Performance of a beacon enabled IEEE 802.15.4-compliant network.

by

Shairmina Shafi

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Faculty of Graduate Studies
University of Manitoba

COMMITTEE SIGNATURE PAGE

This dissertation was presented

by

Shairmina Shafi

It was defended on

Yet to be decided

and approved by

(Signature) ____________________

Committee Chairperson
Dr. Jelena Misic

(Signature) ____________________

Committee Member
Dr. Ekram Hossain

Abstract

The IEEE 802.15.4 standard is mainly defined to enable wireless sensors and personal devices to communicate within a personal operating space (POS) with minimal cost and effort. We analyze the performance of a 802.15.4-compliant network operating in the beacon enabled mode with uplink traffic only and also with both uplink and downlink traffic. We assume that the network is operating in non-saturation mode. The model considers acknowledged transmissions and includes the impact of different parameters such as packet arrival rate, number of devices, and packet size. We measure the performance parameters such as throughput, channel access probability, probability of successful transmission, and access delay. We also focus on certain issues in the standard that lead to serious performance limitations, and suggest some simple modifications of the MAC layer that allows the network to handle higher traffic loads.

Moreover, we investigate the interaction of activity management with the CSMA-CA based MAC layer in a beacon enabled IEEE 802.15.4-compliant sensor network to maximize the network lifetime along with achieving the desired data rate. For the application area of sensor networks, we propose two activity management policies, namely, the centralized policy, and the distributed policy. Through simulation study, we show that the proposed policies can save significant energy to extend network lifetime and also can maintain the desired reliability under various combinations of network size and packet arrival rates.

Acknowledgements

First and foremost, I am thankful to my advisor Dr. Jelena Mišić for accepting me as her Masters student and guiding me with patience, encouragement, and inspiration. Dr. Jelena Mišić introduced me to the area of wireless sensor networks.

I would like to thank Dr. Vojislav Mišić for introducing me with Artifex, simulation programming language. Whenever I faced problems with this language, I got valuable advice from him.

I would like to thank Faculty of Science for supporting me with Faculty of Science Scholarship, Faculty of Graduate Studies for supporting me with University of Manitoba Graduate Fellowship, Dr. Jelena Mišić and the department of Computer Science for supporting me with Grader/marker.

I am also thankful to the thesis committee member, Dr. Ekram Hossain, for being in my thesis committee.

Finally I wish to thank Dr. Attahiru S. Alfa and Dr. Ellen Liu. I got some idea about queueing theory, and simulation output analysis by taking their courses 24.765, and 74.781, respectively.

Dedication

Dedicated to my husband, and my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless personal area networks (WPANs) are short-range wireless networks built from small, energy-efficient devices operating on battery power. This type of network requires little infrastructure to operate, or none at all. Different application areas for such networks have different requirements in terms of data rate, power consumption, and quality of service. IEEE working group has defined three classes of WPANs. High data rate networks for real-time and multimedia applications are supported through IEEE Std 802.15.3 [15], medium data rate networks for cable replacement and consumer devices can use IEEE Std 802.15.1 (Bluetooth) [14], while the recent standard IEEE Std 802.15.4 (ZigBee) [17] is intended to be the key enabler for low complexity, ultra low power consumption, and low data rate wireless connectivity among inexpensive fixed, portable, and moving devices [17]. The scope of this standard is to define the medium access control (MAC) layer and physical (PHY) layer specifications. It is mainly designed to enhance wireless sensor networks.

Wireless sensor networks (WSNs) are collections of large amount of sensor nodes. A sensor node is a simple battery powered device equipped with integrated sensing (vibration, explosive, acceleration, temperature etc.), and wireless communication capabilities. WSNs should be self-organizing to achieve simple installation. In case of self-organization, each sensor node should have the capability to participate in a network without any operator based configuration such as addressing, association, and traffic balancing [11]. Wireless sensor networks are event based systems [23]. In such systems, individual nodes send information to a collecting node known as sink. Based on the collective information, the sink detects an event. Since sensor networks are not operator supervised, and the battery of the sensor nodes can not be replaced, it is important that sink achieves reliable event detection while the power consumption of sensing nodes is minimal. However, WSNs support a wide range of applications in different sectors such as control and monitoring applications in industrial and commercial sectors, precision agriculture, asset and inventory tracking, and security.

In an IEEE 802.15.4 compliant WPAN, a central controller device commonly referred to as PAN coordinator builds a WPAN with other devices within a small physical space known as the personal operating space (POS). A device in the WPAN may be a full function device (FFD) or a reduced function device (RFD). The PAN coordinator must be an FFD whereas an FFD includes all the details of MAC services. An FFD can talk to RFDs or other FFDs. The RFD is an extremely simple device such as a light switch or a passive infrared sensor. An RFD includes a reduced set of MAC services that allow it to talk to only a single FFD at a time.

Depending on the application requirements, the devices and the PAN coordinator can communicate with one another through the same physical channel using two types of network topologies. The topologies are the star topology and the peer-to-peer topology. In the star topology, no direct communication is established among the devices. All devices can communicate with one another via the PAN coordinator. Normally the applications, requiring low latency connections such as home automation, personal health care, toys and games would benefit by using a star topology. In the peer-to-peer topology, the LR-WPAN devices within the POS can directly communicate with one another and also the PAN coordinator [17]. The peer-to-peer topology allows the implementation of more complex network formations include cluster tree, mesh networking, and ring topology. Wireless sensor networks, precision agriculture, security are the possible ap-

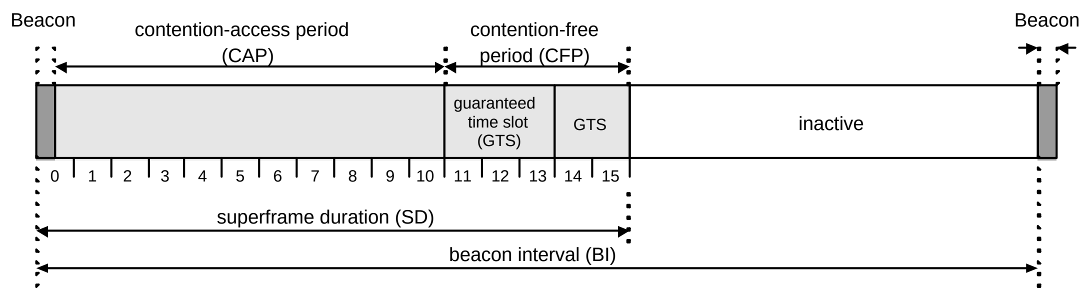


Figure 1.1: The composition of the superframe under IEEE Std 802.15.4 standard (adapted from [17]).

plications that benefit from a peer-to-peer topology. Each PAN coordinator is identified by a unique identifier to enable communications between devices within a network and across independent networks.

The communication between two devices or a device and a PAN coordinator depends on whether the network allows transmission of network beacons. Transmission of beacons provides synchronization between the devices and the PAN coordinator. The standard defines two channel access mechanisms in MAC layer. One is beacon enabled access. The other is non beacon enabled access. Beacon enabled networks use slotted carrier sense multiple access mechanism with collision avoidance (CSMA-CA) while the non beacon enabled networks use simpler, unslotted CSMA-CA. In this work, we will focus on the beacon enabled networks with slotted CSMA-CA; the unslotted access mechanism, being very similar to the one used in IEEE 802.11 standard, will not be considered.

1.1 Timing Structure

In beacon enabled networks, the PAN coordinator divides its channel time into superframes. Each superframe starts with the transmission of a network beacon, followed by an active portion and an optional inactive portion as shown in Fig. 1.1. The coordinator communicates with its PAN members during the active portion and may enter a sleep (low power) mode during the inactive portion. The superframe duration, SD, is equivalent to the duration of the active portion of the superframe, which cannot be longer than the beacon interval BI.

The active portion of each superframe is divided into equally sized slots. The beacon is transmitted at the beginning of slot 0, and the contention access period (CAP) of the active portion starts immediately after the beacon. In each slot, the channel access mechanism is contention based, using the CSMA-CA access mechanism (more details are in Section 3.1.1). A device must complete all of its contention based transactions within CAP of the current superframe.

Within the time slots of the active portion of the superframe, the PAN coordinator may reserve some slots to allow dedicated access to some devices those are less sensitive to delay. These slots are called guaranteed time slots or GTSs. These GTSs together comprise a contention-free period (CFP) [17].

The basic time unit of the MAC protocol is the duration of the so-called backoff period. Access to the channel can occur only at the boundary of the backoff period. The actual duration of the backoff period depends on the frequency band in which the 802.15.4 is operating (more detail are in Section 3.3). Table 1.1 summarizes the basic timing relationships in the MAC sublayer. Note that the constants and attributes of the MAC sublayer, as defined by the standard, are written in italics. Constants have general prefix of "a", e.g. *aUnitBackoffPeriod*, while attributes have a general prefix of "mac", e.g. *macMinBE*.

1.2 Aim of our Research

In this work, we model the WPAN with uplink and downlink transmissions, which is a realistic scenario for a Personal Area Network setting where nodes communicate either with each other or with other devices in an external network (i.e., the Internet) through the PAN coordinator. The main factors that determine performance at the MAC level

Type of time period	Duration	MAC constant/attribute
Modulation symbol	1 data bit in 860 MHz and , 915 MHz bands, 4 data bits in 2.4 GHz band.	N/A
Unit backoff period	20 symbols	*aUnitBackoffPeriod*
Basic superframe slot (SO=0)	three unit backoff periods (60 symbols)	*aBaseSlotDuration*
Basic superframe length (SO=0)	16 basic superframe slots (960 symbols)	*aBaseSuperframeDuration* =*NumSuperframeSlots.* *aBaseSlotDuration*
(Extended) superframe duration SD	$aBaseSuperframeDuration.2^{SO}$	*macSuperframeOrder*,SO
Beacon interval BI	$aBaseSuperframeDuration.2^{BO}$	*macBeaconOrder*,BO
Maximum time to wait for a downlink transmission	1220 symbols	*aMaxFrameResponseTime*
Rx-to-Tx or Tx-to-Rx maximum turnaround time	12 symbols	*aTurnaroundTime*
Time-out value to wait for the acknowledgement	54 or 120 symbols	*macAckWaitDuration*

Table 1.1: Timing structure of the slotted mode MAC protocol (Note that in the beacon enabled mode the values of BO and SO must be less than 15).

include network parameters (size), device parameters (buffer size, packet size), and also traffic parameters such as packet arrival rates. However, performance analysis of such networks at the MAC level are still scarce, in particular of networks adhering to the 802.15.4 standard. Therefore, the performance analysis of this standard is an important issue that helps us to know how the applications that have served as motivation for the creation of this standard could benefit from using such a technology.

Furthermore, the IEEE 802.15.4 standard was mainly designed to enable wireless sensor networks. The features of wireless sensor networks are different in many ways from conventional wireless networks. Among these features, energy consumption is the most significant. Since a wireless sensor node is a battery powered microelectronic device, it can only be equipped with a limited power source [2]. The lifetime of a sensor node strongly depends on battery lifetime whereas battery replacement or recharging of sensor nodes is often impractical in some applications. Therefore, sensor nodes must have the capability to last long time with a limited power source. This can be achieved by applying a proper duty cycle management technique. The duty cycle is the fraction of time a sensor node is allowed to remain active. The low duty cycle operation at the level of individual sensor node can extend the lifetime of the sensors as well as the total network lifetime. In such case, redundant sensors can be used, i.e., the number of deployed sensors covering a given physical area should be larger than minimum number based on the required data rate. The desired data rate received at the network sink, referred to as “event reliability” in [23], can be achieved by adjusting the number of active sensors at any given time.

We assume a sensor node is active when its radio transceiver is on, and inactive when its radio transceiver is off. The impact of turning the radio transceiver off is different from the impact of turning the sensing activity off. In inactive state, a sensor node continues its sensing activity so that it can detect event. In this study, we assume that the radio is duty cycled while the sensing activity is not. The energy consumption during sensing activity is considered to be zero. On the other hand, in active state, the energy consumption may vary depending on the mode that the sensor node operates on. The radio characteristics that the IEEE 802.15.4 standard offers with 250 kbps transmission rate are 31mW as transmit power, 35mW as receive power, and 30mW as idle power [20][16]. Since these three powers vary few percent from one another, for simplicity, we assume that the energy consumption in active state is constant.

In our research, we try to measure the performance of the IEEE 802.15.4 compliant sensor network with and without a duty cycle management technique. We assume that sensor network with star-shaped hierarchical topology is operating under the beacon enabled mode of the IEEE 802.15.4 standard. The star-shaped hierarchical topology and the beacon enabled slotted CSMA-CA regime appear better suited for a sensor network implementation than their peer-to-peer and unslotted CSMA-CA counterparts, respectively. The rationale is simple: the star-shaped topology is better because the PAN coordinator can also act as the network sink which collects the data from individual sensor nodes, and the beacon enabled slotted CSMA-CA regime simplifies synchronization and forwarding of data from the PAN coordinator-cum-network sink for further processing.

Chapter 2

Related Work

The draft of unapproved proposed IEEE 802.15.4 standard was first published in February 2003. This draft is an output of combined efforts of two groups: the IEEE 802 Working Group 15, and Zigbee, a HomeRF spinoff. These two groups formed a task group, to be called 802.15.4 [12]. This task group was officially sanctioned in December 2000 by the IEEE New Standards Committee (NesCom). Since it is a very new standard, very few papers have been published based on this standard. [12] discussed the technical challenges and the system requirements needed to implement such a low cost, low power network solution. They also focused on the applications that could benefit from using such a technology. A brief overview of this standard is presented in [6]. The authors also mentioned suitable applications. Section 2.1 describes several researches that have been done to evaluate the MAC layer performance of various wireless network solutions and section 2.2 describes the design of various power aware protocols for sensor networks.

2.1 Performance Measurement Techniques

Analytical analysis and simulation study are the two concepts to study the behavior of a system. The MAC of the IEEE 802.11 for wireless local network use CSMA-CA mechanism, and also the RTS/CTS mechanism to access the medium. The performance of the IEEE 802.11 standard under various network configurations is measured by [7], [8] through simulation study. An analytical formula derived in [5] to determine the theoretical upper bound of the IEEE 802.11 protocol capacity. The paper showed that the theoretical limits vary according to the network configuration and a result very close to theoretical limits can be obtained by appropriate tuning of the backoff algorithm of the IEEE 802.11 standard. Work presented in [4] modeled the basic channel access mechanism, namely Distributed Coordination Function (DCF) and the optional four way handshaking request to send/clear to send (RTS/CTS) mechanism of the IEEE 802.11 standard in the assumption of saturation conditions and ideal channel conditions (no hidden terminals and capture). In the analysis, the author modeled the protocol using queuing approach and obtained an extremely accurate result. A theoretical analysis carried out by [21] to model the Bluetooth (IEEE 802.15.2 standard) piconet, a relevant technology of the IEEE 802.15.4 standard. To model the two offered scheduling policies of the Bluetooth piconet, and to make a comparison between these two policies, [21] used the theory of M/G/1 queues with vacations. At last, the paper determined that the exhaustive scheduling policy performs better than the limited service policy.

Recently, [20] uses a NS-2 simulator to measure the performance of various features of the IEEE 802.15.4 MAC . They consider a IEEE 802.15.4 compliant star topology network operates in beacon enabled mode. They also consider the devices can access the channel both in contention access period(CAP), and contention free period (CFP). They vary the active or inactive period by setting the parameters SO and BO for a fixed duty cycle expressed by $\frac{2^{SO}}{2^{BO}}$. Their performance evaluation shows that higher traffic load and large number of devices increase collision even though the CSMA-CA mechanism reduce energy cost due to idle listening in the inactive period. However, to the best of our knowledge this is the only attempt that evaluate the MAC performance based on simulation result.

In our simulation model, we try to measure the performance in different approach described in [20]. We consider the superframe only consists of active period and no GTSs are allowed. We measure the performance considering only uplink transmission for a star topology network both in saturation condition and non-saturation condition. This type of network is satisfactory for the sensor applications where devices are trying to send packets to sink for aggregation through the PAN coordinator. In non-saturation regime, we consider that sources generate packets following Poisson distribution instead of a constant averaged rate described in [20]. The simulator is then extended to allow both uplink and downlink transmissions. We observe the performance and identify certain issues in the standard that lead to serious performance limitations. We also suggest that some simple modifications of the MAC layer that allows the network to handle higher traffic loads.

2.2 Power Aware Protocols

Since the power management and conservation is an important concern in wireless sensor networks, researchers are becoming more interested in doing researches in this area. In [18], the authors apply the mathematical paradigm of Gur Game to achieve the desired QoS (the optimum number of sensors active in a given area at any given time and forwarding information to the information collecting sinks, the base stations) of sensor networks. In this approach, the base station determines the number of active nodes by calculating the number of packets collected from the sensor nodes. It then broadcasts to all sensors whether the number of currently active nodes is sufficient to maintain the desired QoS. Depending on this information the sensors take a decision using the probabilistic Gur Game automation whether it should remain active or inactive. The problem of this approach is that the structure of Gur automata limits the number of active sensors to one-half of the total sensors. Besides, this approach consumes a significant energy because all sensors need to listen to the control information from the base station.

To reduce the energy consumption of the above approach [18], an alternative approach is introduced in [9]. In this approach, each sensor transmits its data packets to the base station with a transmission probability T_i, the suffix "i" represents the current state of a sensor node. The higher value of i represents higher transmission probability. Depending on the number of collecting packets, the base station can determine the current QoS information and forward this information as an acknowledgement (ACK) to the transmitting devices. According to the ACK feedback, the sensors then independently update their current states using a simple probabilistic automation. Since all sensor nodes need not to receive control information, the nodes with lower states can go to sleep for a prolonged period. The problem with this approach is the centralized calculation of transmission probabilities.

The authors in [27] use a Voronoi diagram to control the status (turn on or off) of the sensor nodes. The Voronoi diagram decomposes the monitoring area into polygonal regions around each node. To update the Voronoi diagram, each time the node with the smallest area is picked up. The node of that area should be turned off if the corresponding area is smaller than a given threshold. In this situation, the neighbors of that node take the responsibility to monitor the corresponding region. This process continues until at least one node is present with a smaller area than the threshold.

Two types of duty cycle mechanisms, namely, the random sleep type and the coordinated sleep type are introduced in [13]. The coordinated sleep type allows sensors to coordinate with each other to maintain an active-sleep schedule, while the sensors maintain active-sleep schedule independent of each other in the random sleep type mechanism. The authors compensated the performance-energy tradeoff by adding redundancy in sensor deployment and focus on two performance measures within the context of coverage. The performance measures are the extensity of coverage (probability that any given point is not covered by any active sensor), and the intensity of coverage (probability that any given point is not covered by any active sensor for a given period of time longer than n). Both duty cycle mechanisms use a sleep sensor ratio ρ to regulate the average proportion of sleep time where the duty cycle is $1 - \rho$. The random sleep type is simple and attractive, but it cannot adjust the pre-set value of ρ adaptively with the actual node density in the network or in its neighborhood. The coordinated sleep type overcomes this problem by allowing adaptive communication between sensor nodes and defining multiple roles of each sensor. The algorithm is referred to as Role-Alternating, Coverage-Preserving, Coordinated Sleep Algorithm (RACP). The authors show that the RACP can provide continuous coverage with a significantly lower duty cycle. However, the effect of duty cycling within the context of network coverage is not applicable for our work since authors assume multi-hop sensor network, and use sleep schedules to achieve network connectivity.

A MAC protocol S-MAC (sensor MAC) for wireless sensor network is introduced in [28]. This MAC protocol pays more attention on energy conservation and self configuration characteristics of sensor networks rather than per node fairness and latency. The S-MAC protocol carefully handles all the major sources of energy waste such as collision, idle listening, overhearing and control overhead. The protocol encourages sensor nodes to operate in a periodic listen and sleep mode. The periodic sleep mode of this approach provides low duty cycle operation in a multihop network and also increases message latency and reduces throughput.

Besides S-MAC, other famous MACs for sensor network are Self-Organizing Medium Access Control for Sensor networks (SMACS) [25], Eavesdrop-And-Register (EAR) [25], and Power Aware Multi-Access protocol with Signaling for Ad Hoc Networks (PAMAS) [24]. All these MACs assume that all sensor nodes are equivalently powerful and there is no hierarchical structure of the network through some more powerful nodes. These MACs also assume that the nodes have peer-to-peer communications among themselves and build time schedules with the neighbor nodes for transmission. The sensor nodes obey the time schedule to avoid idle listening to and overhearing the channel to save energy. Avoidance of idle listening and unnecessary overhearing require a lot of signaling among the nodes. Therefore, due to high signaling traffic these protocols may not meet the scalability criterion perfectly.

