

Northumbria Research Link

Citation: Richert, Viktor, Issac, Biju and Israr, Nauman (2017) Implementation of a Modified Wireless Sensor Network MAC Protocol for Critical Environments. Wireless Communications and Mobile Computing, 2017. ISSN 1530-8669

Published by: Wiley-Blackwell

URL: <http://dx.doi.org/10.1155/2017/2801204> <<http://dx.doi.org/10.1155/2017/2801204>>

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/35956/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



**Northumbria
University**
NEWCASTLE



UniversityLibrary

Research Article

Implementation of a Modified Wireless Sensor Network MAC Protocol for Critical Environments

Viktor Richert, Biju Issac, and Nauman Israr

School of Computing, Teesside University, Middlesbrough, UK

Correspondence should be addressed to Biju Issac; bissac@ieee.org

Received 29 June 2016; Accepted 4 September 2016; Published 11 January 2017

Academic Editor: Pierre-Martin Tardif

Copyright © 2017 Viktor Richert et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A Wireless Sensor Network (WSN) is a network of many nodes. These nodes are equipped with sensors which communicate wirelessly using techniques for radio frequency transmission. This network helps to measure and record the physical environment variables and to forward these results to a central location known as a sink. As WSN nodes are only supplied by a battery, the primary challenge is to reduce the energy consumption. The MAC layer is responsible for the establishment of a reliable and efficient communication link between WSN nodes and is responsible for energy waste. The newly proposed MAC protocol in this paper uses an improved variant of CSMA which implements weak signal detection (WSD). This technique enables dividing collisions from weak signals and takes appropriate decisions to reduce energy consumption. The CSMA/WSD protocol is presented as a flowchart and implemented in OMNeT++ by using the MiXiM framework structure. Implementation tests are performed to prove the validity of the implemented protocol in different scenarios. Different simulation scenarios show that this protocol offers a higher throughput, a smaller mean backoff time, and less average delay in critical environments.

1. Introduction

A WSN is a network of many small computing nodes. These nodes are equipped with sensors and communicate wirelessly, using radio frequency transmission. The aim of this network is to measure and record the physical environment and to communicate together. Using a multihop communication, it allows the measured data to be forwarded over a long distance to the main location which is known as a sink. To reduce the energy consumption in WSN requires optimisation across all layers. The medium access control (MAC) layer is a part of the data link layer which plays one of the most crucial roles in the communication protocol's overall energy efficiency [1]. Sun et al. [2] have done a survey on energy efficient neighbour discovery in mobile ad hoc and wireless sensor networks.

The establishment of a reliable and efficient communication link between WSN nodes is imperative. As the nodes are equipped with limited resources and power, the following characteristics for WSN MAC protocols should be considered [3] as listed: energy efficiency, reliability, low access delay, and high throughput. To design a new MAC protocol and to prove that it performs well,

the experiments with different WSN scenarios are necessary. A common way to test new applications and protocols is to use appropriate WSN simulation tools [4]. Rajandekar and Sikdar [5] present a survey of the challenges and the current research on medium access control (MAC) layer protocols for supporting machine to machine communications.

The aim of our work is to create a modified MAC protocol in WSN which could overcome the disadvantages of existing protocols in relation to energy efficiency, reliability, low access delay, and high throughput. Further aspects such as security, mobility, and real-time usage are also important but not the essential aspects in this paper. The new MAC protocol will be implemented and evaluated in a WSN simulation tool. This paper describes the result of our work and is divided into six sections. The second section provides related works on common MAC protocols and their characteristics. The third section gives an introduction to OMNeT++ and the used MiXiM framework. The fourth section describes the modified MAC protocol, while the fifth section shows the simulation scenario and the results. The sixth section is the conclusion.

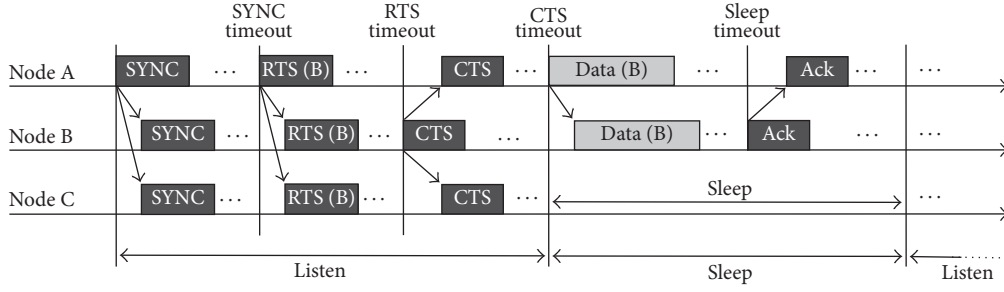


FIGURE 1: Basic S-MAC scheme.

2. Technology and Related Works

There are different types of MAC protocols that can be classified into two categories [6, 7] as CSMA/CA (carrier sense multiple access with collision avoidance) and TDMA (time division multiple access) protocols. CSMA/CA protocols are based on carrier sense multiple access (CSMA) which is a probabilistic technique where each node listens (carrier sense) before it sends and if nobody transmits the node will try to transmit a packet. The term multiple access means that multiple sensor nodes can access the medium at the same time, and a concurrent transmission causes a collision which has to be resolved by a technique such as Binary Exponential Backoff (BEB). In contrast to collision-based protocols, the time division multiple access (TDMA) protocols are known as deterministic protocols. These are using a schedule plan which associates a time slot for each sensor node and enable avoiding collisions and reducing effects of overhearing and idle listening problem [6, 8]. These protocols require the presence of a management authority like a dedicated access point which manages the schedule [8].

