the delay as:

$$D = \left(1 + \frac{\pi_{CC}}{\pi_{SC}} + \frac{\pi_{BC}}{\pi_{SC}}\right) * \overline{T}_{BO} + \frac{\pi_{BC}}{\pi_{SC}} * \overline{T}_{CCA} + (1 + \frac{\pi_{CC}}{\pi_{SC}}) * (n_{cca} + L)$$

This can be simplified to:

$$D = \frac{1}{\pi_{SC}} * \overline{T}_{BO} + \frac{\pi_{BC}}{\pi_{SC}} * \overline{T}_{CCA} + (1 + \frac{\pi_{CC}}{\pi_{SC}}) * (n_{cca} + L) \tag{3.33}$$

3.4 Conclusion

In this chapter, we incorporated variable CCAs that change the contention window from 2 to n_{CCA} into the functionality of the standard MAC, and developed a Markov-based mathematical model to describe the outcome MAC protocol. We also derived mathematical expressions for several performance metrics essential to studying the different aspects of this MAC.

Chapter 4

Model Validation of Variable CCA MAC

4.1 Introduction

In Chapter 3, we provide a detailed mathematical model that describes the functionality of the variable CCA MAC protocol. In this chapter, we present data collected from the simulation and analytical models, and focus on validating the mathematical model to determine if it can predict the performance accurately. The validation is performed against a WSN MAC simulator that mimics the functionality of the IEEE 802.15.4 MAC with the implementation of the variable CCAs. Our main goal in this chapter is to prove the accuracy of our mathematical model; the effect of adding extra CCAs on the functionality of the 802.15.4 MAC will be discussed in Chapter 5.

4.2 WSN MAC Simulator

We have considered developing our own simulator in order to examine the performance of our proposed solution. Developing an in-house simulator has several advantages. First, we can focus our efforts on building a flexible system that can be easily extended to simulate several algorithms and protocols (this is essential to hold comparisons with our solution proposal, draw more accurate conclusions, and envision promising future steps). Second, any general-purpose simulator will need some modifications to make it abide by the details of our proposal. This means that we will be involved in an exhaustive and tedious debugging process that is normal in any software development project. By building our own simulator we can reduce the time of debugging significantly as we will have control over all the details of the simulator. Moreover, we will have the flexibility of specifying relaxed, yet accurate, assumptions that reduces the complexity of the overall design and avoids implementing any other layer of protocols that are supposed, in a real-world scenario, to interact with the MAC sub-layer. The latter fact is highly important because the available general-purpose simulators provide a complete implementation of the entire stack of communication protocols. These implementations are usually complex and support functionality (like routing and modulation) that are out of the scope of our study. Any modifications we may introduce to these implementations may lead to problems in this extra functionality, which complicates the process of development.

We have developed a discrete-time simulation framework to simulate the 802.15.4 MAC sub-layer and study its performance under the proposed changes in the CCA feature. The simulation results are then used to validate the analytical model we have developed in Chapter 3.

The developed MAC simulator assumes WSNs with a star topology and can be extended to support a peer-to-peer topology. The simulator has been designed to abide by several requirements and accomplish a diverse set of objectives. These requirements and objectives are explained in what follows.

The simulator is restricted to implement the functionality of the CSMA-CA algorithm as adopted by the IEEE 802.15.4 standard. That is, we only focus on the specifications and functionality that will be affected by our proposed solution. Also, the simulator should be accurate and capable of producing results that are already published in the literature for other algorithms and protocols. This is an essential requirement that allow us to have confidence in our generated results. Each sensor node is modeled as independent unit (that is, a *structure* in the C language context). The simulator should be flexible enough to allow for the collection of performance parameters while sensor nodes are communicating their packets. The latter should not be on the account of scalability; WSNs may be constituted by thousands of sensor nodes, and therefore, the simulator should be able to generate performance metrics for relatively big networks. Finally, in Figure 4-1 we show the flowchart of our MAC simulator. Based on this flowchart we provide an overview of the operation of the MAC simulator.

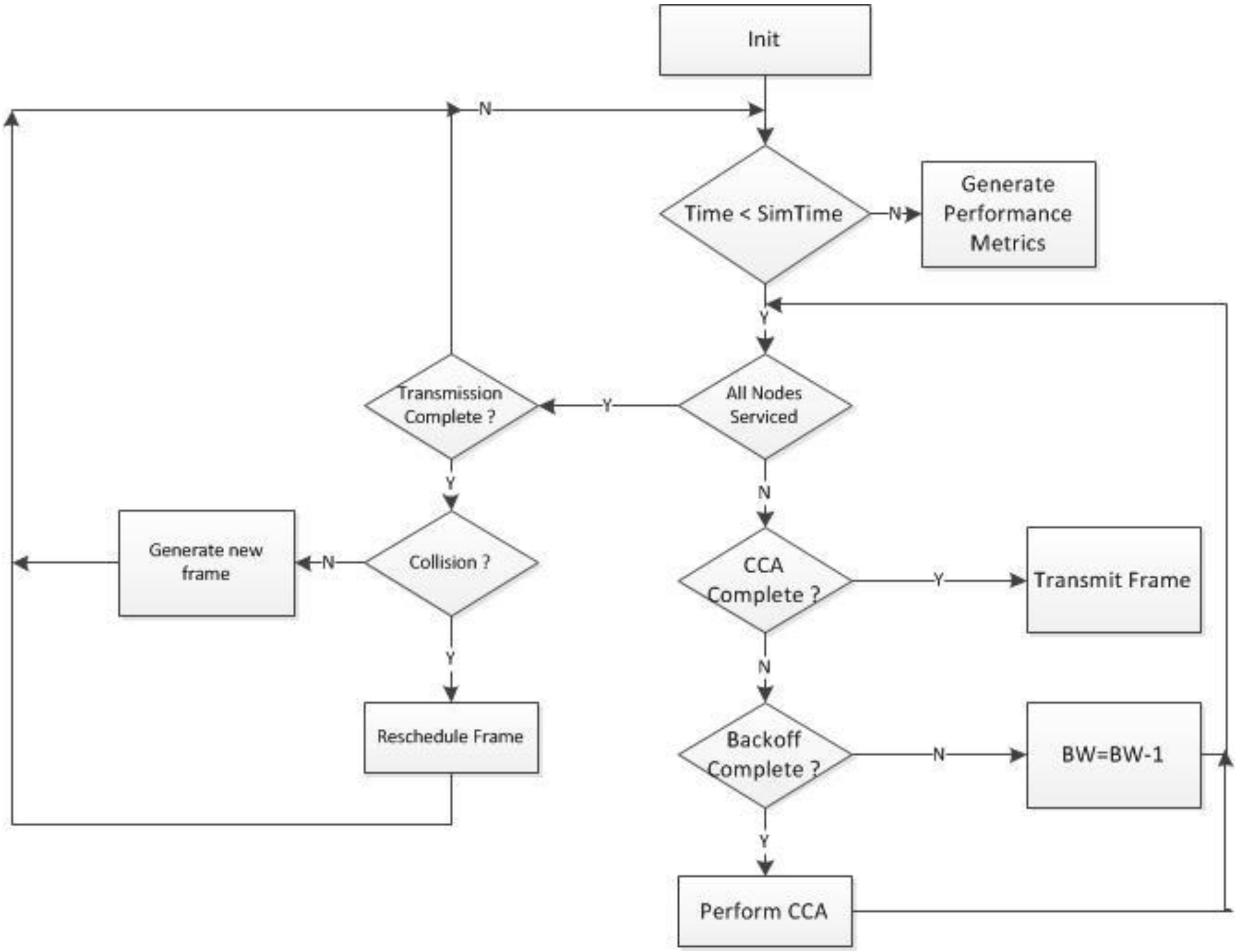


Figure 4-1: Flowchart of the MAC Simulator.

On startup, the simulator sets the total simulation time *SimTime*. This time is used as an upper bound against which the discrete time of the simulation, called *Time*, is iteratively compared. Once *SimTime* is reached, the simulator terminates and the performance metrics are generated for analysis. During the course of simulation, all nodes are given the opportunity of generating a packet and attempting to transmit it. Each transmission attempt is done during a single iteration. That is, during a single iteration, a node generates a packet, backs off, conducts CCAs, and, when idle, transmits the packet. At the end of the iteration, the simulator checks whether all nodes have been serviced. If so, it checks whether the transmission was successful or a collision has occurred. If the transmission was successful, the simulator proceeds to the next iteration (as long as *SimTime* is not crossed). Otherwise, the packet is rescheduled before proceeding to the next iteration (again, provided that *SimTime* is not reached).

The simulation parameters are summarized in Table 4-1 (see [POL08]). The superframe is assumed to have no CFP or inactive period. The traffic conditions in the network under study are assumed to be saturated. The traffic is further assumed to be a non-acknowledged one. The confidence interval (CI) is considered to be at 95%; in collecting our data, five simulation runs are conducted. We note that confidence interval of the simulation results is very low. Therefore, we have omitted it from the graphs.

Table 4-1: Simulation Parameters

Average Power Consumed (mW)	Rx	40
	Tx	30
	CCA	40
	Sleep	0.8
Durations	1 timeslot	0.32 ms (80 bits)
	Frame Length (L)	14 timeslots
	ACK Frame Length (L_{ACK})	2 timeslots
	Simulation Time	320 s
IEEE 802.15.4 Settings	*macMaxCSMABackoffs*	5
	macMaxFrameRetries	4
	macMinBE	3
	macMaxBE	5

In the following sections we will validate our Markov-based model against the performance metrics: probability of collision, channel utilization, channel idle time, channel collision time, delay, total energy consumption, energy wasted in collisions, and reliability. The accuracy of the developed Markov-based model is assessed through the *coefficient of variation of the root-mean-square deviation RSMD* (*CV-RMSD*). This coefficient measures the differences between the curves generated by the mathematical model and the simulations. The coefficient is expressed as follows:

$$CV(RSMD) = \frac{\sqrt{\frac{\sum_{i=1}^{n}(V_{th} - V_s)^2}{n_s}}}{\bar{V}}$$

where, V_{th} is the value generated by the mathematical model, V_s is the value collected from the simulations, $\bar{V}$ is the mean of the considered sample values, and n_s is the size of the sample values under study. Lower values of $CV(RMSD)$indicate that the mathematical model has a high capability of predicting the real behavior of the network.

4.3 Probability of Collision

The probability of collision is expressed as:

$$p_c = 1 - (1 - \tau)^{N-1}$$

In Figures 4-3, 4-4, and 4-5 we show the expected behavior of p_c as a function of the number of nodes N. We show the theoretical behavior and the simulation-based one given that nodes are configured with 2 CCAs (Figure 4-2), 5 CCAs (Figure 4-3), and 8 CCAs (Figure 4-4). We can clearly see that we have an exact match between the two curves in all three figures. The accuracy of the theoretical model is confirmed by noticing the low values of CV-RMSD in the figures; 1.24% in Figure 4-2, 1.36% in Figure 4-3, and 2.81% in Figure 4-4.

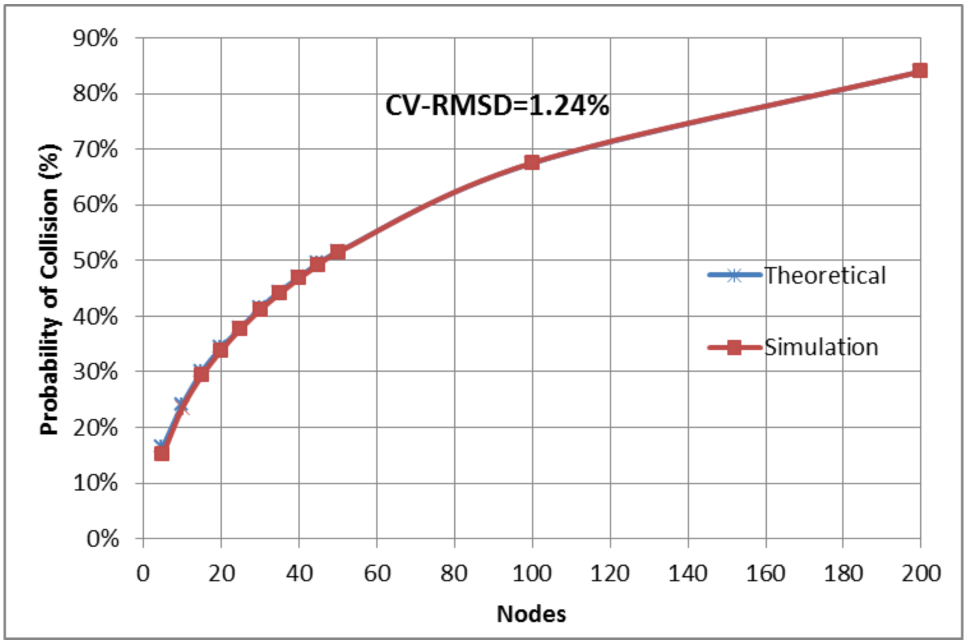


Figure 4-2: Theoretical and simulation-based performance in terms of the probability of collision with 2 CCAs.

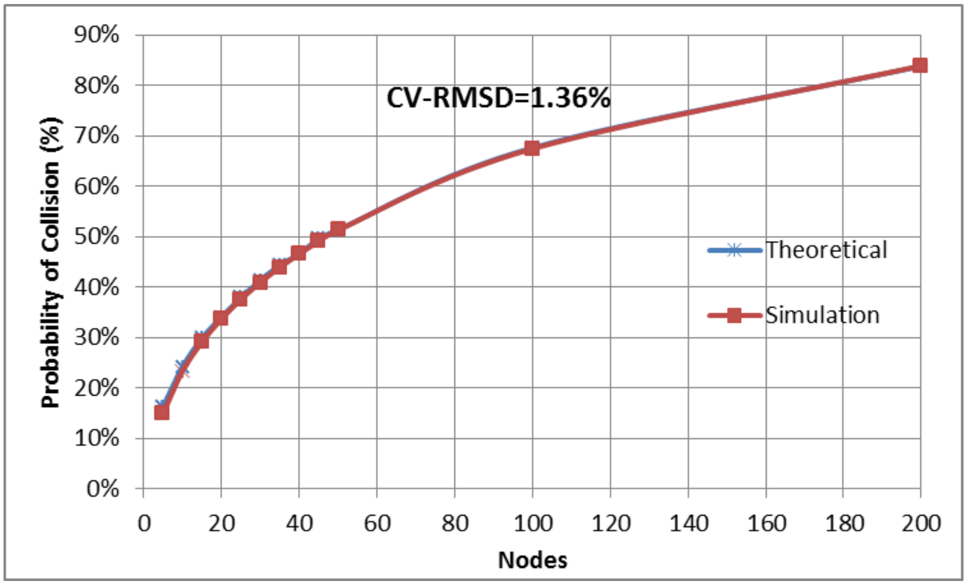


Figure 4-3: Theoretical and simulation-based performance in terms of the probability of collision with 5 CCAs.

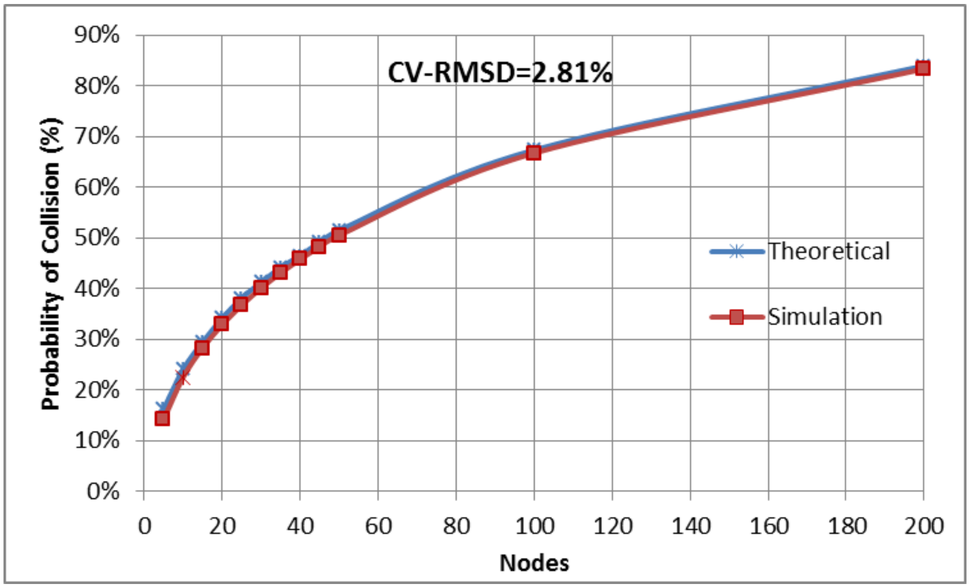


Figure 4-4: Theoretical and simulation-based performance in terms of the probability of collision with 8 CCAs.

4.4 Channel Utilization

According to equation (3.13), the theoretical expression of the channel utilization is given by the following formula:

$$U = N * L * \tau * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i)$$

In Figure 4-5, Figure 4-6, and Figure 4-7 we show both theoretical and simulation-based performance in terms of the channel utilization parameter. The number of CCAs considered is 2, 5, and 8 for the graphs in Figure 4-5, Figure 4-6, and Figure 4-7, respectively. The graphs clearly show that our Markov-based model is able to accurately predict the behavior of the network under the variable CCAs approach. The accuracy, however, is stronger for N values larger than 10 nodes. The accuracy of our model is confirmed through the values of the CV-RMSD; 5.62% with 2 CCAs, 5.92% with 5 CCAs, and 6.14% with 8 CCAs.

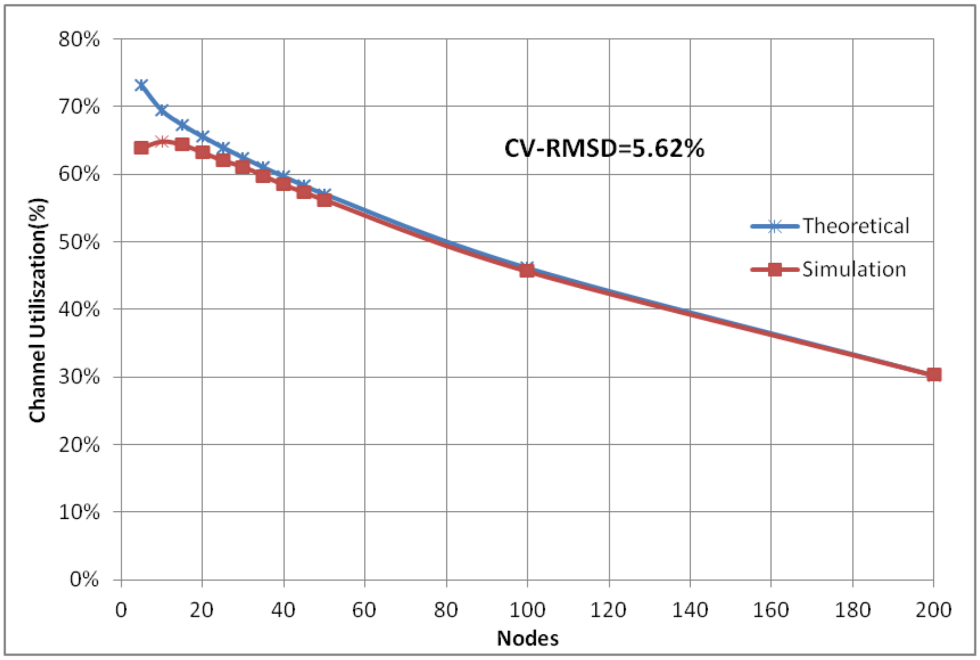


Figure 4-5: Theoretical and simulation-based performance in terms of the channel utilization with 2 CCAs.

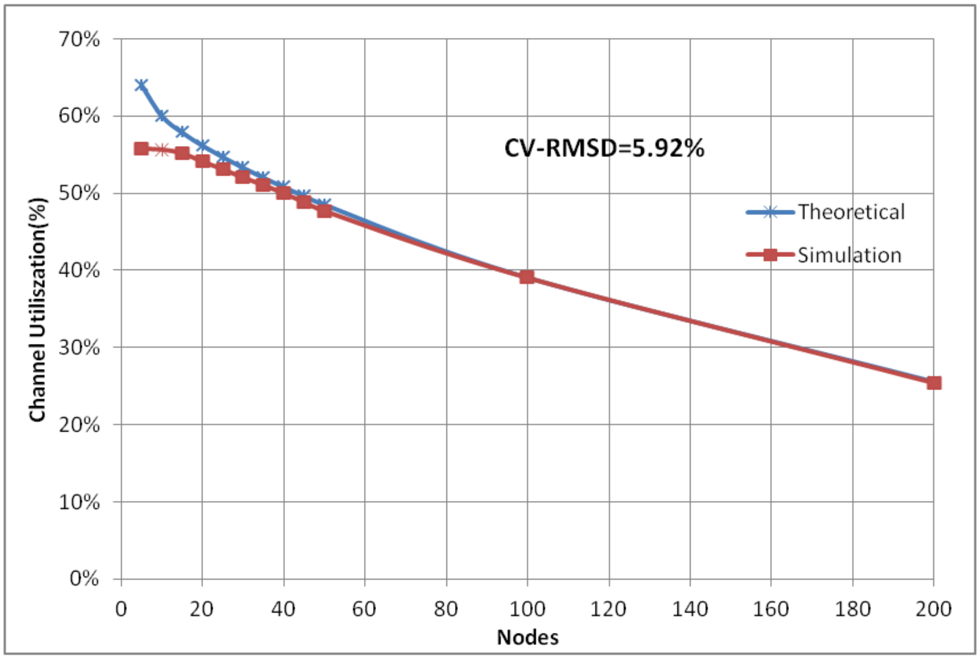


Figure 4-6: Theoretical and simulation-based performance in terms of the channel utilization with 5 CCAs.

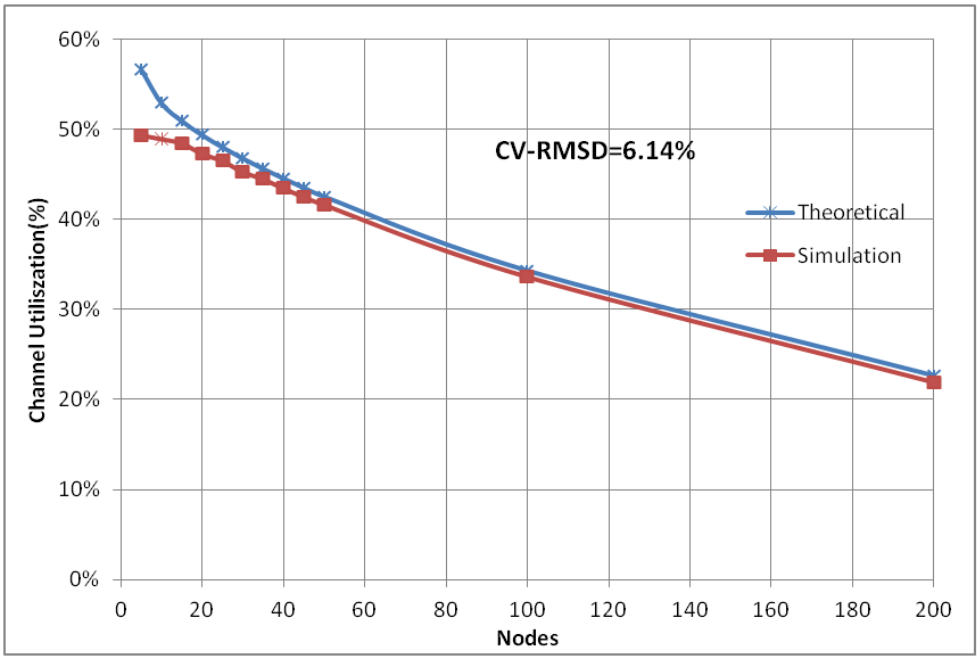


Figure 4-7: Theoretical and simulation-based performance in terms of the channel utilization with 8 CCAs.