On the other hand, the industrial standard IEEE 802.15.4 includes many features for enabling low power consumption and low cost implementation. To manage duty cycle, the IEEE 802.15.4 standard proposes to allow the network to remain inactive for fixed period of time. [20] shows low duty cycle operation can save significant energy by offering higher latency and lower bandwidth. On the other hand, our proposed algorithms allow some devices to sleep instead of allowing the total network to sleep at any given time, thus achieving continuous event monitoring. Moreover, we also carefully observe how the continuous fixed event reliability can be maintained with the change of network size (occurred due to node failures) as well as channel conditions.

Chapter 3

Basic Description of IEEE 802.15.4 MAC

3.1 Channel Access Mechanisms

As in other contention based access control schemes, transmissions will be attempted only when the medium is clear, but withheld if there is channel activity, or when contention occurs. The standard defines two channel access mechanisms in MAC layer. One is beacon enabled access. The other is non beacon enabled access. In beacon enabled access, the coordinator periodically sends beacon frame out to all its devices. When a device receives a beacon frame, it will synchronize the communication according to the beacon frame. In non beacon enabled access, coordinator does not send any beacon. Beacon enabled networks use slotted carrier sense multiple access mechanism with collision avoidance (CSMA-CA) while the non beacon enabled networks use simpler, unslotted CSMA-CA.

3.1.1 Slotted CSMA-CA Algorithm

The flowchart shown in Fig. 3.1 describes the CSMA-CA protocol in more detail. The algorithm is invoked when a packet is ready to be transmitted. Three variables are maintained for each packet:

- NB is the number of times the algorithm was required to backoff due to the unavailability of the medium during channel assessment.
- CW is the contention window, i.e., the number of backoff periods that need to be clear of channel activity before the packet transmission can begin; and
- BE is the backoff exponent which is related to the number of backoff periods a device should wait before attempting to assess the channel.

The algorithm begins by initializing NB to zero and CW to 2. The variable NB takes the value from 0 to m (m = *macMaxCSMABackoff* - 1), while the variable CW takes the value 0, 1, and 2.

If the device operates on battery power, as indicated by the attribute *macBattLifeExt*, the parameter BE (the backoff exponent which is used to calculate the number of backoff periods before the node device attempts to assess the channel) is set to 2 or to the constant *macMinBE*, which is less; otherwise, it is set to *macMinBE* (the default value of which is 3).

The algorithm then locates the boundary of the next backoff period; as mentioned above, all operations must be synchronized to backoff time units.

In step (2), the algorithm generates a random waiting time k in the range $0..2^{BE}$ - 1 backoff periods. The value of k is then decremented at the boundary of each backoff period. Note that the counter will be frozen during the inactive portion of the beacon interval, and the countdown will resume when the next superframe begins. When this counter becomes zero, the device must make sure the medium is clear before attempting to transmit a frame. This is done by listening to the channel to make sure no device is currently transmitting. This procedure, referred to as Clear Channel Assessment (CCA), has to be done in two successive backoff periods.

If the channel is found to be busy at the second CCA, the algorithm simply repeats the two CCAs starting from step (3). However, if the channel is busy at the first CCA,

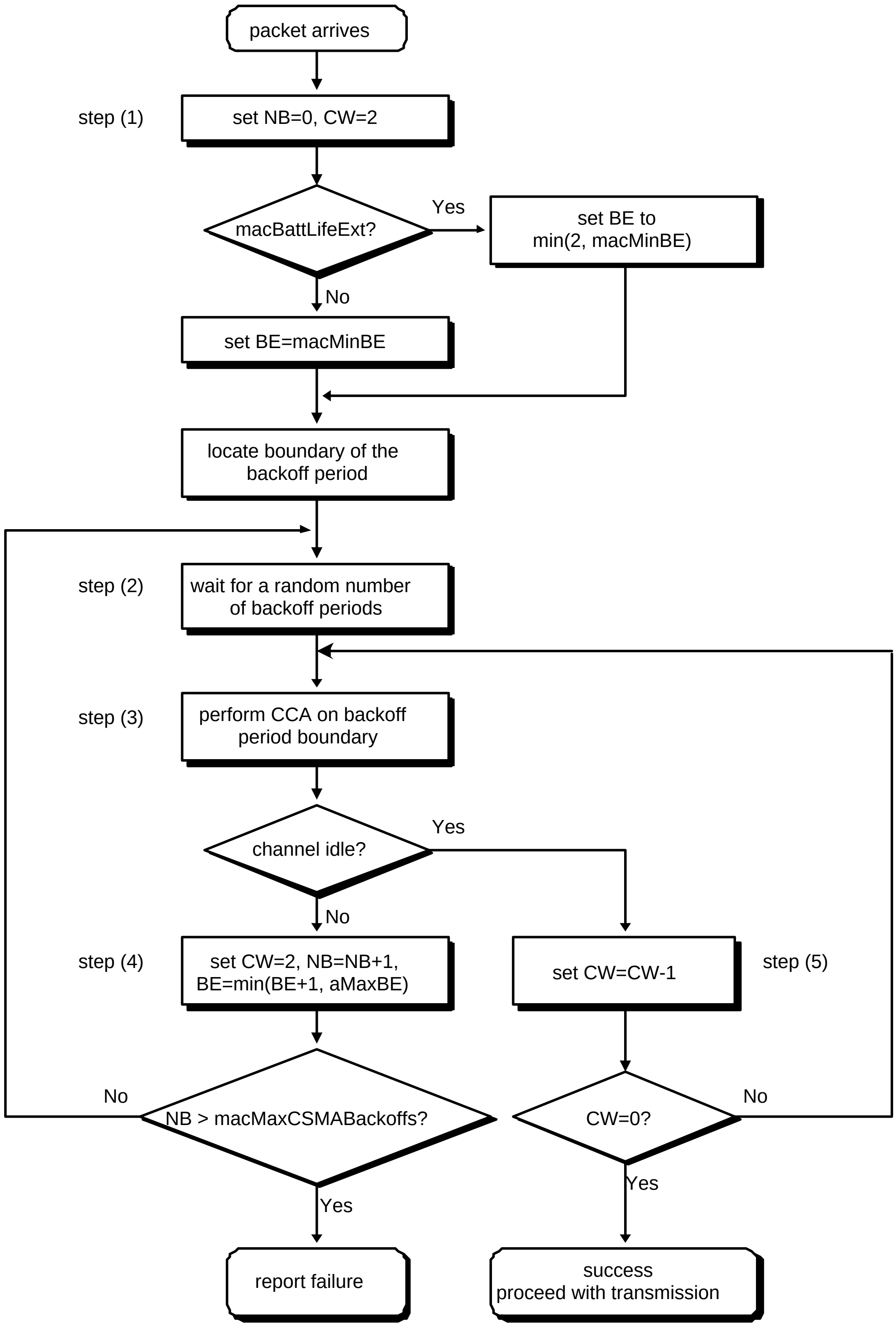


Figure 3.1: Operation of the slotted CSMA-CA algorithm in the beacon enabled mode(adapted from [17]).

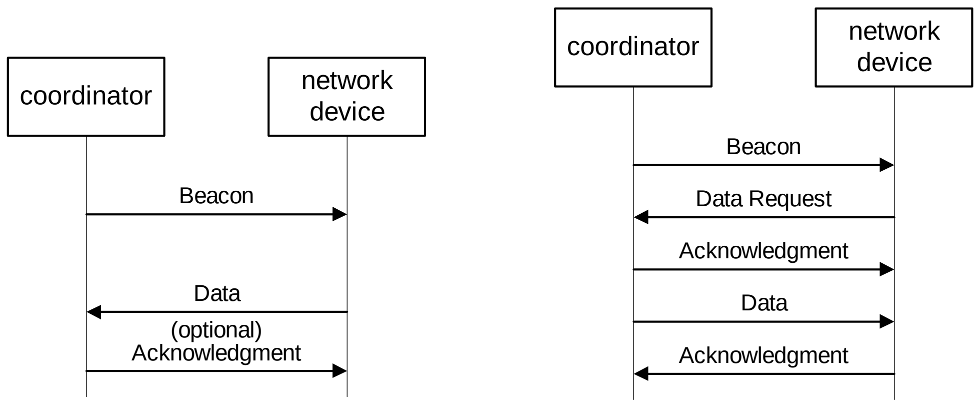
the values of NB and BE are increased by one, while CW is reset to 2, and another random wait is initiated; this is step (4) in the flowchart. In this case, when the number of retries is below or equal to *macMaxCSMABackoffs* (the default value of which is 5), the algorithm returns to step (2), otherwise it terminates with a channel access failure status. Failure will be reported to the higher protocol layers, which can then decide whether to re-attempt the transmission as a new packet or not. In our model, we assume that the transmission will be reattempted until the final success.

If both CCAs report that the channel is idle, packet transmission may begin. Before undertaking step (3), the algorithm checks whether the remaining time within the CAP area of the current superframe is sufficient to accommodate the CCAs, the data frame, the proper interframe spacing, and the acknowledgement. If this is the case, the algorithm proceeds with step (3); otherwise it will simply pause until the next superframe, and resume step (3) immediately after the beacon frame.

3.2 On uplink and downlink communication

In a beacon enabled network, uplink data transfers from a node to the coordinator are synchronized with the beacon, in the sense that both the original transmission and the subsequent acknowledgment must occur within the active portion of the same superframe, as shown in Fig. 3.2(a). Uplink transmissions always use the CSMA-CA mechanism outlined above.

Data transfers in the downlink direction, from the coordinator to a node, are more complex, as they must first be announced by the coordinator. In this case, the beacon frame will contain the list of nodes that have pending downlink packets, as shown in Fig. 3.2(b). When the node learns there is a data packet to be received, it transmits a MAC command requesting the data. The coordinator acknowledges the successful reception of the request by transmitting an acknowledgement. After receiving the acknowledgement, the node listens for the actual data packet for the period of *aMaxFrameResponseTime*, during which the coordinator must send the data frame.



(a) Uplink transmission. (b) Downlink transmission.

Figure 3.2: Uplink and downlink data transfers in beacon enabled PAN.

According to the standard, it is allowed to send the data frame 'piggybacked' after the request acknowledgement packet, i.e., without using CSMA-CA. However, two conditions have to be fulfilled: the coordinator must be able to commence the transmission of the data packet between *aTurnaroundTime* and *aTurnaroundTime* + *aUnitBackoffPeriod*, and there must be sufficient time in the CAP for the message, appropriate interframe spacing, and acknowledgement; if either of these is not possible, the data frame must be sent using the CSMA-CA mechanism [17]. While the first condition depends on the implementation platform, the second depends on the actual traffic; thus some data frames will have to be sent using CSMA-CA. For uniformity, we consider a more generic approach by assuming that slotted CSMA-CA is used for all downlink transmissions, although the case where CSMA-CA is not used could be accommodated with ease. Furthermore, downlink transmissions that do not use the CSMA-CA mechanism would cause additional collisions and thus lead to the deterioration of network performance.

While the use of acknowledgement is optional (i.e., it is sent only if explicitly requested by the transmitter), in this work we assume that all the transmissions are acknowledged.

In this case, the receiving node must acknowledge successful reception of the data frame within a prescribed time interval, otherwise the entire procedure (starting from the announcement through the beacon frame) has to be repeated.

According to the Section 7.5.6.4.2 of the 802.15.4 standard [17], the transmission of an acknowledgement frame shall commence at the backoff period boundary between *aTurnaroundTime* and *aTurnaroundTime* + *aUnitBackoffPeriod* after the data frame, which amounts to a delay of 12 to 32 symbol periods. Since one backoff period takes 20 symbols, this time interval may include at most one backoff period at which the channel will be assessed idle. However, a node that has finished its random countdown will need at least two CCAs before attempting transmission: while the first one may find the medium idle in between the data frame and the acknowledgment, the second one will coincide with the acknowledgement and cause the CSMA-CA algorithm to revert to the next iteration of the backoff countdown. Consequently, the acknowledgement packet cannot possibly collide with the data packet sent by another node.

It should be noted that the Section 7.5.6.7 of the standard stipulates that the data packet originator should wait for an acknowledgement for at most *macAckWaitDuration*, which amounts to 54 or 120 symbols, depending on the actual channel number. If the acknowledgement packet is not received within *macAckWaitDuration* after the original data frame, the originator may safely assume that the frame has been lost and initiate retransmission.

3.3 Impact of the physical layer

IEEE 802.15.4 standard can work in three frequency bands, namely in 868 MHz band with raw data rate of 20kbps, in 915 MHz band with 40kbps and in 2450MHz band known as (Industrial, Scientific and Medical - ISM) with 250kbps. Since we perceive that future sensing applications will need large bandwidth we will consider the third band only. This band is already hosting wireless LAN/PAN standards such as 802.11b and 802.15.1 (Bluetooth) and a lot of interference is expected. IEEE 802.15.4 standard in the 2450 MHz range (ISM band) uses 16-ary quasi-orthogonal modulation technique. Four data bits represent one modulation symbol and that symbol is further encoded into 32 bit chip sequence. There are 16 nearly-orthogonal Pseudo-Noise chip sequences. Each chip sequence is modulated onto the carrier using offset quadrature phase shift keying (O-QPSK). Since the chip rate is 2Mcps and raw data rate is 250kbps the processing gain is 8. According to properties of DSSS systems [10] for one user, this results in maximum supported ratio of bit energy to the noise power spectral density of $\frac{E_b}{N_0} = 8$. According to the properties of QPSK, the Bit Error Rate is given with the expression [10]:

$$BER = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

where $Q(u) \approx e^{-u^2/2}(\sqrt{2\pi}u)$, $u \gg 1$.

Therefore, without the interference, we should expect BER slightly less than 10^{-4}. This is confirmed in the Section 6.1.6 of the standard where Packet Error Rate (PER) of 1% is expected on packets which have 20 octets including MAC and physical level headers. However, in the presence of interference in the ISM band, it is more realistic to expect BER around 10^{-3} and Packet Error Rate equal to $PER = 1 - (1 - BER)^X$ where X is packet length including MAC and physical layer header expressed in bits.

However, the transmission will be corrupted by noise when either data packet is corrupted or acknowledgement packet is corrupted. The probability that data transmission is not corrupted is then equal to:

$$\delta = (1 - BER)^{X_d + X_a} \tag{3.1}$$

where X_d and X_a are lengths, in bits, of data packet and acknowledgement packet respectively (including all headers).

Chapter 4

Performance Measurements

4.1 The Network Model

We focus on PANs that operate in the ISM band at 2.4GHz because of much higher bandwidth with raw data rate 250kbps, and with SO=0, BO=0. In that case, one modulation symbol corresponds to 4 data bits, *aUnitBackoffPeriod* has 10 bytes, while *aBaseSlotDuration* has 30 bytes; as *aNumSuperframeSlots* is 16, the *aBaseSuperframeDuration* is exactly 480 bytes. Furthermore, we have assumed the following values for the parameters of the CSMA-CA MAC algorithm:

1. The minimum value of backoff exponent *macMinBE* is set to three.
2. The maximum value of backoff exponent *aMaxBE* is set to five.
3. The maximum number of backoff attempts is set to five, i.e. *macMaxCSMABackoffs* = 4.

We assume that the basic beacon length using short device addresses of four bytes and including the MAC and physical layer headers, is 17 bytes and that it does not contain Guaranteed Time Slot announcement or beacon payload. For convenience, we have rounded it to 20 bytes i.e. the length of the beacon frame is 2 backoff periods. However, we assume that the pending address field can contain up to seven short addresses.

The packet size includes all physical layer and MAC layer headers, and it is expressed as a multiple of unit backoff periods. We also assume that the physical layer header has 6 bytes and that the MAC layer header and Frame Check Sequence fields have a total of 9 bytes. Therefore, the minimum MAC and physical layer header is 15 bytes or 1.5 backoff periods. Such a short MAC header implies that the destination addressing mode subfield (bits 10-11) within the frame control field is set to 0 and that the source addressing mode field (bits 14-15) is set to short address mode. This means that packet is directed to the coordinator with the PAN identifier as specified in the source PAN identifier field.

According to the standard, the duration of the MAC command frame for a data request is 16 bytes, but we have rounded it to 20 bytes (i.e., two backoff periods) for simplicity; furthermore, both our simulation model use the backoff period as the smallest unit of granularity. In the same manner, the duration of acknowledgment was set to one backoff period, as its duration is 11 bytes.

Note that we have assumed that the MAC sublayer will retry packet transmission until the acknowledgement is received. As the standard currently prescribes a maximum of three retries, our calculations will tend to overestimate the packet service time under high loads. However, the 802.15.4 cluster is more likely to operate under low to moderate loads, and the error due to this difference will be negligible in practice.

4.2 Analytical Model of slotted CSMA-CA algorithm

The analytical model for MAC layer operation is developed based on the network model and the slotted CSMA-CA algorithm described in Section 3.1 and 4.1. We consider a PAN with n devices in non-saturation regime, in which each network node accepts new packets via a finite size buffer. When the buffer is empty, the device will not attempt any transmission; when the buffer is full, the device will simply reject new packets coming from the upper layers of the protocol stack.

Clearly, the packet queue in the device buffer should be modeled as a $M/G/1/K$

queuing system. In our model packet arrivals follow the Poisson process with the average arrival rate λ_i. The length of the device buffer is L packets. An important characteristic of such system is that the probability π_0 that the queue is empty immediately after the packet departure is not equal to the probability P_0 that the queue is empty at an arbitrary time.

Let the PGF of the data packet length be $G_p(z) = \sum_{k=2}^{12} p_k z^k$, where p_k denotes the probability of the packet size being equal to k backoff periods or $10 \cdot k$ bytes. Then, the mean data packet size is $G_p'(1)$ backoff periods.

Let the PGF of the time interval between packet transmission and subsequent acknowledgement be $t_{ack}(z) = z^2$; actually its value is between *aTurnaroundTime* and *aTurnaroundTime* + *aUnitBackoffPeriod* [17], but we round the exponent to the next higher integer for simplicity. Also, let $G_a(z) = z$ stands for the PGF of the acknowledgment duration. We note that the timing prescribed by the standard precludes the possibility that an acknowledgment will collide with the transmission of a packet from another device.

Then, the PGF for the total transmission time of the data packet will be denoted with $D_d(z) = z^2 G_p(z) t_{ack}(z) G_a(z)$, while its mean value is $\overline{D_d} = 2 + G_p'(1) + t_{ack}'(1) + G_a'(z)$.

To analyze the behavior of the PAN in this case, we will introduce a number of random variables. First, $b(t)$ represents the value of the backoff time counter which at the beginning of the backoff period can take any value in the range $0 \mathinner{..} 2^{BE} - 1$. When the counting starts, it decrements at the boundary of each backoff unit period. The value $b(t)$ will be frozen during the inactive portion of the beacon interval, and countdown will resume when the next superframe begins.

Second, $n(t)$ represents the value of NB at time t; it belongs to the range $0 \mathinner{..}$ *macMaxCSMABackoff* $- 1$, while $c(t)$ represents the value of CW at time t; it may be 0, 1, or 2.

Finally, $d(t)$ represents the current value of "delay line counter" which is started if transmission can not be finished within the current superframe. Namely, the standard

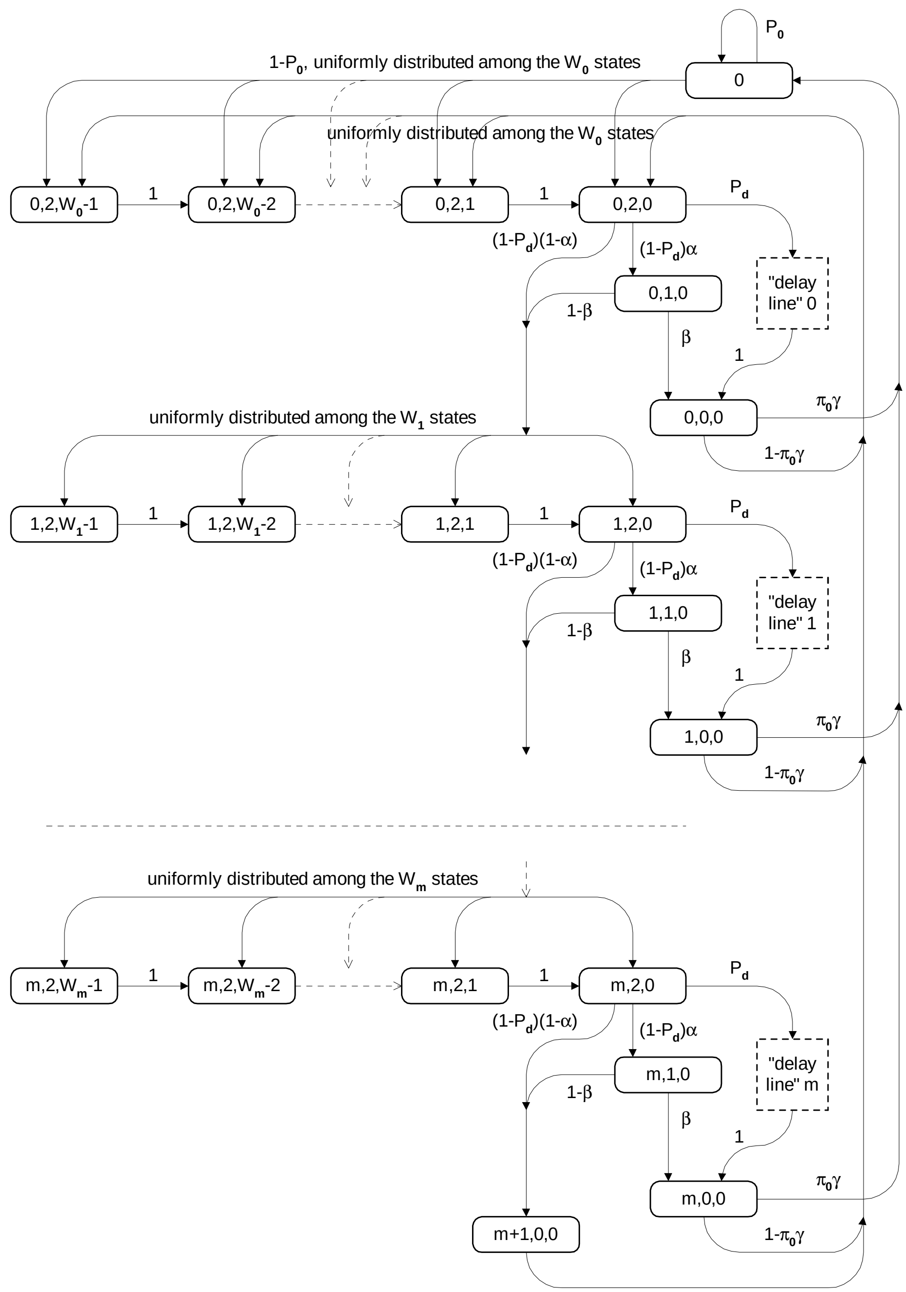


Figure 4.1: Markov chain model of slotted CSMA-CA algorithm in non-saturation regime with finite buffer.