Some related works on MAC protocol are as follows. Li et al. [9] proposed a novel delay tolerant MAC protocol (DTMAC) that used distributed coupon collection algorithm in an underwater wireless sensor network. The throughput of DTMAC greatly outperforms that of MAC protocol with RTS/CTS scheme in most underwater scenarios. Buratti and Verdone [10] proposed a novel contention-based medium access control (MAC) protocol L-CSMA meant for linear topologies. L-CSMA assigns different levels of priority to nodes where nodes closer to the destination have higher priority when accessing the channel. The model is validated through simulations. Zheng et al. [11] proposed a novel MAC protocol known as wireless arbitration (WirArb) which allows each user channel access based on their different priority levels and it supports multiple users. The results show that the protocol provides high performance. Shen et al. [12] proposed PriorityMAC, a priority-enhanced MAC protocol, designed for critical traffic in industrial wireless sensor and actuator networks. The experiments show that PriorityMAC efficiently handles different traffic categories with different latency requirements. Peng and Cui [13] proposed a mobile-supporting mechanism for MAC protocols, where the decision to update the neighbours of a mobile node is made dynamically based on the mobile speed. The simulation results demonstrate that the mechanism achieves

a better performance. Huang et al. [14] proposed an energy efficient receiver-centric MAC protocol called RC-MAC that combines duty cycling and receiver-centric scheduling, giving high throughput.

2.1. MAC Protocols. The following discussion describes some common MAC protocols in WSN. A common trend to reduce the power consumption is to let the nodes sleep as soon as possible. This avoids idle listening as soon as possible but causes synchronisation overhead. The majority of the presented protocols in this section uses different techniques to enable a long sleep session as soon as possible. The majority of WSN nodes are equipped with limited hardware to reduce production costs. Furthermore, they use only one transceiver which receives or transmits at the same time (half-duplex mode). Therefore, all the described protocols in this section are usable for nodes with only one transceiver.

2.1.1. IEEE 802.11. This MAC protocol is a CSMA/CA-based protocol and implements control packets to avoid a collision as soon as possible. The Power Save Mode (PSM) reduces idle listening by periodically entering into the sleep state. Unfortunately, this PSM is not suitable for multihop networks such as WSN [6] and is therefore not usable in WSN.

2.1.2. Sensor S-MAC. This MAC protocol is CSMA/CA-based protocol and a well-known protocol in WSN [8]. It is a modification of the IEEE 802.11 protocol which overcomes previous disadvantages of this [6].

Figure 1 shows the basic scheme of this protocol. It is divided into two parts, a listening session and a sleep session. The listening session enables sensor nodes to communicate with other nodes to exchange some control packets. During the sleep session, the nodes turn their radio off to save energy [6]. The first part of the listening session is the synchronisation by sending SYNC packets. Every node maintains a schedule table which stores all the schedules (listen and sleep sessions) of all its known neighbours. During the synchronisation, a node broadcasts its schedule to all its neighbours [15]. After that, every node has a schedule of its neighbours and can use this to send data to the neighbours. Before a node wants to send data to its neighbour to arrange a data exchange, it knows the neighbour schedule, sends an RTS packet and waits for neighbour's answer. If the neighbour is ready to receive it sends a CTS packet and the data

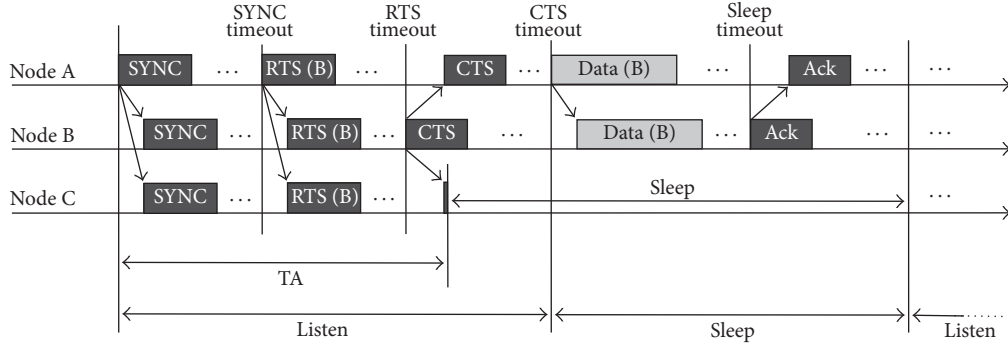


FIGURE 2: Basic T-MAC scheme.

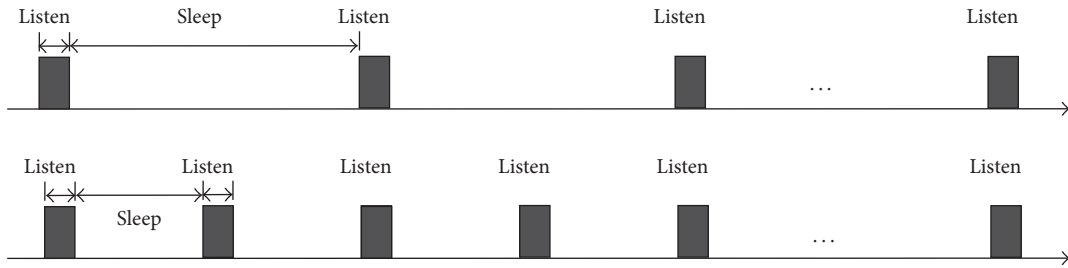


FIGURE 3: Dynamic duty cycling in DSMAC.

transmission can start immediately [15]. Nodes which are not participating in any data transmission change to sleep mode to save energy.

Using synchronisation (SYNC) packet, every node now has the schedule of its neighbour and can arrange a data transfer, and this enables the usage of this protocol in a multihop network [15]. It is required that every node maintains its scheduling table after a certain number of scheduled synchronisations. Therefore, every node should listen for a full period to find the neighbours who may have different time schedules. This causes packet overhead and is the disadvantage of this protocol [16].