4.5 Channel Idle Time

According to equation (3.14), the theoretical expression of the channel idle time is:

$$T_{idle} = 1 - \alpha_1$$

Figure 4-8, Figure 4-9, and Figure 4-10, show the theoretical and simulation-based performance in terms of channel idle time. Again, 2, 5, and 8 CCAs are considered in Figure 4-8, Figure 4-9, and Figure 4-10, respectively. The match in performance is quite clear on each of these graphs. The CV-RMSD values further confirm the accuracy of the mathematical model (7.75% with 2 CCAs, 4.13% with 5 CCAs, and 3.15% with 8 CCAs). As mentioned earlier, the accuracy is better for N values beyond 10 nodes.

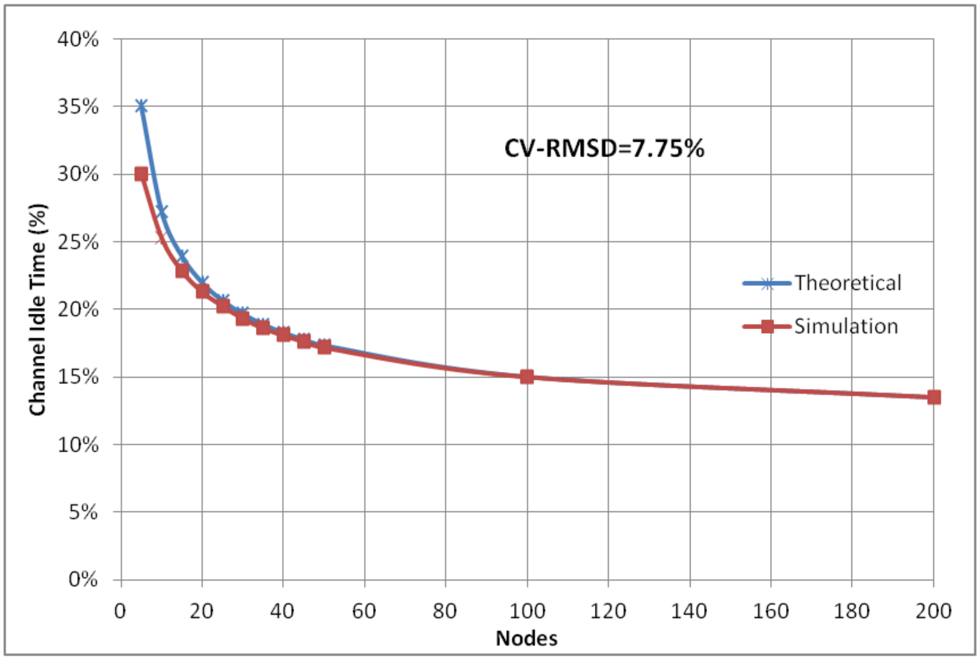


Figure 4-8: Theoretical and simulation-based performance in terms of the channel idle time with 2 CCAs.

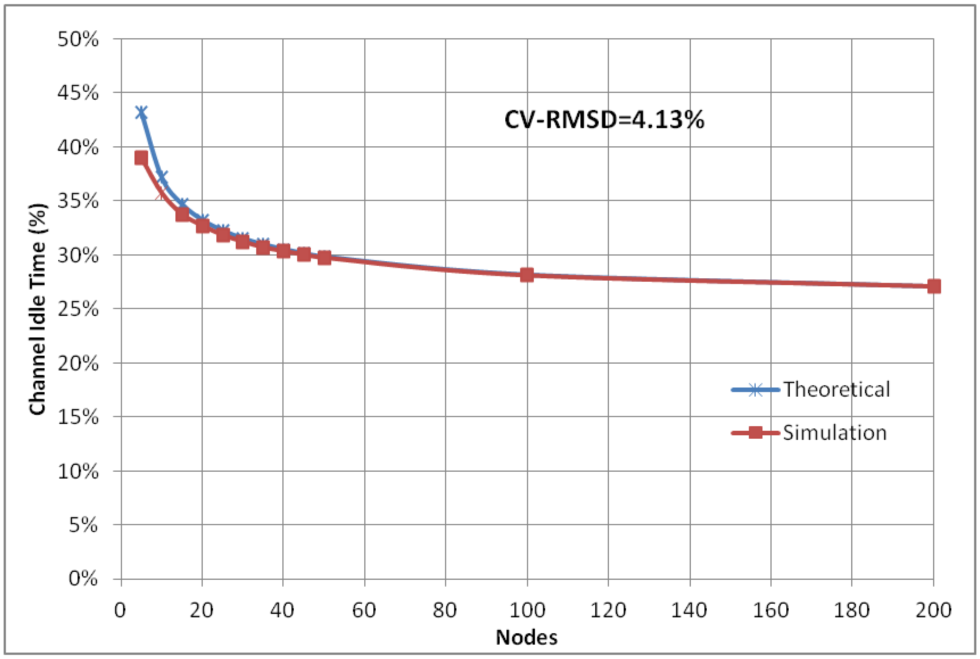


Figure 4-9: Theoretical and simulation-based performance in terms of the channel idle time with 5 CCAs.

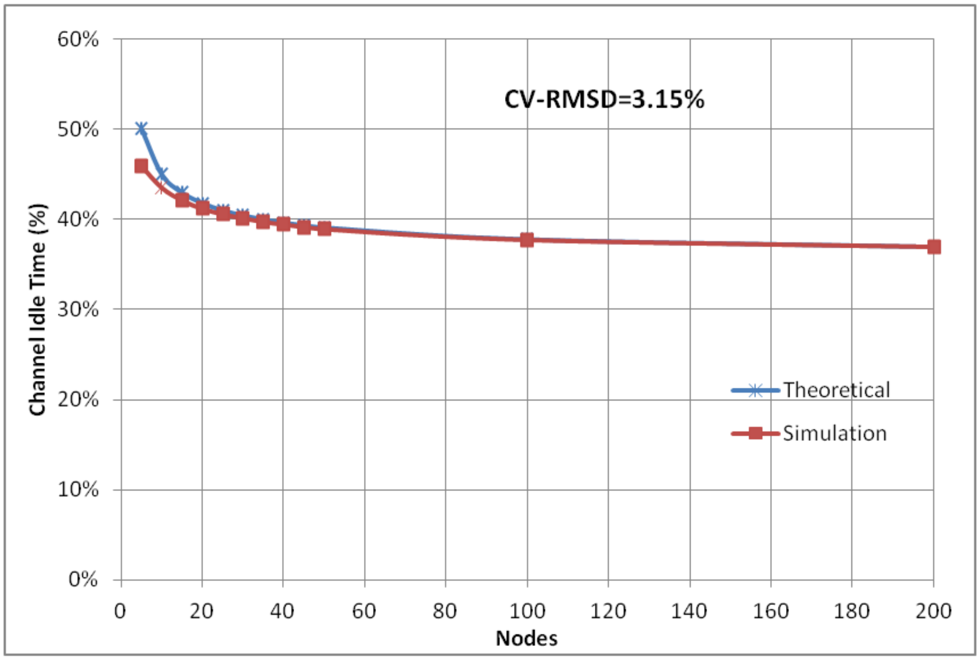


Figure 4-10: Theoretical and simulation-based performance in terms of the channel idle time with 8 CCAs.

4.6 Channel Collision Time

According to equation (3.15), the theoretical expression of the channel collision time is:

$$T_{collision} = \alpha_1 - N * L * \tau * (1-\tau)^{N-1} * \prod_{i=1}^{n_{CCA}} (1-\alpha_i)$$

The performance in terms of channel collision time is illustrated in Figure 4-11 (2 CCAs), Figure 4-12 (5 CCAs), and Figure 4-13 (8 CCAs). The CV-RMSD values achieved are 11.8% (2CCAs), 10.88% (5 CCAs), and 13.47% (8 CCAs). The ability of the Markov-based model to match the network's behavior, as confirmed by the simulations, is clear in the provided graphs (especially as N grows in value).

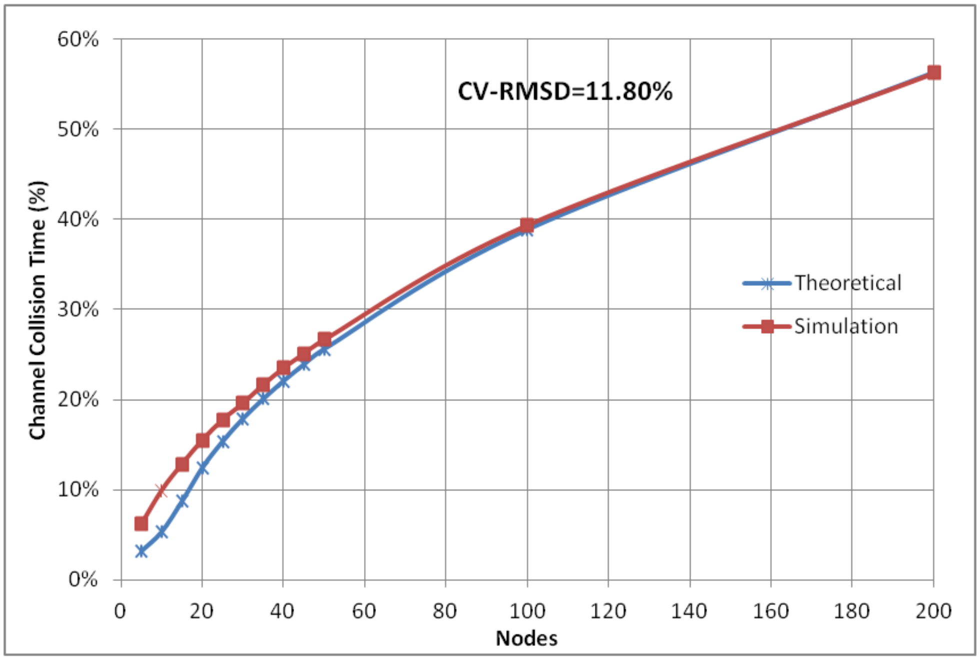


Figure 4-11: Theoretical and simulation-based performance in terms of the channel collision time with 2 CCAs.

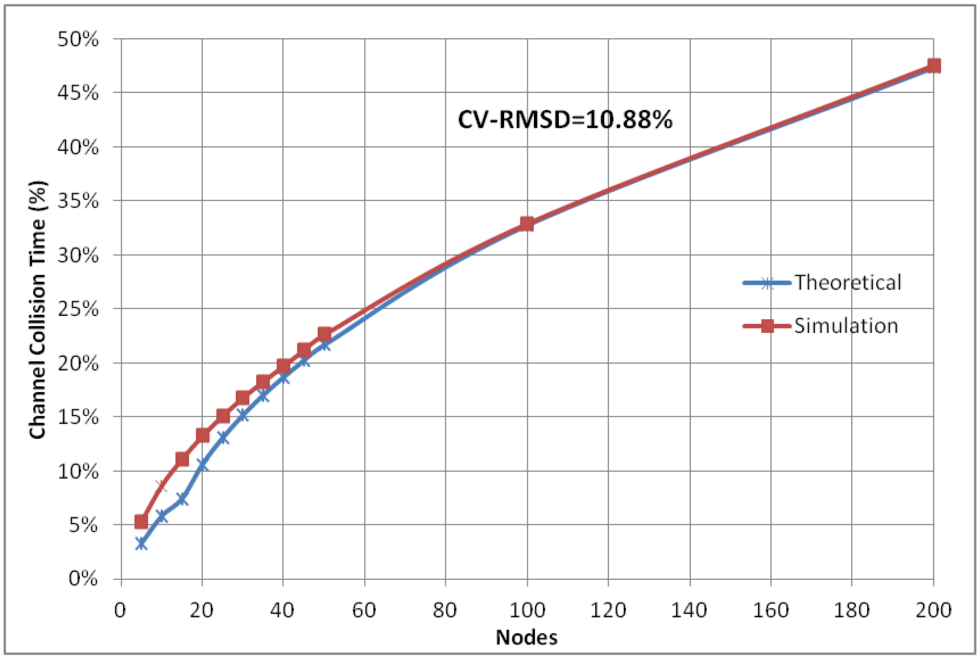


Figure 4-12: Theoretical and simulation-based performance in terms of the channel collision time with 5 CCAs.

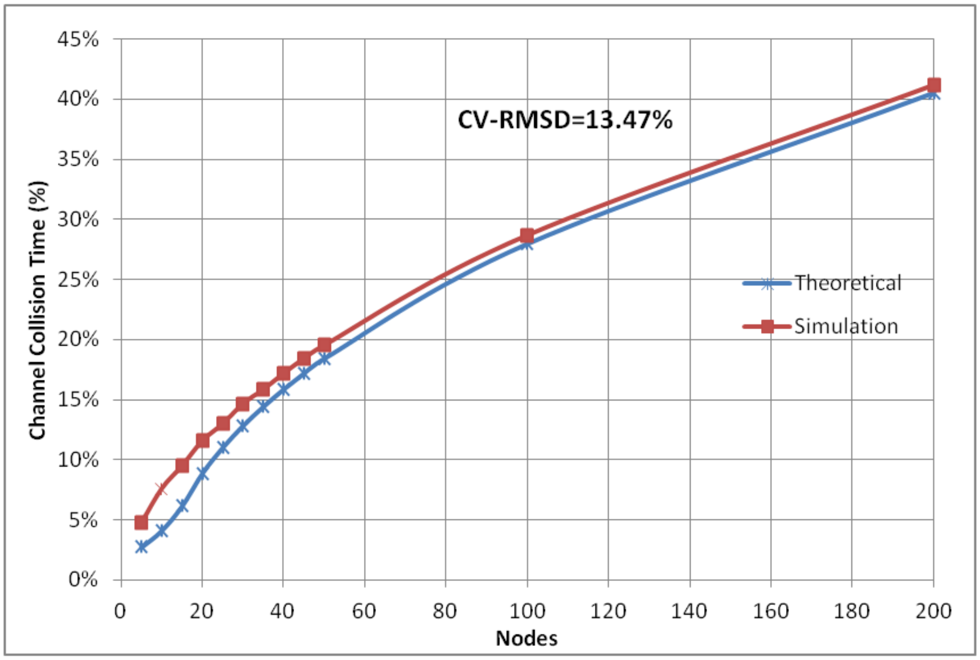


Figure 4-13: and simulation-based performance in terms of the channel collision time with 8 CCAs.

4.7 Delay

Recall that the theoretical expression of the delay is given by equation (3.33):

$$D = \frac{1}{\pi_{SC}} * \overline{T}_{BO} + \frac{\pi_{BC}}{\pi_{SC}} * \overline{T}_{CCA} + (1 + \frac{\pi_{CC}}{\pi_{SC}}) * (n_{cca} + L)$$

The encountered delay in networks applying the 802.15.4 Mac with variable CCAs is studied in Figure 4-14, Figure 4-15, and Figure 4-16. These figures show the theoretical and simulation-based performance with 2, 5, and 8 CCAs, respectively. We can see that our model is able to achieve a perfect match in Figure 4-14 and Figure 4-15, with a slight deviation from simulations, at larger N, in Figure 4-16. The CV-RMSD in these graphs are 1.89% (2 CCAs), 1.17% (5 CCAs), and 6.53% (8 CCAs), which indicate a high level of accuracy.

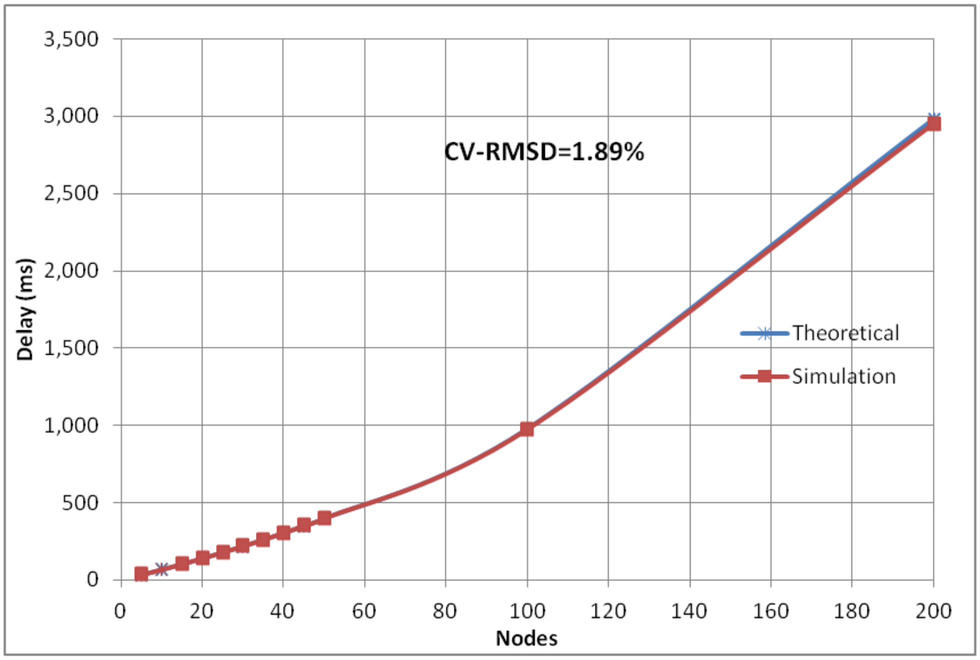


Figure 4-14: Theoretical and simulation-based performance in terms of the delay with 2 CCAs.

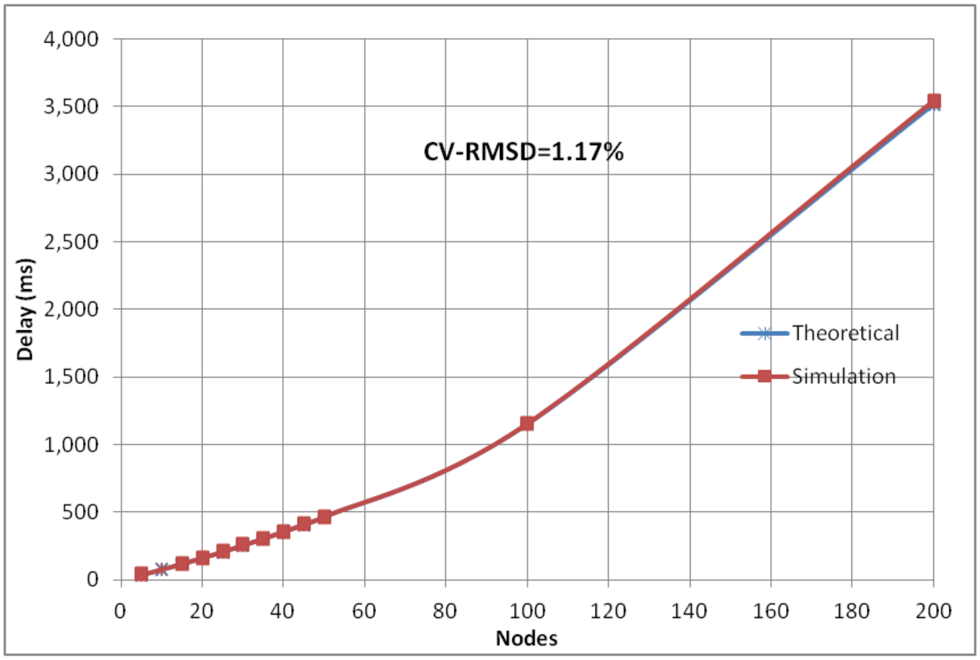


Figure 4-15: Theoretical and simulation-based performance in terms of the delay with 5 CCAs.

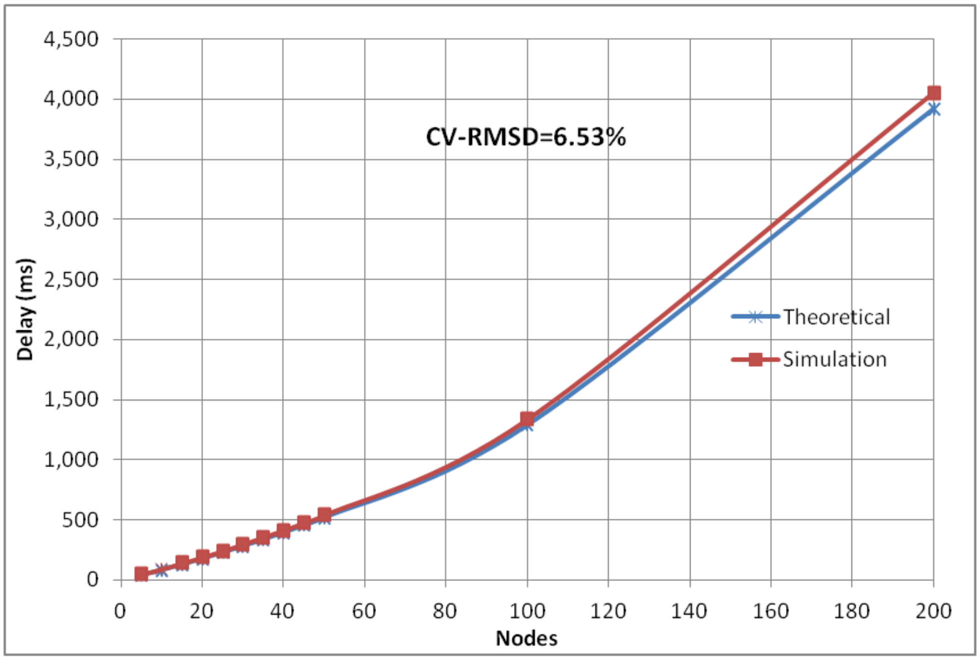


Figure 4-16: Theoretical and simulation-based performance in terms of the delay with 8 CCAs.