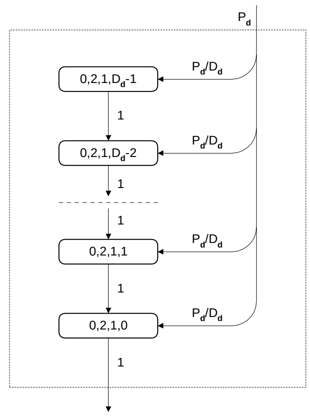


Figure 4.2: Delay lines from Fig. 4.1

prescribes that packets that cannot fit in the remaining part of the superframe have to wait until the next superframe. Assuming fixed packet size equal to G_p backoff periods (including MAC and physical layer headers), the duration of period between the packet and its acknowledgement equal to $SIFS$ backoff periods, and the duration of acknowledgement packet equal to G_{ack}, the number of backoff periods which are necessary to complete transmission in the current superframe is equal to $D_d = 2 + G_p + SIFS + G_{ack}$ (we neglect the beacon frame size here). The probability that the MAC sublayer will be unable to complete transmission and, thus, have to go through the delay line, is $P_d = D_d/SD$, where SD is superframe size.

The process $\{n(t), c(t), b(t), d(t)\}$ defines the state of the device at backoff unit boundaries. The discrete-time Markov chain which depicts this process is presented in Fig. 4.1.

The non-null transition probabilities can be described with the following equations:

$$
\begin{aligned}
P\{0,2,k \mid i,0,0\} &= \frac{1-\pi_0\gamma}{W_0}, \\
&\qquad \text{for } \ i=0\,..\,m;\ k=0\,..\,2^{BE}-1 \\
P\{0,2,k \mid 0\} &= \frac{1-P_0}{W_0}, \\
&\qquad \text{for } \ i=0\,..\,m;\ k=0\,..\,2^{BE}-1 \\
P\{0 \mid i,0,0\} &= \pi_0\gamma, \\
&\qquad \text{for } \ i=0\,..\,m \\
P\{0,2,k \mid m+1,0,0\} &= \frac{1}{W_0}, \\
&\qquad \text{for } \ k=0\,..\,2^{BE}-1 \\
P\{i,2,k-1 \mid i,2,k\} &= 1, \\
&\qquad \text{for } \ i=1\,..\,m;\ k=1\,..\,2^{BE}-1 \\
P\{i,1,0 \mid i,2,0\} &= \alpha(1-P_d), \\
&\qquad \text{for } \ i=0\,..\,m \\
P\{i,0,0 \mid i,1,0\} &= \beta, \\
&\qquad \text{for } \ i=0\,..\,m \\
P\{i+1,2,k \mid i,2,0\} &= \frac{(1-\alpha)(1-P_d)}{W_{i+1}}, \\
&\qquad \text{for } \ i=0\,..\,m;\ k=0\,..\,2^{BE}-1 \\
P\{i+1,2,k \mid i,1,0\} &= \frac{1-\beta}{W_{i+1}}, \\
&\qquad \text{for } \ i=0\,..\,m;\ k=0\,..\,2^{BE}-1 \\
P\{i,2,0,l \mid i,2,0\} &= \frac{P_d}{D_d}, \\
&\qquad \text{for } i=0\,..\,m;\ l=0\,..\,D_d-1 \\
P\{i,2,0,l-1 \mid i,2,0,l\} &= 1, \\
&\qquad \text{for } i=0\,..\,m;\ l=0\,..\,D_d-1 \\
P\{i,0,0 \mid i,2,0,0\} &= 1, \\
&\qquad \text{for } i=0\,..\,m
\end{aligned}
\tag{4.1}
$$

To reduce the notational complexity, we have shown the last tuple member $d(t)$ only within the "delay line" when it can be non-zero, and omitted it otherwise. The idle state with no packets to transmit is denoted only as state x_0. The probabilities $P\{n(t+1) = i, c(t+1) = j, b(t+1) = k-1, d(t+1) = l-1 \mid n(t) = i, c(t) = j, b(t) = k, d(t) = l\}$ are written simply as $P\{i, j, k-1, l-1 \mid 0, j, k, l\}$. Also, the constant *macMaxCSMABackoff* which represents the maximum value of the variable *NB* is denoted with m. let W_0 stand for $2^{macMinBE}$, and let i represents the current value of *NB* during the execution of the algorithm ($i = 0 \,..\, m$). The maximum value of the random waiting time (expressed in units of backoff periods) that corresponds to i will be $W_i = W_0 2^{min(i, 5-macMinBE)}$; note that the value of *BE* is limited to *aMaxBE* $= 5$. (The detailed explanation of all the parameters can be found in the standard [17].)

The first transition probability in the set (4.1) represents the probability to choose a random duration of the backoff period after a channel access. Since the range to choose from is 0 to $2^{BE_{min}} - 1$, this probability is equal to $1/W_0$. Note that the random backoff period always precedes the packet transmission, regardless of whether the packet to be transmitted is brand new or it is simply an earlier packet that could not be transmitted due to collision.

The second probability corresponds to the case when the previous attempt to transmit a packet has been unsuccessful, and the device begins to perform the algorithm again; this probability is equal to $1/W_0$. The third equation shows the transition probability to inactive state. The fourth is the transition probability to the retry attempt. The fifth equation shows the probability that the backoff time is decremented after each *aUnitBackoffPeriod.*

The sixth equation determines the probability α that the channel is sensed to be idle when the backoff counter reaches zero. The seventh equation determines the probability β that the channel is sensed to be idle after it was already sensed idle for one backoff period. Note that, when the backoff counter reaches zero and CCA senses a busy channel, the ongoing packet transmission may have started one or more backoff periods earlier. However, when the first CCA senses that the medium is idle but the second one finds it busy, that packet transmission must have started in that same backoff period.

Therefore, the corresponding probabilities that the medium is not idle differ, and so do the probabilities α and β.

The eighth and ninth equations describe the probabilities that the device chooses another random backoff in the range $0 \mathinner{..} W_{i+1}-1$ upon sensing the channel to be busy. The last three equations describe the delay line which is entered if MAC does not have enough backoff periods to complete transmission in the current superframe.

Let $x_{i,j,k,l} = \lim_{t\to\infty} P\{n(t) = i, c(t) = j, b(t) = k, d(t) = l\}$ for $i = 0 \mathinner{..} m, j = 0, 1, 2, k = 0 \mathinner{..} 2^{BE}-1, l = 0 \mathinner{..} D_d - 1x$, be the stationary distribution of the chain (when l=0, it will be omitted). In order to simplify the notation, let us first derive the auxiliary variables C_1, C_2, and C_3.

$$\begin{aligned}
x_{0,1,0} &= x_{0,2,0}(1-P_d)\alpha = x_{0,2,0}C_1 \\
x_{1,2,0} &= x_{0,2,0}(1-P_d)(1-\alpha\beta) = x_{0,2,0}C_2 \\
x_{0,0,0} &= x_{0,2,0}((1-P_d)\alpha\beta + P_d) = x_{0,2,0}C_3
\end{aligned} \tag{4.2}$$

Then, the stationary probability distribution of this Markov chain may be described with:

$$\begin{aligned}
x_0 &= x_{0,0,0}\frac{\pi_0\gamma\left(1-C_2^{m+1}\right)}{(1-P_0)(1-C_2)} \\
x_{i,0,0} &= x_{0,0,0}C_2^i, \\
&\qquad \text{for } \ i = 0 \mathinner{..} m \\
x_{i,2,k} &= x_{0,0,0}\frac{W_i - k}{W_i}\cdot\frac{C_2^i}{C_3}, \\
&\text{for } \ i = 1 \mathinner{..} m; \ k = 0 \mathinner{..} W_i - 1 \\
x_{i,1,0} &= x_{0,0,0}\frac{C_1 C_2^i}{C_3}, \\
&\qquad \text{for } \ i = 0 \mathinner{..} m \\
x_{0,2,k} &= x_{0,0,0}\frac{W_0 - k}{W_0 \ C_3}, \\
&\qquad \text{for } \ k = 1 \mathinner{..} W_0 - 1 \\
x_{m+1,0,0} &= x_{0,0,0}\frac{C_2^{m+1}}{C_3} \\
\sum_{l=0}^{D_d-1} x_{i,2,0,l} &= \frac{x_{0,0,0}C_2^i P_d(D_d-1)}{2C_3}
\end{aligned} \tag{4.3}$$

The value of $x_{0,0,0}$ may be obtained from the constraint that the sum of all the probabilities in the Markov chain must be equal to one,

$$\begin{aligned} x_0 + \sum_{i=0}^{m} \sum_{k=0}^{W_i-1} x_{i,2,k} + \sum_{i=0}^{m} x_{i,0,0} + \sum_{i=0}^{m} x_{i,1,0} \ + \\ x_{m+1,0,0} + \sum_{i=0}^{m} \sum_{l=0}^{D_d-1} x_{i,2,0,l} \ = \ 1 \end{aligned} \tag{4.4}$$

Then, the total probability to access the medium is

$$\tau = \sum_{i=0}^{m} x_{i,0,0} = x_{0,0,0} \frac{1 - C_2^{m+1}}{1 - C_2} \tag{4.5}$$

However, the probability to access the medium when the transmission is deferred to the next superframe due to insufficient time, $\tau_1 = \frac{P_d}{C_3}\tau$, differs from the probability to access the medium in the current superframe, $\tau_2 = \left(1 - \frac{P_d}{C_3}\right)\tau$. At any moment, q stations out of n are not-delayed due to insufficient space and $n-1-q$ are delayed to the start of next superframe. Numbers q and $n-1-q$ follow a binomial distribution with probability $P_q = \binom{n-1}{q}(1-P_d)^q P_d^{n-1-q}$.

In order to calculate the probability α that the medium is idle on the first CCA test, we have to find the mean number of busy backoff periods within the superframe; this number will be divided into the total number of backoff periods in the superframe wherein the first CCA can occur. Note that the first CCA will not take place if the remaining time in the superframe is insufficient to complete the transaction, which amounts to $SD - \overline{D_d} + 1$ backoff periods. Then, the probability that any (one or more) packet transmissions will take place at the beginning of the superframe is $n_1 = 1 - (1-\tau_1)^{(n-1-q)}(1-\tau_2)^q$, and the number of busy backoff periods due to these transmissions is $n_1(G_p'(1) + G_a'(1))$.

The occupancy of the medium after the first transmission time can be found by dividing the superframe into chunks of $\overline{D_d}$ backoff periods and calculating the probability of transmission within each chunk. As the total arrival rate of non-deferred packets is $q\tau_2$, the probability that the number of transmission attempts during the period $\overline{D_d}$ will be non-zero is $n_2 = \overline{D_d}\, q(\tau_2)$. The total number of backoff periods in which the first CCA can be occur is $SD - \overline{D_d} + 1$. The probability that the medium is idle at the first CCA is

$$\alpha = \sum_{q=0}^{n-1} P_q \left(1 - \left(\frac{n_1 \overline{D_d}}{SD - \overline{D_d} + 1} + \frac{n_2(SD - 2\overline{D_d} + 1)}{SD - \overline{D_d} + 1}\right) \cdot \frac{(G'_p(1) + G'_a(1))}{\overline{D_d}}\right) \tag{4.6}$$

The probability that the medium is idle on the second CCA for a given node is, in fact, equal to the probability that neither one of the remaining $(n-1)$ nodes, nor the coordinator, have started a transmission in that backoff period. The second CCA can be performed in any backoff period from the second backoff period in the superframe, up to the period in which there is no more time for packet transmission, which amounts to $SD - \overline{D_d} + 1$. The probability in question is

$$\beta = \sum_{q=0}^{n-1} P_q \left(\frac{1}{SD - \overline{D_d} + 1} + \frac{(1-\tau)^{(n-1)}}{SD - \overline{D_d} + 1} + \frac{SD - \overline{D_d} - 2}{SD - \overline{D_d} + 1}(1-\tau)^q\right) \tag{4.7}$$

Finally, the probability γ that a packet will not collide with other packet(s) that had successful first and second CCAs can be calculated as the probability that there are no accesses to the medium by the other nodes or the coordinator during the period of one complete packet transmission time. (Note that a collision can happen in $SD - \overline{D_d} + 1$ consecutive backoff periods starting from the third backoff period in the superframe.)

$$\gamma = \sum_{q=0}^{n-1} P_q \left(\frac{(1-\tau)^{(G'_p(1)+2+t'_{ack}(1)+G'_a(1))(n-1)}}{SD - \overline{D_d} + 1} + \frac{SD - \overline{D_d}}{SD - \overline{D_d} + 1}(1-\tau)^{(G'_p(1)+2+t'_{ack}(1)+G'_a(1))q}\right) \tag{4.8}$$

4.3 Simulation Model

To investigate the performance of an 802.15.4 compliant network through simulation modeling, we use a simulator which was built using the object-oriented Petri net simulation engine Artifex by RSoft Design, Inc. [1]. This tool is suitable to model discrete event systems. Using Artifex, developers can easily build non ambiguous models of a system and validate them by running on a simulator. This language allows developers to integrate standard programming languages like C and C ++ code within the Artifex language environment.

Case of uplink transmissions only

We observe the behavior of an IEEE 802.15.4-compliant network both in saturation condition and non-saturation condition. In saturation condition, it is assumed that the devices always have a packet waiting to be transmitted. This assumption corresponds to the similar approach to modeling 802.11 networks presented in [4]. Fig. 4.3 shows the MAC performance in saturation condition as the function of the number of devices and packet sizes.

Figs. 4.3(a) and 4.3(b) show that the probabilities τ_1 and τ_2 for the delayed packets, deferred to the next superframe due to lack of space in the current superframe, and for the nondelayed packets, respectively. With increasing packet size, τ_1 is increasing while τ_2 is decreasing. The observation implies that larger packet size increases the total transmission time D_d. Larger D_d increases the probability of deferred transmission to the next superframe. When two or more nodes defer their transmissions, they find a free channel in the first two backoff periods following the beacon frame. As a result, they start transmissions in the third backoff period of the current superframe, thus causing collision. The affected nodes then reattempt their transmissions by setting the three random variables NB, CW, and BE to their initial values, and starting random backoff countdown to proceed the CSMA-CA algorithm. With the increasing value of NB, BE increases and the corresponding range (0 to $2^{BE} - 1$) of random waiting time increases. Since a larger number of backoff periods D_d is wasted at the beginning of the superframe due to collision, the probability that most of the nodes finish their countdown closer to the end of the superframe by generating shorter random number with lower value of NB. As a result, the probability of deferred transmissions increases with a lower value

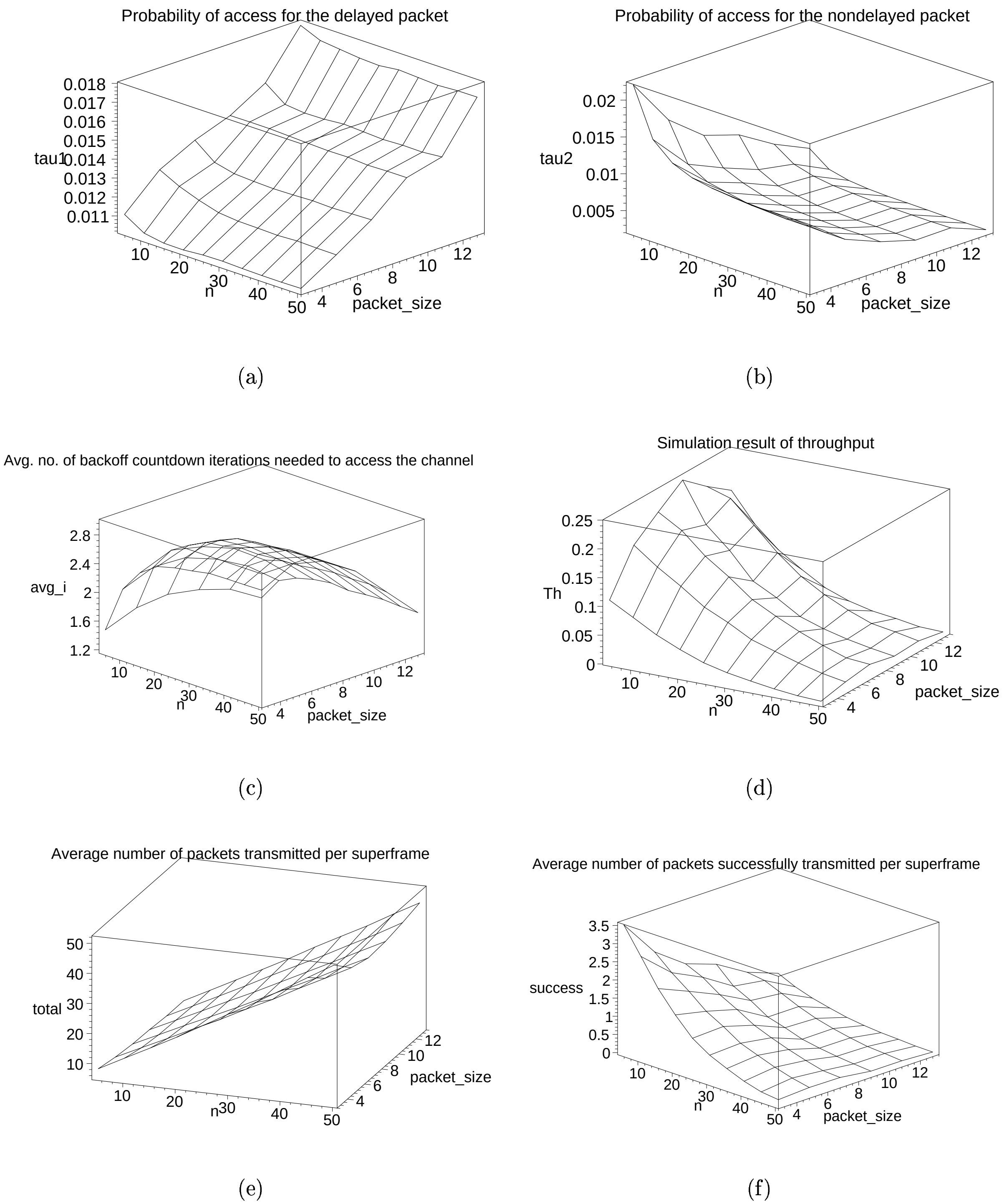


Figure 4.3: Probabilities of access the medium for delayed and nondelayed packets, average no. of iterations needed to access the medium, throughput, average packet transmissions per superframe, and average successful packet transmissions per superframe in saturation condition.

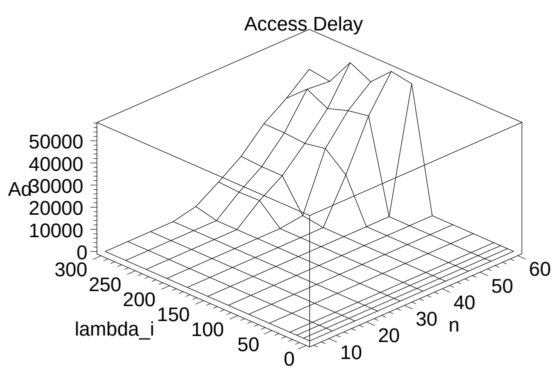
of NB or we can say that with a lower order of backoff countdown iteration. Fig. 4.3(c) validates the observation.