2.1.3. Timeout T-MAC. This protocol is based on the S-MAC protocol and overcomes the S-MAC protocol by less energy consumption in terms of idle listening [6]. In contrast to the S-MAC, this protocol keeps the listening session in a variable length, which depends on the network load. It defines an interval T_A which is the minimum amount of idle listening per frame. If a node wants to synchronise with its neighbour for a data transfer using SYNC and RTS/CTS, this T_A interval is still higher than the exchange of these control packets. Figure 2 shows that node A is arranging a data transmission with node B. Node A has another contention interval T_C than node C. If node B is ready for data transmission, it replies using a broadcast CTS packet. The interval T_A should be long enough for node C to receive the beginning of the CTS packet. After that, node C turns automatically into sleep mode to avoid idle listening [17]. In a unidirectional connection, say in a node-to-sink communication, the T-MAC protocol has an early-sleeping problem.

2.1.4. DSMAC. This protocol is based on the S-MAC protocol and adds a dynamic duty-cycle feature (see Figure 3). As all nodes share their one-hop latency values, every node calculates the average value. If this value is too high, a node will decide to shorten the sleep time and announce this in the SYNC period. The major aim in this extension is to decrease the latency for delay-sensitive applications [18].

However, in contrast to the initial S-MAC, frequent sleep periods increase the idle listening and the additional announcements in the SYNC period increase the overhead.

2.1.5. WiseMAC. This protocol is a nonpersistent CSMA protocol which uses preamble sampling to reduce idle listening. Figure 4 shows the function principle of this protocol. Every node measures the received signal strength regularly (sampling) to check the activity. All nodes have the same constant sampling period but independent timings for sampling.

If node A wants to transmit data to another node B, it waits for the next wake-up period of node B and sends a preamble P first. If node B wakes up and measures that the medium is busy, it continues to listen and receive the subsequent data packet. After this successful data transmission, node B sends an acknowledgment packet. In case that the medium is idle, node B would change automatically to sleep mode and sleep for the next sample.

In an initial state, a node defines a preamble size which is equal to the sampling period. This preamble will be transmitted and the size of a full period will ensure that every neighbour sees this preamble during this wake-up period. This long preamble causes more power consumption. To reduce power consumption, every node learns the sampling

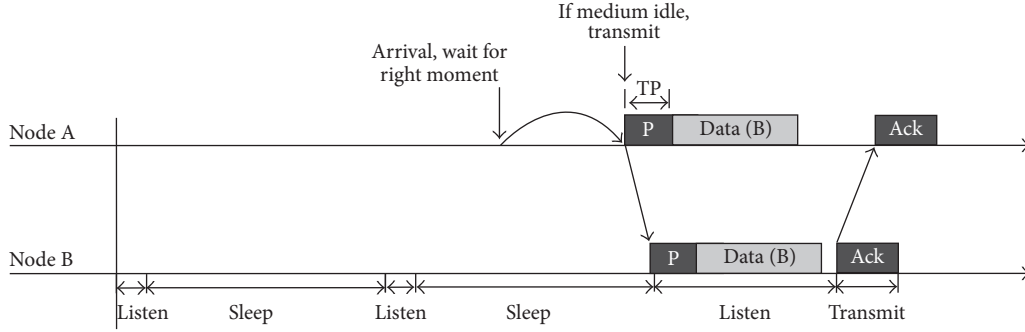


FIGURE 4: Basic WiseMAC scheme.

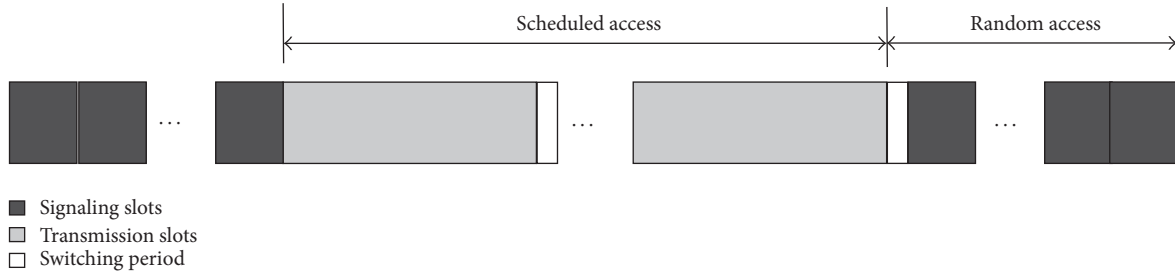


FIGURE 5: Time slot organisation in TRAMA.

schedule of its direct neighbours and minimises the preamble as far as possible. Using piggybacking mechanism into the acknowledgment packet, every node spreads out its neighbour's sampling schedule. This enables every node to learn or refresh the sampling schedule of its neighbours to reduce the preamble [1].

2.1.6. TRAMA. This protocol is a TDMA based protocol which reduces energy by ensuring a collision-less transmission and more usage of node's low power sleep mode [6]. This protocol consists of three components [21]:

- (i) Neighbour Protocol (NP): collecting information's about neighbouring nodes
- (ii) Schedule Exchange Protocol (SEP): exchanging information about two-hop neighbour and their schedule
- (iii) Adaptive Election Algorithm (AEA): deciding the transmitting and receiving nodes for the next round during the current time slot by using neighbourhood and schedule information. All other nodes are in sleep mode during this time.

The time slots in this protocol are organised and shown in Figure 5 [21]. It divides these slots which are collision-free and usable for data transmission.

To avoid collisions and the hidden node problem, every node has a knowledge of its two-hop neighbour. Using the NP, a node spreads out its one-hop neighbour information to its direct neighbours. This enables a node to know about its neighbour's neighbour (the two-hop neighbour) [21]. The nodes transmit the NP during random access periods to permit node additions and deletions. If a node does not hear from its neighbour for a certain period, it times

out this neighbour and shares this information with other neighbours. This makes the network more dynamic [21].