4.8 Total Energy Consumption

The theoretical expression of the average power consumption by each node is given by equation (3.20):

$$E_{total} = P_{idle}\frac{W-1}{2}b_{1,1} + P_{CCA}\left(1+\sum_{i=2}^{n_{CCA}}\prod_{j=1}^{i-1}(1-\alpha_j)\right)b_{1,1} + P_{tx}*L*\prod_{i=1}^{n_{CCA}}(1-\alpha_i)*b_{1,1} + P_{rx}*L$$
$$*(1-\tau)^{N-1}*\prod_{i=1}^{n_{CCA}}(1-\alpha_i)*b_{1,1}$$

The impact of using additional CCAs on the total energy consumption is depicted in Figure 4-17, Figure 4-18, and Figure 4-19 for 2, 5, and 8 CCAs, respectively. With CV-RMSD values of 10.25% (2 CCAs), 9.93% (5 CCAs), and 8.10% (8 CCAs), we can easily see the accurate match between the theoretical values, predicted by our mathematical model, and the simulation-based values.

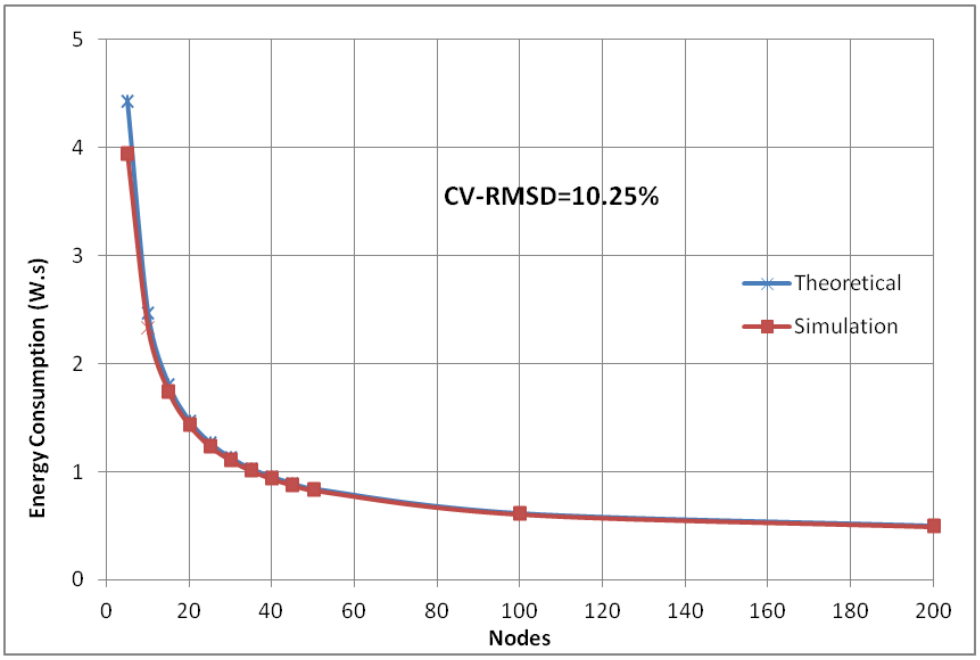


Figure 4-17: Theoretical and simulation-based performance in terms of the total energy consumption with 2 CCAs.

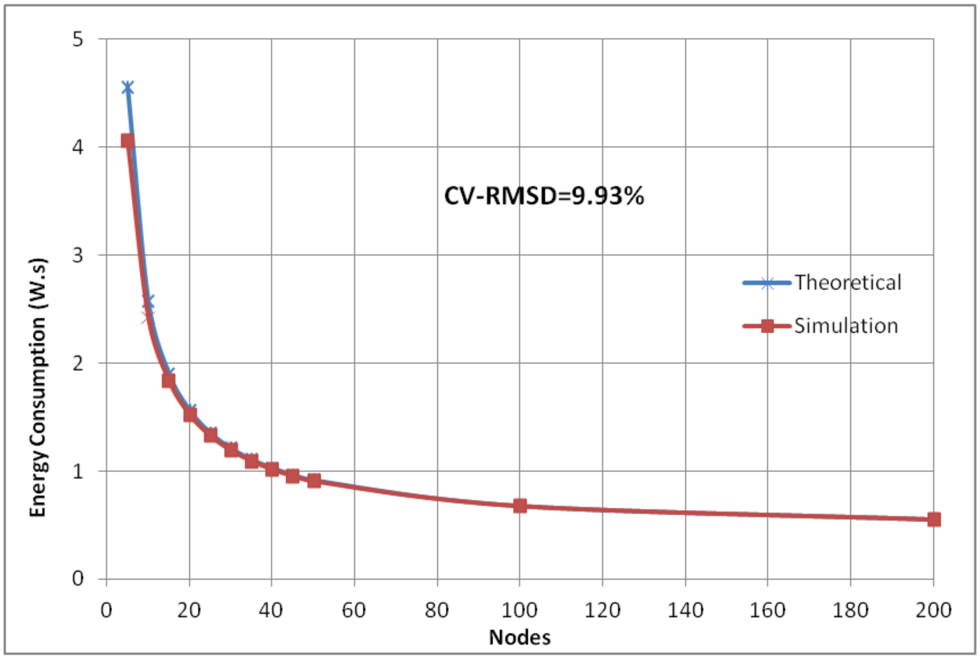


Figure 4-18: Theoretical and simulation-based performance in terms of the total energy consumption with 5 CCAs.

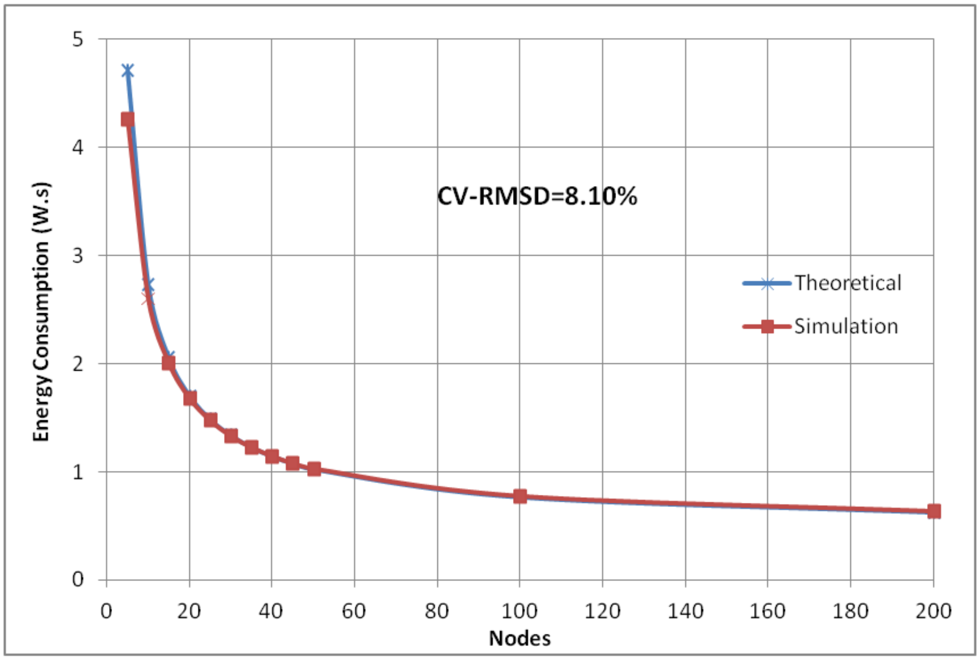


Figure 4-19: Theoretical and simulation-based performance in terms of the total energy consumption with 8 CCAs.

4.9 Energy Wasted in Collisions

The theoretical expression of the average energy wasted in collisions is given by equation (3.21):

$$E_c = P_{tx} * L * (1 - (1 - \tau)^{N-1}) * \prod_{i=1}^{n_{CCA}} (1 - \alpha_i) * b_{1,1}$$

In Figure 4-20 (2 CCAs), Figure 4-21 (5 CCAs), and Figure 4-22 (8 CCAs) we show the performance in terms of the energy wasted in collisions. Again, we can see the accuracy of our mathematical model in predicting the behavior of our system. The achieved CV-RMSD are 2.69%, 2.15%, and 2.24% for 2, 5, and 8 CCAs, respectively.

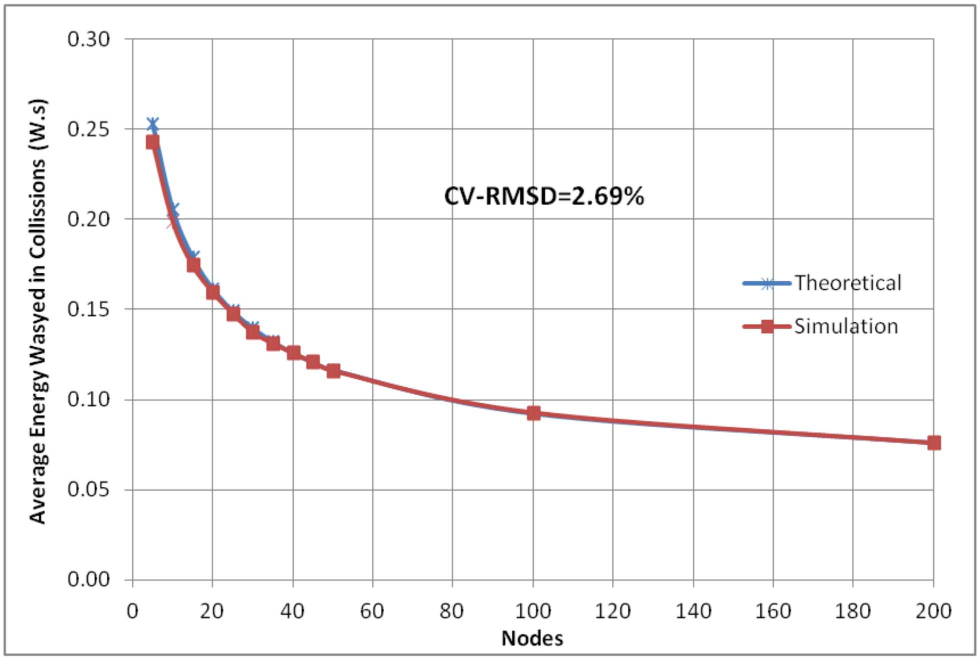


Figure 4-20: Theoretical and simulation-based performance in terms of the energy wasted in collisions with 2 CCAs.

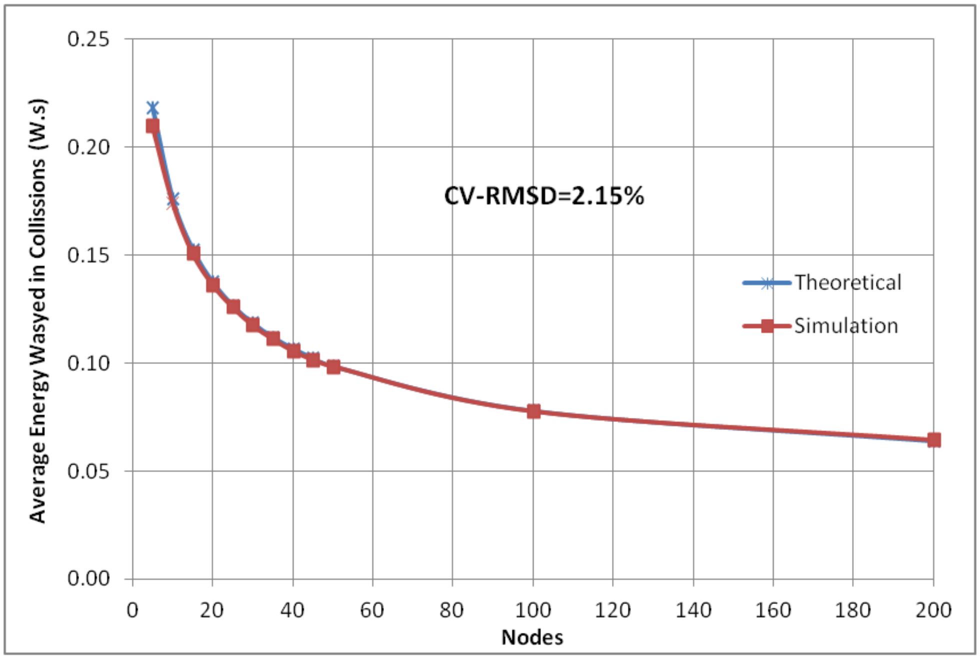


Figure 4-21: Theoretical and simulation-based performance in terms of the energy wasted in collisions with 5 CCAs.

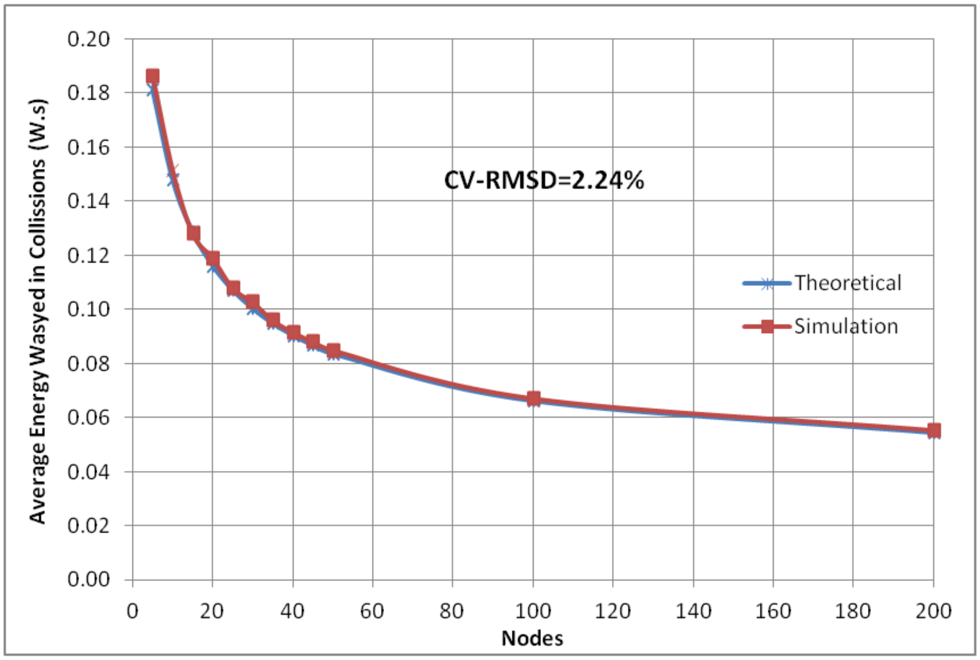


Figure 4-22: Theoretical and simulation-based performance in terms of the energy wasted in collisions with 8 CCAs.

4.10 Transmission Reliability

The theoretical expression of the reliability is given by equation (3.32):

$$R = \frac{1}{1 + \frac{(1-\pi_{BC})\pi_{BC}{}^{m+1}}{(1-\pi_{BC}{}^{m+1})\pi_{SC}} + \frac{\pi_{CC}{}^{n+1}}{(1-\pi_{BC})^{n+1} - \pi_{CC}{}^{n+1}}}$$

Figure 4-23, Figure 4-24, and Figure 4-25 illustrate the theoretical and simulation-based performance in terms of the reliability. The theoretical model is highly capable of describing the system's behavior as reflected by the simulations result. This fact is even implied by noticing the values of the CV-RMSD; 1.71% (2 CCAs), 2.14% (5 CCAs), and 3.66% (8 CCAs).

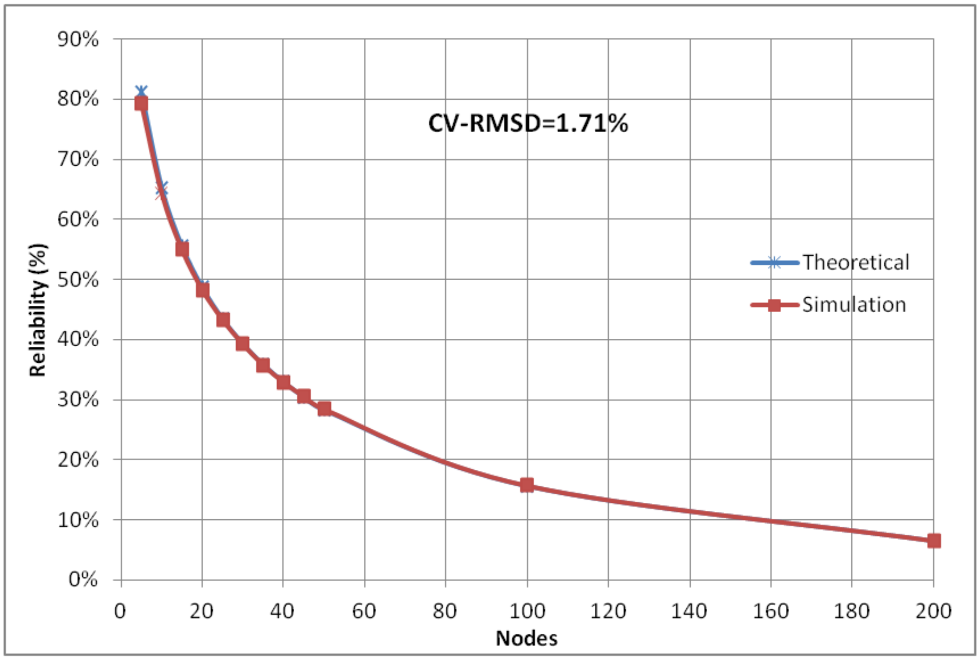


Figure 4-23: Theoretical and simulation-based performance in terms of the reliability with 2 CCAs.

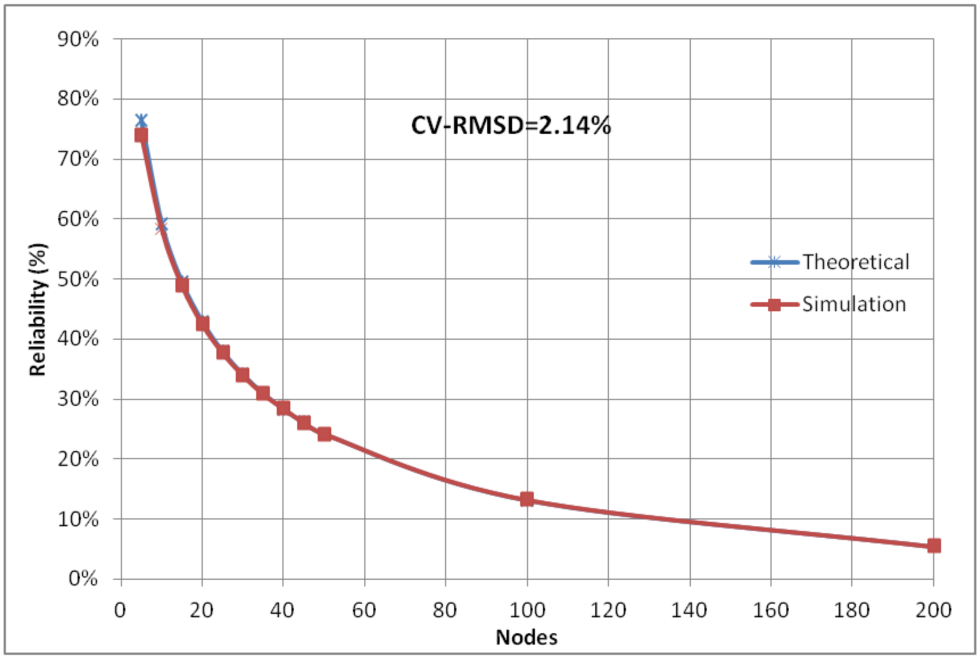


Figure 4-24: Theoretical and simulation-based performance in terms of the reliability with 5 CCAs.

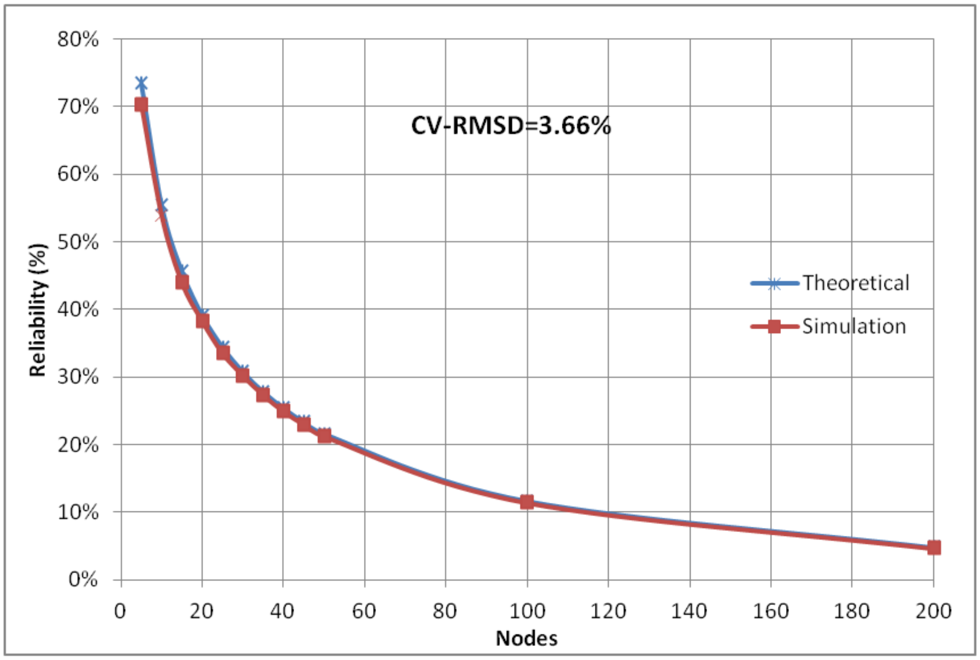


Figure 4-25: Theoretical and simulation-based performance in terms of the reliability with 8 CCAs.

4.11 Protocol Fairness

After discussing the accuracy of our Markov-based model, we investigate whether incorporating variable CCAs affects the known feature of 802.15.4 MAC of being fair in treating the different nodes.

We examine the fairness of the 802.15.4 MAC for different values of CCAs through the fairness index (Jain's formula) introduced in [JAI84]:

$$fairness\ index = \frac{(\sum x_i)^2}{N \sum {x_i}^2}$$

where, N is the number of nodes in the network, and x_i is the medium share of the *i*th node. A fair protocol is one that can achieve a fairness index close to 1. Such a protocol allows for a fair sharing of the wireless channel among the contending nodes. In contrast, as the value of the fairness index decreases (towards zero), the protocol become less fair and tends to favor certain nodes over others in their access to the medium. In Figure 4-26 we show the performance of the MAC protocol with variable CCAs in terms of fairness. It is evident that this protocol is fair and allowing the nodes an equal opportunity of medium access.