We will discuss the behavior of deferred transmission in more detail. For example, $D_d = 16$ backoff periods are required to transmit a comparatively larger packet of size 110 bytes. Since a superframe consists of only 48 backoff periods, at most 2 packets can be transmitted during each superframe considering the time needed to execute CSMA-CA algorithm. When two or more devices defer transmissions, a collision occurs at the beginning of the superframe that wastes 16 backoff periods. The devices then take attempt to retransmit and can access the channel only if they can perform the first CCA within the next 16 backoff periods among the remaining 32 backoff periods, and find an idle channel. The devices that generate the smallest random number in the first countdown iteration, succeed to access the channel. Other devices find a busy channel and generate another random number with next higher order iteration. The biggest random numbers that a device can generate at the first two backoff iterations are 7 and 15. Therefore, the probability to generate a summation of random numbers greater than 16 at the first two iterations or the probability to defer transmission is 0.164. With probability 0.836, the devices may generate another longer random number (since BE = 5 in third or fourth iteration) that ultimately leads the devices to finish backoff countdown closer to the end of the next superframe, and thus increase the probability of deferral. On the other hand, if collision occurs at the second 16 backoff periods then the devices participating in collision defer their transmissions with probability 1 after executing the random countdown at the first backoff iteration. Therefore, each collision increases the probability of deferred transmission, thus leading to even more collisions and more deferred transmissions with lower order iteration.

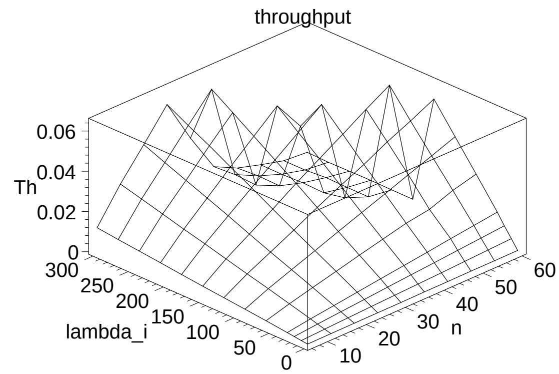
More collisions also occur when large number of devices participates in the network in saturation condition as shown in Figs. 4.3(e) and 4.3(f). We notice that with the increasing number of devices the average number of transmissions per superframe increases and the number of successful transmissions per superframe decreases. With larger packet size, most transmissions occur at the beginning of the superframe while for shorter packet size transmissions occur throughout the superframe. Since all devices always have a packet ready for transmission, the probability to generate the same random number in the first iteration of the countdown procedure increases with increasing number of devices. Collision will happen only if more than one device generates the same smallest random number as the first backoff value. The probability to generate the same smallest random number increases when more than 7 devices participate in the network since the upper limit of the backoff value of first iteration is 7. Furthermore, the participation of large number of devices in a collision increases the probability of another collision. As a result, we can say that large number of devices implies large number of transmissions and large number of collisions. The highest throughput (around 25%) shown in Fig. 4.3(d) occurs with the packet size 9 and with the number of devices around 5, where the highest successful transmission (3.5 packets per superframe) occurs with the packet size 3 and the same number of devices shown in Fig. 4.3(f).

We also observe the MAC performance in non-saturation condition and finite buffer condition assuming that packet arrivals follow the Poisson process with average arrival rate λ_i. The length of the device buffer is set to $L = 3$ packets. Figs. 4.4, 4.5, and 4.6 present access delay, throughput, and blocking probability for a varying number of devices and packet arrival rate for packet sizes 3, 9, and 13 backoff periods, respectively. Packet arrival rate is expressed in packets per minute i.e., the label "60" corresponds to 60 packets per minute, or one packet per second.

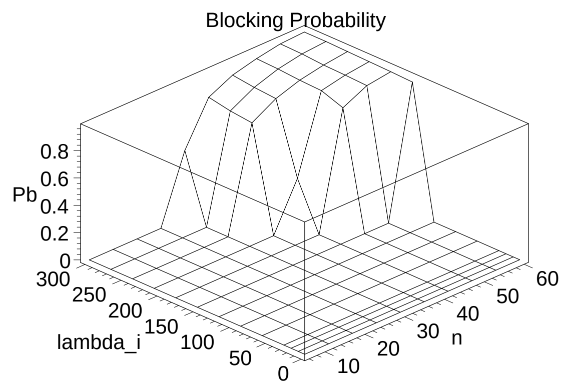
By analyzing the figures shown in Fig. 4.4, we can say that the network operates in unstable condition when more than 20 devices participate with high arrival rate. Beyond certain arrival rate, very few packets or almost no packets are transmitted successfully which declines the throughput abruptly shown in Fig. 4.4(b). As a result, retransmission of a packet occurs repeatedly. This ultimately causes the drop of newly arrived packets due to buffer overflow and increases the blocking probability (Fig. 4.4(c)). The access delay that includes the queuing delay and service time of a packet considering all retrans-



(a)



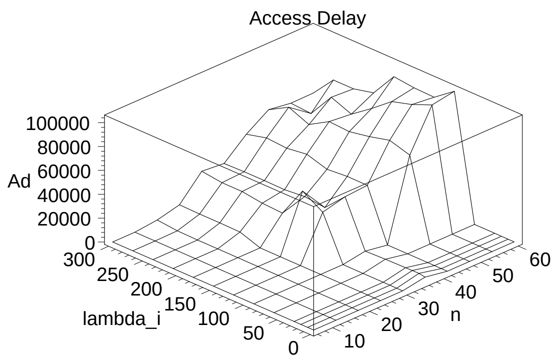
(b)



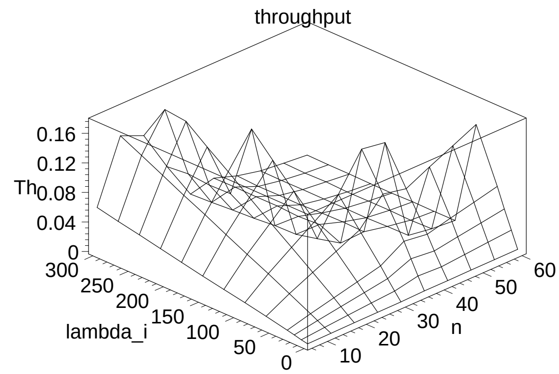
(c)

Figure 4.4: MAC performance as the function of the number of devices and packet arrival rate with packet size 3 in non-saturation condition.

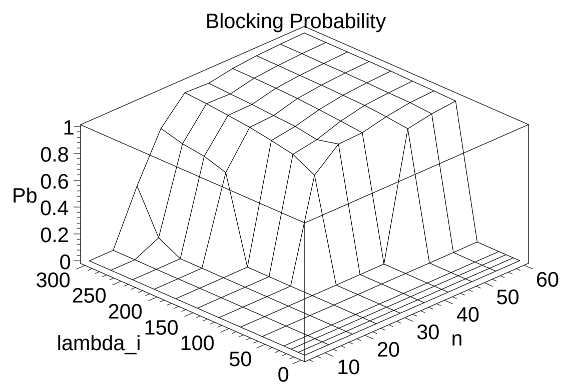
mission raises all of a sudden to a very high value. Similar situation also happens when we consider packet size 9, and 13. The larger packet size leads the network to operate in unstable condition with lower number of devices and lower arrival rate compare to lower packet size.



(a)



(b)



(c)

Figure 4.5: MAC performance as the function of the number of devices and packet arrival rate with packet size 9 in non-saturation condition.

Case of Uplink and Downlink transmissions

When we consider both uplink and downlink transmissions which is presented in Section 3.2, we see the following states can be identified for the PAN coordinator node:

1. The coordinator may be transmitting the beacon.

2. The coordinator may be listening to its nodes and receiving data or request packets.

3. The coordinator may be transmitting the downlink data packet as a result of pre-

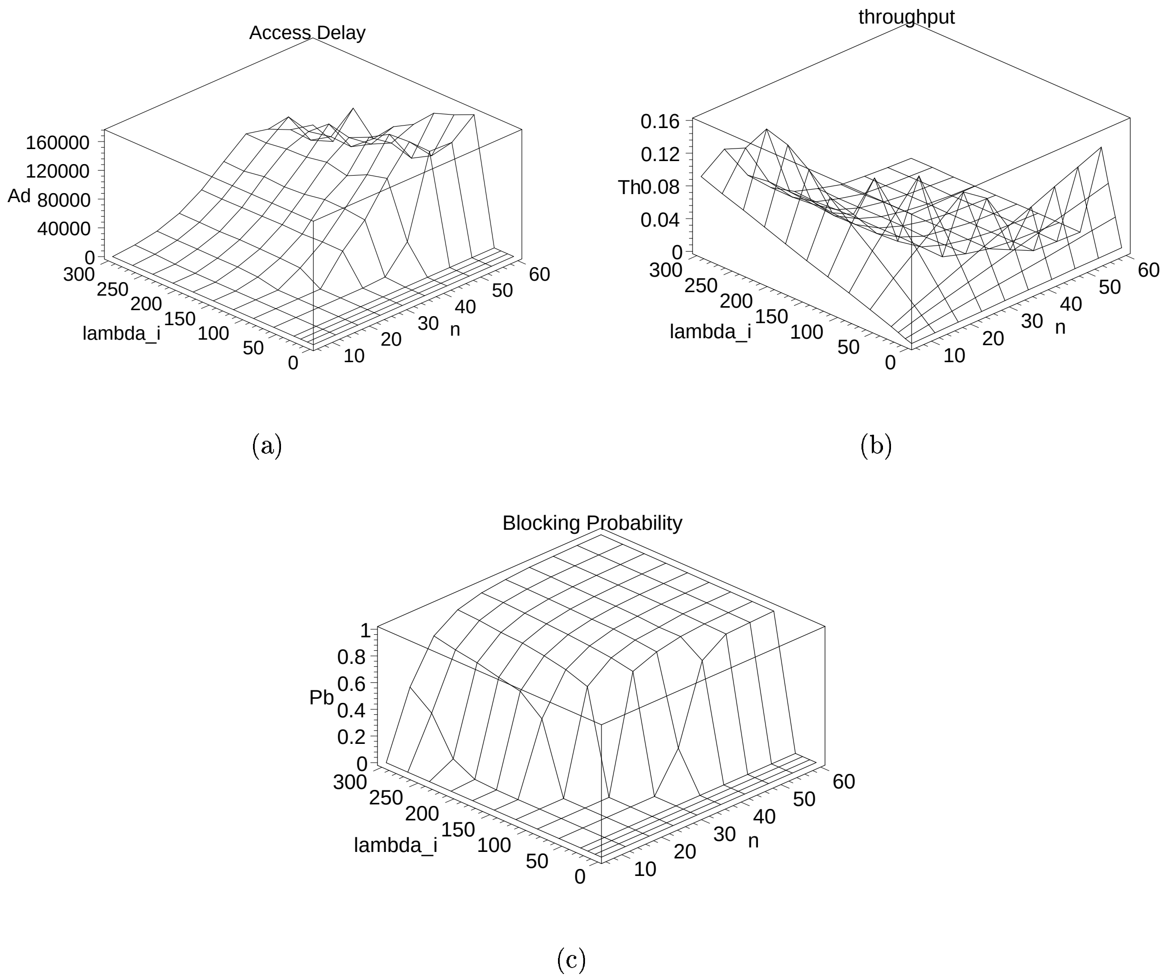


Figure 4.6: MAC performance as the function of the number of devices and packet arrival rate with packet size 13 in non-saturation condition.

viously received request packet. As soon as downlink transmission is finished coordinator switches to the listening mode.

Similarly, an arbitrary (non-coordinator) node in the network can be in one of the following states:

1. The node may be transmitting an uplink data packet.

2. The node may be transmitting an uplink request packet.

3. The node may be waiting for a downlink packet.

4. The node may also be in an idle state, without any downlink or uplink transmission pending or in progress.

After defining the states, we assume that each device will operate in non-saturation conditions and finite buffer conditions. The buffer can be empty or nonempty. A device discards any new packet when the buffer already full. Packets waiting in the queue are served as first come, first served basis. We also consider the interarrival time between two consecutive packets and we assume that they are exponentially distributed. The PAN coordinator maintains finite buffer for each device in the downlink queue.

After a successful uplink or downlink transmission, the node enters into the idle state if both downlink and uplink data queues for the device are empty. The node will leave the idle state upon the arrival of a packet to either queue. In case of simultaneous packet arrival to both queues, the downlink transmissions have priority over the uplink ones.

Each downlink transmission must be preceded by successful transmission of a data request packet. Those packets may experience collisions, or they may arrive while the coordinator is executing backoff countdown and thus will be ignored. Upon receipt of a request, the coordinator will acknowledge it; the absence of acknowledgment means that the node must repeat the request transmission procedure.

If the downlink transmission is not successful because of collision or a time-out at the destination node, the coordinator will not repeat the process again. After waiting a *macAckWaitDuration* period, the coordinator switches to receiving mode and listen to the data packet or MAC command request packet. Note that, after a successful transmission of MAC command frame the node will turn on it's receiver. If the duration of the downlink packet service time exceeds $aMaxFrameResponseTime = 61$ backoff periods, the node will time-out and the packet will not be received; consequently, the request has to be repeated.

If the downlink transmission was successful and the downlink queue towards the node is not empty, node will start a new downlink transmission cycle. If the downlink queue was empty but the uplink queue contained a packet, the node will initiate the uplink transmission cycle. Due to the priority considerations, the uplink data transmission will be started only if the downlink data queue is empty. If there was a downlink packet arrival during the uplink transmission, then as soon as the uplink transmission was finished, the node will synchronize with the beacon and attempt transmission of a request packet.

Since the standard allows at most seven stations to be advertised in the beacon, we assume that the coordinator will advertise nodes in round-robin fashion in the case if it has more than seven downlink packets.

Fig. 4.7 shows the probabilities that the medium is idle on the first CCA, second CCA, the probability of success γ of overall transmission , the probability of blocking the incoming request and data packets by the PAN coordinator, and the probability of receiver time-out at a destination node after a *aMaxFrameResponseTime* period.

As can be seen, α, β, and γ reach lower (saturation) bounds at moderate loads for network size between 10 and 20 nodes. The higher blocking probability (above 80%) impinges on the lower bound for the success probability close to zero, which means that, in this regime, virtually no packet is able to reach its destination. This may be explained by packet collisions and blocking at the coordinator, which decrease the number of downlink packets to be processed by the coordinator and reduce the impact of downlink transmissions.

Fig. 4.8 shows the uplink and downlink access probabilities. The flattening of uplink access probability indicates that the onset of saturation regime, in which case all accesses to the medium are contributed by the request packets that do not succeed. A rather dramatic decrease of downlink access probability for the coordinator may be observed as well; it is caused by the inability of the coordinator to receive any correct data requests due to collisions and blocking.

This observation is also confirmed by the diagrams that depict the throughput and the uplink service time. As can be seen, both the throughput and the uplink service times deteriorate rapidly when the network enters saturation.

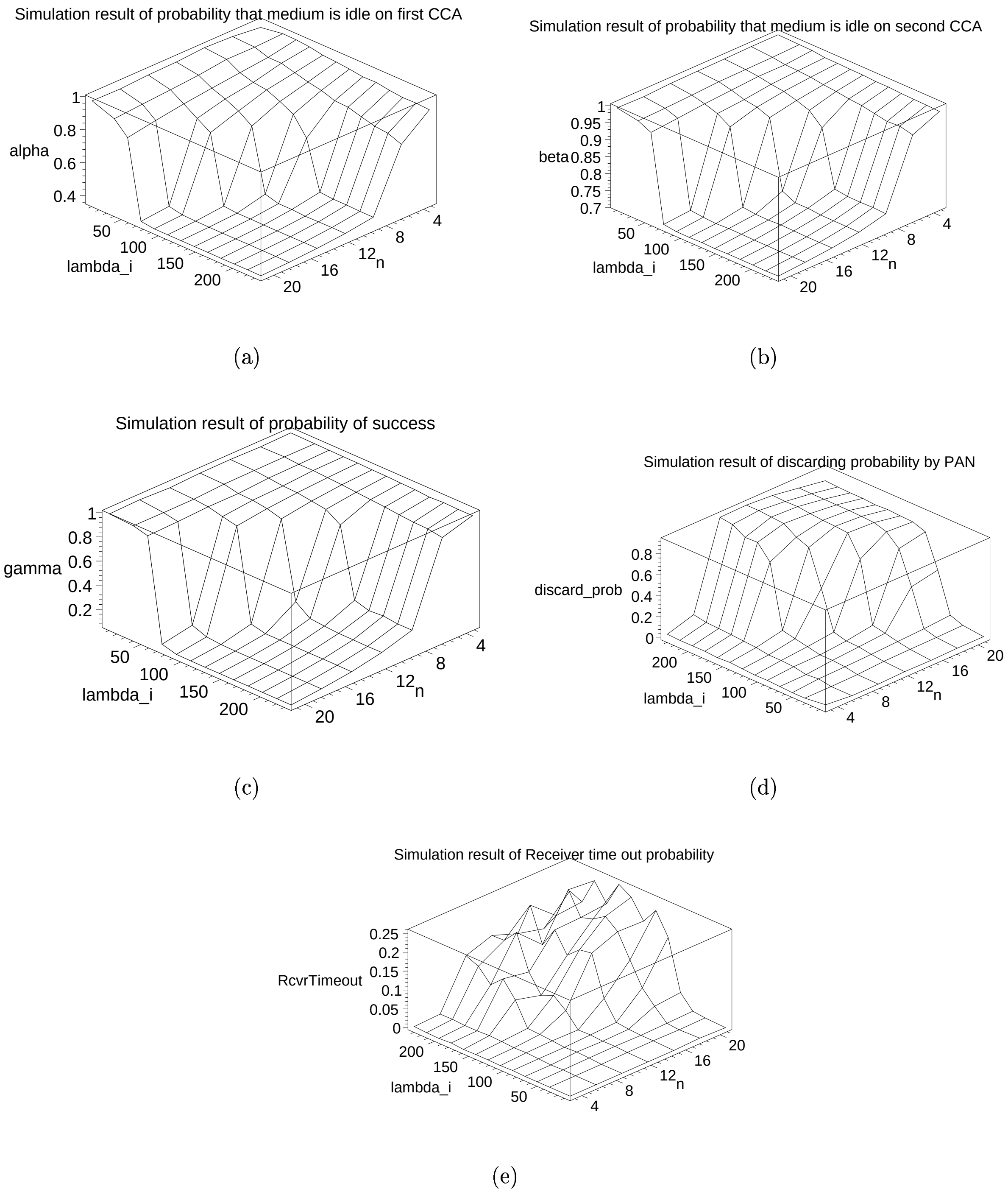


Figure 4.7: Probabilities that medium is idle on first, second CCA, probability of success, the blocking probability at the coordinator, and the time-out probability at the node for default MAC parameters *macMaxCSMABackoffs*= 4 and *macMinBE*= 3.