2.1.7. DMAC. Similar to TRAMA, this protocol is TDMA based. The assumption is that, in many applications, the major traffic consists of a data collection from several nodes, and it forwards it to a sink by the use of a unidirectional tree. This is the reason why this protocol is designed and optimised for data gathering trees in WSN. Figure 6 shows that this protocol divides the time into small slots. In a network with n nodes, every node receives and transmits after n -sleep slots. A tree which has the depth d enables a source node to transmit its data to the sink node (root) in maximum d time intervals [19].

A multihop transmission through the tree requires that the node's path to the sink (e.g. V_2 , V_3 , and V_4) has bigger transmitting intervals as these are additionally forwarding previous neighbour's data. Using the more data flag, a node can inform the next node that the duty cycle is bigger; either its buffer is not empty, or it has received a packet from the previous hop. A node which is a father of two children (e.g., V_8) cannot forward data from these two nodes concurrently. This problem is solved by data prediction scheme which let the nodes schedule an additional receiving slot to forward both data packets separately. Dai et al. [22] designed an energy efficient MAC protocol for data gathering in linear wireless sensor networks. Results show that it provides better energy efficiency and long lifetime than the existing DMAC protocol.

In summary, the CSMA-based protocols T-MAC and DSMAC are adapted from S-MAC protocol and react well to topology changes. However, the T-MAC protocol is more

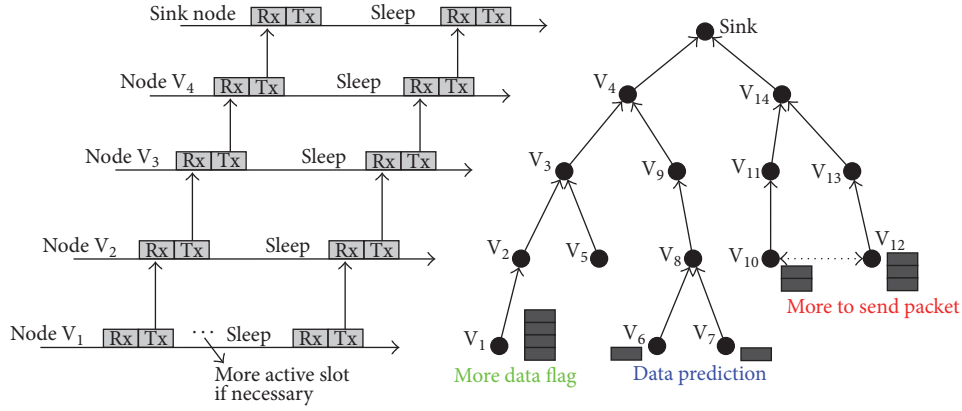


FIGURE 6: DMAC in a data gathering tree [19].

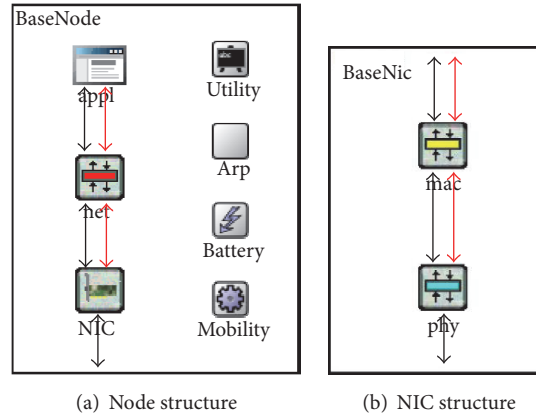


FIGURE 7: MiXiM's node and NIC structure [20].

energy efficient because it avoids idle listening more and the DSMAC uses a dynamic duty cycle which has good adaptivity to topology changes and therefore a smaller delay. The WiseMAC protocol is similar to the DSMAC protocol, but it consumes more because it is more vulnerable to collisions and has a long preamble. The TDMA protocols TRAMA and DMAC require time synchronisation which increases the transmission delay. However, this synchronisation makes them more efficient and increases the throughput. The different types of protocols are proposed for WSN, but none of them was accepted as standard. The choices of the appropriate MAC protocol depend on the application, and this is one reason why there is no specific one standardised MAC protocol for WSN. The standardisation at other lower layers such as physical layer and sensor hardware is another reason why none of these protocols is standardised [18].

3. OMNET++ and MiXiM Framework

OMNeT++ is used for simulation because it is a certified simulation environment used by many research institutions for protocol evaluation. The mixed simulator (MiXiM) is a combination and extension of various simulation frameworks

which were developed for mobile and wireless simulations in OMNeT++ [20].

The MiXiM framework uses the model structure to create a WSN node with all necessary layers and additional modules for physical hardware behaviour. Figure 7(a) shows the internal structure of the node module. It combines the upper application layer (appl), the network layer (netw), and a Network Interface Card (NIC). The NIC in Figure 7(b) consists of further submodules, the MAC layer (mac) and the physical layer (phy).

The grouping of the physical and MAC layer design in one NIC is required because these in some communication protocols such as IEEE 802.15.4 are more coupled together [20]. The exchange of the NIC enables exchanging the communication protocol easily. Shaukat and Hashim [23] have done a detailed analysis of the design using OMNeT++ (MiXiM Framework) for Mobile Wireless Sensor Network and discusses in detail the mobile node's architecture and the mobile node's movement.

4. CSMA/WSD Protocol Design

This section describes the new protocol idea and supports the detailed description with a flowchart. The proposed MAC

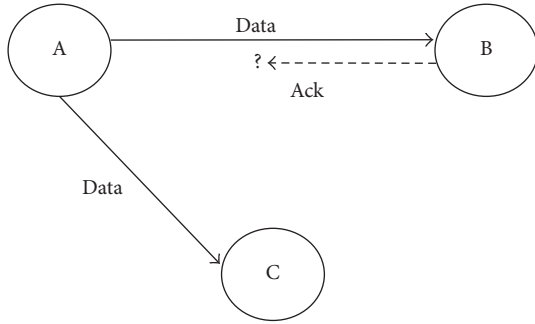


FIGURE 8: Data transmission with no Ack reply.

protocol in this research uses an improved variant of CSMA by using weak signal detection (WSD) to divide collisions from weak signals and take appropriate decisions to reduce energy consumption.