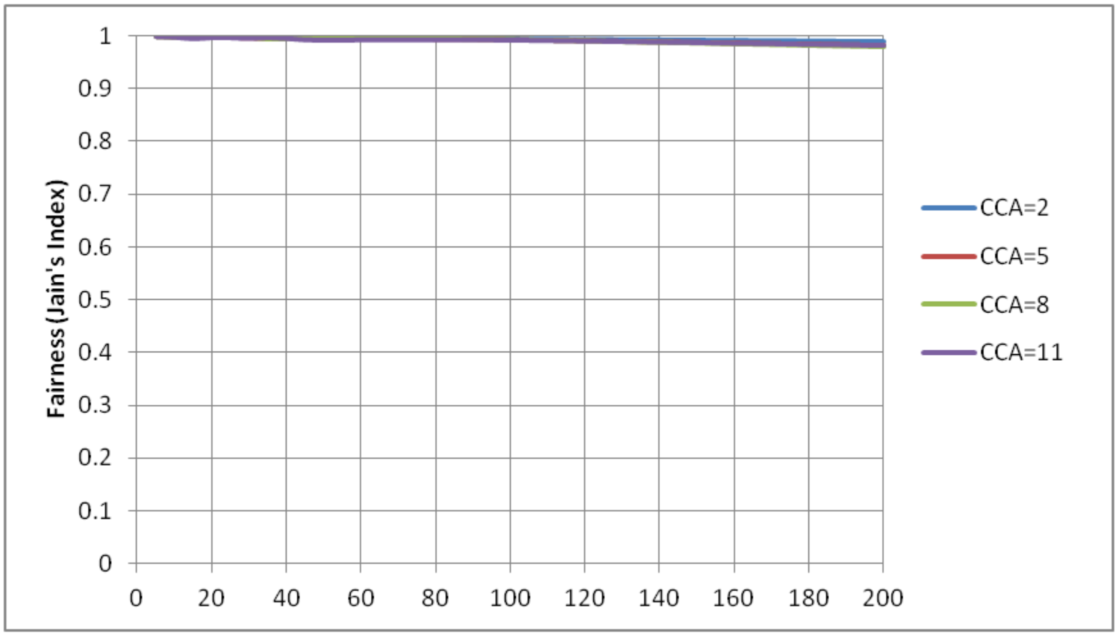


Figure 4-26: Fairness of the 802.15.4 MAC with variable CCAs.

4.12 Conclusion

In this chapter, we introduced a WSN MAC simulator to study the behavior of the 802.15.4 MAC protocol with variable CCAs implemented. The simulator is used to validate the Markov-based model we developed in Chapter 3. The results demonstrate the superior capability of our mathematical model to predict the behaviour of the designed MAC protocol, and a significant degree of matching between the theoretical data and the simulation-based data. Since we have proven the accuracy of our mathematical model, in the next chapter we work on designing a novel MAC protocol that can provide enhanced performance in terms of various performance metrics, such as energy consumption, channel utilization and delay.

Chapter 5

Semi-Persistent CSMA/CA for Efficient and Reliable Communication in WSN

5.1 Introduction

In Chapter 2, we highlighted the fact that the CCA feature of the 802.15.4 standard requires more research to fully exploit and enhance the performance of the of the MAC sub-layer. In Chapter 3, we described our proposal for a variable CCA MAC that can operate in a semi-persistent fashion. In this chapter, we first examine the effect of the variable CCAs on various performance metrics, using our developed MAC simulator. Based on the conclusions from this analysis, we develop a novel hybrid MAC protocol that is a combination of the persistent and non-persistent modes of CSMA-CA. The proposed protocol takes advantage of the semi-persistent feature of the variable CCA MAC protocol. We develop both the analytical and simulation models of the proposed protocol, and then examine the effect of the SP-MAC on two protocols: the Binary Exponential Backoff and the Adaptive Backoff Algorithm.

5.2 Analysis of Variable CCA MAC

To conduct the analysis of the Variable CCA MAC protocol, we focus on the following performance metrics: channel utilization, delay, reliability, and energy consumption. Figures 5-1, 5-2, 5-3, and 5-4 show the simulation results of these metrics, as functions of the number of nodes N, respectively. In all of these figures we can clearly observe that there is performance degradation as the number of CCAs increases. In particular, if we check the performance in terms of channel utilization in Figure 5-1, we can see that we achieve a utilization of 30% at N

=200 with 2 CCAs. At the same network size, the utilization is 25% with 5 CCAs and 21% with 8 CCAs. If we refer to Figure 5-2, the delay with N = 200 and 2 CCAs is 2974 ms, while it is 3536 ms with 5 CCAs and 4047 ms with 8 CCAs. Next, in Figure 5-3 the reliability at N = 50 is 28% with 2 CCAs, 24% with 5 CCAs, and 21% with 8 CCAs. Finally, in Figure 5-4 we can see that at N = 25 the energy consumption with 2 CCAs, 5 CCAs, and 8 CCAs is 1.23 W.s, 1.32 W.s, and 1.47 W.s, respectively.

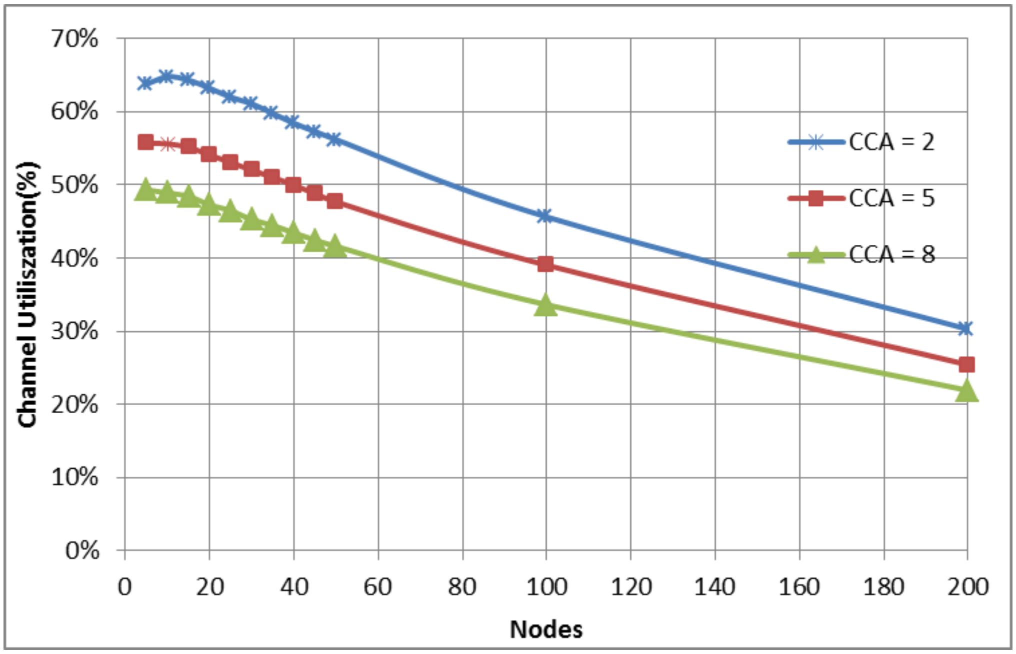


Figure 5-1: Impact of increasing the value of the contention window on the channel utilization

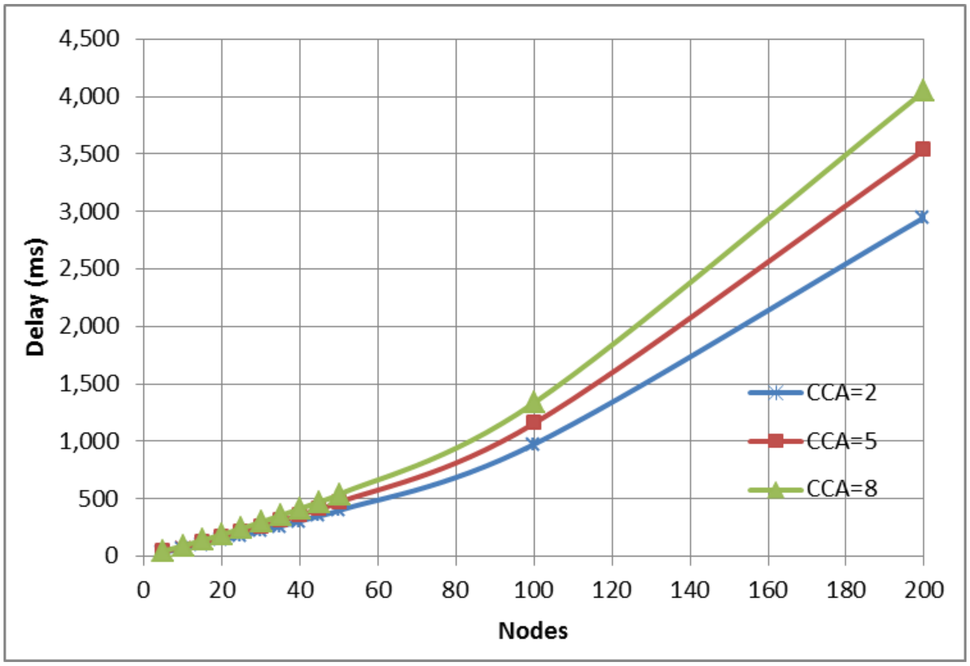


Figure 5-2: Impact of increasing the value of the contention window on the transmission delay

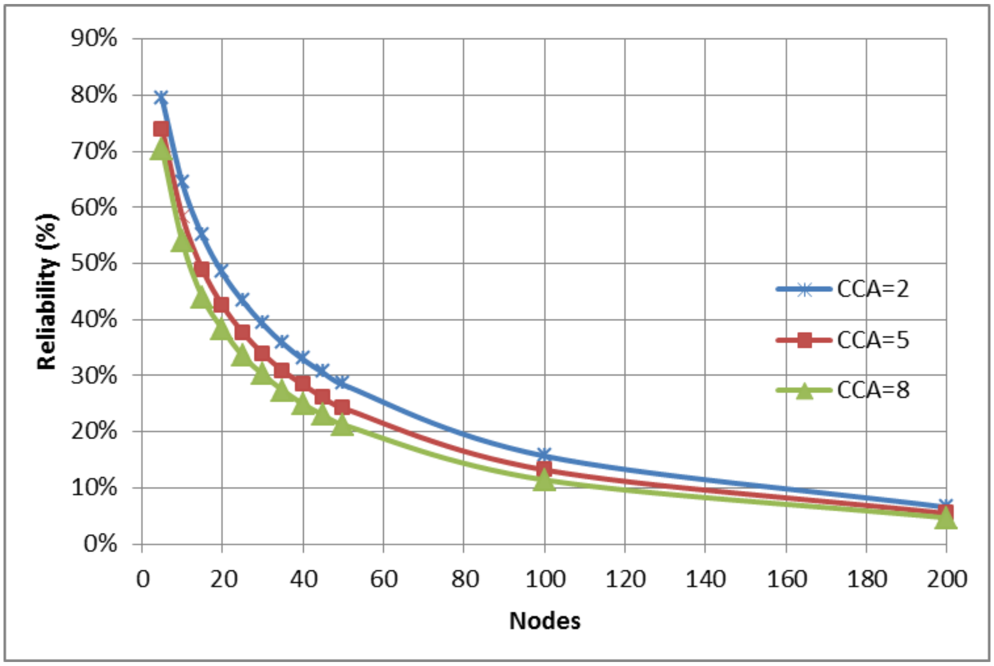


Figure 5-3: Impact of increasing the value of the contention window on the reliability

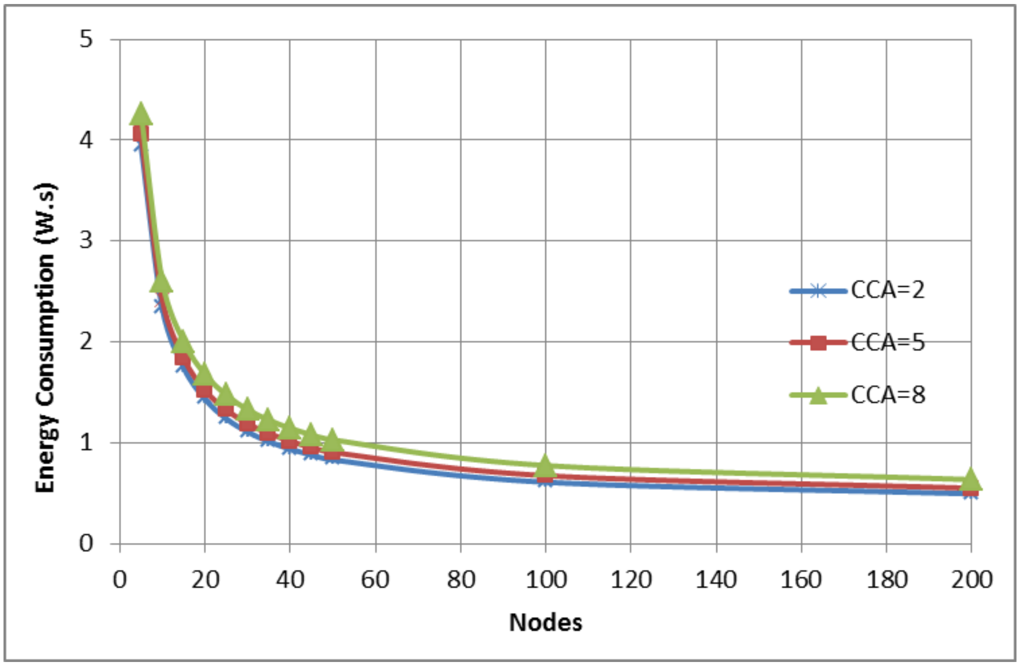


Figure 5-4: Impact of increasing the value of the contention window on the average energy consumption

The reason behind this behavior is that as two nodes happen to have the same number of CCAs and conduct CCA1 (state $S_{1,1}$) at the same time, they will suffer from a collision, regardless of the number of CCAs they are supposed to perform. This means that configuring nodes with a constant value of CCAs to conduct will deteriorate the overall performance. Therefore, increasing to the number of CCAs performed to be constant for all nodes leads to more latency in the network without improving the performance metrics.

5.3 Semi-Persistent MAC

In a shared communication medium environment, the main reason behind collisions is when two nodes, that are about to conduct the same number of CCAs, start their first CCA at the same time. In such a scenario, the probability of collision can be expressed as $1-(1-\tau)^{N-1}$, where τ is the probability of a node initiating its first CCA and N is the number of nodes sharing the medium.

Both 802.11 MAC and 802.15.4 MAC protocols work on mitigating the consequences of sharing the wireless medium by employing intelligent CSMA-CA mechanisms. With 802.11, the approach is to keep listening to the medium during the backoff to ensure that is idle before decrementing the backoff counter. With 802.15.4, however, the approach is to sleep during backoff and then conduct two CCAs before attempting to transmit the packet. Our aim in this chapter is to devise a hybrid MAC protocol that benefits from the access schemes of both standards.

- **IEEE 802.11 CSMA/CA Access Method**

In 802.11-based networks, a node has to persistently monitor the wireless medium to check if any ongoing transmission is in progress. This node cannot send any packet before confirming that the channel is free. If it senses a transmission, it has to backoff for a random period of time (selected from the interval [0, CW], where CW is in the range [15, 1023] in units of aSlotTime (set at 50 μs)). The node sets its backoff timer to the selected value and then decreases the timer, in steps of aSlotTime, before starting a transmission. While the timer is decreasing, the node keeps on sensing the medium. If the medium becomes busy at any time slot, the node will freeze its timer. The timer is not resumed before the medium becomes idle again. Once the backoff timer expires, the node sends its packet.

- **IEEE 802.15.4 Access Method**

With 802.15.4-based networks, the node has to conduct 2 CCAs, after a backoff period, before sending any packet. The wireless medium should be sensed idle during these two CCAs. Otherwise, the packet cannot be sent and the node backs off again. The main difference compared to 802.11 is that nodes do not sense the medium during the backoff period, they go into a sleep mode.

From these descriptions we can see that 802.11 implements a *persistent access mode* CSMA-CA while 802.15.4 implements a *non-persistent access mode* CSMA-CA.

In our view, there should be a balance between the two methodologies: the persistent mode can be beneficial in reducing the probability of collisions, while the non-persistent mode can be effective in reducing the consumption of the node's power resources. That is, we aim at devising a *semi*-persistent version of CSMA-CA that incorporates this new hybrid functionality. This functionality is based on increasing the number of CCAs beyond the standard 2 CCAs. The node intending to send a packet has to firstly backoff, similar to 802.15.4, for a random period of time. After that, it starts conducting CCAs, the number of which is selected randomly from the interval [2, CCA_MAX], where CCA_MAX is set to 11 in our simulation framework. In this manner, the node gets better opportunities of accessing the wireless medium, and avoids getting back into the backoff states that leads to increases in the transmission delays. The flowchart of the semi-persistent MAC is given in Figure 5-5. The main difference between the SP-MAC and the standard protocol is that the contention window CW is generated randomly between 2 and a maximum value CCA_MAX. This way, the nodes are given more opportunities to monitor the shared channel to detect any ongoing transmission before attempting to transmit the frame. It is expected that the increase of the number of clear channel assessments performed before a frame transmission will have an impact on the energy consumption; however, the decrease in the collisions will balance the energy consumption and have a positive impact on the total energy consumption.

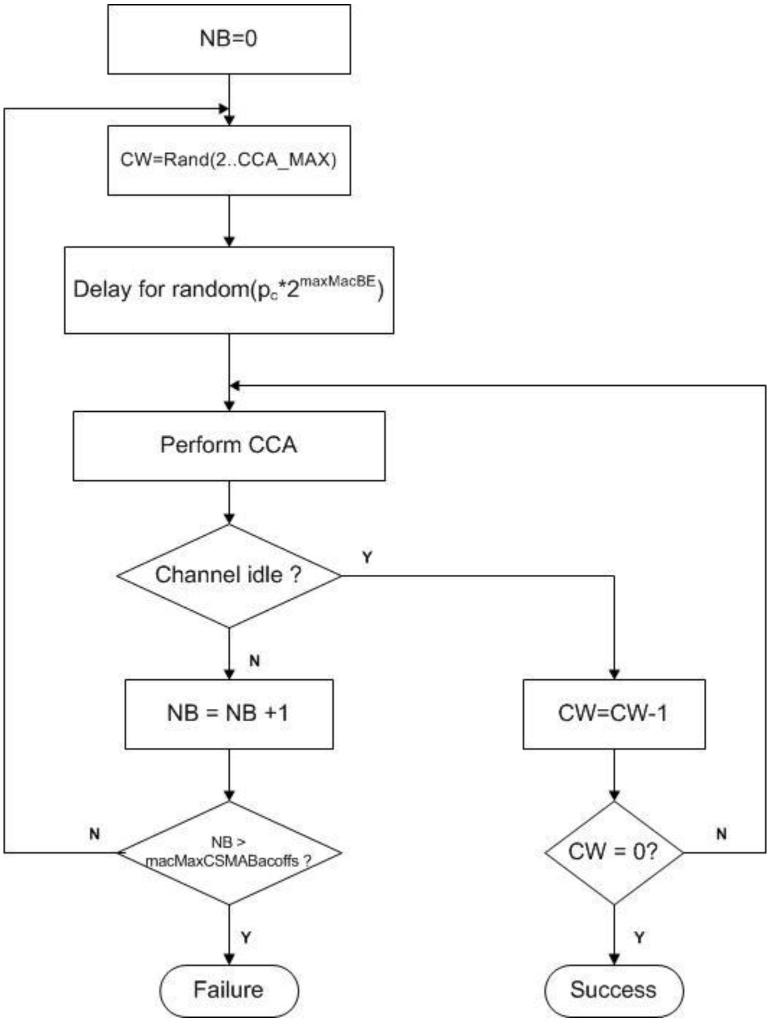


Figure 5-5: Flowchart of SP-MAC

5.4 Analytical Expression of the Probability of Collision

The probability of collision plays a crucial role in determining the performance of a MAC protocol. In this section we derive the analytical expression of the probability of collision for the semi-persistent MAC. We show that this quantity decreases significantly when the semi-persistent feature is incorporated into the MAC protocol.

We can compute the probability of collisions by firstly finding its complement. That is, the probability of having no collision, denoted as p_{nc}.

The current node generates uniformly a contention window, denoted as cca, between [2, CCA_MAX]; therefore, the probability of non collision is the average of the probability of non collision of each value of the possible values of the contention window. Therefore, we can express the probability of non collision as follows:

$$p_{nc} = \frac{1}{CCA} \sum_{cca=2}^{CCA\_MAX} p_{nc}^{cca}$$

where p_{nc}^{cca} is the probability of non collision for the chosen value of the contention window. We will focus on what follows on the derivation of p_{nc}^{cca}.

Since the generation of the contention window is uniformly distributed over the interval [2,CCA_MAX], the number of nodes that have a contention window equal to a specific value of the contention window is equal to : $\frac{N-1}{CCA\_MAX-1}$.

In order that the current node doesn't collide with nodes who have a similar value of the contention window (cca), the latter nodes should be at any state other than $S_{1,1}$. This probability can be written as:

$$(1-\tau)^{\frac{N-1}{CCA\_MAX-1}} \tag{5.1}$$

Likewise, in order that the current node doesn't collide with nodes that have a lower value of the contention windows, say i with i<cca, the latter nodes should not be in the backoff state $S_{0,cca-i}$. This probability can be written as:

$$(1-b_{0,cca-i})^{\frac{N-1}{CCA\_MAX-1}}$$

According to equation (3.11), the probability of being a state $S_{0,cca-i}$ is equal to:

$$b_{0,cca-i}=\frac{W+i-cca}{W}$$

Therefore, the probability can be now expressed as:

$$(1-\frac{W+i-cca}{W}\tau)^{\frac{N-1}{CCA\_MAX-1}} \tag{5.2}$$

Last, in order for the current node not to collide with nodes that have a greater value of the contention window, say i where i>cca, the latter nodes should not be in the state $S_{1,i+1-cca}$. This probability can be expressed as:

$$(1-\tau\prod_{k=1}^{i-cca}(1-\alpha_k))^{\frac{N-1}{CCA\_MAX-1}} \tag{5.3}$$

From the previous derivations (5.1, 5.2, and 5.3), we can express p_{nc}^{cca} as:

$$p_{nc}^{cca}=\prod_{i=2}^{cca-1}(1-\frac{W+i-cca}{W}\tau)^{\frac{N-1}{CCA\_MAX-1}}*\prod_{i=cca}^{CCA\_MAX}(1-\tau\prod_{k=1}^{i-cca}(1-\alpha_k))^{\frac{N-1}{CCA\_MAX-1}} \tag{5.4}$$

And the probability of non collision as:

$$p_{nc}=\frac{1}{CCA\_MAX-1}\sum_{cca=2}^{CCA\_MAX}\prod_{i=2}^{cca-1}(1-\frac{W+i-cca}{W}\tau)^{\frac{N-1}{CCA\_MAX-1}}*\prod_{i=cca}^{CCA\_MAX}(1-\tau\prod_{k=1}^{i-cca}(1-\alpha_k))^{\frac{N-1}{CCA\_MAX-1}} \tag{5.5}$$

Since, the probability of collision is the complement of the probability of non collision, p_c is given by the following equation

$$p_c = 1 - \frac{1}{CCA\_MAX - 1} \sum_{cca=2}^{CCA\_MAX} \prod_{i=2}^{cca-1} \left(1 - \frac{W+i-cca}{W} \tau\right)^{\frac{N-1}{CCA\_MAX-1}} * \prod_{i=cca}^{CCA\_MAX} \left(1 - \tau \prod_{k=1}^{i-cca} (1 - \alpha k)\right) N-1 CCA\_MAX-1 \tag{5.6}$$

Note that in case we have only two CCA states (i.e., the standard case), the probability of collision reduces to the following expression given by equation (3.5):

$$p_c = 1 - (1 - \tau)^{N-1}$$

It can be easily seen that the probability of collision of the semi-persistent MAC is lower than the standard value $p_c = 1 - (1 - \tau)^{N-1}$. Consequently, the performance metrics that depend on the value of the probability of collision gets improved as we increase the value of the contention window. This will be detailed in the following sections.