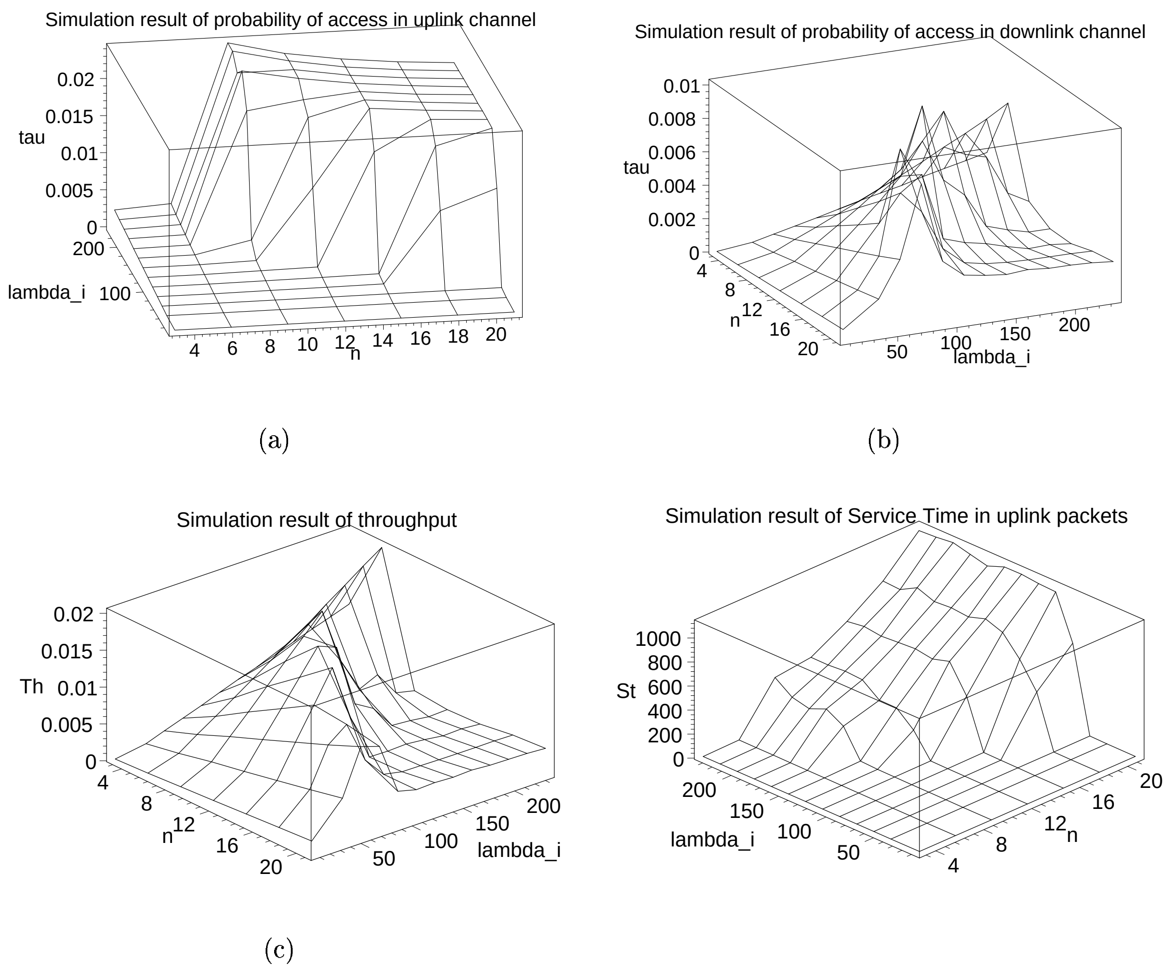


Figure 4.8: Uplink and downlink access probabilities, throughput, and uplink packet service time for default MAC parameters *macMaxCSMABackoffs*= 4 and *macMinBE*= 3.

Chapter 5

Avoiding the bottlenecks of the 802.15.4 MAC

5.1 Bottlenecks of the MAC layer

From the performance analysis of the MAC discussed in Chapter 4, a number of problems can be identified that may affect performance.

5.1.1 Congestion of deferred uplink transmissions

The first of those problems occurs in the situation wherein two or more nodes have to defer their transmissions because the remaining time in the current superframe is insufficient. All such nodes will start their CCAs immediately following the beacon frame. In the first two backoff periods, the channel will be found idle. Consequently, all the nodes will conclude that the channel is free and start their transmission in the third backoff period. This will result in a collision, and the CSMA-CA algorithm in all affected nodes will revert to the random backoff countdown phase.

Furthermore, the probability that a packet transmission will be deferred depends on the index of the current backoff countdown iteration. As the NB increases, so does BE and the corresponding range of the possible waiting time. Longer random backoff countdowns will finish closer to the end of the superframe, and thus increase the probability that the remaining time would not suffice for the two CCAs, packet transmission, and acknowledgment. Therefore, each collision increases the probability of deferral, thus leading to even more collisions later on.

5.1.2 Underutilization of certain portions of the superframe

Further consequence of the deferral of transmissions is the reduced utilization of the medium, as the probability of a transmission in a given backoff interval will decrease towards the end of the superframe. Also, the mandatory two CCAs before even attempting the transmission mean that no packet can be transmitted in the first two backoff periods of the superframe. However, the reduction in utilization caused by these effects will be noticeable only for short superframes (e.g., when the *macSuperFrameOrder* has a value of zero, superframe duration is only 48 backoff periods). For longer superframes, this reduction will not cause significant losses.

5.1.3 Congestion of data requests

A similar kind of problem may occur when the coordinator has pending packets for two or more of the nodes. As explained above, the beacon frame can accommodate at most seven such nodes. All nodes that recognize their address in the beacon frame, but have no pending uplink data transmission in progress, will immediately start the MAC algorithm from Fig. 3.1.1. The first step in this algorithm is the random backoff countdown; since the range of the first iteration of the countdown procedure is small (0 . . 7 only), two or more nodes may choose the same number of backoff periods for the countdown. After the countdown and the two required CCAs, such nodes may start their transmission simultaneously and collide. A similar situation may occur in the second and higher-order iterations, but the corresponding probabilities are smaller.

5.1.4 Interaction of data requests with deferred transmissions

Furthermore, packet requests for downlink transmissions might collide with the deferred data transmissions from the previous superframe. The collisions of this type are effectively a combination of the previous two. It should be noted that the frequency of all aforementioned types of collisions will increase with traffic intensity.

5.1.5 Blocking of data requests

The third problem is somewhat subtler, as it is due to the fact that a typical 802.15.4 coordinator is likely to have a single radio interface. (While this holds for other nodes in the cluster as well, the constraints imposed by the beacon-enabled mode of operation make the limitations of the coordinator more critical.) As explained above, when the coordinator announces a pending packet for a node, this node will send a data request packet using the procedure from Fig. 3.2(b). Upon successful reception of the request, the coordinator will acknowledge it and immediately begin the random backoff countdown in order to send the data frame. During the countdown, the coordinator will not receive subsequent requests, even though it may have switched its radio to reception. (The switchover is certainly needed for subsequent CCAs, however it does mean that no more acknowledgments can be sent, even if received without collision; and the standard does not mention that incoming requests are to be queued or registered in any way.)

In this manner, all data requests besides the first one are effectively blocked, even in the absence of any collisions, and will have to be retransmitted. Retransmission, by default, consumes bandwidth and increases the probability of further collisions. Due to the blocking of requests, the pending downlink data transfers will be delayed, which will compromise the stability of the corresponding queue at the coordinator, or incur the risk of buffer overflow and data loss in case this queue is implemented with a finite buffer.

5.2 Possible remedies

The observations in Chapter 4 indicate that the servicing policy of the coordinator is critical to achieve acceptable network performance when both uplink and downlink transmissions are considered (Section 4.3). Two parameters seem critical: blocking of uplink data requests by the coordinator and the default time-out at individual nodes of 61 back-off periods. If those parameters could be changed, the stable region of the network could be extended and the problem defined in Section 5.1.5 could be reduced.

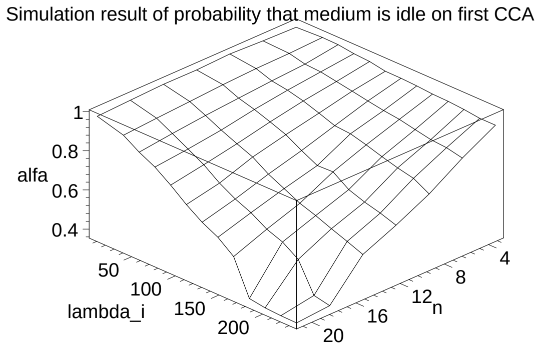
In order to verify this observation, we have modified the simulator. As for the blocking of data request packets, a simple solution is to allow the coordinator to listen to such packets throughout the countdown that leads to the transmission of acknowledgment packets. (As mentioned above, its radio should be switched to reception anyway, as it needs to perform the two CCAs before transmitting the acknowledgment packet.)

Data request packets received during the countdown should be queued and acted upon after the current request is acknowledged. The small span of the first countdown iteration means that probably only one or two request packets may be received during the coordinator countdown, which further means that even small buffers would suffice for this purpose. However, the lack of immediate acknowledgment to second and later request packets might lead their originating nodes to conclude that those packets have been lost and initiate re-transmission, which would basically render the queuing useless (and cancel any improvement due to queuing). Fortunately, this can be catered to by simply extending of the receiver time-out. Furthermore, the avoidance of some re-transmissions would reduce the probability of collisions and improve the utilization of the medium, thus leading to improved throughput (or reduced power consumption) of the entire network.

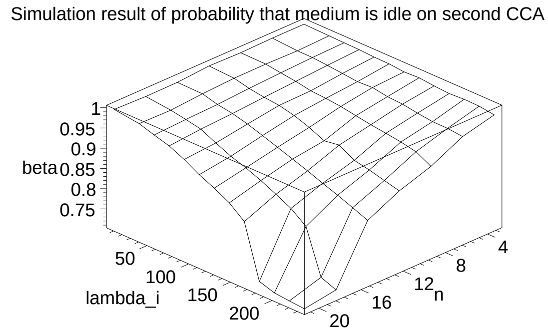
Since the coordinator now queues the requests, it is needed to follow a service discipline to serve the requests. We consider the coordinator would serve requests with FIFO discipline and at the same time we extend the receiver time-out of a node waiting for downlink transmission to 660 backoff periods instead of 61 backoff periods. The adequate time-out period prohibits a device of sending MAC command frame within a short time interval for the same downlink packet when a collision occurs. As a result, the collision occurred due to the MAC command frame is reduced. Since the PAN coordinator serves only one packet from a downlink queue during a single visit before switches to another queue without considering whether the transmssion is successful or not, the FIFO scheduling with extended time-out period reduces the probability of receiver time-out at the destination node. Therfore, the retransmission of the downlink packets is reduced which ultimately reduces the misuse of the channel.

The results that we obtained are presented in Fig. 5.1. We observe the graceful degradation of performance and the extension of the range of network parameters (i.e., the number of nodes and the packet arrival rate) in which the network operates in non-saturation regime, than with the parameters set up as stipulated in the standard. However, in practice the allowable queueing of packets at the coordinator and time-out value for listening at the node should be set according to the known ratio of uplink and downlink traffic in the network.

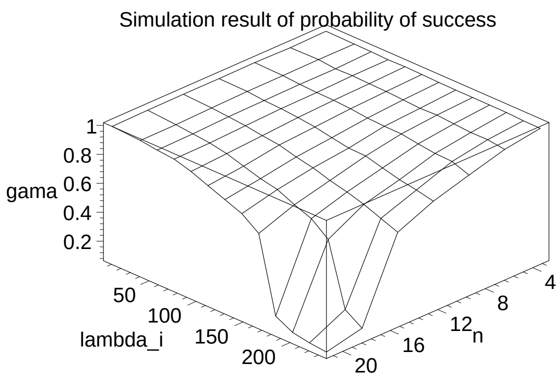
For further improvement of the MAC, we try to alleviate the contention of deferred transmissions immediately after the beginning of the superframe discussed in Section 5.1.1. The deferred packets should wait for a variable period so as to ensure that the CCAs and subsequent transmission attempts are spread over time, rather than clustered in the same backoff interval. Two simple solutions would be possible to define the variable waiting period for the deferred packets. The first solution is to make the waiting time equal to the D_d backoff periods. For illustration, we assume that two devices are trying to access the channel. In order to avoid deferred transmission, they must finish their countdown within the SD - D_d backoff periods of the current superframe, where SD is the superframe size. We assume that both devices fail to do that and one finishes countdown at the end of SD - $D_d + 2$ backoff period boundary while the other finishes countdown at the end of SD - $D_d + 4$ backoff period boundary of the current superframe. Therefore, the two devices start countdown for D_d backoff periods. The countdown procedure finishes at the second and fourth backoff boundary of the next superframe. The device that



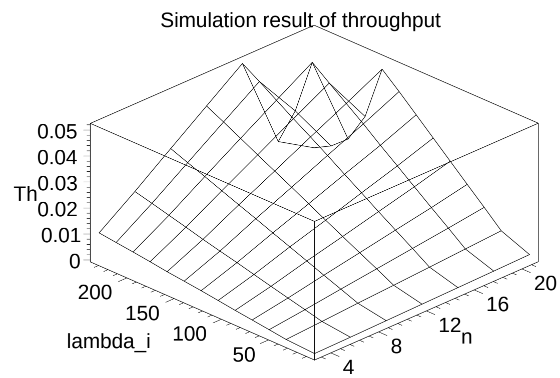
(a)



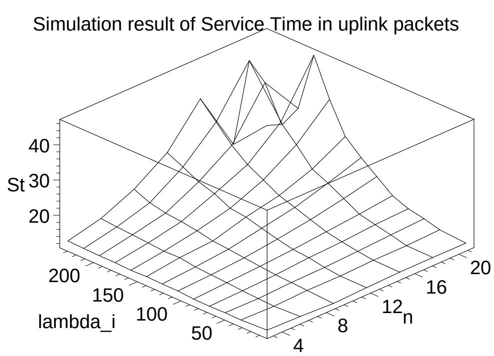
(b)



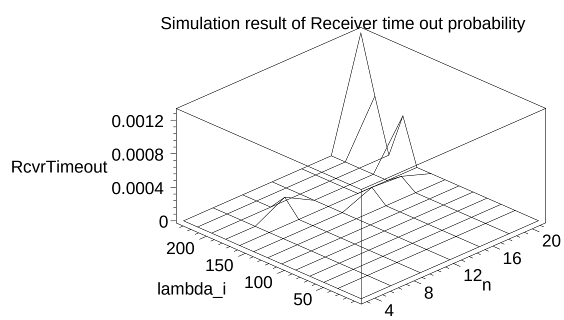
(c)



(d) Throughput with the improved MAC.



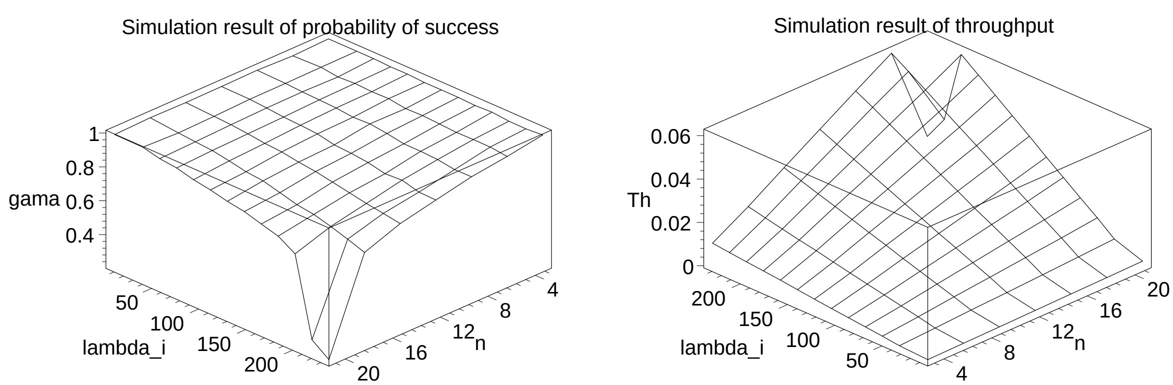
(e)



(f)

Figure 5.1: Performance with the improved MAC.

finishes countdown at the second backoff boundary would succeed to access the channel. The other device then finds a busy channel and starts CSMA-CA algorithm again, thus avoiding collision. In such case, collision may occur only if the above two devices would finish their countdown at the same backoff period in the previous superframe and would be unable to complete transmission. The results thus obtained are presented in Fig. 5.1 which clearly shows a significant improvement of the medium behavior by reducing the collisions. This solution also reduces the problem discussed in Section 5.1.2, and in Section 5.1.4 because these two problems arise as a consequence of deferred transmissions. Another solution would be to restart the random backoff countdown iteration cycle. In this case, when devices find insufficient space to complete transmission in the current superframe, they would generate another random number and start countdown in the same manner as to find a busy channel.



(a) (b)

Figure 5.2: Medium behavior with the improved MAC

Chapter 6

Duty Cycle Management

Among the most important requirements for sensor networks is the maximization of the lifetime of individual node and, by extension, the overall network lifetime. Sensor lifetime can be extended at the hardware level through the use of low power chips and high capacity power sources, but also at the network level by adjusting the frequency and ratio of active and inactive periods of individual sensor node [23]. Since the devices of the IEEE 802.15.4-compliant network are small battery powered devices, their lifetime can be extended at the network level. The approach is supported by the 802.15.4 standard [17] in its beacon enabled mode with slotted CSMA-CA, where the interval between two beacons is divided into active and inactive parts, and the sensors can switch to low power mode during the inactive period.

However, the minimum acceptable level of information flow from the network is dictated by the nature of the sensing application. In many cases, such as surveillance, health care, and structural health monitoring, continuous monitoring is required, and letting the entire network sleep for prolonged periods is simply out of the question. In such cases redundant sensors can be used, i.e. the number of sensors covering a given physical area should be larger than the minimum number based on the required data rate. Then, activity management can be applied at the level of individual sensor node, by sending them to sleep for variable time intervals. The desired packet rate received at the network sink can be achieved by adjusting the number of active sensors; at any given time, some of the sensors are active while others sleep.

Depending on the manner in which this adjustment is managed, we distinguish between centralized and distributed approaches. In the centralized approach, the network coordinator calculates the sleep intervals and instructs the sensors accordingly; while this approach allows for simpler sensors to be used, the computation time and memory requirements for the coordinator can be prohibitively high. In the distributed approach, the load is shared: the coordinator just monitors the aggregate rate of information flow and sends this information to sensors, which then use it to individually determine their sleep intervals. (We assume that the network sink—the destination toward which all sensors send the sensed data—also acts as the network coordinator.) However, in our research work we investigate both of the approaches.

6.1 Activity scheduling policies

In both approach, we consider sensing applications in which redundant sensors are used to achieve the desired value of event reliability. We assume that each sensor has a small buffer for data packets obtained by sensing the appropriate physical variable of interest. The buffer is managed in a push-out manner, i.e., the newly arrived packets are always admitted; if the buffer is full, the packet at the head of the buffer is discarded in order to make space for the newly arrived packet. In this manner, the sensing application will always receive the most recent data, regardless of how long the node may have been inactive.

We also assume that individual nodes sleep for a random time interval, the duration of which (i.e., the sleep time) is a geometrically distributed random variable regulated with probability P_{sleep}. When a node wakes up and finds an empty buffer, the MAC takes a vacation and the node goes to sleep.

When designing the packet scheduling during the active period of the node, several options are available. The simplest one, similar to the so-called 1-limited scheduling discipline used in traditional polling systems [26], requires the node to send one packet only and, then, immediately go back to sleep. The main shortcoming of this policy is its relative inefficiency, as the node may have to wait quite a long time before it gets a chance to send an uplink packet.

Another option is to keep the node active for as long as there are unsent packets in its uplink buffer; this option is similar to the exhaustive scheduling discipline in polling systems [26]. While this approach improves the apparent efficiency of the network (i.e., the node will still wait once, but then it may send several packets to the coordinator), it is not without problems of its own. First, assuming that the data from a single measurement of the sensed variable can fit in a single packet, sending several packets means that most of the data actually sent are, in fact, outdated. Second, the short time interval between successive packets increases the temporal correlation of sensed data; in other words, the expenditure in bandwidth and energy rises faster than the amount of information transferred to the sink, and the efficiency is reduced. This policy may be unacceptable in applications where controlled reliability means controlled inter-packet spacing; it may also compromise security because a malfunctioning node is allowed to inject large amounts of data into the network.

A promising compromise between the two policies described above, we have decided to use the so-called Bernoulli scheduling [26]. In this approach, at the end of each packet transmission the node checks its uplink buffer. If it is empty, the node immediately goes to sleep; if there are packets to send, the node transmits the next packet from the buffer with the probability P_{ber}, or goes to sleep with the probability $1 - P_{ber}$. (The limiting values of $P_{ber} = 0$ and 1 correspond to 1-limited and exhaustive scheduling policies, respectively.) Therefore, two regulating parameters are needed: one of them, P_{sleep}, determines the duration of the inactive period; the other, P_{ber}, regulates the duration of the active period.

It should come as no surprise that the regulation of the inactive period is the dominant mechanism in the situation when there are plenty of sensors (i.e., many more than the minimum number required to achieve the desired reliability). However, when individual nodes begin to cease functioning, because of battery exhaustion or for other reasons, the remaining nodes will have to extend their activity to achieve satisfactory reliability, and the importance of the Bernoulli mechanism will increase. We will use this tradeoff to advantage, as will be seen in the discussions that follow.

In order to investigate the feasibility and behavior of activity management policies with both centralized and distributed approach, we use a slightly modified version of the network model described in Section 4.1. The updated simulator considers only the uplink transmission where all data traffic is sent from the individual node to the coordinator.

6.2 Centralized Approach

With centralized approach, we assume that the network coordinator is aware of the number of nodes n and packet arrival rates λ_i: nodes that participate in the network have to be admitted first [17], and every packet received carries the source node address, which makes it simple to estimate the packet arrival rates. We can also assume that the coordinator knows the required reliability R as well as the packet size, both of which are set by the application.

Then the coordinator calculates the P_{sleep} and broadcasts in each beacon, so that all nodes can learn of it and adjust their sleep times accordingly. The analytical computation of P_{sleep} is derived in [22]. In our simulator, we use only the value of P_{sleep} obtained from the analytical result. The calculation has to be re-done when significant changes of reliability are sensed, most likely because of failure of some nodes.