4.1. Protocol Idea. Apart from the TDMA based protocols which try to avoid collisions, the majority uses CSMA as a base to handle collisions. Even the standardised communication protocols like ZigBee and Z-Wave use CSMA to access the medium and handle collisions. The CSMA method was involved in the early 1970 and is widely used in the wired (802.3) and wireless (802.11) Ethernet technologies. It can detect concurrent access by two or more participants which cause a collision [24].

The collision detection in wired Ethernet is archived by participants who transmit a packet and continue listening for incoming signal (collisions) and in the case of a collision, these participants emit jamming signals which notify all other participants that a collision is detected. This enables timely feedback to resolve the concurrent access. This timely collision detection for wireless media is hard to detect because the strongest signal from the closest source always dominates the receiver circuitry. The receiver which is close to a transmitter is not able to receive another concurrent transmission and detect collision [24] like in wired scenario.

Some attempts use the second antenna to detect and abort collisions during the transmission immediately. This approach is known as Carrier Sense Multiple Access with Collision Notification (CSMA/CN) and obtains part of CSMA/CD's benefits in wireless environments [25]. But the major problem is still the cost for the second antenna. The most WSN applications are still using only one antenna and a simple transceiver for communication to save costs in the design and the implementation phase. The CSMA/CA should be a solution to this problem because the receiver will send an Ack to confirm that the data packet has arrived successfully. If a collision occurs, the transmitter which waits for an Ack will identify the occurrence of a collision after a timeout. However, there are several other reasons why a data transmission is not confirmed by an Ack reply [24].

Figure 8 shows that node A sends a data packet to node B, but this node does not reply to node A by an Ack. However, wireless signal transmissions are influenced by multiple external factors such as reflection and absorption.

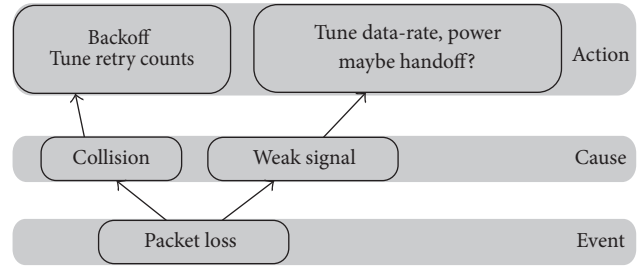


FIGURE 9: Example of a loss diagnosis [24].

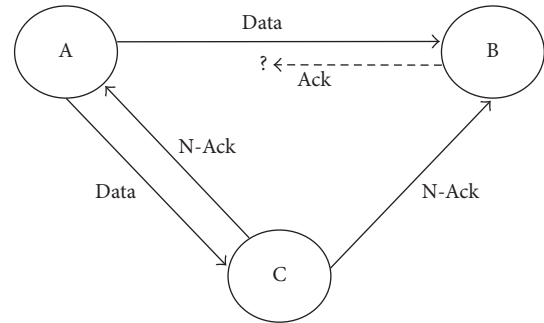


FIGURE 10: Data transmission with N-Ack reply.

These effects can influence the data as well as the possible Ack response. Following CSMA/CA protocol, this case is a collision, and the BEB algorithm will increase a CW and retransmit the packet at a randomly chosen time. This brings an additional delay which affects the throughput of the network, and the nodes are confronted with longer idle listening periods which cause higher energy consumption.

If node A could recognise the reason for packet loss, it could take appropriate strategy to handle in this situation. Rayanchu et al. [24] call the problem of determining the cause of packet loss as loss diagnosis. Figure 9 shows that different actions are performed after a loss diagnosis. In the case of a collision, a BEB algorithm will be performed and in the case of a weak signal some parameters such as data rate and transmission power will be tuned.

The previous approach was used in WiFi where only a small number of nodes were used. This work concentrates on WSN where multiple nodes are communicating together in a mesh network. The approach in this work uses exactly the characteristics of a mesh network where a node usually has multiple neighbours.

As already shown in Figure 10, node A was not receiving any Ack from node B. However, node C receives the data successfully and knows it was addressed to its neighbour B. As node B never replied with Ack, node C assumes that the data packet was forwarded to B without any collisions between A and B. The only problems why node B was not able to reply with an Ack are as follows:

- (i) Node B is off or in sleep mode.
- (ii) The data packet from A to B has a weak signal.
- (iii) The Ack reply from B to C has a weak signal.

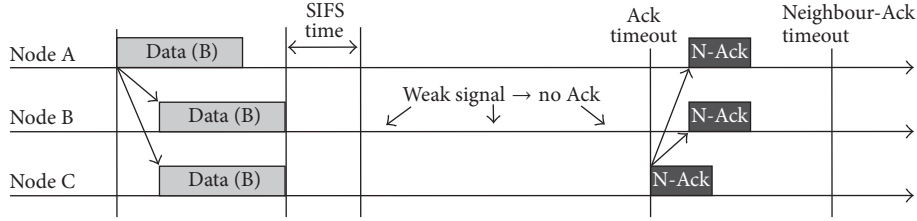


FIGURE 11: CSMA/WSD behaviour under weak signal conditions.