5.5 Analytical Model

Based on the derivations of chapter 3, we express the performance metrics for the semi-persistent MAC protocol. The main difference is that the maximum value of the number of CCAs, denoted in the equations as n_{cca}, is equal to the average value of the contention window, i.e. $\frac{CCA\_MAX+2}{2}$.

- **Channel Utilization**

Based on equation (3.13), the channel utilization can be expressed as:

$$U = N * L * \tau * (1 - p_c) * \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1 - \alpha_i) \tag{5.7}$$

- **Channel Idle Time**

Based on equation (3.14), the channel idle time of the semi-persistent MAC protocol is given by the following equation:

$$T_{idle} = 1 - \alpha_1 \tag{5.8}$$

- **Channel Collision Time**

From equation (3.15), the channel collision time is expressed as:

$$T_{collision} = \alpha_1 - N * L * \tau * (1 - p_c) * \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1 - \alpha_i) \tag{5.9}$$

- **Energy Consumption**

Based on equation (3.20), we can derive the equation of the total energy consumption as:

$$E_{total} = P_{idle} \frac{W-1}{2} \tau + P_{CCA} \left(1 + \sum_{i=2}^{\frac{CCA\_MAX+2}{2}} \prod_{j=1}^{i-1} (1 - \alpha_j) \right) \tau + P_{tx} * L * \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1 -$$

$$\alpha i*\tau+Prx*L*(1-pc)*i=1CCA\_MAX+221-\alpha i*\tau \tag{5.10}$$

- **Energy Wasted in Collisions**

Similarly the energy wasted in collisions can be expressed from equation (3.21) as:

$$E_c = P_{tx} * L * p_c * \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1 - \alpha_i) * \tau \tag{5.11}$$

- **Reliability**

From equation (3.32), we can derive the equation of the reliability for the semi-persistent MAC protocol.

$$R = \frac{1}{1+\frac{(1-\pi_{BC})\pi_{BC}{}^{m+1}}{(1-\pi_{BC}{}^{m+1})\pi_{SC}}+\frac{\pi_{CC}{}^{n+1}}{(1-\pi_{BC})^{n+1}-\pi_{CC}{}^{n+1}}} \qquad (5.12)$$

With

$$\pi_{BC} = \sum_{i=1}^{\frac{CCA\_MAX+2}{2}} \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)$$

$$\pi_{CC} = p_c \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1-\alpha_i)$$

$$\pi_{SC} = (1-p_c) \prod_{i=1}^{\frac{CCA\_MAX+2}{2}} (1-\alpha_i)$$

- **Delay**

The delay of the semi-persistent MAC protocol can be derived from equation (3.33) as:

$$D = \frac{1}{\pi_{SC}} * \overline{T}_{BO} + \frac{\pi_{BC}}{\pi_{SC}} * \overline{T}_{CCA} + \left(1+\frac{\pi_{CC}}{\pi_{SC}}\right) * \left(\frac{CCA\_MAX+2}{2} + L\right) \qquad (5.13)$$

5.6 Impact of SP-MAC on the efficiency of MAC protocols

To study how the semi-persistent MAC would improve the overall performance in the WSN, we incorporate the concept of semi-persistency in two protocols, namely, the standard 802.15.4 MAC and the Adaptive Backoff Algorithm (ABA) defined in [KHA11b]. The standard 802.15.4 MAC utilizes the Binary Exponential Backoff (BEB) algorithm that has been described in

Chapter 1. ABA is a backoff algorithm that intends to involve the level of the activities over the wireless medium into the calculation of the backoff window. This is accomplished by updating the contention window according to the following expression:

$$W = P_c * W_{max} \tag{5.14}$$

This equation states that the upper limit of the interval from which the backoff window is selected (W) is directly dependent on the *steady-state* value of the probability of collision (P_c). The increases in the probability of collision are indicative of a busy channel and bigger traffic intensity. Thus, equation (5.14) states that a node should increase the interval from which it selects its backoff window. This extended interval reduces the probability of having multiple nodes using the same backoff window, which reflects in reduced likelihood of collisions. On the other side, as the probability of collision decreases, the wireless medium becomes less busy, and thus, as equation (5.14) dictates, the nodes should decrease the range of the backoff window selection interval. In this manner, the protocol manages to associate the size of the backoff window with the level of traffic over the communication medium.

In the following sub-section we investigate the benefits of incorporating the semi-persistency concept into the functionality of both BEB and ABA. In what follows, we refer to these versions of BEB and ABA as semi-persistent BEB (SP-BEB) and semi-persistent ABA (SP-ABA). We present our simulation results and analyze their indications. We focus on the metrics: probability of collision, channel utilization, channel idle time, channel collision time, transmission delay, transmission reliability, total energy consumption, and average energy wasted in collisions. In these simulations we assume that the size of the contention window varies between 2 and 11.

5.6.1 Probability of Collision

In Figure 5-6 we show the performance in terms of the probability of collision as a function of the network size. We can clearly see that both SP-BEB and SP-ABA can achieve a promising performance. In particular, while the use of BEB and ABA leads to a probability of collision of 98% and 67% (at N =100), respectively, SP-BEB and SP-ABA reduce these probabilities to 57% and 43%, respectively.

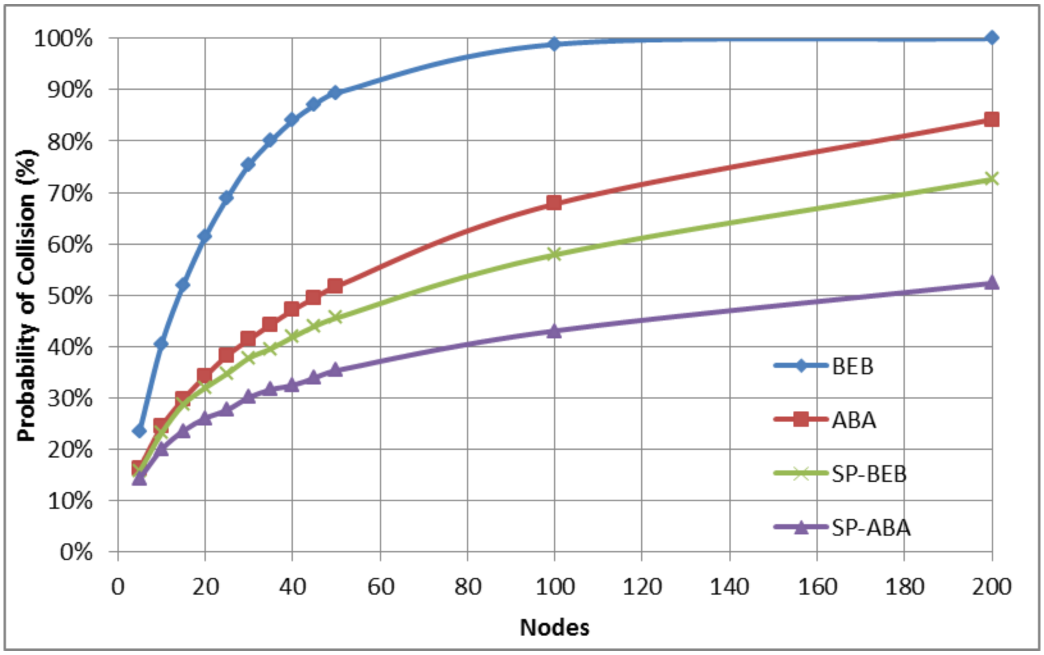


Figure 5-6: Probability of Collision of BEB, ABA, SP-BEB, and SP-ABA under unacknowledged traffic.

5.6.2 Channel Utilization

In Figure 5-7 we illustrate the performance in terms of the channel utilization. The improvements in this metric are apparent in this figure. For example, at N = 200, we BEB and ABA achieve a channel utilization of 0.01% and 29.8%, respectively. However, SP-BEB and SP-ABA boost these values to 42.53% and 55.31%, respectively. To the best of the author's knowledge, the latter level of 55.31%, at N = 200, for the channel utilization is not matched in the literature.

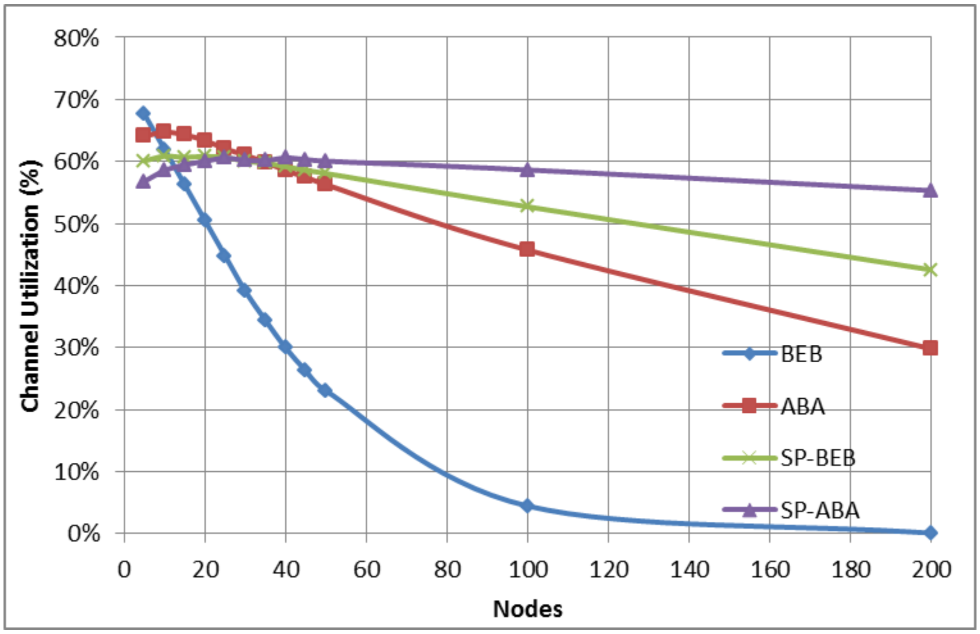


Figure 5-7: Channel Utilization of BEB, ABA, SP-BEB, and SP-ABA.

5.6.3 Channel Idle Time

The performance in terms of the channel idle time is shown in Figure 5-8. We can see that the semi-persistency feature is leading to increases in this metric. In particular, at N = 50, BEB achieves a channel idle time of 13.09% while SP-BEB leads to a value of 21.84%. With ABA, the channel idle time is 17.21% and it increases to 25.07% with SP-ABA (both at N =50).

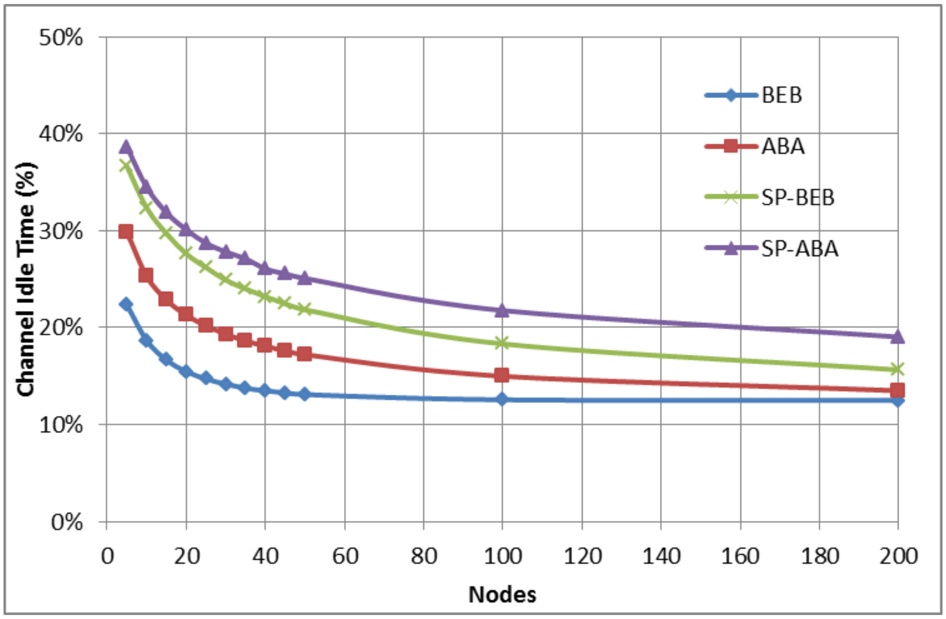


Figure 5-8: Channel Idle Time of BEB, ABA, SP-BEB, and SP-ABA.

5.6.4 Channel Collision Time

In Figure 5-9 the performance in terms of the channel collision time is shown. It is quite clear that the use of the semi-persistency concept is leading to significant reductions in the total channel time wasted due to collisions. At N = 200, 87.4% of the channel time is wasted in collisions with BEB. SP-BEB reduces that collisions time to 42.45%. Similarly, while ABA achieves channel collision time of 56.61% (at N = 200), SP-ABA reduces that to 25.63%.

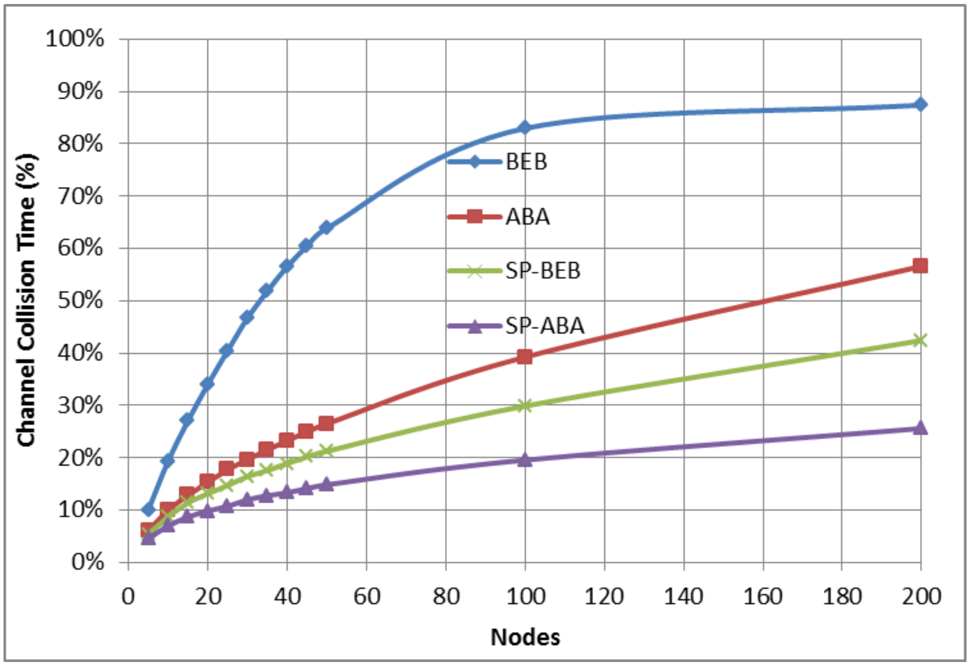


Figure 5-9: Channel Collision Time of BEB, ABA, SP-BEB, and SP-ABA.

5.6.5 Transmission Delay

In Figure 5-10 we graph the behavior of the transmission delay. The trend of the shown curves clearly illustrates the effect of using the semi-persistency concept (note that the curve of BEB is not shown because the collected results are too large, compared to the other algorithms, and could not be fit with the used scale). At N = 100, BEB and ABA results in transmission delays of 10018.7 ms and 971.2 ms, respectively. However, SP-BEB and SP-ABA achieve transmission delays of 849.7 ms and 763.8 ms, respectively.

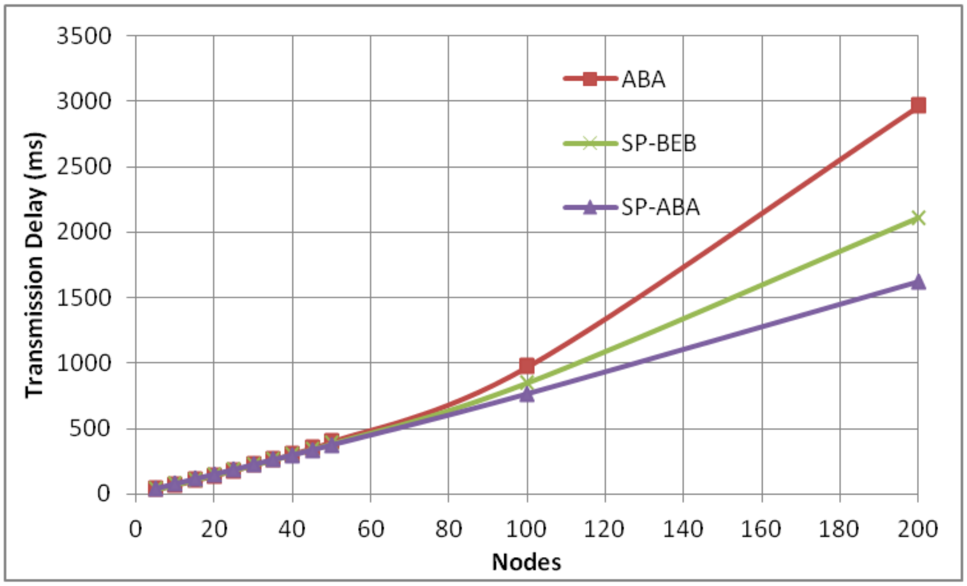


Figure 5-10: Transmission Delay of ABA, SP-BEB, and SP-ABA.

5.6.6 Transmission Reliability

In Figure 5-11 we illustrate the behavior of the transmission reliability as the network size increases. At N = 50, we can observe that with BEB the transmission reliability reaches 4.36% while with ABA it becomes 19.83%. As we introduce SP-BEB and SP-ABA, the values jump to 10.45% and 28.28%, respectively.

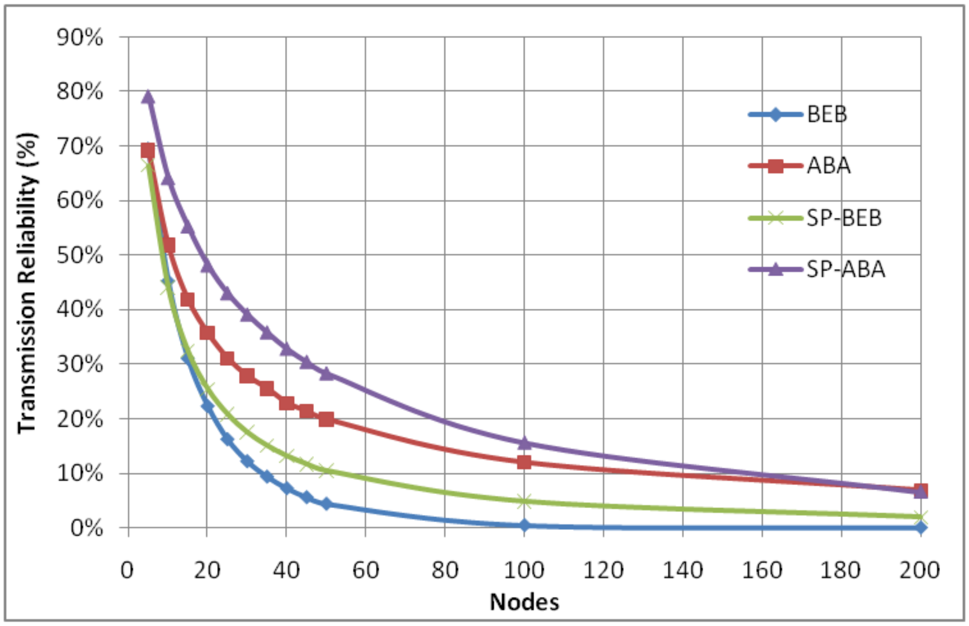


Figure 5-11: Transmission Reliability of BEB, ABA, SP-BEB, and SP-ABA.

5.6.7 Total Energy Consumption

The total energy wasted under the different protocols is shown in Figure 5-12. In this figure, at N = 200, BEB and ABA lead to a total energy consumption of 1.19 W.s and 0.49 W.s, respectively. With SP-BEB and SP-ABA the values are 0.97 W.s and 0.6 W.s, respectively. We should mention that the energy consumption with ABA is lower than SP_ABA because the calculation of the energy consumption includes the energy spent during packet reception. Since ABA wastes more time in collisions, the channel utilization is lower and therefore the energy consumed during packet reception is lower than SP-ABA. This will be clarified more in the next sub-section when studying the average energy wasted in collisions.

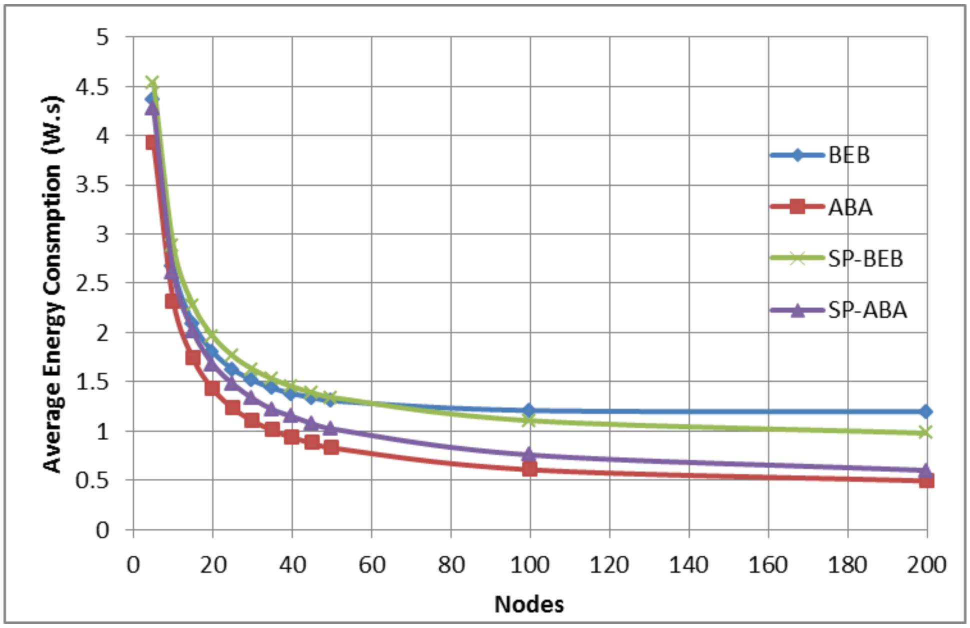


Figure 5-12: Total Energy Consumption of BEB, ABA, SP-BEB, and SP-ABA.