To demonstrate the operation of this control mechanism, we use the calculated sleeping probabilities needed to maintain the reliability at $R = 5$ packets per second, under varying values of different network and traffic parameters. We have assumed $P_{ber} = 1$ i.e., the exhaustive, non-gated service discipline where the device will serve all the packets from its buffer before going to sleep, including those that arrive during the active period. The packet size has been fixed at 9 backoff periods, while the device buffer had a fixed size of $L = 3$ packets.

Fig. 6.1 shows event reliability, individual (per-node) reliability, and mean number of active nodes as functions of the number of nodes and packet arrival rate. Fig. 6.1(b) shows that, due to the control mechanism, the reliability per node is virtually independent of the packet arrival rate – it depends on the number of nodes only.

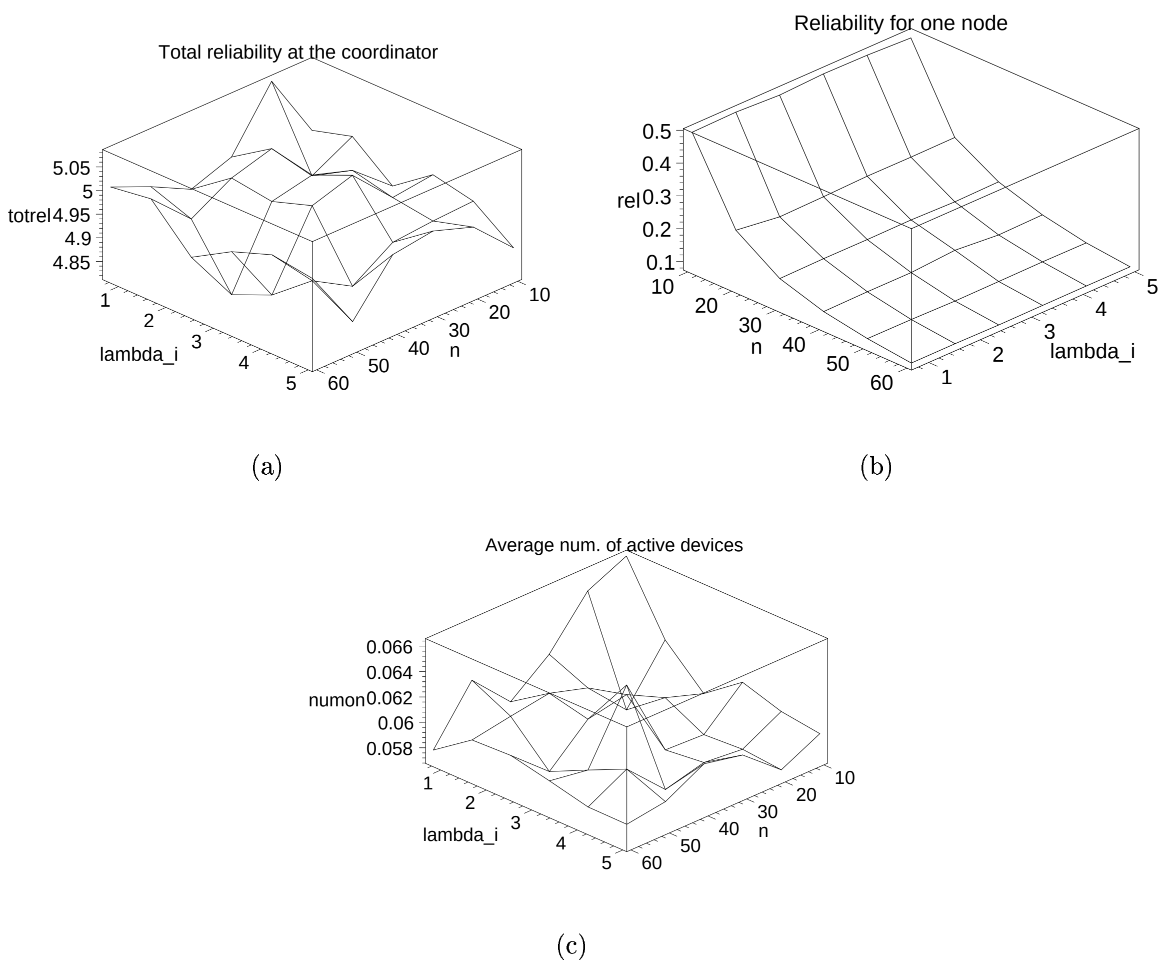
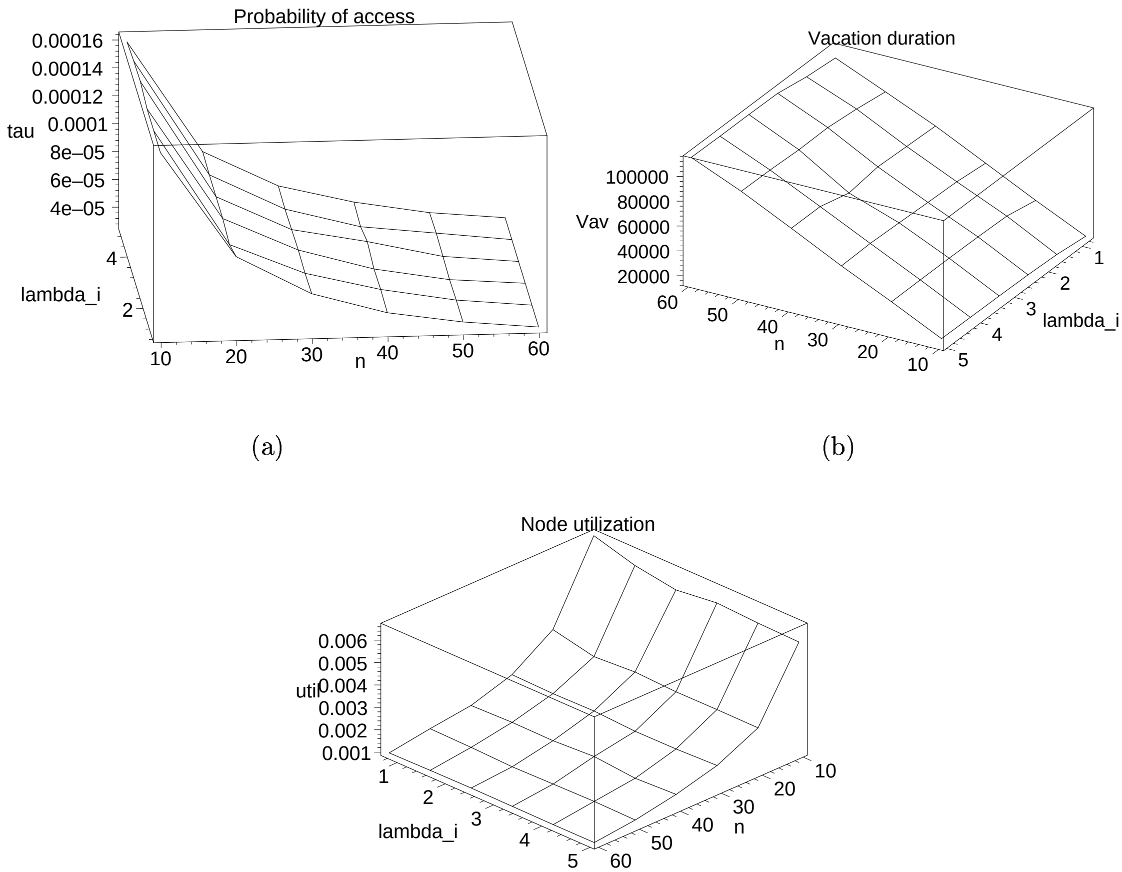


Figure 6.1: Network performance under controlled reliability and exhaustive service discipline ($P_{ber} = 1$).



(a) (b)

(c)

Figure 6.2: Node activity under controlled reliability and exhaustive service discipline ($P_{ber} = 1$).

The other point worth noting is that the mean number of active devices is very low, which indicates that the duty cycle (μ) of individual node is low. Diagrams shown in Fig. 6.2(c) confirms this observation, with node utilization well below 1% in the entire observed range of network size and traffic intensity. This property translates into long network lifetime: a single AAA battery can power an off-the-shelf radio transceiver drawing $10mA$ for two years, provided the duty cycle below 0.5% is maintained [11].

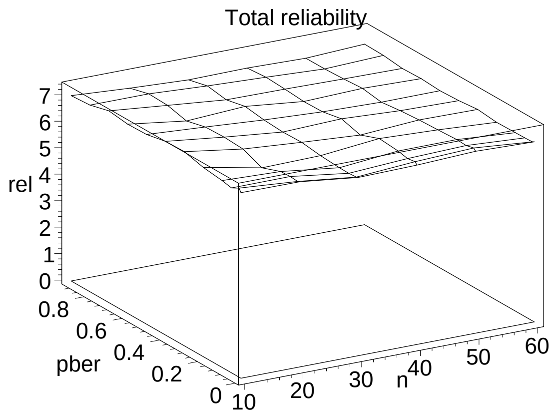
Furthermore, we note from Fig. 6.2(b) that the mean sleep duration is very long, of the order of tens of thousands of backoff periods. At the 250kbps data rate, one backoff period corresponds to 320 μs; the mean sleep duration is therefore of the order of seconds.

Fig. 6.2(c) could also be used to estimate the network size required to achieve the given node duty cycle and network reliability. For example, if the duty cycle below 0.1% and reliability at the coordinator of 5 packets per second are required, the minimum number of sensors in the network must be at least 60. Of course, the actual number would have to be higher because of node failures; appropriate allowances could be made on the basis of known hardware reliability for each sensor node.

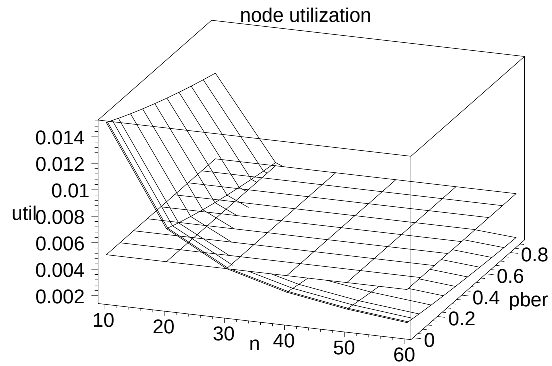
Because of the problem of exhaustive service discipline discussed in Section 6.1, we further try to implement another approach where the network size and probability P_{ber} are variable. The packet arrival rate was set to one packet per second for each node and the reliability R was set to 7 packets per second. As explained above, the sleep probabilities are obtained from the analytical result and used in simulator. Besides this, we also try to incorporate the impact of physical layer where the transmission is not only corrupted by collision but also by noise. Therefore, the network operates in the ISM band with raw data rate 250 kbps and $BER = 10^{-3}$ (details are described in Section 3.3).

As can be seen from Fig. 6.3(a), this simple control mechanism manages to maintain the reliability at the desired value, or close to it, in a wide range of values for the number of nodes and probability P_{ber}. From Fig. 6.3(c), mean duration of sleep periods is quite long, of the order of tens of thousands of backoff periods.

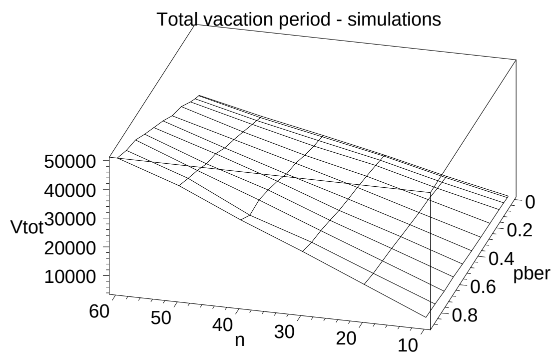
When the reliability R is fixed, individual node utilization depends on the number of nodes as well as on P_{ber}, as can be seen in Fig. 6.3(b). To better illustrate the relationship between P_{ber} and utilization, the plane that corresponds to the utilization of 0.5% is plotted on the same diagram. Note that at least 20 nodes are needed in order to keep the utilization below this value. When the number of nodes begins to drop toward 20, the value of P_{ber} has to increase and quickly approaches 1 – which corresponds to the exhaustive scheduling within one node.



(a) Collective reliability.



(b) Node utilization.



(c) Mean sleep duration.

Figure 6.3: Network performance as a function of number of devices and Bernoulli service discipline (P_{ber}) under centralized control with reliability fixed at $R = 7$.

A better solution would be to operate the network in dual mode, i.e., use 1-limited scheduling with $P_{ber} = 0$ while the number of sensors is sufficiently high (at least 30, using the data from Fig. 6.3) but switch to true Bernoulli scheduling with $P_{ber} > 0$ when the number of sensors drops below 30. (A higher switchover point of around 40 to 50 nodes would provide some safety margin.) In this manner, the network operates with maximum efficiency when possible, and achieves graceful degradation if efficiency cannot be maintained.

Note that the diagrams from Fig. 6.3(b) allow us to estimate the minimum network size needed to achieve the given duty cycle μ – it suffices to set $P_{ber} = 0$. Of course, new calculations and new measurements would be required for different values of reliability and packet arrival rate.

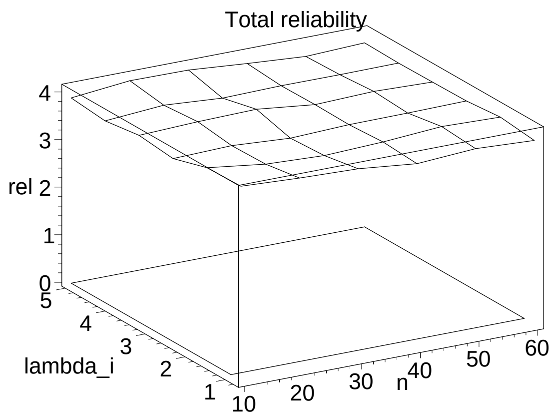
While the centralized approach is able to control the network reliability within a few percent, it requires excessive computational resources. Namely, given the number of live nodes, as well as the required reliability and node utilization, the coordinator has to use the complete analytical model [22] to minimize P_{ber} and calculate the corresponding P_{sleep}. As nodes die over time, the optimum values change, and the calculations have to be repeated after each change of network size. These may well be beyond the capabilities of coordinator nodes, in particular when they run on battery power.

6.3 Distributed Approach

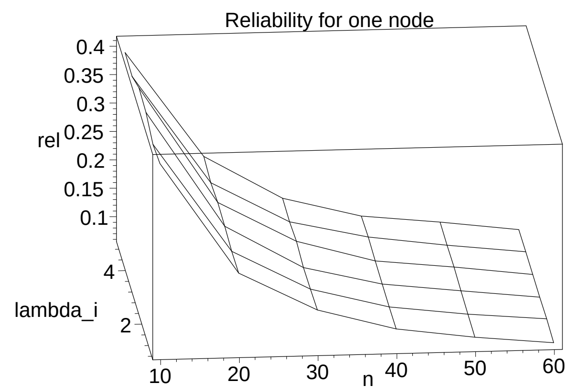
Fortunately, the algorithm can be implemented in a distributed fashion. The coordinator has to calculate individual reliability based on its knowledge of the number of live nodes n and required collective reliability R. Then the coordinator sends the information to the devices through beacon frame. Each node should keep track on the transmission success probability γ it has experienced. Based on it's transmission success probability γ, a device recalculates its individual reliability $r = R/(n\gamma)$ to obtain the average period between transmissions. This period corresponds to the average sleep time. Note that the sleep time is geometrically distributed and the mean sleep time is $t_{boff}/(1 - P_{sleep}) = 1/r$. Therefore, each sensor node starts with $P_{sleep} = 1 - rt_{boff}$ where $t_{boff} = 0.00032s$. This approach is computationally much less demanding, for both the coordinator and individual node.

To demonstrate the operation of this control mechanism, the simulator has to calculate the sleep probabilities and network QoS needed to maintain the reliability at R=4 packets per second, under varying network size and packet arrival rate. We have assumed 1-

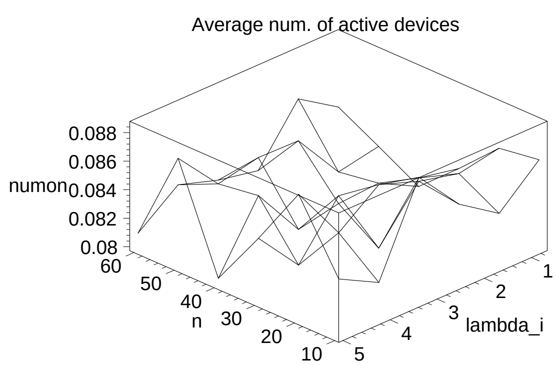
limited service discipline i.e., the device will serve only one packet from its buffer during each active period. We also assume that the nodes obtain only the required value of individual reliability, and calculate P_{sleep} individually. When sleep probability is known, random sleep periods will be generated according to geometric distribution. In this way, simultaneous attempts to access the medium and, consequently, the likelihood of collisions, is minimized.



(a)



(b)



(c)

Figure 6.4: Network performance under controlled reliability and 1-limited service discipline ($P_{ber} = 0$).

Fig. 6.4 shows event reliability, individual (per-node) reliability, and mean number of

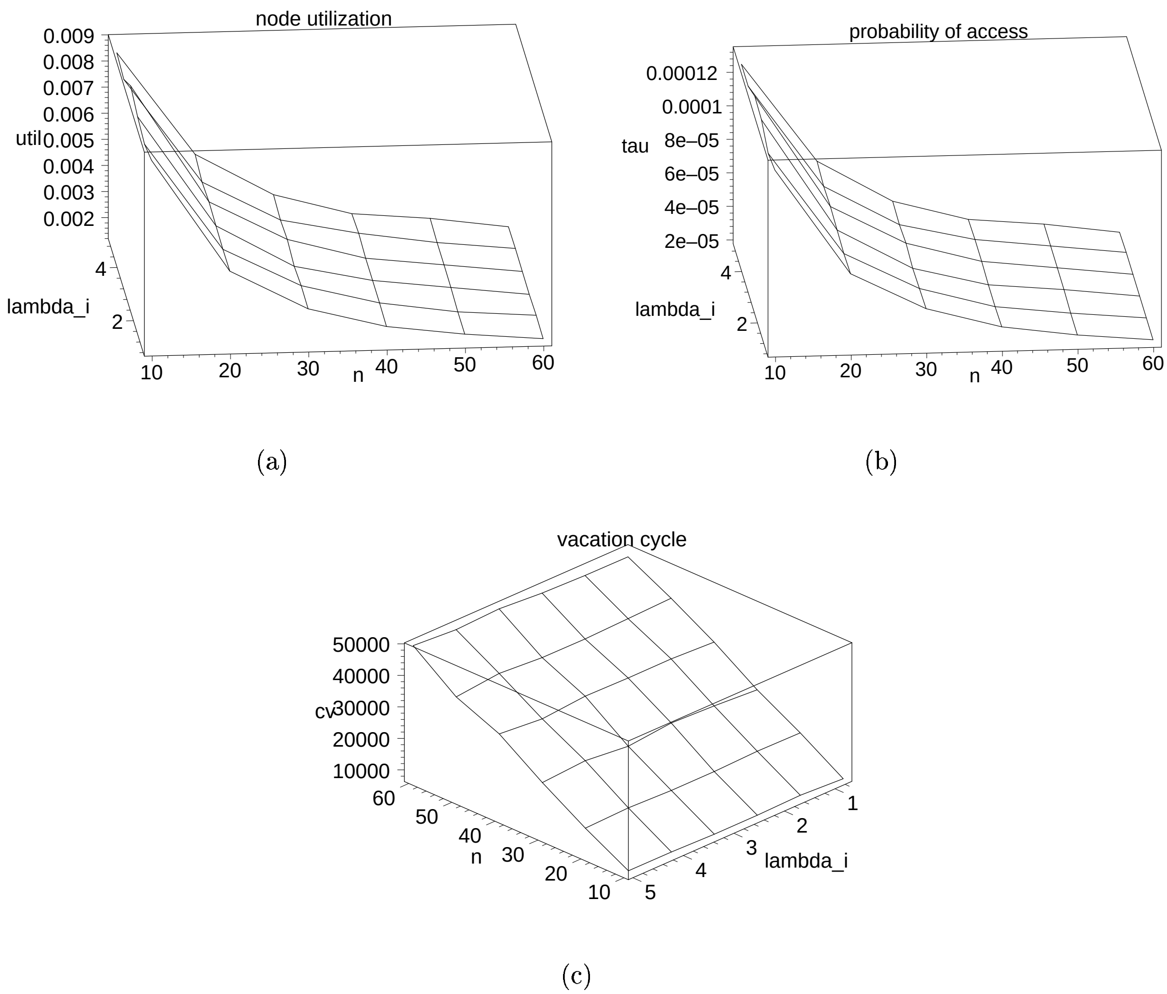


Figure 6.5: Node activity under controlled reliability and 1-limited service discipline (P_{ber} = 0).

active nodes as functions of the number of nodes and packet arrival rate. The Fig. 6.4(a) shows that the event reliability is controlled within a few percent. Fig. 6.4(b) shows that, due to the control mechanism, the reliability per node is virtually independent of the packet arrival rate – it depends on the number of nodes only.