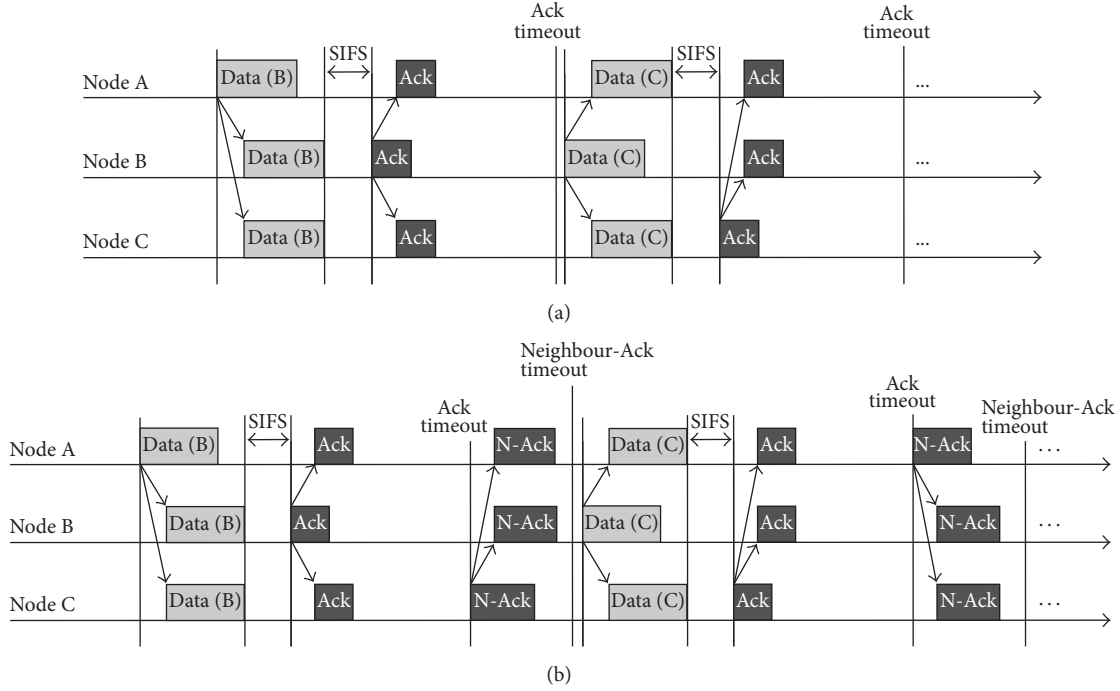


FIGURE 12: Throughput compares between CSMA and CSMA/WSD.

- (iv) Another node which is not visible for A and C was causing a collision with the original data or Ack packet.

The last case is only visible to node B and is not a collision between these three nodes. As node C knows that there is no collision between node A and B, it assumes that there could only be a weak signal and sends a neighbour-Ack (N-Ack) packet after a timeout to node A.

After receiving N-Ack packet, node A recognises that C is not the destination node and it assumes this packet has not arrived properly at node B. Using the BEB algorithm in this scenario would increase the time and energy consumption by idle listening. Instead of this, node A should retransmit the data packet immediately and does not increase CW.

Figure 11 shows the timely behaviour of the protocol idea. If node A sends a packet, it defines an Ack timeout and within that it expects an Ack from node B. In this case, due to the weak signal, node B will be unable to respond with an Ack. After the Ack timeout, node C sends a neighbour-Ack to show

that it does not receive an Ack at all which offers node A to retransmit the packet immediately.

The Short Interframe Space (SIFS) is a small time interval between data and Ack packet. It ensures the sender (node A) has enough time to switch from transmitting to receiving mode and to receive the Ack reply. Initially, this new protocol idea decreases the throughput slightly in an ideal environment with no packet loss. Figure 12 compares both protocols, the original CSMA (a) and CSMA/WSD (b) which have the same Ack timeout. However, the new protocol waits additionally for the neighbour-Ack which decreases the protocol throughput slightly.

Theoretically, the new idea results in a small mean backoff time and a high throughput. If the wireless channel causes packet loss due to a weak signal, the neighbour-Ack provides support by keeping the backoff time reduced. This enables transmitting more packets in a short period and therefore increases the throughput. Figure 13 shows the actions which the CSMA/WSD protocol performs after a loss diagnosis. In the case of a collision a BEB algorithm will be triggered to increase the retransmission time and in the case of a

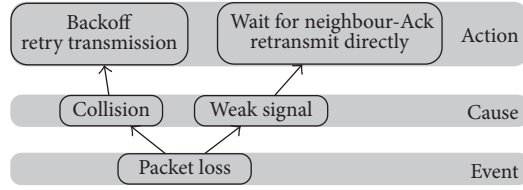


FIGURE 13: CSMA/WSD loss diagnosis.