5.6.8 Average Energy Wasted in Collisions

In Figure 5-13, we study the performance in terms of the average energy wasted in collisions. We notice that the semi-persistency concept provides important enhancements to the performance in terms of this metric. For example, at N = 50, while BEB and ABA waste 0.37 W.s and 0.11 W.s, respectively, in collisions, SP-BEB and SP-ABA reduce these values to 0.09 W.s and 0.06 W.s, respectively. It is interesting to notice that BEB performs the worst in terms of the power it wastes due to collisions. Apparently, when operating BEB most of the energy is lost in collisions.

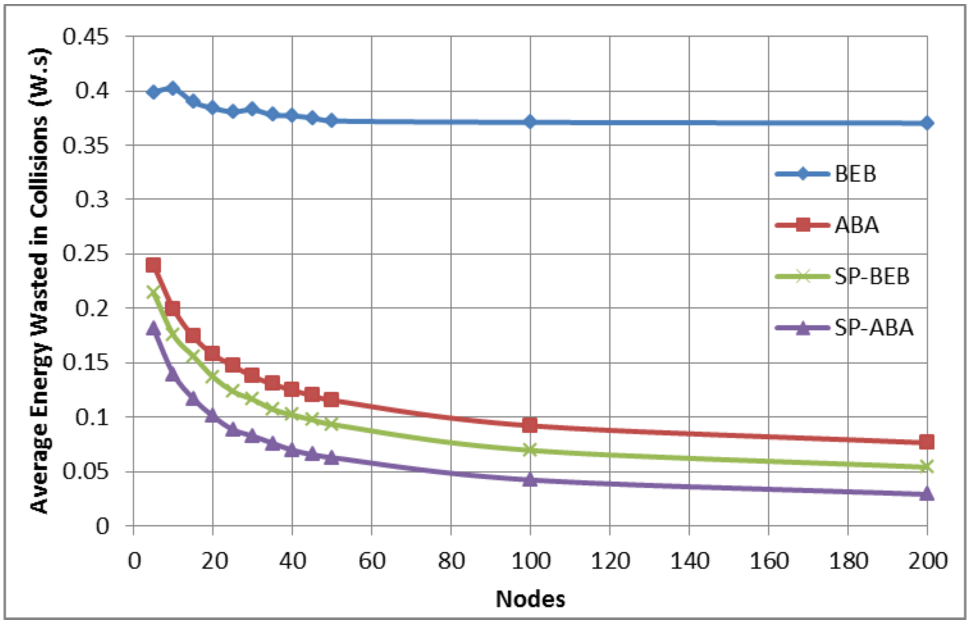


Figure 5-13: Average Energy Wasted in Collisions of BEB, ABA, SP-BEB, and SP-ABA.

5.7 Discussion

The collected simulation results clearly prove that the use of the semi-persistency concept can lead to important enhancements in the performance. We have noticed that we are able to boost the channel utilization and transmission reliability. Furthermore, we have observed significant reductions in the transmission delay, channel collision time, energy consumption, and energy wastage due to collisions.

As we can see from Table 5-1, the integration of the SP-MAC into existing protocols leads to significant gain in the performance especially for networks with a large number of sensor nodes.

For example, for a network size of 200 nodes, the probability of collision is reduced by 27% for the binary exponential backoff protocol, and by 37% for SP-ABA. Likewise, the throughput of ABA is significantly boosted by 85%. We note that ABA protocol is one of the leading MAC protocols in terms of channel utilization. As mentioned in section 5.6.7, the increase in energy consumption of SP-ABA compared to ABA protocol is due to the fact that we include energy during reception in the calculation of the total energy. All other metrics show a significant improvement of the performance.

Table 5-1: Performance Gain of SP-MAC for N=200

Performance Metric	**SP-BEB**	**SP-ABA**
Probability of collision	-27.45%	-37.85%
Channel Utilization	-	85.63%
Channel Idle Time	25.14%	41.10%
Channel Collision Time	-51.43%	-54.71%
Transmission Delay	-99.75%	-45.29%
Reliability	-	4.76%
Energy Consumption	-18.32%	21.78%
Energy Collisions	-85.43%	-62.01%

We should mention that the effect of introducing more CCAs (beyond the standard 2) is apparent in the increases of the channel idle time. In other words, the nodes are forced to spend more time in the clear channel assessment phase before being able to access the medium for packet transmission. The increases in this idle time are beneficial as they reduce the stress on the communication channel, which reduces the probability of packet collisions.

We have already demonstrated in Chapter 4 that increasing the number of CCAs does not affect the fairness of the system. This means that all the nodes still have an equal opportunity to access the medium under both SP-BEB and SP-ABA.

5.8 Conclusion

In this chapter, we proposed a hybrid MAC protocol for beacon-enabled 802.15.4 based wireless sensor networks that incorporates some aspects of 802.11 MAC into the operation of 802.15.4 MAC. With SP-MAC, the sensor nodes can alternate between backoff periods and listening periods, which provides a new opportunity to save energy during backoff. While in listening mode, a node can avoid collisions with other nodes that are sending their packets. Our simulations have shown that the semi-persistency feature can lead to significant enhancements in the performance of both the standard Binary Exponential Backoff (BEB) and the Adaptive Backoff Algorithm (ABA). We provided simulation results to show that the proposed semi-persistent CSMA-CA can greatly improve performance over standard 802.15.4 MAC protocols with binary exponential backoff and adaptive backoff algorithm, in terms of probability of collision, channel utilization, energy consumption, delay, and transmission reliability.

Chapter 6

Conclusions and Future Research

6.1 Concluding Remarks

In this thesis, we thoroughly examined how to exploit the Clear Channel Assessment (CCA) feature of the IEEE 802.15.4 MAC protocol to enhance its performance. The IEEE 802.15.4 is the de facto standard for current Wireless Sensor Networks (WSNs), and it defines the specifications for the MAC sub-layer that conform to the specific characteristics of these networks. While it has been highly praised for its features and performance, several drawbacks of this standard have been highlighted in the literature; thus, we proposed enhancements to the design of the standard MAC protocol.

After surveying the literature in Chapter 2, in order to better understand the methodologies used in the research community to enhance the standard MAC functionality, we found that few studies addressed the CCA feature, and this prompted us to explore new opportunities to modify the standard and boost the performance of the WSN.

Our achievements in this thesis can be summarized as follows: In Chapter 3 we investigated the impact of increasing the number of CCAs to more than two, as specified by the standard. This increase was of a static nature; that is, the nodes in the network were configured with a fixed number of CCAs higher than two. The aim of this configuration was to understand the consequences on the overall performance of the network, and to prepare a mathematical model to describe the overall system based on a Markov chain. We designed a Variable CCA MAC protocol that takes advantage of both 802.11 and 802.11.5 CSMA/CA models. Based on Markov Chain modeling, we derived the theoretical expressions of the performance metrics, namely the probability of collision, channel utilization, channel idle time, channel collision time, reliability, energy consumption and transmission delay. All these performance metrics are essential to the operation of a WSN network.

In Chapter 4, we validated our Markov-based model against the performance of the network, as projected by a WSN MAC simulator. The mathematical model not only showed a perfect match with the simulation-based performance; it also proved that our designed MAC simulator worked correctly and accurately, and that it can used to conduct performance analysis of WSN MAC protocols reliably.

In Chapter 5, we analyzed the performance of the Variable CCA MAC by studying the performance metrics of three settings of the maximum value of the contention window: CCA=2, CCA=5 and CCA=8. As expected, the performance metrics degrade as the value of the maximum contention window is increased. This can be explained by the fact that the Variable CCA protocol sets the contention window to a fixed value; thus, the expression of the probability of collision remains constant and does not change. However, the extra clear channel assessments that were conducted add to the latency of the system, and degrade overall performance metrics such as throughput, energy consumption and reliability. Based on the conclusions from our mathematical modeling and simulation studies, we designed a semi-persistent MAC protocol (SP-MAC) that sets the number of CCAs to be conducted before transmitting a frame as a random value, between two and the maximum value CCA_MAX. According to this protocol, each node will generate a different number of CCAs based on a uniform distribution. We developed the analytical expression of the probability of collision of the SP-MAC protocol, and proved mathematically that such a system can effectively reduce the probability of collision over the communication medium; as the level of collisions decreases, the performance improves. The new MAC protocol can be considered a *hybrid* protocol, because in addition to being solely based on IEEE 802.15.4, it adopts some of the functionality of the IEEE 802.11 MAC protocol which, in particular, is classified as a persistent protocol in which nodes continue to listen to the wireless medium while decreasing their backoff counters. Conversely, the IEEE 802.15.4 MAC is classified as a non-persistent protocol; as the nodes do not listen to the medium before their backoff counters expire.

Our new protocol can be considered *semi*-persistent, since the nodes do not listen to the medium during their backoff. Instead, they conduct multiple CCAs (which can be more than two) and do not proceed to the next CCA before ensuring that the medium is idle. We conducted extensive simulations to examine the performance of the new hybrid MAC protocol compared to the standard binary exponential backoff protocol and the adaptive backoff algorithm, a leading protocol that has been proven in previous studies to outperform most of the published MAC protocols for wireless sensor networks. We focused on the following performance metrics: channel utilization, total energy consumption, energy consumption due to collisions, channel idle time, channel collision time, delay and reliability. We proved that incorporating the semi-persistent feature into the operation of the target MAC protocol can boost the performance of the underlying MAC protocol, particularly for dense WSN networks with a large number of sensor nodes. We demonstrated that the probability of collision was reduced by 27% for BEB and 37% for ABA. The channel utilization of the ABA protocol was boosted by 85%, reaching 55% for a network of 200 nodes. This result is unmatched in other studies that focused on enhancing the performance of the WSN MAC protocol.

6.2 Future Research

We believe the following areas are promising for future research:

1. Incorporate the semi-persistent MAC into the operation of other MAC protocols for wireless sensor networks, to study the effect of the semi-persistency feature on the performance of the targeted protocols.
2. Build the semi-persistent protocol on sensor platforms to collect the experimental results to facilitate compare them with the theoretical and simulation result sets. Building a test-bed of these sensors to conduct various research studies is a challenging undertaking.
3. Our work assumed an error-free wireless channel, but in reality the wireless medium is inherently noisy. It could be fruitful to study the effect of noise on the functionality of our proposed solution, and its ability to maintain acceptable performance under these conditions.
4. Study the performance of the designed protocol under acknowledged and non-saturated traffic conditions, which is a more realistic scenario for many WSN-based applications. The analytical model for unsaturated of traffic would be based on a stochastic arrival process.
5. Differentiate services in the CAP of 802.15.4 by providing different priority to data flows, in order to guarantee certain levels of performance for each type of traffic. Higher priority streams would be configured to perform a small number of CCAs, while low priority streams would perform a greater number of CCAs.

Bibliography

[ABD12] N. Abdeddaim, F. Theoleyre, M. Heusse, A. Duda, "Adaptive IEEE 802.15.4 MAC for Throughput and Energy Optimization," In the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 223-230, 20-23 May, 2013.

[AKY02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," Computer Networks, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[ANA11] G. Anastasi, M. Conti, and M. D. Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," IEEE Transactions on Industrial Informatics, vol. 7, no. 1, pp. 52-65, Feb. 2011.

[BAL12] C. Balarengadurai and S. Saraswathi, "Detection of exhaustion attacks over IEEE 802.15.4 MAC layer using fuzzy logic system," In the 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 527-532, Nov. 2012.

[BAN14] J. I. Bangash, A. H. Abdullah, M. H. Anisi, and A. W. Khan, "A Survey of Routing Protocols in Wireless Body Sensor Networks", Sensors 14, no. 1: 1322-1357, 2014.

[BAR07] P. Baronti, P. Pillaia, V. Chooka, S. Chessa, A. Gotta, and Y. Hua, "Wireless Sensor Networks: A Survey on the State of the Art and the 802.15.4 and ZigBee Standards," Computer Communications, vol.30, no.7, pp. 1655–1695, May 2007.

[CAM11] C. E-A Campbell, K-K Loo, H. A. Kurdi, S. Khan, "Comparison of IEEE 802.11 and IEEE 802.15.4 for Future Green Multichannel Multi-radio Wireless Sensor Networks," International Journal of Communication Networks and Information Security (IJCNIS), vol. 3, no. 1, pp. 96-103, Apr. 2011.

[CAS10] E. Casilari, J. Hurtado-Dueñas, and J.M. Cano-García, "A Study of Policies for Beacon Scheduling in 802.15.4 Cluster-Tree Networks," in the Proceedings of the 9th WSEAS International Conference on Applied Computer Science, pp. 124-129, Hangzhou, China, Apr. 2010.

[DEP11] R. de Paz Alberola and D. Pesch, "Duty Cycle Learning Algorithm (DCLA) for 802.15.4 Beacon-Enabled Wireless Sensor Networks," Elsevier Ad Hoc Networks, doi:10.1016/j.adhoc.2011.06.006, pp. 217–232, available online: Jul. 7th, 2011.

[DHA05] M. Dhanaraj, B.S. Manoj, C.S.R. Murthy, "A New Energy Efficient Protocol for Minimizing Multi-Hop Latency in Wireless Sensor Networks", Third IEEE International Conference on Pervasive Computing and Communications, PerCom 2005, , pp: 117 - 126, 2005.

[FAR13] M. O. Farooq and T. Kunz "BEAR: Bandwidth estimation-based admission control and routing for IEEE 802.15.4-based networks," In the 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC), pp. 1-4, Apr. 2013.

[FRA11] M. D. Francesco, G. Anastasi, M. Conti, S. K. Das, and V. Neri, "Reliability and Energy-Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach," IEEE Journal on Selected Areas in Communictaions, vol. 29, no. 8, pp. 1508 –1524, Sept. 2011.

[FRA14] A. Fragkiadakis, V. Angelakis, and E. Z. Tragos, “Securing Cognitive Wireless Sensor Networks: A Survey,” International Journal of Distributed Sensor Networks, vol. 2014, 2014.

[GHO13] K. El Gholami, N. Elkamoun and K. M. Hou “FF-MAC : Fast forward IEEE 802.15.4 MAC protocol for real-time data transmission,” In the International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 4, No. 1, pp. 25-32, 2013.

[GIL11] M. H. S. Gilani, I. Sarrafi, and M. Abbaspour, “An adaptive CSMA/TDMA hybrid MAC for energy and throughput improvement of wireless sensor networks,” Elsevier Ad Hoc Networks, doi:10.1016/j.adhoc.2011.01.005, pp. 1-8, available online: Jan. 11th, 2011.

[GUE14] M. Guennoun and H. T. Mouftah, “Semi-Persistent CSMA/CA for Efficient and Reliable Communication in Wireless Sensor Networks”, Proc. of 2014 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE2014), pp CN-3.4.1-CN-3.4.6, Toronto, Canada, May 4-May 7, 2014.

[GUP14] M. Gupta, C.R. Krishna, D. Prasad, SEEDS: Scalable Energy Efficient Deployment Scheme for homogeneous Wireless Sensor Network", 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Page(s): 416 - 423, 2014.

[HA07] J. Y. Ha, T. H. Kim, H. S. Park, S. Choi, and W. H. Kwon, “An Enhanced CSMA-CA Algorithm for IEEE 802.15.4 LR-WPANs,” IEEE Communications Letters, vol. 11, no. 5, pp. 461-463, May 2007.

[HUA08] 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, vol. 19, no. 5, pp. 641-651, May 2008.

[HUA13] D. C. Huang, Y.-W. Lee, and H.-W. Wu “An efficient GTS allocation scheme for IEEE 802.15.4 MAC layer,” In Comput. Sci. Inf. Syst., Vol. 10, No. 2, pp. 651-665, 2013.

[JAR10] S. Jardosh, P. Ranjan, D. Rawal, “Prioritized IEEE 802.15.4 for Wireless Sensor Networks,” Proceedings of the 6th Conference on Wireless Advanced (WiAD’10), pp. 1-7, London, UK, Jun. 2010.

[JAR13] S. Jardosh and P. Ranjan “EPCAP: Explicit prioritized channel access protocol for IEEE 802.15.4 based wireless sensor networks,” In the IEEE Wireless Communications and Networking Conference (WCNC), pp.1-6, 7-10 Apr., 2013.

[JIN11] H. Jing and H. Aida, “An Analytical Approach to Optimization of Throughput for IEEE 802.15.4 Slotted CSMA/CA Networks,” Proceedings of the 8th Annual IEEE Consumer Communications and Networking Conference (CCNC’11), pp. 1021-1025, Las Vegas, Nevada, USA, Jan. 2011.

[JOH14] B. Johny, A. Anpalagan,Body Area Sensor Networks: Requirements, Operations, and Challenges, IEEE Potentials, Volume: 33 , Issue: 2, pp. 21 - 25, 2014.

[JAI84] R. Jain, D. Chiu, and W. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems”, DEC-TR-301, Sept. 1984.

[JUN08] B. H. Jung, J. W. Chong, C. Y. Jung, S. M. Kim, and D. K. Sung, “Interference Mediation for Coexistence of WLAN and ZigBee Networks,” in Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC’08), pp. 1-5, Cannes, France, Sept. 2008.

[KHA11a] M. Khanafer, M. Guennoun, H. T. Mouftah, “Priority-Based CCA Periods for Efficient and Reliable Communications in Wireless Sensor Networks,” Wireless Sensor Network, vol. 4, no. 2, pp. 45-51, Feb. 2011.

[KHA11b] M. Khanafer, M. Guennoun, H. T. Mouftah, “An Efficient Adaptive Backoff Algorithm for Wireless Sensor Networks,” Proceedings of IEEE Global Communications Conference (GLOBECOM ’11), pp. 1-6, Houston, Texas, USA, Dec. 2011.

[KHA11c] M. Khanafer, M. Guennoun, H. T. Mouftah, “Adaptive Sleeping Periods in Slotted IEEE 802.15.4 for Efficient Energy Savings: Markov-Based Theoretical Analysis,” Proceedings of IEEE International Conference on Communications (ICC ’11), pp. 1-6, Kyoto, Japan, Jun. 2011.

[KHA11d] M. Khanafer, M. Guennoun, H. T. Mouftah, “Extending Beacon-Enabled IEEE 802.15.4 to achieve Efficient Energy Savings: Simulation-Based Performance Analysis,” Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security Program (NTMS ’11), pp. 1-5, Paris, France, Feb. 2011.

[KHA12] M. Khanafer, “Design of Efficient MAC Protocols for IEEE 802.15.4-based Wireless Sensor Networks,” PhD thesis, University of Ottawa, Ottawa, Ontario, Canada, 2012.

[KIM06] T. H. Kim and S. Choi, “Priority-Based Delay Mitigation for Event-Monitoring IEEE 802.15.4 LR-WPANs,” IEEE Communications Letters, vol. 10, no. 3, pp. 213-215, Mar. 2006.

[KHA13] M. Khanafer, M. Guennoun and H. T. Mouftah, “A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks”, IEEE Communication Surveys and Tutorials, Volume 16, Issue 2, pp 856-876, December 2013.

[KIM10] M. Kim and C.-H. Kang, “Priority-Based Service-Differentiation Scheme for IEEE 802.15.4 Sensor Networks in Nonsaturation Environments,” IEEE Transactions on Vehicular Technology, vol. 59, no. 7, pp. 3523-3535, Sept. 2010.

[KIM12] W. Kim,“Short Clear Channel Assessment in Slotted IEEE 802.15.4 Networks,” Wireless Personal Communications, http://link.springer.com/article/10.1007/s11277-012-0841-x., Sept. 2012.

[KHA13] Mounib Khanafer, Mouhcine Guennoun, Hussein T. Mouftah, A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks, IEEE Communication Surveys and Tutorials, Issue 99, December 2013.

[KOU06] A. Koubâa, M. Alves, and E. Tovar, “i-GAME: an Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks,” in

Proceedings of the 18th Euromicro Conference on Real-Time Systems (ECRTS'06), pp. 183-192, Dresden, Germany, Jul. 2006.

[KOU08] A. Koubâa, A. Cunha, M. Alves, and E. Tovar, "TDBS: A Time Division Beacon Scheduling Mechanism for ZigBee Cluster-Tree Wireless Sensor Networks," Real-Time Systems, vol. 40, no. 3, pp. 321-354, 2008.

[KOU09] A. Koubâa, R. Severino, M. Alves, and E. Tovar, "Improving Quality-of-Service in Wireless Sensor Networks by Mitigating "Hidden-Node Collisions"," IEEE Transactions on Industrial Informatics, vol. 5, no. 3, pp. 299-313, Aug. 2009.

[LEE10] B.-H. Lee, R.-L. Lai, H.-K. Wu, and C.-M. Wong, "Study on Additional Carrier Sensing for IEEE 802.15.4 Wireless Sensor Networks," Sensors, vol. 10, no. 7, pp. 6275–6289, Jun. 2010.

[LEE12] B.-H. Lee, M. U. H. Al Rasyid, and H.-K. Wu, "Analysis of superframe adjustment and beacon transmission for IEEE 802.15.4 cluster tree networks," EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 219, pp. 1-12, Jul. 2012.

[LI11] X. Li, C. J. Bleakley, and W. Bober, "Enhanced Beacon-Enabled Mode for Improved 802.15.4 Low Data Rate Performance," Wireless Networks, Online First, published online: Sept. 10th, 2011.

[MAR10] P. D. Marco, P. Park, C. Fischione, and K. H. Johansson, "TREnD: a Timely, Reliable, Energy-efficient and Dynamic WSN Protocol for Control Applications," in Proceedings of the IEEE International Conference on Communications (ICC'10), pp. 1-6, Cape Town, South Africa, May 2010.

[MOR11] K. Mori, K. Naito, and H. Kobayashi, "Distributed Backoff Mechanism for Traffic Adaptive Active Period Control in Cluster-based IEEE 802.15.4 WSNs," Proceedings of IEEE 73rd Vehicular Technology Conference (VTC 2011-Spring), pp. 1-5, Budapest, Hungary, May 2011.

[MOR14] V. Mor, H. Kumar, "Energy efficient wireless mobile networks: A review", 2014 International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), Page(s): 281 - 285, 2014.

[MUT09] P. S. Muthukumaran, R. de Paz, R. Špinar, and D. Pesch, "MeshMAC: Enabling Mesh Networking over IEEE802.15.4 Through Distributed Beacon Scheduling", In J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 561–575, 2009.

[NA08] C. Na, Y. Yang, and A. Mishra, "An Optimal GTS Scheduling Algorithm for Time-Sensitive Transactions in IEEE 802.15.4 Networks," Computer Networks, vol. 52, no. 13, pp. 2543–2557, Sept. 2008.

[NDI09] E. D. N. Ndih, N. Khaled, G. D. Micheli, "An Analytical Model for the Contention Access Period of the Slotted IEEE 802.15.4 with Service Differentiation," in Proceedings of IEEE International Conference on Communications (ICC'09), pp. 1-6, Dresden, Germany, Jun. 2009.