The mean number of active devices shown in Fig. 6.4(c) is very low, which indicates that the duty cycle of individual nodes is low. Diagrams shown in Figs. 6.5(a) and 6.5(c) confirm this observation, with node utilization well below 1% in the entire observed range of network size and traffic intensity and with very long mean sleep duration, of the order of tens of thousands of backoff periods.

From the above discussion, it is clear that the distributed lightweight implementation of the control policy offers superior performance over other techniques where for fixed reliability, there is no freedom in choosing duty cycle of the device.

Although the above approach gives us feasible performance with low computational load of the devices, we again try to change the algorithm to adapt with Bernoulli packet scheduling during the active period. The updated algorithm can be implemented in the following way. The coordinator has to determine utilization and calculate only individual reliability r based on its knowledge of the number of live nodes and send it in the beacon frame. Each node should start by deducing the average sleep time under $P_{ber} = 0$. This is possible since under geometric probability distribution of sleep time, the average value of sleep time in seconds is $t_{boff}/(1 - P_{sleep}) = 1/r$. The node should start with $P_{sleep} = 1 - rt_{boff}$ and $P_{ber} = 0$ and monitor utilization of its transmitter/receiver. Such monitoring is possible by finding the ratio of the number of backoff periods while the node was active during the specified recent monitoring window of backoff periods and the total size of the monitoring window. If utilization is below the required limit (e.g. $\mathcal{U} = 0.005$), P_{ber} can be left at zero value. We assume that initially the number of sensors is sufficient to maintain required utilization with $P_{ber} = 0$. However, as the time passes, some sensors will die, and coordinator has to broadcast updates on individual reliability which grows with each sensor's death.

If utilization, at some point of time increases above required limit $\mathcal{U}$ due to increased requested individual reliability, node should start to gradually increase P_{ber} and increase sleeping probability until its utilization drops below the limit. A simple way to calculate P_{ber} and P_{sleep} is:

$$P_{ber} = (\rho' - \mathcal{U})/\rho', \qquad \rho' > \mathcal{U} \tag{6.1}$$

$$P_{sleep} = 1 - rt_{boff}(1 + P_{ber}) \tag{6.2}$$

If the required individual reliability is too large to be maintained with requested utilization even with $P_{ber} = 1$ the node continues working with $P_{ber} \approx 1$ until it dies. This approach is computationally very lightweight for both the coordinator and the nodes.

We have implemented the distributed activity management mechanism under the assumption that each battery has a fixed lifetime, expressed in backoff periods. The lifetime is measured through a dedicated counter which is decreased by one for each backoff period in which the node has been active. The power consumption during transmission generally differs from that during reception, depending on the construction of the radio (as has been assumed in, for example, [3]). For simplicity, however, we have assumed that the power consumption is the same whenever the radio is turned on; this assumption is additionally justified by the fact that the transmission is a dominant mode in 802.15.4 networks with uplink traffic – nodes only have to receive the beacon and the acknowledgment frames. We have also considered the impact of physical layer with $BER = 10^{-3}$.

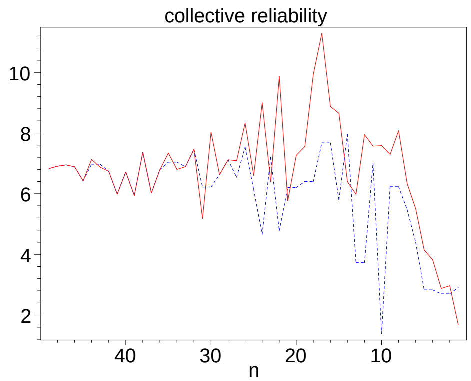
For comparison, measurements were taken using both pure 1-limited scheduling (i.e., P_{ber} is always zero) and adaptive Bernoulli scheduling. Network reliability was set to $R = 7$ packets per second, the same value as before. The network starts with 50 nodes with equally charged batteries; we used the value of 10^5 backoff periods for each one. Measurements of network parameters were taken for the periods between two successive node deaths (all times are expressed in backoff periods). Fig. 6.6 shows a number of performance measures, with the number of live sensors in the network shown on the horizontal axis.

The diagram of event reliability, Fig. 6.6(b), shows that distributed activity management with adaptive Bernoulli scheduling is every bit as capable of maintaining the network reliability at the specified level, as its centralized counterpart. However, the individual node utilization and remaining energy level, shown in Figs. 6.6(b) and 6.6(d), are much improved and clearly demonstrate graceful degradation of network performance as individual sensors die.

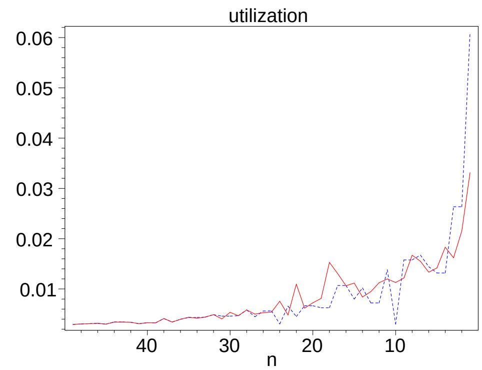
The diagram of the value of P_{ber}, Fig. 6.6(c), shows the adaptive scheduling in action;

as long as the number of live sensors is above 30, P_{ber} is kept at zero and there is no significant difference between adaptive and pure 1-limited scheduling. When the number of live sensors drops below the threshold, 1-limited scheduling becomes more efficient – at the expense of being unable to maintain the reliability at the desired level. However, adaptive scheduling maintains the desired reliability of $R = 7$ even when the number of sensors drops below 10! Overall, adaptive Bernoulli scheduling extends the useful lifetime of the network for about seven node lifetimes, or around 6% in terms of absolute time.

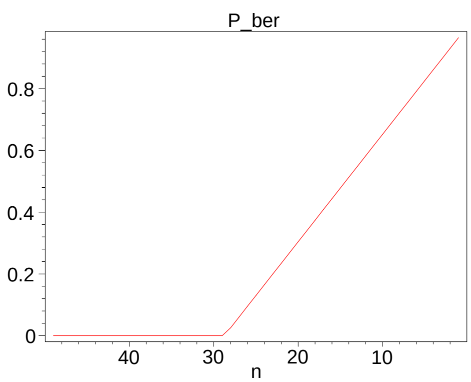
This difference notwithstanding, both scheduling techniques increase the total network lifetime by more than two orders of magnitude compared to the case without duty cycle management, whilst keeping the reliability at the desired level for the major portion of the network lifetime. This is confirmed by the diagrams in Figs. 6.6(e) and 6.6(f); for clarity, vertical axes use the logarithmic scale instead of a linear one.



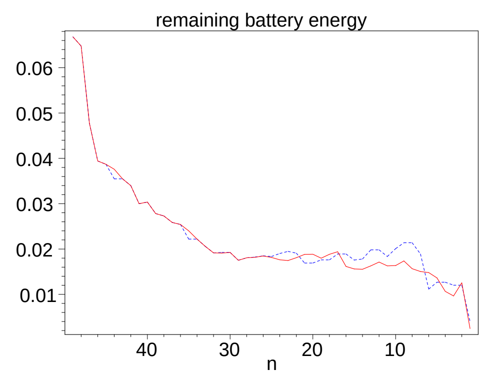
(a) Network reliability.



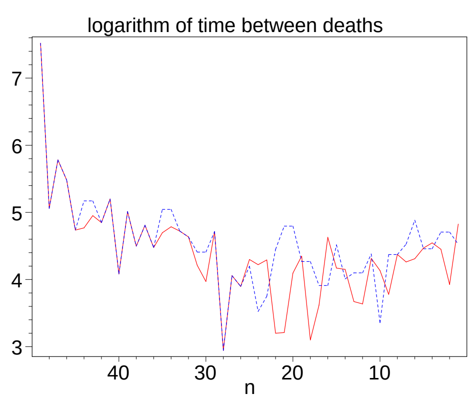
(b) Individual node utilization.



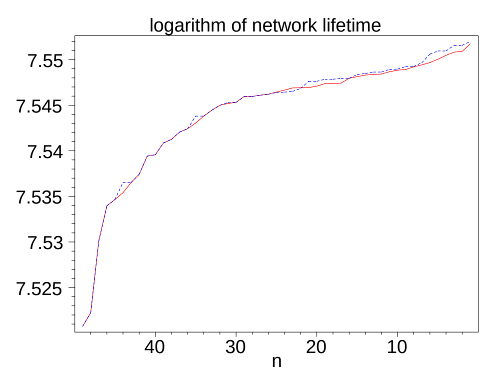
(c) Bernoulli scheduling probability P_{ber}.



(d) Percentage of remaining energy.



(e) Logarithm of time between successive node deaths.



(f) Logarithm of network lifetime.

Figure 6.6: Pertaining to the performance of 802.15.4 sensor network with distributed activity management. Dotted line denotes results obtained under pure 1-limited scheduling, solid line shows results obtained under adaptive Bernoulli scheduling.

Chapter 7

Simulator Design and Analysis

7.1 Simulator Design

Artifex Petri nets [1] provide graphical user interface to build simulation model with a combination of different kind of objects. A object, instance of a class, with its behavior and interfaces can be defined by using some graphical elements. The most common elements are the *transition*, the *place*, and the *link*. The rectangles, the circles and the arrows shown in Fig. 7.2 represent the transitions, the places, and the links, respectively. Transitions are actually the processing unit of the model where users write their codes. Places are data stores containing units of information called tokens. Each transition is connected to one or more places and is triggered when it fetch tokens from the connecting places. Tokens are structured data. Links are connections between places and transitions.

Objects interact with each other through interfaces. Interface is a set of input and output places. The input place of one object is linked with the output place of another object. For exchanging information, one object sends token to its output place and then token is immediately passed to connecting input place of another object. Through Artifex, it is also possible to split a complex design into multiple pages based on the functionalities.

The Fig. 7.1 represents the model of our IEEE 802.15.4-compliant network in Artifex.

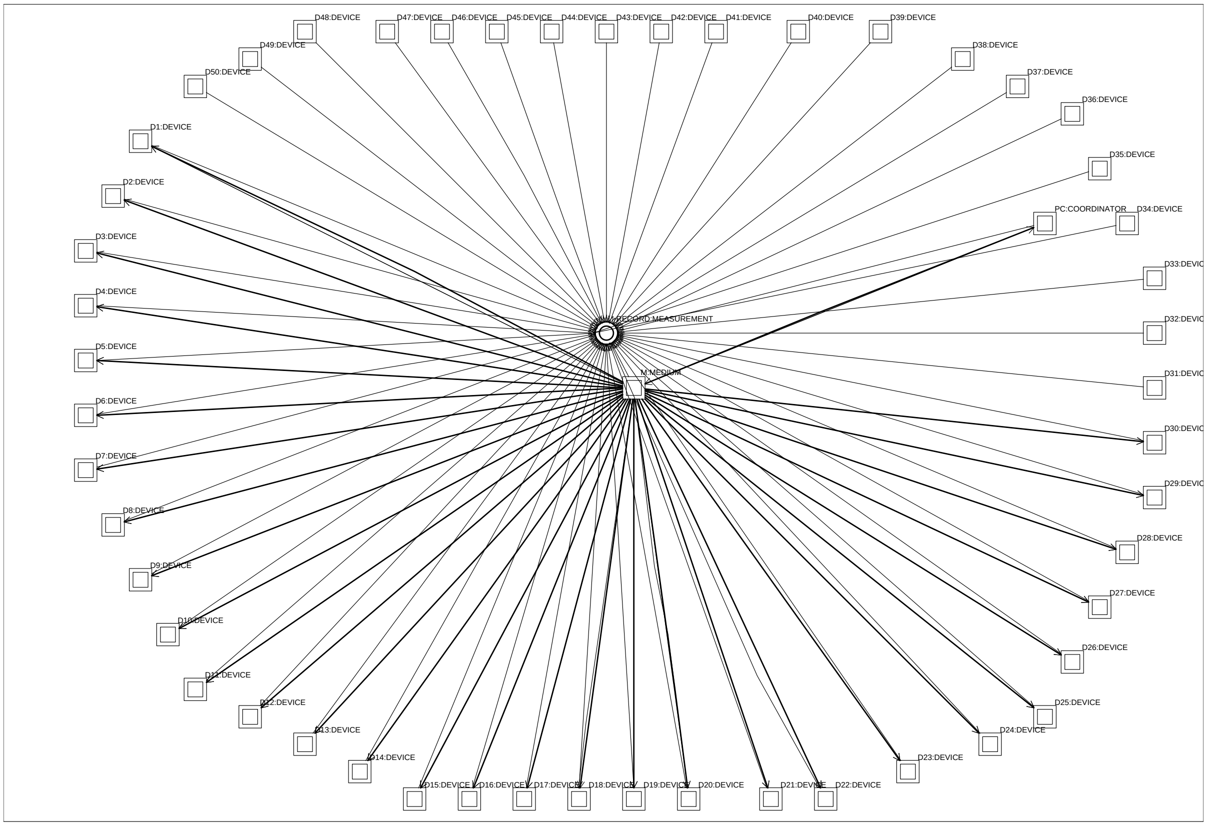


Figure 7.1: IEEE 802.15.4-compliant star-shaped network in simulator.

In this model, devices (represented as D1, D2, and so on) are connected to the medium (M) and medium is connected to the PAN coordinator (PC). Then the three entities (device, medium, and coordinator) are connected to another page through *RECORD* : *MEASUREMENT* output place. The page contains all statistical counters which are updated by the three entities when an event occurs. Each device is identified with a unique identifier. The PAN coordinator also has an identifier. Since we have different versions of our simulator such as for uplink transmissions only, for both uplink and downlink transmissions, and also for managing duty cycling, we will not discuss all of them. We will discuss only one version of the simulator which includes the basic functions of the IEEE 802.15.4 MAC.

7.1.1 PAN coordinator

The coordinator contains two input places and three output places. The input places are *RECEIVE_DATA* : *PACKET*, and *GET* : *BACKOFF* where the output places are *SEND* : *BEACON*, *SEND_OUT* : *PACKET*, and *RECORD* : *MEASUREMENT* shown in Fig. 7.2. The "PACKET", "BACKOFF", and "MEASUREMENT" are three data (token) types. The using of data type with the name of the input and output places indicate the type of token that they can receive from the medium, and send to the medium.

The coordinator generates beacon after every 48 backoff periods since we assume SO = 0, and BO = 0. Each backoff period is a unit of time. To implement unit of time, Artifex offers real time, and virtual time. We model time in our simulator using the concept of virtual time. The coordinator sends beacon to the medium through *SEND* : *BEACON* output place and receives packet from the medium through *RECEIVE_DATA* : *PACKET* input place.

The coordinator consists of three pages: *MAIN_PAGE*, *DOWNLINK_QUEUES*, and the *CSMA_CA_ALGORITHM* page. In page *DOWNLINK_QUEUES*, the coordinator maintains finite size buffers for each device to queue the downlink packets. The page *CSMA_CA_ALGORITHM* mainly executes the CSMA-CA algorithm to access the channel for downlink packets. We will describe it briefly in Section 7.1.3.

The coordinator deals with three kinds of packets: data packets, MAC command request packets (MCRF), and acknowledgement (ACK) packets. After receiving data packet, the coordinator checks the destination identifier (id) field of the packet. If the buffer of the destination device is already full, it discards the packet through the place *FILTER* : *PACKET*. Otherwise it accepts the packet and passes it to the *DOWNLINK_QUEUES* page for storing in the destination device buffer for downlink transmission. At the same time, it also initiates acknowledgement by triggering *ACK_GEN* transition, and sends the ACK to the source device through medium.

When a MCRF arrives at the input place *RECEIVE_DATA* : *PACKET* of the coordi-

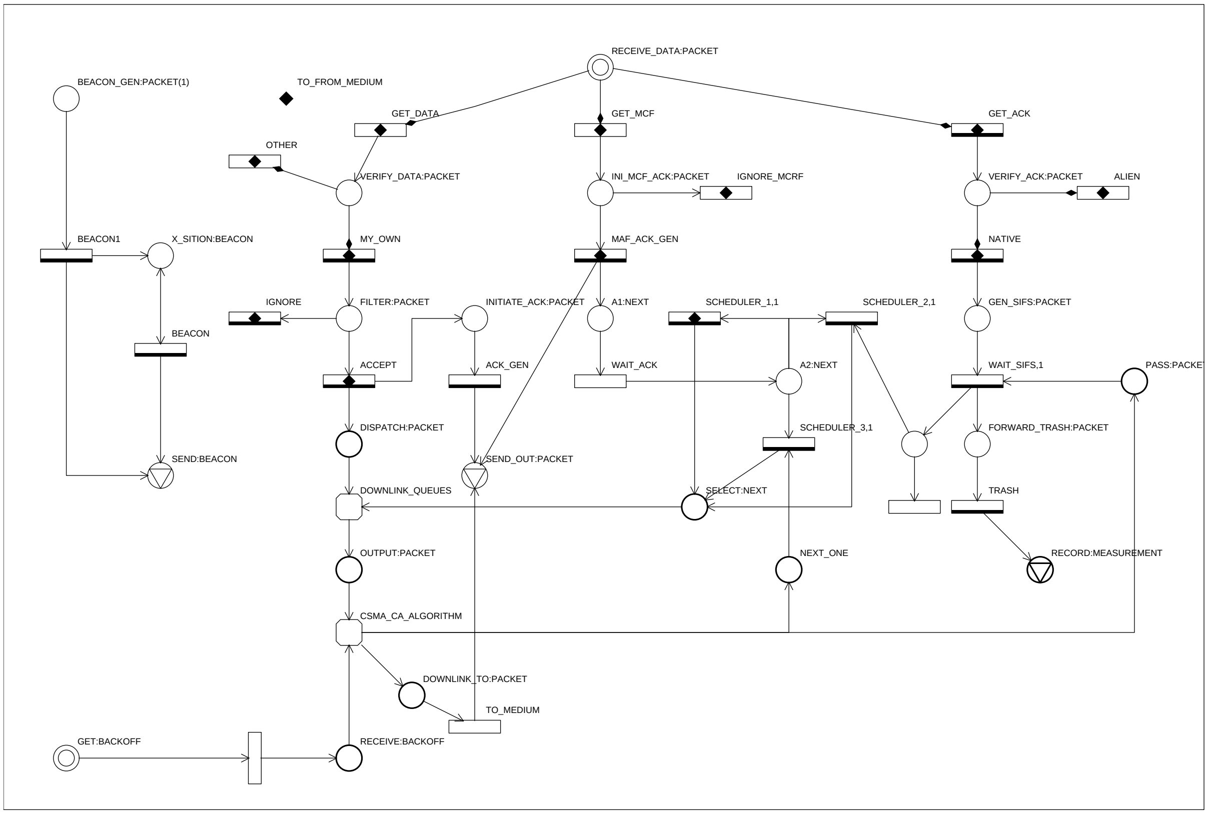


Figure 7.2: Functionalities of the PAN coordinator.

nator, it can be received (*MCF_ACK_GEN*) or rejected (*IGNORE_MCRF*) depending on the current status of the coordinator. If the coordinator is busy with the countdown procedure of the CSMA-CA algorithm, the packet will be rejected. Otherwise, it will be accepted. By receiving the MCRF, the coordinator acknowledges the source device and starts CSMA-CA algorithm to initiate downlink transmission. For a successful transmission, the coordinator receives an ACK.

7.1.2 Medium

Medium shown in Fig. 7.3 links the devices to the coordinator and vice versa. When it receives a beacon from the coordinator, it will start a clock. The clock is implemented by using the transition *DELAY_BACKOFF*. The clock releases a token of data type

BACKOFF to the devices and the coordinator (SEND: BACKOFF) after every one time unit (backoff period).

If the medium receives two or more data packets at a time, the transition *COLLISION* is triggered. After a collision, the medium declares that it is busy by sending a token through the *SEND* : *BACKOFF* output place to the devices and the coordinator. The transition *KEEP_BUSY_MEDIUM* waits a time period (packet size + t_{ack} + ACK size) before delivering the packets to the place *DESTROY* : *PACKET*. After destroying the packets, the medium declares that it is idle. On the other hand, if it receives only one packet at a time, it forwards the packet to the destination (may be the coordinator or a device). Before forwarding the packet, the medium holds the packet (DELAY), and declares that it is busy for a time period equal to the packet size. Then it sends the packet to the coordinator (*SEND_TO_PAN* : *PACKET*) or the devices (*SEND_TO_DEVICES* : *PACKET*) based on the destination address and declares that the medium is idle. In case of receiving an ACK packet, the medium also remains busy for a time period equal to the ACK size.