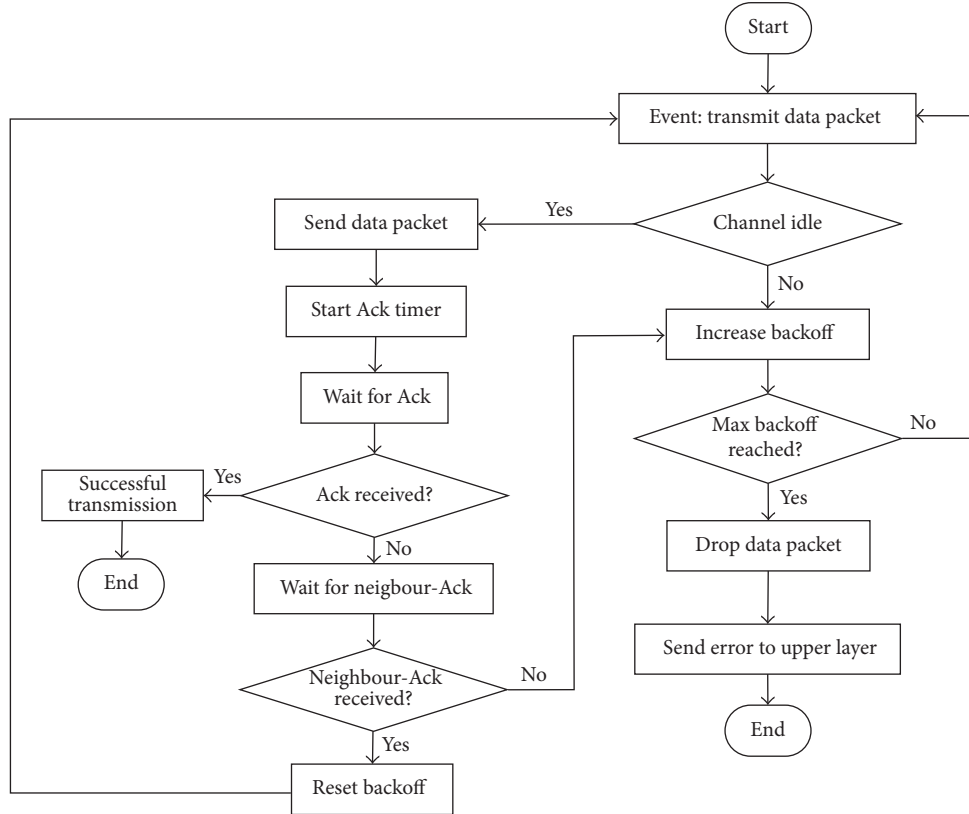


FIGURE 14: Flowchart for transmitting.

weak signal, a neighbour-Ack will be received and the packet directly retransmitted.

4.2. Protocol Description. Both the flowcharts give a diagrammatic representation of the two base functionalities of the CSMA/WSD protocol, namely, “transmit” and “receive.” As in Figure 14 if a node has to transmit a data packet, it goes through all steps in the given flowchart. It checks first whether the channel is idle or not and then it transmits the data packet; otherwise, it increases the backoff time. After a data packet is transmitted, the node waits for an Ack. If there is no Ack received, the node waits for a neighbour-Ack. If both Acks are not received, the backoff time increases.

The flowchart in Figure 15 gives an overview of the process in the receiving nodes. Each receiving node checks first if the packet is directed to it. If yes, a node starts an SIFS timer, and after the timer runs out, it sends an Ack. For packets which are not directed to the node, the node checks whether the

destination node is already sending an Ack back to the source or not, and it sends a neighbour-Ack.

5. Implementation and Simulation

5.1. Preparation of the Simulation Environment. The implemented protocol requires the simulation of collisions and weak signal. To ensure that the MiXiM framework can simulate this kind of behaviour, different scenarios were created and simulated. The MiXiM framework provides the simulation of real environment behaviour such as signal strength, attenuation, and delay. The ZigBee protocol is one of the most used communication protocols in WSN. It implements the physical and MAC layer of the 802.15.4 standard. To prepare the simulation environment, values such as maximum distance or SNR which are based on the 802.15.4 standard are used in this section. To decide if a packet is received correctly, a signal quality criterion

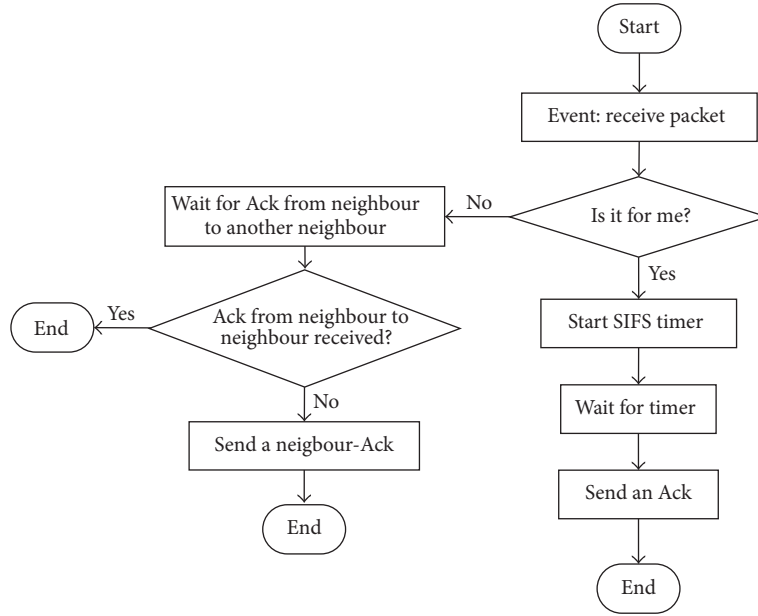


FIGURE 15: Flowchart for receiving.

should be defined. The used decider in both scenarios is the *SNRThresholdDecider* method in MiXiM which drops frames below a defined SNR value. Figure 16 shows the BER over the SNR for different communications standards.

As all scenarios are based on the 802.15.4, only this curve is necessary. But to find an appropriate BER value, some reference or recommendation is required. Based on a user guide for WSN node product which uses 802.15.4 standard transceiver, the commonly used threshold to determine the sensitivity is a PER of 1% [26]. We are using the following formula for the connection between PER and BER:

$$PER = 1 - (1 - BER)^{packet_length}. \quad (1)$$

It allows calculating the necessary BER:

$$BER = 1 - \sqrt[packet_length]{1 - PER}. \quad (2)$$

The used packet length in all scenarios is 568 bits.

$$BER = 1 - \sqrt[568]{1 - 0.01} = 0.0000176941 \approx 1.8 \cdot 10^{-5}. \quad (3)$$

Using Figure 16, an approximate SNR value of 1.5 dB is as follows:

$$SNR = 10^{1.5 \text{ dB}/10} \approx 1.4125. \quad (4)$$

This value is used as a criterion for all the following scenarios.

5.1.1. Scenario 1: Collision in MiXiM. This scenario simulates the collision between two packets which are transmitted at the same time. As a collision scenario in WSN requires at least three nodes where two nodes send at the same time, the third node is listening and detects the collision. Figure 17 shows the scenario where node[0] and node[2] send data at the same time and node[1] is receiving.