[NEU05] M. Neugebauer, J. Plonnigs, and Klaus Kabitzsch, "A New Beacon Order Adaptation Algorithm for IEEE 802.15.4 Networks," in Proceedings of the 2nd European

Workshop on Wireless Sensor Networks (EWSN'05), pp. 302-311, Istanbul, Turkey, Jan. 2005.

[PAR10] P. Park, P. D. Marco, C. Fischione, and K. H. Johansson, "Adaptive IEEE 802.15.4 Protocol for Reliable and Timely Communications," in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'10), pp. 327-338, Stockholm, Sweden, Apr. 2010.

[POL08] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. Van der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," IEEE Transactions on Wireless Communications, vol. 7, no. 9, pp. 3359-3371, Sept. 2008.

[RAW14] P. Rawat, K. Deep, H. Chaouchi, J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies", The Journal of Supercomputing, Volume 68, Issue 1 , pp 1-48, April 2014.

[RAM06] I. Ramachandran and S. Roy, "On the Impact of Clear Channel Assessment on MAC Performance," in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'06), pp. 1–5, San Francisco, California, USA, Nov. 2006.

[RAM07] I. Ramachandran and S. Roy, "Clear Channel Assessment in Energy-Constrained Wideband Wireless Networks," IEEE Wireless Communications Magazine, vol. 14, no. 3, pp. 70-78, Jun. 2007.

[RAY06]M. Raya, I. Aad, J.-P. Hubaux, and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behaviour in IEEE 802.11 Hotspots," IEEE Transactions on Mobile Computing, vol. 5, no. 12, pp. 1691-1705, Dec. 2006.

[ROY10] F. Royo, T. Olivares, M. Lopez-Guerrero, L. Orozco-Barbosa, and A. M. Ortiz, "Enhancing Collision Resolution for Large-Scale IEEE 802.15.4 Networks," Proceedings of the 3rd Joint IFIP Wireless and Mobile Networking Conference (WMNC'10), pp. 1-6, Budapest, Hungary, Oct. 2010.

[SER10] P. Serrano, A. Banchs, V. Targon, and J. F. Kukielka, "Detecting Selfish Configurations in 802.11 WLANs," IEEE Communications Letters, vol. 14, no. 2, pp. 142-144, Feb. 2010.

[SEV10] R. Severino, M. Batsa, M. Alves, and A. Koubaa, "A Traffic Differentiation Add-On to the IEEE 802.15.4 Protocol: Implementation and Experimental Validation over a Real-Time Operating System," Proceedings of 13th Euromicro Conference on Digital System Design: Architectures, Methods, and Tools (DSD'10), pp. 501-508, Lille, France, Sept. 2010.

[SHE09] S.-T. Sheu, Y.-Y. Shih, and W.-T. Lee, "CSMA/CF Protocol for IEEE 802.15.4 WPANs," IEEE Transactions on Vehicular Technology, vol. 58, no. 3, pp. 1501-1516, Mar. 2009.

[SHI07] S. Y. Shin, I. Ramachandran, S. Roy, and W. H. Kwon, "Cascaded Clear Channel Assessment: Enhanced Carrier Sensing for Dynamic Spectrum Access," in the Proceedings of the IEEE International Conference on Communications (ICC'07), pp. 6532-653, Glasgow, Scotland, Jun. 2007.

[SHI11] A. Shi, G. Tan, G. Chen, and L. Xu, "An Improved CSMA/CA Protocol for Real-Time Abnormal Events Monitoring," Journal of Computational Information Systems, vol. 7, no. 9, pp. 3299-3308, Sept. 2011.

[SHI13] S.Y. Shin, "A Novel Method for Service Differentiation in IEEE 802.15.4: Priority Jamming," International Journal of Computers, Communications and Control, vol. 8, no. 1, pp. 127-135, Feb. 2013.

[SHR10] B. Shrestha, E. Hossain, S. Camorlinga, R. Krishnamoorthy, and D. Niyato, "An Optimization-Based GTS Allocation Scheme for IEEE 802.15.4 MAC with Application to Wireless Body-Area Sensor Networks," in Proceedings of 2010 IEEE Conference on Communications (ICC'10), pp. 1-6, Cape Town, South Africa, May 2010.

[SIN14] D.P. Singh, V. Bhateja, S.K. Soni, "Prolonging the lifetime of wireless sensor networks using prediction based data reduction scheme", International Conference on Signal Processing and Integrated Networks (SPIN), pp: 420 - 425, 2014.

[SUN13] Q. Sun, X. You, D. Xie, and Y. Qiu "Modified beacon-enabled IEEE 802.15.4 MAC for energy critical and bandwidth," In the Journal of Computational Information Systems Vol. 9, No. 4, pp. 1483-1491, 2013.

[SUT98] R.S. Sutton and A.G. Barto, "Reinforcement Learning: an Introduction", Artificial Intelligence, The MIT Press, Cambridge, Massachusetts, 1998.

[TAK10] M. Takaffoli, E. Elmallah, and W. Moussa, "Scheduled Access Using the IEEE 802.15.4 Guaranteed Time Slots," in Proceedings of IEEE International Conference on Communications (ICC'10), pp. 1-5, Cape Town, South Africa, May 2010.

[TUM14] A.E. Tumer, M. Gunduz, "An improved leach protocol for indoor wireless sensor networks", International Conference on Signal Processing and Integrated Networks (SPIN), 2014, Page(s): 432 - 437, 2014.

[TSE11] H.-W. Tseng, S.-C. Yang, P.-C. Yeh, and A.-C. Pang, "A Cross-Layer Scheme for Solving Hidden Device Problem in IEEE 802.15.4 Wireless Sensor Networks," IEEE Sensors Journal, vol. 11, no. 2, pp. 493-504, Feb. 2011.

[TYT12] L. Tytgat, O. Yaron, S. Pollin, I. Moerman, and Piet Demeester, "Avoiding collisions between IEEE 802.11 and IEEE through coexistence aware clear channel assessment," Journal on Wireless Communications and Networking (EURASIP), vol. 2012, no. 1, pp. 137-152 , Apr. 2012.

[VAL10] M. Valero, A. Bourgeois, and R. Beyah, "DEEP: A Deployable Energy Efficient 802.15.4 MAC Protocol for Sensor Networks," Proceedings of IEEE International Conference on Communications (ICC'10), pp. 1-6, Cape Town, South Africa, May 2010.

[WAN11] F. Wang, D. Li, and Y. Zhao, "On Analysis of the Contention Access Period of IEEE 802.15.4 MAC and its Improvement," Wireless Personal Communications, Online First, published online: Apr. 2011.

[WON10a] C.-M. Wong and W.-P. Hsu, “An Additional Clear Channel Assessment for IEEE 802.15.4 Slotted CSMA/CA Networks,” Proceedings of IEEE International Conference on Communication Systems (ICCS’10), pp. 62-66, Singapore, Nov. 2010.

[WOO08] S. Woo, W. Park, S.Y. Ahn, S. An, and D. Kim, “Knowledge-Based Exponential Backoff Scheme in IEEE 802.15.4 MAC,” Lecture Notes in Computer Science (LNCS), vol. 5200, pp. 435-444, 2008.

[XIA11] F. Xia, R. Hao, Y. Cao, and L. Xue, “A Survey of Adaptive and Real-Time Protocols Based on IEEE 802.15.4,” International Journal of Distributed Sensor Networks, vol. 2011, p. 11, Sep. 2011.

[YED08] K. Yedavalli and B. Krishnamachari, “Enhancement of the IEEE 802.15.4 MAC Protocol for Scalable Data Collection in Dense Sensor Networks,” in Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT’08), pp. 152–161, Berlin, Germany, Apr. 2008.

[YAN14] Q. Yang, S. He, J. Li, J. Chen, Y. Sun, "Energy-Efficient Probabilistic Area Coverage in Wireless Sensor Networks", IEEE Transactions on Vehicular Technology, Volume:PP , Issue: 99, January 2014.

[YEN08] L.-H. Yen, Y. W. Law, and M Palaniswami, “Risk-Aware Beacon Scheduling for Tree-based ZigBee/IEEE 802.15.4 Wireless Networks,” in Proceeding of the 4th Annual International Conference on Wireless Internet (WICON’08), no. 79, Maui, Hawaii, Nov. 2008.

[YUA10a] W. Yuan, X. Cui, and I. G. M. M. Niemegeers, “Distributed Adaptive Interference-Avoidance Multi-channel MAC Protocol for ZigBee Networks,” in Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT’10), pp. 415-419, Bradford, West Yorkshire, UK, Jun. 2010.

[YUA10b] W. Yuan, J.-P. M. G. Linnartz, and I. G. M. M. Niemegeers, “Adaptive CCA for IEEE 802.15.4 Wireless Sensor Networks to Mitigate Interference,” in Proceedings of the IEEE 2010 Wireless Communications and Networking Conference (WCNC’10), pp. 1-5, Sydney, Australia, Apr. 2010.

[ZHA10] X. Zhao, W. Zhang, W. Niu, Y. Zhang, and L. Zhao, “Power and Bandwidth Efficiency of IEEE 802.15.4 Wireless Sensor Networks,” Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing (UIC’10), pp. 243-251, Xi’an, China, Oct. 2010.

[ZHU11a] J. Zhu, Z. Tao, and C. Lv, “Delay Analysis for IEEE 802.15.4 CSMA/CA Scheme with Heterogeneous Buffered Traffic,” in Proceedings of the 3rdInternational Conference on Measuring Technology and Mechatronics Automation (ICMTMA’11), vol. 1, pp. 834-845, Shangshai, China, Jan. 2011.

[ZHU11b] J. Zhu, Z. Tao, and C. Lv, “Performance Evaluation of IEEE 802.15.4 CSMA/CA Scheme Adopting a Modified LIB Model,” Wireless Personal Communications, Online First: http://wwW.springerlink.com/content/712431107jwx1h66/fulltext.pdf, pp. 1-27, Jan. 2011.

[ZIG06] IEEE Std 802.15.4-2006, September, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Available online at: http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf (Last accessed in May 2014).

Appendix A

Theoretical Model of Variable CCA

```
% Theoretical model of Variable CCA protocol with CCA=8
function x = V_CCA_8
global L;
global Wmax;
global N;
global m;
global maxBE;
global minBE;
Ts=0.32; %0.32ms

L = 14;
m = 4;
n = 3;
maxBE = 8;
minBE = 4;
Wmax = 2^maxBE;
nCCA = 8;
i = 0;
Pbo  = 0.8;     % average power consumption (in mW) during "idle-listen" state
Pcca = 40;      % average power consumption (in mW) during "channel sensing"
state (it is referred to as Psc in Park's paper)
Pt   = 30;      % average power consumption (in mW) during "transmit" state
Pr   = 40;      % average power consumption (in mW) during "receive" state

for N = [5 10 15 20 25 30 35 40 45 50 100 200]

i = i + 1;

NN(i) = N;

        x0 = [0.001; 0.001; 0.001; 0.001; 0.001; 0.001; 0.001; 0.001; 0.001];
        % x0 = [a1 ; a2 ; a3 ; a4 ; a5 ; a6 ; a7; a8 ; t]
        x = fsolve(@V_CCA_8_CCA_fun,x0);

        a1 = x(1);
        a2 = x(2);
        a3 = x(3);
        a4 = x(4);
        a5 = x(5);
        a6 = x(6);
        a7 = x(7);
        a8 = x(8);
        t = x(9);
```

```
k = 0;
while(a1 < 0 || a1 > 1 || a2 < 0 || a2 > 1 || t < 0 || t > 1 || a3 < 0 || a3
> 1 || a4 < 0 || a4 > 1 || a5 < 0 || a5 > 1 || a6 < 0 || a6 > 1 || a7 < 0 ||
a7 > 1 || a8 < 0 || a8 > 1)
        k = k + 0.01;
        x0 = [0.001+k; 0.001+k; 0.001+k; 0.001+k; 0.001+k; 0.001+k; 0.001+k;
0.001+k; 0.001+k]; % x0 = [a1 ; a2 ; a3 ; a4; a5; t]
        x = fsolve(@V_CCA_8_CCA_fun,x0);

        a1 = x(1);
        a2 = x(2);
        a3 = x(3);
        a4 = x(4);
        a5 = x(5);
        a6 = x(6);
        a7 = x(7);
        a8 = x(8);
        t = x(9);
end

b11 = t;
Pc = 1-(1-t)^(N-1);
tau(i)=t;
Pcol(i)=Pc;

%-----Channel Utilization-------------------------------------
U(i)=NN(i)*L*t*((1-t)^(NN(i)-1))*(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)*(1-
a6)*(1-a7)*(1-a8);


%-----------------------------------------------------------
%Idle time

I(i) = 1-a1;

%-----Collision Time ---------------------------------------

C(i)= 1-U(i)-I(i);

%-----Delay--------------------------------------------------

T_CCA(i) = 0;
for ii=1:nCCA
    ss1=0;
    for kk=ii:nCCA
        pp1=1;
        for rr=ii:kk-1
            pp1=pp1*(1-x(rr));
        end
        ss1=ss1+x(kk)*pp1;
    end
    ppp1=1;
    for jjj=1:ii-1
        sss1=0;
```

```
        for kkk=jjj:nCCA
            pppp1=1;
            for rrr=jjj:kkk-1
                pppp1=pppp1*(1-x(rrr));
            end
            sss1=sss1+x(kkk)*pppp1;
        end
        ppp1=ppp1*(1-x(jjj)/sss1);
    end
    T_CCA(i)=T_CCA(i)+ii*x(ii)*ppp1/ss1;
end

PI_BC=0;
for ii=1:nCCA
    pp1=1;
    for jj=1:ii-1
        pp1=pp1*(1-x(jj));
    end
    PI_BC=PI_BC+x(ii)*pp1;
end

PI_BC


T_BO(i)=Pc*Wmax/2;
PI_CC=1;
for ii=1:nCCA
    PI_CC=PI_CC*(1-x(ii));
end
PI_CC=PI_CC*Pc

PI_SC=1;
for ii=1:nCCA
    PI_SC=PI_SC*(1-x(ii));
end
PI_SC=PI_SC*(1-Pc)

coeff=(1+PI_CC/PI_SC+PI_BC/PI_SC)
D(i)=(1+PI_CC/PI_SC+PI_BC/PI_SC)*T_BO(i)+(PI_BC/PI_SC)*T_CCA(i)+(1+PI_CC/PI_S
C)*(nCCA+L);


%------------------------------------------------------------

%-----Power measurements-----------------------------------

Ebo(i) = Pbo*((Pc*Wmax-1)/2)*b11;

Ecca(i) = Pcca*((2-a1)+(1-a1)*(1-a2)+(1-a1)*(1-a2)*(1-a3)+(1-a1)*(1-a2)*(1-
a3)*(1-a4)+(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)+(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-
a5)*(1-a6)+(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)*(1-a6)*(1-a7))*b11;

Et(i) = Pt*L*(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)*(1-a6)*(1-a7)*(1-a8)*b11;
```

```
Er(i) = Pr*L*(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)*(1-a6)*(1-a7)*(1-a8)*(1-
Pc)*b11;

Etot(i) = (Ts)*(Ebo(i) + Ecca(i) + Et(i) + Er(i));

Ec(i) = (Ts)*(Pt*L*(1-a1)*(1-a2)*(1-a3)*(1-a4)*(1-a5)*(1-a6)*(1-a7)*(1-
a8)*b11*Pc);

%-----Reliability-----------------------------------
x = PI_BC;
y = PI_CC;
R(i) = 1/(1+((1-x)*x^(m+1)/((1-x^(m+1))*(1-x-y))) + (y^(n+1))/((1-x)^(n+1)-
y^(n+1)));
%---------------------------------------------------
% Calculate the probability of collision for SP-MAC
pc(i)=Pc;

end

% -------
% Print peformance metrics
pc'
U'
I'
C'
D'
R'
Etot'
Ec'
```

```
% Solving ncca+1 non linear equations to find the network operating point

function F = V_CCA_8_CCA_fun(x)
global L;
global Wmax;
global N;

F= [
    -x(1)+(1-(1-x(9))^(N-1))*L*(1-x(1))*(1-x(2))*(1-x(3))*(1-x(4))*(1-
x(5))*(1-x(6))*(1-x(7))*(1-x(8));
    -x(2)+(1-(1-x(9))^(N-1))*(1-x(2))*(1-x(3))*(1-x(4))*(1-x(5))*(1-x(6))*(1-
x(7))*(1-x(8));
    -x(3)+(1-(1-x(9))^(N-1))*(1-x(3))*(1-x(4))*(1-x(5))*(1-x(6))*(1-x(7))*(1-
x(8));
    -x(4)+(1-(1-x(9))^(N-1))*(1-x(4))*(1-x(5))*(1-x(6))*(1-x(7))*(1-x(8));
    -x(5)+(1-(1-x(9))^(N-1))*(1-x(5))*(1-x(6))*(1-x(7))*(1-x(8));
    -x(6)+(1-(1-x(9))^(N-1))*(1-x(6))*(1-x(7))*(1-x(8));
    -x(7)+(1-(1-x(9))^(N-1))*(1-x(7))*(1-x(8));
    -x(8)+(1-(1-x(9))^(N-1))*(1-x(8));
    -x(9)*((1-(1-x(9))^(N-1))*Wmax+1+2*L*(1-x(1))*(1-x(2))*(1-x(3))*(1-
x(4))*(1-x(5)*(1-x(6))*(1-x(7))*(1-x(8)))+2*((1-x(1))+(1-x(1))*(1-x(2))+(1-
x(1))*(1-x(2))*(1-x(3))+(1-x(1))*(1-x(2))*(1-x(3))*(1-x(4))+(1-x(1))*(1-
x(2))*(1-x(3))*(1-x(4))*(1-x(5))+(1-x(1))*(1-x(2))*(1-x(3))*(1-x(4))*(1-
x(5))*(1-x(6))+(1-x(1))*(1-x(2))*(1-x(3))*(1-x(4))*(1-x(5))*(1-x(6))*(1-
x(7))))+2;
   ];
```