7.1.3 Device

The device consists of three pages, namely, the *MAIN_PAGE*, the *CSMA_CA_ALGORITHM*, and the *SOURCE_POISSON*. The *SOURCE_POISSON* page is actually responsible to generate packets following Poisson distribution and also to generate MAC command request packet if a device finds its address to the pending address list of the beacon frame. This page also maintains two separate finite buffers; one buffer is for data packet while the other buffer is for request packets.

The CSMA-CA algorithm is executed in the *CSMA_CA_ALGORITHM* page shown in Fig. 7.4. After receiving a packet from the *SOURCE_POISSON* page, the device initializes (INITIALIZE) three variables NB, BE, and CW; and generates a random number. Then after receiving each backoff period (*FORWARD* : *BACKOFF*), the device decrements the random number by one. When the random number goes down to 0, the

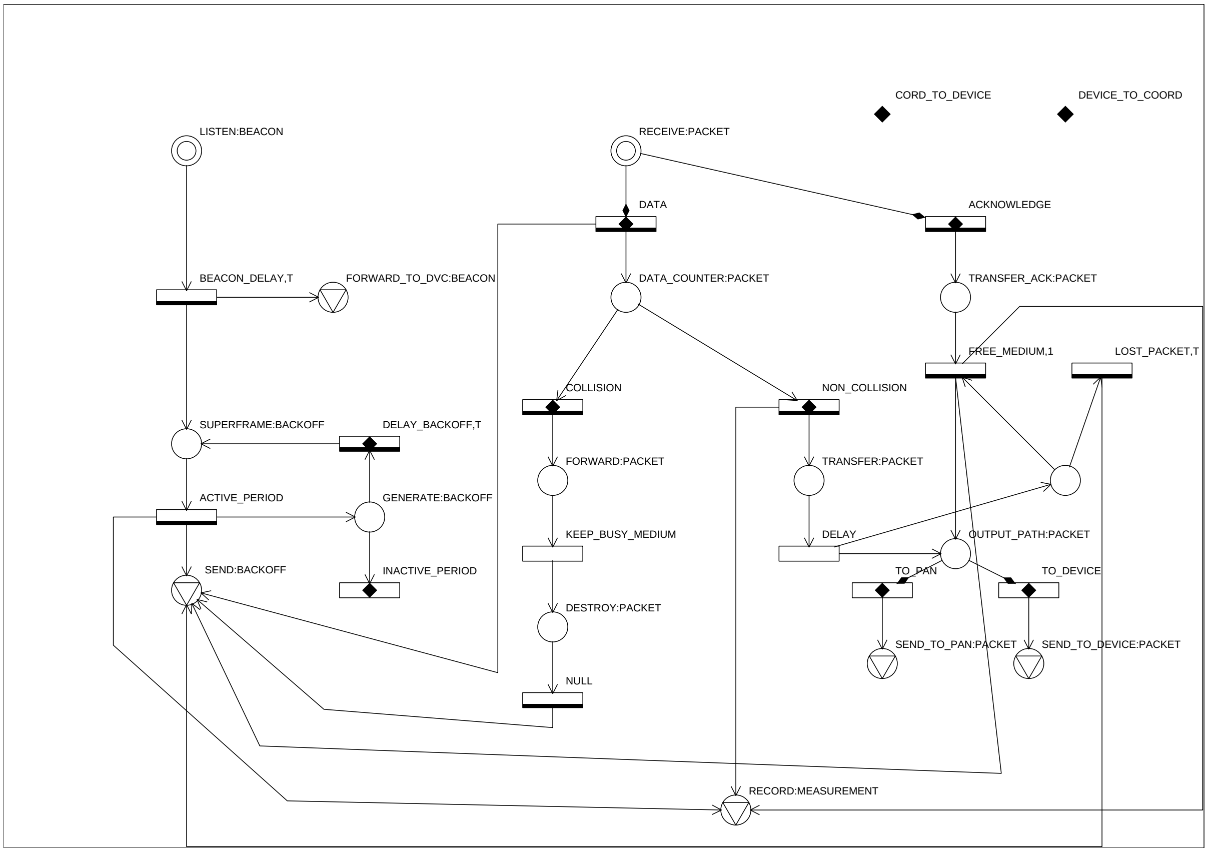


Figure 7.3: Functionalities of the medium.

transition *WAIT_ACTIVE_PERIOD* determines whether the current superframe has sufficient time for further processing. The token (data packet) is then forwarded to the *WAIT_INACTIVE_PERIOD* transition to defer transmission; otherwise it is forwarded to the *NOT_WAIT* transition to assess the channel. After two successful CCAs, it passes the packet to the *MAIN_PAGE* to send the packet to the medium. Before transmitting the packet, the device makes a copy of the packet for retransmission in case of collision.

There are two input places (*WELCOME_PACKET*) and two output places (*SEND_OUT* : *PACKET*, *RECORD* : *MEASUREMENT*) in the page denoted by *MAIN_PAGE* (Fig. 7.5). A device receives a token and knows about a backoff boundary from the input place *RECEIVE* : *BACKOFF*. Then it passes the token to the *CSMA_CA_ALGORITHM* page. Data and ACK packets are received through the input place *WELCOME_PACKET*.

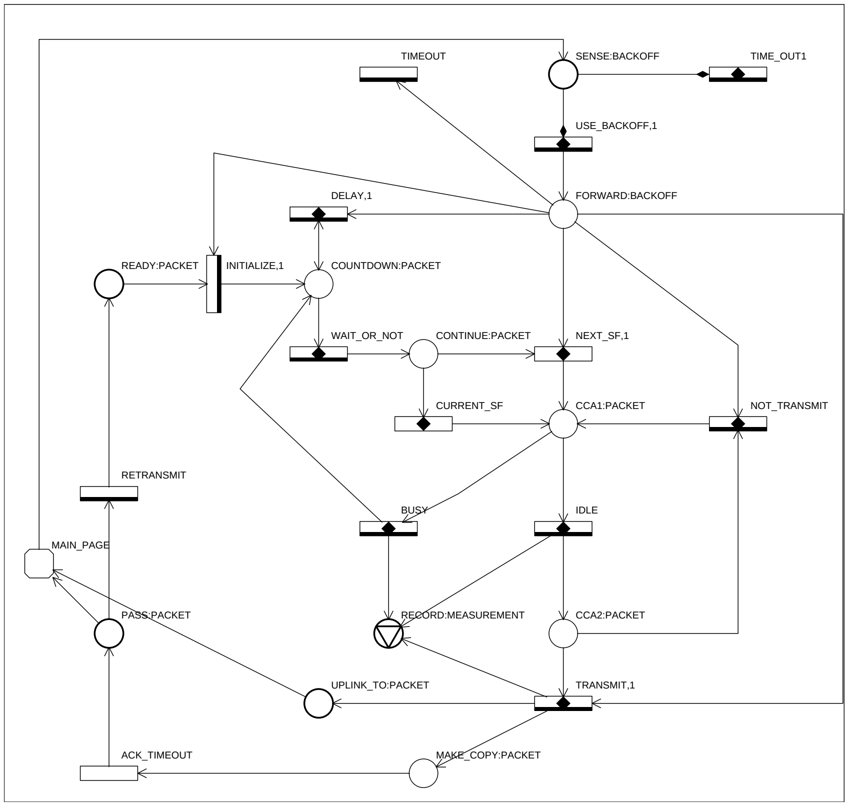


Figure 7.4: Functionalities of the CSMA-CA algorithm.

The transition *WAIT_SIFS* is triggered after receiving an acknowledgement of a data packet and starts CSMA-CA algorithm (*PROCESS_NEXT*) if there is any packet in any of the two buffers. On the other hand, if the device receives an acknowledgement of a MAC request packet, it holds the token (ACK packet) on the place $SUCCESS : PACKET$ and turns on its receiver for a period of 61 backoff periods. Within this time period, if a device receives a data packet, it generates an acknowledgement and sends it to the medium using the output place $SEND\_OUT : PACKET$. At the same time, it will trigger the transition *TIME_OUT* to release the token stored in the place $SUCCESS : PACKET$, and destroy it. Otherwise, after 61 backoff units, the transition *IGNORED* is triggered which turns off the receiver and turns on the transmitter of the device.

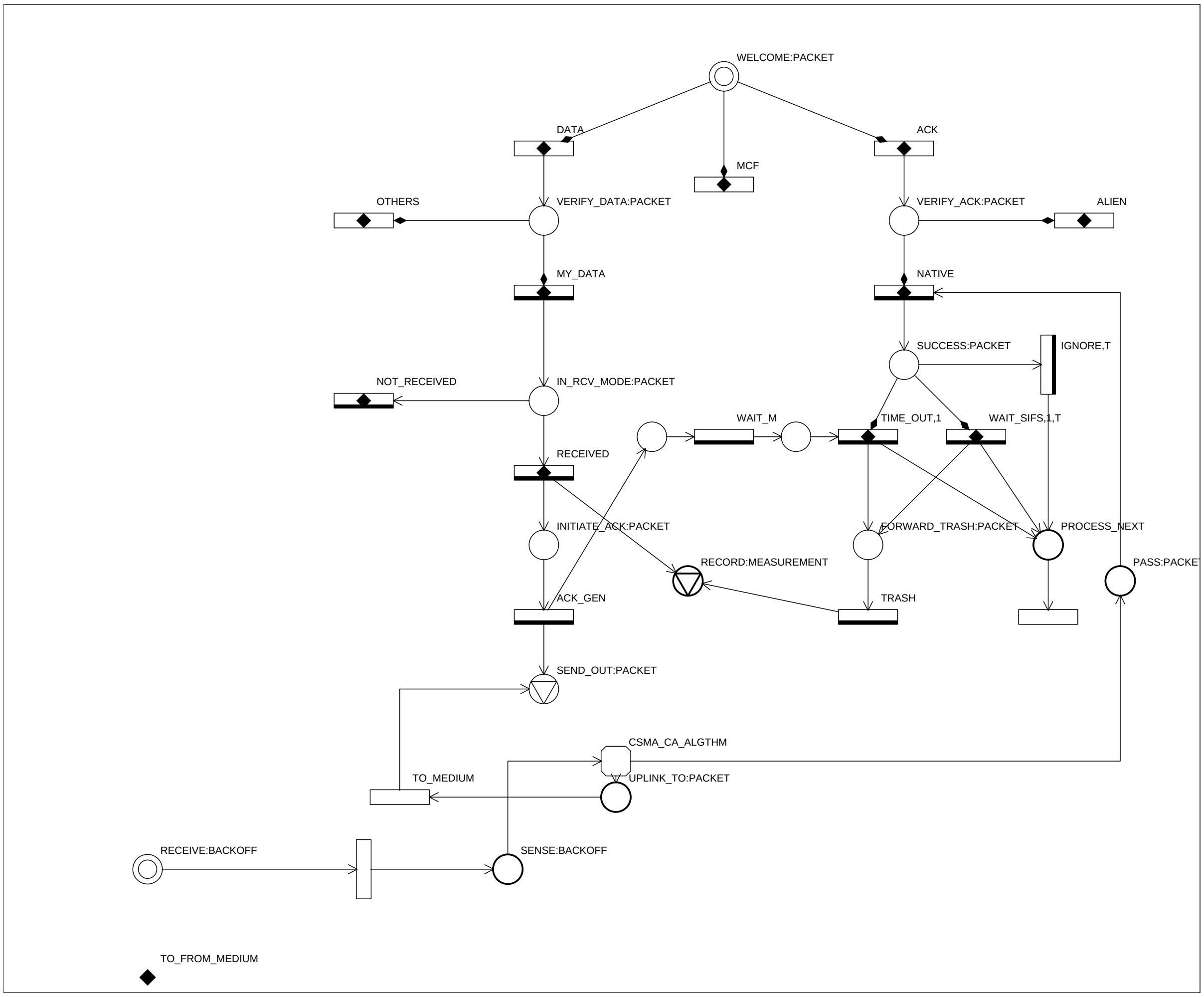


Figure 7.5: Functionalities of the device.

7.2 Analysis of simulation output

Since we are interested in steady-state performance, it is needed to estimate the transient period. In such case, results of the initial part of the simulation should not be included in the final computations. This initial part is called the transient state [19]. Since we have to take so many measurements, we will not discuss all of them. Here we will only discuss how we obtained transient interval and the stopping criterion (length of run) to take the measurements in case of uplink transmissions in non-saturation condition (Section 4.3).

From Fig. 4.4, it is clear that we use two factors: one is number of devices, and the other one is arrival rate. The set of the number of devices contain 10 elements, where the set of arrival rates contains 13 elements. As a result, for each performance parameter, we have to take 130 measurements (points). Since it is difficult to calculate the transient interval and the stopping criterion for each measurements, we measure only for 5 devices with a arrival rate 10 packets per minute. Then, we apply the same transient interval and the length of run to take the remaining 129 measurements. The reason behind to choose this combination is that with high arrival rate, and with large number of devices, a steady state result would be obtained earlier than with low arrival rate, and with small number of devices.

For getting the transient interval, 6 replications are considered where each replication consists of 60 observations. We run each replication 60,000 backoff periods i.e., the time interval between two consecutive observations is 1000 backoff periods. We measure the probability of access (τ) in each observation to obtain the transient interval. In each replication, the seeds of random number generator of the slotted CSMA-CA algorithm is changed. Then we use the moving average method for different moving window parameter (W) and observe that the steady state performance starts around 29th observation. This observation implies that the transient interval is 29,000 backoff periods which is shown in Fig. 7.6.

After getting the transient interval, the next step is to determine the length of run. The proper length of run could be obtained by running the simulation in various time length. In each simulation run, the statistical counter should be reset after the transient interval. We run the simulation until the mean response narrows to a desired width. If the sample mean is $\overline{\tau}$ for m observations and its variance is s=Var($\overline{\tau}$), then the width of 90% confidence interval is $2t_{m-1,0.95}\frac{s}{\sqrt{m}}$ [19]; where $t_{m-1,0.95}$ is the 0.95-quantile of a t-variate with m-1 degrees of freedom. By using the above technique, the length of run that we obtain in case of uplink transmissions is equal to 150,000 backoff periods where the width of the 90% confidence interval for τ is within 5% of the mean.

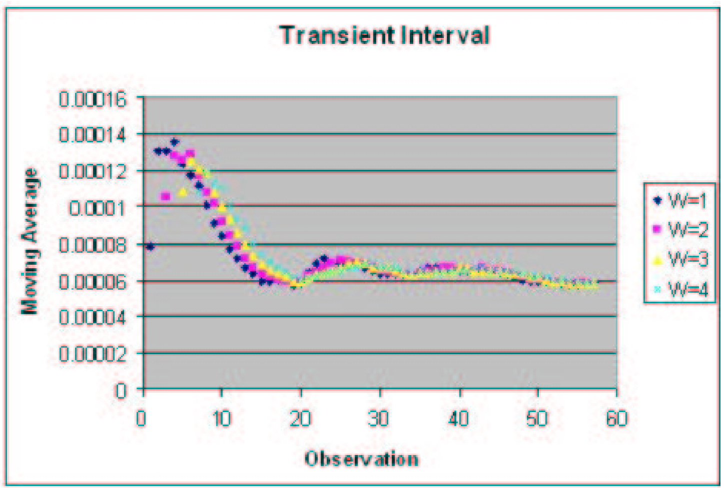


Figure 7.6: Transient Interval of the simulation in case of uplink transmissions.

Chapter 8

Conclusion

The IEEE 802.15.4 standard was developed to provide a low cost wireless solution to a wide variety of applications. To deploy this standard on the targeted applications, it is important to evaluate the performance of this standard. We have modeled the network behavior with uplink traffic and also both uplink and downlink traffic. We have shown that the default parameter values set up in the standard lead to low performance and abrupt change from non-saturation to saturation regime. We have also identified three problems in the current MAC definition that contribute to this performance problem.

We have also shown that the network coordinator can handle only a comparatively small amount of downlink traffic and that the number of nodes and their traffic load should be chosen with the goal of keeping the operating point of the network well away from the saturation point. Fortunately, in most sensor network applications the majority of the traffic occurs in the uplink direction but nonetheless, the stability limit for cluster coordinator should be closely monitored. Finally, for applications which have moderate amount of downlink traffic we have proposed a modification of the standard that leads to improved performance.

To manage the activity of sensor nodes, we have described centralized and distributed algorithms for a sensor network with star topology operating under 802.15.4 standard in beacon enabled slotted CSMA-CA mode. While the centralized and distributed mechanisms offer comparable accuracy with respect to event reliability, the later offers significant advantages in terms of computational complexity. We have introduced the Bernoulli scheduling of inactive and active periods of sensor nodes. The effect of noise errors from physical layer is included in the model. The results show that Bernoulli scheduling offers graceful degradation of network performance in the presence of node failures. It is superior over 1-limited techniques since it can extend the network lifetime. It is also superior to exhaustive scheduling since it limits the amount of temporal and spatial correlation in sensed data.

Bibliography

[1] *Rsoft Design Group*. San Jose, CA, 2003. Artifex v.4.4.2.

[2] I. F. Akyildiz, Y. Sankarasubramaniam W. Su, and E. Cayirci. Wireless sensor networks: a survey. *Communications Magazine, IEEE.*, 40(8):102–114, August 2002.

[3] M. Bhardwaj and A. P. Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. *Infocom02*, 3:1587–1596, June 2002.

[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Special Areas in Communications - Wireless Series*, 18:535–547, March 2000.

[5] F. Cal, M. Conti, and E. Gregori. IEEE 802.11 Wireless LAN: Capacity analysis and protocol enhancement. *Proceedings of INFOCOM*, 1:142–149, March/April 1998.

[6] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, M. Naeve, B. Heile, and V. Bahl. Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine*, 40(28):70–77, August 2002.

[7] H. S. Chhaya and S. Gupta. Performance of asynchronous data transfer methods of IEEE 802.11 MAC protocol. *IEEE Personal Communications*, 3:8–15, October 1996.

[8] B. P. Crow, I. Widjaja, J. G. Kim, and P. Sakai. Investigation of the IEEE 802.11 Medium Access Control (mac) sublayer functions. *Proceedings of INFOCOM*, 1:126–133, April 1997.

[9] J. Frolik. QoS control for random access wireless sensor networks. *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, 3:1522 – 1527, March 2004.

[10] V. K. Garg, K. Smolik, and J.E. Wilkes. In *Applications of CDMA in Wireless/Personl Communications*, Prentice Hall, Upper Saddle River, NJ, 1998.

[11] J. A. Gutierrez, E. H. Callaway, Jr., and R.L. Barrett. *Low-Rate Wireless Personal Area Networks.*, 2004. IEEE Press.

[12] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile. IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks. *IEEE Network*, 15(5):78–85, Sept/Oct. 2001.

[13] C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random and Coordinated sleep algorithms. *In 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.

[14] Standard for part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPAN). IEEE standard 802.15.1, IEEE, New York, NY, 2002.

[15] Standard for part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPAN). IEEE standard 802.15.3, IEEE, New York, NY, 2003.

[16] Data Sheet for CC2420 2.4 GHz IEEE 802.15.4/Zigbee RF Transceiver. Technical report, available online at http://www.chipcon.com/files/CC2420_Data_Sheet_1_2.pdf.

[17] Standard for part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN). IEEE Std 802.15.4, IEEE, New York, NY, 2003.

[18] R. Iyer and L. Kleinrock. QoS control for sensor networks. *Communications, 2003. ICC '03. IEEE International Conference on*, 1:517–521, May 2003.

[19] Raj Jain. *The art of computer systems performance analysis.* Newyork/ Chichester/ Brisben/ Toronto/ Singapore, 1991. John Wiley & Sons, Inc.

[20] G. Lu, B. Krishnamachari, and C. S. Raghavendra. Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks. *Performance, Computing, and Communications, 2004 IEEE International Conference on*, April 2004.

[21] J. Mišić and V. B. Mišić. Modeling bluetooth piconet performance. *IEEE Communication Letters*, 7:18–20, Jan. 2003.

[22] J. Mišić, S. Shafi, and V. B. Mišić. Maintaining reliability through activity management in 802.15.4 sensor networks. *In Proc. Qshine 2005*, August 2005.

[23] Y. Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. *In* Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing., pages 177 – 188, June 2003.

[24] S. Singh and C. S. Raghavendra. PAMAS: power aware multi-access protocol with signalling for ad hoc networks. *ACM-CCR*, 5:5 – 26, July 1998.

[25] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. A self organizing wireless sensor network. *Proc. 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999.

[26] H. Takagi. *queueing analysis.* vol. 1: Vacation and Priority Systems*, 1991.*

[27] M.A.M. Vieira, L.F.M. Vieira, and L.B. Ruiz. Scheduling nodes in wireless sensor networks: a voronoi approach. Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on*, October 2003.*

[28] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. Networking, IEEE/ACM Transactions on*, 12:493 – 506, June 2004.*