Appendix B

Simulation Module of SP-MAC

```
/*@ Mouhcine Guennoun
  @ Simulation of the CSMA/CA module for SP-MAC
*/
#include <iostream>
using namespace std;
#include <cmath>
#include <ctime>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//We assume that the queues are always full (saturated mode)
//but we can use a stochastic process to model packet arrivals

int CSMA_MOD;
int CCA_MAX;
#undef  DEBUG
//#define DEBUG

#define MAX_DELAY 256
#define INT64 __int64
#define alpha 0.5
#define beta  0.8
#define MAX   2000  //MAX number of nodes in the network
#define L 14       //Length of packets in time slots
#define Lack  2
#define CW_TR 16
#define ACK   0
#define QUEUE 0
#if  ACK
  #define CW_ACK 1  //CW for ack frames
  #define ACK_TIMEOUT (Lack+CW_ACK) //Time needed to detect a collision
#else
  #define CW_ACK 0  //CW for ack frames
  #define ACK_TIMEOUT 0 //Time needed to detect a collision
#endif
#define MacMinBE  3 //Min Backoff Exponent
#define aMaxBE    8 //Maximum Backoff Exponent
#define macMaxCSMABackoffs  4 //Maximum of Backoffs before frame transmission fails
#define macMaxFrameRetries  3 // Maximum number of retries in case of a collision
#define CCA_Duration  1 //How many slots CCA is performed
#define SIM_TIME  1000000 //The total simulation time


ofstream fileT;
ofstream fileE;
ofstream fileEC;
```

```
ofstream fileR;
ofstream fileF;
ofstream fileS;
ofstream fileC;
ofstream fileI;
ofstream fileA;
ofstream fileD;
ofstream fileDL;
ofstream fileAF;
ofstream fileCF;
ofstream filePC;

INT64  tick;
typedef struct {
  INT64 start; //start time on which the medium is busy
  int   lengthFrame;
  bool  collision; //Current Status of the medium
  bool  idle;
  bool  sending;
  int   timeCollission;
  int   timeSending;
  int   timeIdle;
}medium;

typedef struct{
  int   id;     //The id of the node
  INT64 ST;     //The time when the frame starts being transmitted
  int   NB;     //Number of backoffs
  int   CW;     //Number of CCA
  int   CW_start; //Start Value Number of CCA
  int   BE;     //Backoff exponent
  int   frameLength; //Length of the frame to be transmitted
  int   backoffPeriod;  //Random delay before frame can be transmitted
  int   receiver; //Destination station
  int   retries;
  int   remainingDelayIBEB;
  double  pc;   //Probability of collision
  int         time_generated;
}frame;

void printFrame(frame f){
  cout<<"["<<f.id<<"]"<<"\t";
  cout<<"ST="<<f.ST<<"\t";
  cout<<"NB="<<f.NB<<"\t";
  cout<<"CW="<<f.CW<<"\t";
  cout<<"BE="<<f.BE<<"\t";
  cout<<"BP="<<f.backoffPeriod<<"\t";
  cout<<"RC="<<f.receiver<<"\t";
  cout<<"FL="<<f.frameLength<<endl;
}

void printFrameTable(frame *fT,int NUM_NODES){
  for(int node=0;node<NUM_NODES;node++)
    printFrame(fT[node]);
}
```

```
void printStatus(frame *fT,medium channel,int success,int collision,INT64 tick,int
NUM_NODES,double nodeStats[][MAX]){
  cout<<"Current Time="<<tick<<endl;
  cout<<"Channel Start Busy Time="<<channel.start<<endl;
  cout<<"Length of Current Frame="<<channel.lengthFrame<<endl;
  cout<<"Channel Status=";
  if(channel.collision)
    cout<<"Collision"<<endl;
  if(channel.idle)
    cout<<"Idle"<<endl;
  if(channel.sending)
    cout<<"Transmitting"<<endl;
  printFrameTable(fT,NUM_NODES);
  cout<<"Collision="<<(100.0*channel.timeCollission)/SIM_TIME<<"%
Idle="<<(100.0*channel.timeIdle)/SIM_TIME<<"%
Throughput="<<(100.0*channel.timeSending)/SIM_TIME<<"%"<<endl;
  cout<<"Verification:
"<<channel.timeCollission+channel.timeIdle+channel.timeSending<<endl;
  for(int node=0;node<NUM_NODES;node++){

cout<<"["<<node<<"]="<<nodeStats[0][node]+nodeStats[1][node]+nodeStats[3][node]+nodeStats
[4][node]+nodeStats[5][node]<<endl;
  }
}

int getBackoffPeriod(frame &f,double nodeStats[][MAX]){
  double delay,pc;
  switch(CSMA_MOD){
    case 0: //Original BE with variable cca (SP-MAC Plain)
      return (rand()%(int)(pow(2.0,f.BE)));
    case 1: //Original BE Extended BEB* (priority based)
      return (rand()%(int)(pow(2.0,f.BE)));
    case 2: // SP-ABA with backoff window equal to pc*Wmax
      //probability of collision
      pc=(1.0*nodeStats[1][f.id])/(nodeStats[0][f.id]+nodeStats[1][f.id]);
      pc=beta*f.pc+(1-beta)*pc; //Exponential Average
      f.pc=pc;
      delay=pc*MAX_DELAY;
      return rand()%(1+((int)ceil(delay)));
    default:
      printf("Error CSMA_MOD\n");
      exit(0);
  }
}

void generateACK(frame frameTable[],int node){
  int receiver=frameTable[node].receiver;
  frameTable[receiver].ST=-1;
  frameTable[receiver].frameLength=Lack;
  frameTable[receiver].CW=CW_ACK;
  frameTable[receiver].NB=0;
  frameTable[receiver].BE=MacMinBE;
  frameTable[receiver].backoffPeriod=0; //Since it's an ACK
  frameTable[receiver].receiver=node;
}
```

```
void generateFrame(frame frameTable[],int node,int delay,int NUM_NODES,double
nodeStats[][MAX],int retries=0){
  frameTable[node].ST=-1;
  frameTable[node].frameLength=L;
  double pc=nodeStats[1][node]/(nodeStats[0][node]+nodeStats[1][node]);
  if(CSMA_MOD==1){
    frameTable[node].CW_start=2+(int)((macMaxCSMABackoffs+macMaxFrameRetries)*pc);
    frameTable[node].CW=frameTable[node].CW_start;
  }else
    frameTable[node].CW=2+rand()%(CCA_MAX-2); //SP-MAC
  //if(tick>SIM_TIME-10) cout<<"CW="<<frameTable[node].CW<<"\n";
  frameTable[node].NB=0;
  frameTable[node].retries=retries;
  //Since completed transmission is successful after receiving ACK
  frameTable[node].BE=MacMinBE;
  frameTable[node].backoffPeriod=delay+getBackoffPeriod(frameTable[node],nodeStats);
  do{
    frameTable[node].receiver=rand()%NUM_NODES;
  }while(frameTable[node].receiver==node);
  frameTable[node].time_generated=tick;
}
void rescheduleFrameAfterCollision(frame frameTable[],int node,double nodeStats[][MAX]){
  frameTable[node].ST=-1;
  frameTable[node].frameLength=L;
  if(CSMA_MOD==1){
    //frameTable[node].CW_start--;
    frameTable[node].CW=max(2,frameTable[node].CW);
  }else
    frameTable[node].CW=2+rand()%(CCA_MAX-2);
  frameTable[node].NB=0;
  frameTable[node].retries++;
  frameTable[node].BE=MacMinBE;
  //Since completed transmission is successful after receiving ACK
  frameTable[node].backoffPeriod=getBackoffPeriod(frameTable[node],nodeStats);
}

int runSimulator(int NUM_NODES){
  int success=0,collision=0;
  int  totalCollided=0,nbCollisions=0;
  int nbCollisionDiscarded=0,nbBackoffDiscarded=0;

  tick=0; //The current time of the simulation
  frame frameTable[MAX];
  frame f;
  medium channel;
  /* col 0 Successful Transmission, col 1 collisions, col 2 successful receive,
  col 3 CCA, col 4 idle, col 5 ACK, col 6 Failurs (assuming MaxFrameRetries=1)*/
  double nodeStats[7][MAX]={0};
  double accessFailures[MAX]={0};
  double collisionFailures[MAX]={0};
  double delays[MAX]={0};
  double nb_cycles=0;
  double cycles[MAX][3]={0}; //0 Success 1 Busy 2 Collision
  //Initialize the frame table
  channel.start=-2*L; //So channel will be idle
  channel.collision=false;
  channel.idle=true;
```

```
  channel.sending=false;
  channel.lengthFrame=0;
  channel.timeCollission=0;
  channel.timeIdle=0;
  channel.timeSending=0;
  //Init
  for(int node=0;node<NUM_NODES;node++){
    //Init stations stat
    nodeStats[0][node]=1;
    nodeStats[1][node]=1;
    //Init frame
    f.id=node;
    f.pc=0;
    f.BE_KEB=MacMinBE;
    //Insert frame into frameTable
    frameTable[node]=f;
    generateFrame(frameTable,node,0,NUM_NODES,nodeStats);
  }
  //Start the simulation
  while(tick<SIM_TIME){
    //Service each node
    //First check who needs to transmit
    for(int node=0;node<NUM_NODES;node++){
      //Check if node is not trasmitting
      if((frameTable[node].ST!=-1) && ((frameTable[node].ST+frameTable[node].frameLength-
1)>=tick))
        //Node is transmitting Packet
        continue;
      if(frameTable[node].CW==0){
        //Transmit frame
        channel.start=tick;
        channel.sending=true;
        channel.idle=false;
        if(channel.lengthFrame<frameTable[node].frameLength)
          channel.lengthFrame=frameTable[node].frameLength;
        frameTable[node].ST=tick;
        frameTable[node].CW=20; //so it won't be transmitted again
        //We check for collisions at the end
      }
    }
    //Next do CCA
    for(int node=0;node<NUM_NODES;node++){
      //Check if node is not trasmitting
      if((frameTable[node].ST!=-1) &&
(frameTable[node].ST+frameTable[node].frameLength>=tick))
        //Node is transmitting Packet
        continue;
      //Check delay counter
      if(frameTable[node].backoffPeriod>0){
        frameTable[node].backoffPeriod--;
        nodeStats[4][node]++; //Idle
        continue;
      }
      //CCA
      if(frameTable[node].CW>0){
        //Perform Clear Channel Assessment
        //Increment how many times the node made a CCA
```

```
        nodeStats[3][node]++;
        //counts how many cycles performed by a node
        if(frameTable[node].CW==CCA_MAX) nb_cycles++;
        if(!channel.idle){ //channel is busy
          frameTable[node].NB++;
          cycles[node][1]++; //Busy cycle
          if(frameTable[node].NB<=macMaxCSMABackoffs){
            if(CSMA_MOD==1){
              //frameTable[node].CW_start--;
              frameTable[node].CW=max(2,frameTable[node].CW);
            }else
              frameTable[node].CW=2+rand()%(CCA_MAX-2);
            frameTable[node].BE=min(frameTable[node].BE+1,aMaxBE);
            frameTable[node].backoffPeriod=getBackoffPeriod(frameTable[node],nodeStats);
          }else{
            //Start Over (Failure)
            delays[node]+=1+tick-frameTable[node].time_generated;

generateFrame(frameTable,node,0,NUM_NODES,nodeStats,frameTable[node].retries);
            //Failed to deliver frame discarded due to MAX backoffs
            nodeStats[6][node]+=frameTable[node].frameLength;
            //Number of channel access failures (Reached max backoffs)
            accessFailures[node]++;
            nbBackoffDiscarded++;
          }
        }else{
          frameTable[node].CW--;
        }
      }
    }
    //Check if there is a collision
    for(int node1=0;node1<NUM_NODES;node1++){
      if(frameTable[node1].ST==-1)
        continue; //Node is not transmitting
      for(int node2=node1+1;node2<NUM_NODES;node2++){
        if(frameTable[node1].ST==frameTable[node2].ST){
          channel.collision=true;
          channel.sending=false;
        }
      }
    }
    bool completed=false;
    int nodeT[MAX]={0};
    //Check if a node has finished transmission
    for(int node=0;node<NUM_NODES;node++){
      if((frameTable[node].ST!=-1) && (frameTable[node].ST+frameTable[node].frameLength-
1==tick)){
        //Transmission completed
        completed=true;
        //Keep track of the node who completed a successful transmission
        nodeT[node]=1;
      }
    }
#ifdef DEBUG
    printStatus(frameTable,channel,success,collision,tick,NUM_NODES,nodeStats);
#endif
```

```
if(completed){
  if(channel.collision){
    int check=0;
    collision++;
    for(int node=0;node<NUM_NODES;node++){
      if(nodeT[node]==0)
        continue; //Station not involved
      //Update collision stat for the node
      nodeStats[1][node]+=frameTable[node].frameLength;
      cycles[node][2]++; //Collision cycle
      check++;
      //Reschedule Frame for Node
      rescheduleFrameAfterCollision(frameTable,node,nodeStats);
      if(frameTable[node].retries>macMaxFrameRetries){
        frameTable[node].retries=0;
        //Discarded due to collision
        nodeStats[6][node]+=frameTable[node].frameLength;
        //Number of failures due to max collision retries
        collisionFailures[node]++;
        nbCollisionDiscarded++;
      }
    }
    if(check<2){
      cout<<"Error: Less than two stations were involved in a collision\n";
      exit(0);
    }
    totalCollided+=check;
    nbCollisions++;
  }else{
    //Successful Transmission
    int check=0;
    success++; //Should happen only for one node
    for(int node=0;node<NUM_NODES;node++ ){
      if(nodeT[node]==0) continue;
      check++;
       if(frameTable[node].frameLength==Lack)
        nodeStats[5][node]+=frameTable[node].frameLength; //Ack
      else{
        //Successful transmission
        nodeStats[0][node]+=frameTable[node].frameLength;
        //Successful cycle
        cycles[node][0]++;
      }
      //Succecssful reception
      nodeStats[2][frameTable[node].receiver]+=frameTable[node].frameLength;

      //Generate The ACK if it's not already an ACK
      if(ACK && frameTable[node].frameLength==L){
        //Generate an ack at the receiver side
        generateACK(frameTable, node);
      }
      //If regular packet add delay to receive ACK
      if(ACK && frameTable[node].frameLength==L)
        generateFrame(frameTable,node,CW_ACK+Lack,NUM_NODES,nodeStats);
      else{
        delays[node]+=1+tick-frameTable[node].time_generated;
        //Should be modified when we use queue (since delay will be increased)
```

```
            generateFrame(frameTable,node,0,NUM_NODES,nodeStats);
          }
        }
        if(check!=1){
          cout<<"Error: More than one station transmitted successfully at the same 
time\n";
          exit(0);
        }
      }
    }
    if(channel.collision)
      channel.timeCollission++;
    if(channel.sending)
      channel.timeSending++;
    if(channel.idle)
      channel.timeIdle++;
#ifdef DEBUG
    int timeSending=0;
    for(int node=0;node<NUM_NODES;node++)
      timeSending+=nodeStats[0][node]-1;
    cout<<"Channel Sending Time:"<<channel.timeSending<<endl;
    cout<<"Verification: timeSending="<<timeSending<<endl;
    system("pause");
#endif
    //Update status if completed
    if(completed){
      completed=false;
      channel.idle=true;
      channel.collision=false;
      channel.sending=false;
    }
    tick++; //Increase time by one unit
  }
  cout<<"\n";
  //Calculate the fairness index based on the number of successful transmissions using 
Jain Index
  double sum=0,squareSum=0,fairnessIndex;
  for(int node=0;node<NUM_NODES;node++){
    sum+=nodeStats[0][node];
    squareSum+=nodeStats[0][node]*nodeStats[0][node];
  }
  fairnessIndex=(sum*sum)/(NUM_NODES*squareSum);
  //Print to file
  fileF<<fairnessIndex<<"\t";

  //Calculate the average energy
  //For this calculation we assume that a node is either in Tx, Rx, CCA, or idle
  double Rx=40,Tx=30, idle=0.8,cca=40;
  double energy[MAX]={0},avgEnergy=0;
  double energyC[MAX]={0},avgEnergyC=0;
  for(int node=0;node<NUM_NODES;node++){
    //We count energy consumed during simulations
    energy[node]=(nodeStats[0][node]+nodeStats[5][node])*(Tx)+nodeStats[1][node]*(Tx)
              +nodeStats[3][node]*cca+nodeStats[4][node]*idle + nodeStats[2][node]*(Rx);
    avgEnergy+=energy[node];
  }
  avgEnergy/=(1000*NUM_NODES); //Calculate as a function of slot=0.32ms
```

```
//Convert it to second
avgEnergy*=(0.32/1000); //Now we have a Watt
//Print Average energy
fileE<<avgEnergy <<"\t";

for(int node=0;node<NUM_NODES;node++){
  //We count energy transmissions
  energyC[node]=nodeStats[1][node]*(Tx);
  avgEnergyC+=energyC[node];
}
avgEnergyC/=(1000*NUM_NODES); //Calculate as a function of slot=0.32ms
//Convert it to second
avgEnergyC*=(0.32/1000); //Now we have a Watt
//Print Average energy
fileEC<<avgEnergyC <<"\t";
//Calculate the reliability. We define reliability as success/(success+collisions)
double R[MAX]={0},totalR=0;
for(int node=0;node<NUM_NODES;node++){
  R[node]=(1.0*nodeStats[0][node])/(nodeStats[0][node]+nodeStats[6][node]);
  totalR+=R[node];
}
totalR/=NUM_NODES;
//Print Reliability
fileR<<totalR <<"\t";

//Calculate Throughput
double throughput=(1.0*channel.timeSending)/SIM_TIME;
fileT<<throughput<<"\t";

//Calculate collisions rate
fileC<<(1.0*channel.timeCollission)/SIM_TIME<<"\t";

//calculate average delay to send a frame
double delay=0;
double n_sc=0; //number successful cycles
for(int node=0;node<NUM_NODES;node++)
  n_sc+=cycles[node][0];

n_sc/=NUM_NODES;
delay = SIM_TIME/n_sc;

fileDL<<delay<<"\t";

//Calculate Idle rate
fileI<<(1.0*channel.timeIdle)/SIM_TIME<<"\t";


//Print to final simulation results
fileS<<fairnessIndex<<"\t"<<throughput<<"\t"<<avgEnergy<<"\t"<<totalR<<"\t"<<endl;
int totalData=0,totalACKs=0;
for(int node=0;node<NUM_NODES;node++){
  totalData+=nodeStats[0][node];
  totalACKs+=nodeStats[5][node];
}

//Calculate Data Throughput
fileD<<(1.0*totalData)/SIM_TIME<<"\t";
```

```
  //Calculate ACK throughput
  fileA<<(1.0*totalACKs)/SIM_TIME<<"\t";

  //Calculate Access Failures
  int totalAF=0;
  for(int node=0;node<NUM_NODES;node++){
    totalAF+=accessFailures[node];
  }
  fileAF<<(1.0*totalAF)/(totalData/L)<<"\t";

  //Calculate Collision Failures

  int totalCF=0;
  for(int node=0;node<NUM_NODES;node++){
    totalCF+=collisionFailures[node];
  }
  fileCF<<(1.0*totalCF)/(totalData/L)<<"\t";


  //Calculate Probability of Collision

  double pc=0;
  for(int node=0;node<NUM_NODES;node++){
    pc+=(1.0*nodeStats[1][node])/(nodeStats[1][node]+nodeStats[0][node]);
  }
  pc/=NUM_NODES;
  filePC<<pc<<"\t";


cout<<"NODES="<<NUM_NODES<<"\tThroughput="<<(100.0*channel.timeSending)/SIM_TIME<<"%\n";

cout<<"NODES="<<NUM_NODES<<"\tData="<<(100.0*totalData)/SIM_TIME<<"%\tACK="<<(100.0*total
ACKs)/SIM_TIME<<"%\n";

cout<<"NODES="<<NUM_NODES<<"\tCollisions="<<(100.0*channel.timeCollission)/SIM_TIME<<"%\n
";
  cout<<"NODES="<<NUM_NODES<<"\tIdle="<<(100.0*channel.timeIdle)/SIM_TIME<<"%\n";

cout<<"NODES="<<NUM_NODES<<"\tReliability="<<(1.0*success)/(success+nbCollisionDiscarded+
nbBackoffDiscarded)<<"\n";

  double PI_SC=0;
  for (int node=0;node<NUM_NODES;node++)
        PI_SC+=(1.0*cycles[node][0])/(cycles[node][0]+cycles[node][1]+cycles[node][2]);
  PI_SC/=NUM_NODES;
  double PI_BC=0;
  for (int node=0;node<NUM_NODES;node++)
        PI_BC+=(1.0*cycles[node][1])/(cycles[node][0]+cycles[node][1]+cycles[node][2]);
  PI_BC/=NUM_NODES;
  double PI_CC=0;
  for (int node=0;node<NUM_NODES;node++)
        PI_CC+=(1.0*cycles[node][2])/(cycles[node][0]+cycles[node][1]+cycles[node][2]);
  PI_CC/=NUM_NODES;

  cout << "PI_SC="<< PI_SC<<endl;
  cout << "PI_BC="<< PI_BC<<endl;
```

```
  cout << "PI_CC="<< PI_CC<<endl;
  cout<<"Throughput2="<<n_sc*NUM_NODES*L/SIM_TIME<<endl;

  return 0;
}
int main(){
  int nodesT[15]={5,10,15,20,25,30,35,40,45,50,100,200,400,800,1600};
  int CCA_V[5]={2,5,8,11};
  srand(time(0));
  char fileName[80];
  char prefix[80];
  char test[80];
  if(ACK)
    strcpy(prefix,"ACK_");
  else
    strcpy(prefix,"NOACK_");
  if(QUEUE)
    strcat(prefix,"QUEUE_");
  else
    strcat(prefix,"SATUR_");
  strcat(prefix,itoa(L,test,10));
  strcat(prefix,"_");
  strcat(prefix,itoa(macMaxCSMABackoffs,test,10));
  strcat(prefix,"_");
  strcat(prefix,itoa(macMaxFrameRetries,test,10));
  strcat(prefix,"_");

  strcpy(fileName,prefix);
  strcat(fileName,"Throughput.txt");
  fileT.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"Energy.txt");
  fileE.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"EnergyCollisions.txt");
  fileEC.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"Reliability.txt");
  fileR.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"Fairness.txt");
  fileF.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"SimulationResults.txt");
  fileS.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"Collisions.txt");
  fileC.open(fileName);

  strcpy(fileName,prefix);
  strcat(fileName,"Idle.txt");
```

```
fileI.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"ACK.txt");
fileA.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"Data.txt");
fileD.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"Delay.txt");
fileDL.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"AccessFailure.txt");
fileAF.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"CollisionFailure.txt");
fileCF.open(fileName);

strcpy(fileName,prefix);
strcat(fileName,"ProbabilityCollision.txt");
filePC.open(fileName);

for(int index=0;index<12;index++){
  fileT<<nodesT[index]<<"\t";
  fileE<<nodesT[index]<<"\t";
  fileEC<<nodesT[index]<<"\t";
  fileR<<nodesT[index]<<"\t";
  fileF<<nodesT[index]<<"\t";
  fileS<<nodesT[index]<<"\t";
  fileC<<nodesT[index]<<"\t";
  fileI<<nodesT[index]<<"\t";
  fileA<<nodesT[index]<<"\t";
  fileD<<nodesT[index]<<"\t";
  fileDL<<nodesT[index]<<"\t";
  fileAF<<nodesT[index]<<"\t"; //Access failure
  fileCF<<nodesT[index]<<"\t"; //Collision Failure
  filePC<<nodesT[index]<<"\t"; //Probability of Collision


  for(CSMA_MOD=0;CSMA_MOD<=2;CSMA_MOD++){
    for(int index2=3; index2<=3; index2++){
      CCA_MAX=CCA_V[index2];
      runSimulator(nodesT[index]);
    }
  }
  fileT<<"\n";
  fileE<<"\n";
  fileEC<<"\n";
  fileR<<"\n";
  fileF<<"\n";
  fileS<<"\n";
  fileC<<"\n";
  fileI<<"\n";
```

```
    fileA<<"\n";
    fileD<<"\n";
    fileDL<<"\n";
    fileAF<<"\n";
    fileCF<<"\n";
    filePC<<"\n";
  }
  fileT.close();
  fileE.close();
  fileR.close();
  fileF.close();
  fileEC.close();
  fileS.close();
  fileC.close();
  fileI.close();
  fileA.close();
  fileD.close();
  fileDL.close();
  fileAF.close();
  fileCF.close();
  filePC.close();

  system("pause");
  return 0;
}
```

Appendix C

Confidence Interval Computation

The confidence interval (CI) is a quantification method used to measure the uncertainty in collected samples of data. It consists of a range of values in which a generated sample lies with some probability. This probability is usually equal to 0.95, which means that the collected samples lay in an interval with a confidence of 95% which is called CI. The end points of the CI are called confidence limits. The confidence limits for a *normally* distributed sample of data n are calculated as:

$$\mu \pm z \frac{\sigma}{\sqrt{n}} \tag{A.1}$$

where, μ is the mean value of a sample of data n, σ is the standard deviation of n, and z is the significance level. Equation A-1 shows that the confidence interval is localized at the mean value μ of the collected data.

The level z specifies the area within the normal distribution, which corresponds to the desired confidence level, see Figure A-1. In order to cover the 95% CI for this kind of distribution, we need to exclude 5% of the total area. Therefore, we should exclude 2.5% on both sides of the mean μ. Then, we need to find the area that corresponds to 95% of n. This area can be found using the z-table that contains the areas which correspond to the desired confidence level, see Table A-1. In order to read z from this table, we need to specify the percentage of the sample data needed to achieve a 95% of confidence. This corresponds to 1-0.025 = 0.975. Thus, from the table we can see that z = 1.96 (once we locate the 0.975 in the table, we read the first two digits of z from the leftmost column. Then, we get the third digit from the first row of the column where 0.975 is located).

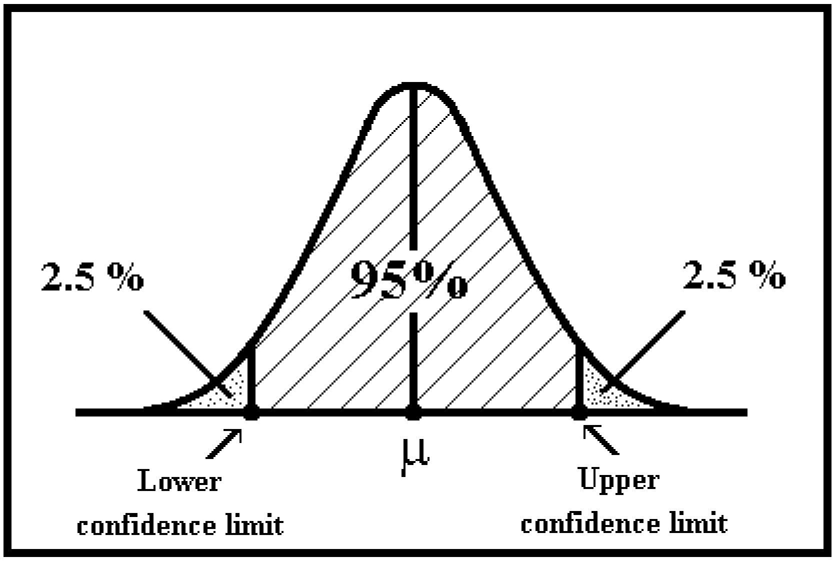


Figure A-1: The normal distribution of the sample data *n*.

Table A-1: z-table.

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07
1.8	0.96407	0.96485	0.96562	0.96638	0.96712	0.96784	0.96856	0.96926
1.9	0.97128	0.97193	0.97257	0.97320	0.97381	0.97441	0.97500	0.97558
2	0.97725	0.97778	0.97831	0.97882	0.97932	0.97982	0.98030	0.98077