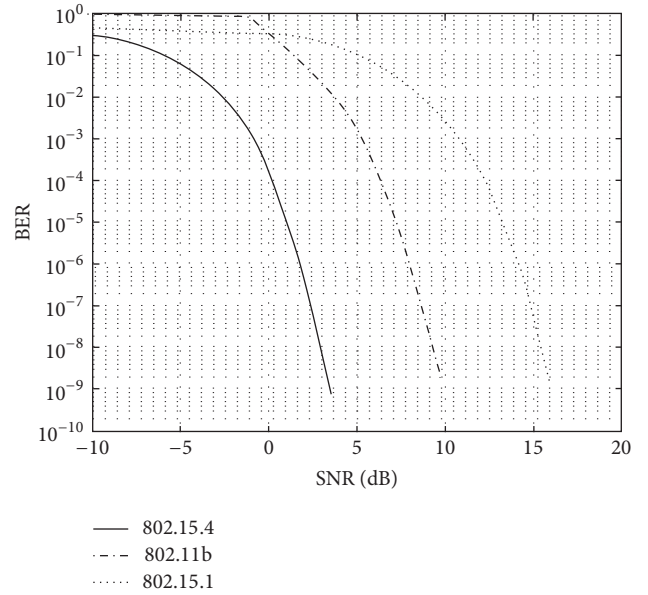


FIGURE 16: Theoretical BER to SNR for different communication protocols [27].

In accordance with the BEB algorithm, the nodes increase the contention window by detecting collisions. The CSMA MAC layer in MiXiM uses for every sent packet an Ack reply to confirm a successful transmission. If node[0] and node[2] send at the same time, a collision occurs, and node[1] cannot reply with an Ack because the data is corrupted. Both sender nodes recognise this unsuccessful data transmission and use the BEB algorithm to increase the contention window. In this bigger contention window, both nodes chose a random time for retransmission to avoid another collision.

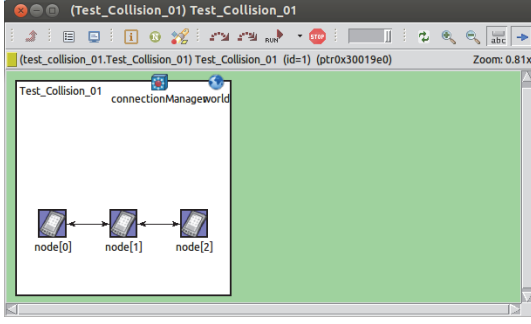


FIGURE 17: Collision scenario.

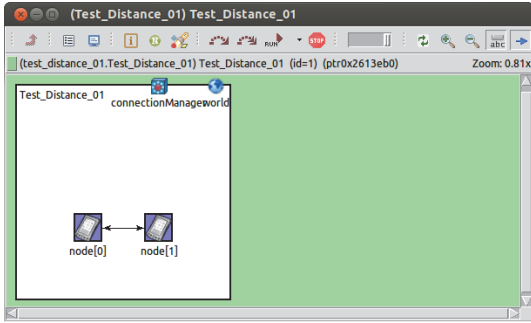


FIGURE 18: Weak signal scenario.

Originally the probability of collisions decreases if the contention window grows and the collision should be resolved. However, in this scenario, the CSMA MAC layer was manipulated to let all nodes choose the last possible retransmission time within the contention window. This will let the nodes collide consequently and enable resolving the situation if the BEB algorithm increases the contention window correctly. It is visible that the contention window increases the first ten retransmissions exponentially and reaches a value of 1024. It uses this value for the next six retransmissions. After 16 unsuccessful retransmissions, a MAC Error message will inform the upper layers.

5.1.2. Scenario 2: Weak Signal in MiXiM. This scenario (see Figure 18) simulates a transmission between two nodes where the distance causes a weak signal. The SNR of this signal is below the allowed threshold, and that causes packet loss.

The MiXiM framework provides different wireless channel modes which influence the signal strength. To simulate channel behaviour in WSN two different models are used simultaneously.

BreakpointPathlossModel. This model simulates the basic attenuation influence on the signal depending on the distance of the node. Figure 19 shows the theoretical and experimental results. The experimental environment consists of two nodes which communicate with each other at different distances. Both nodes measure the signal power of received signals by using their built-in Radio Signal Strength Indicator (RSSI). However, this measurement result shows the relative signal strength.

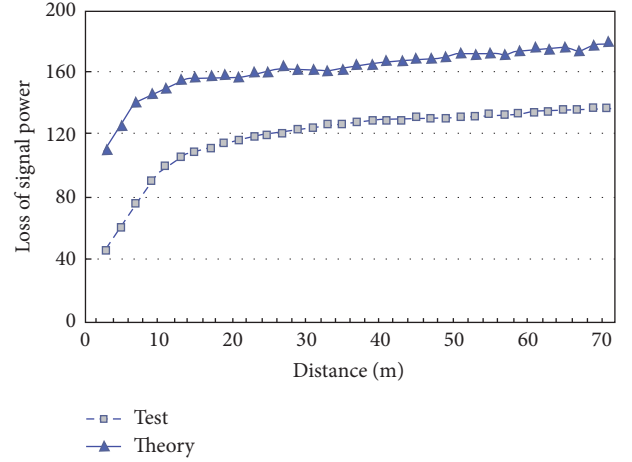


FIGURE 19: Signal attenuation in a theoretical model and real measurement [28].

The transceiver CC2430 used a receiving sensitivity of -90 dBm and the propagation distance of the signal is 75 meters [28]. In accordance with the datasheet of the transceiver CC2430, the $RSSI_OFFSET$ is approximately -45 [29, p. 168]. This means that the real measurement results in dBm are 45 dB below the shown test results in Figure 24. For example, the measurement result of 75 m distance would have a test result of approximately 125 dB. Subtracting 45 dB enables the value of 90 dB which is already the maximum receiving sensitivity of the transceiver. Furthermore, the experimental result shows that near the area of 20 meters the signal energy weakens quickly and above 20 meters it weakens slowly [28].

This behaviour requires a breakpoint model which simulates the attenuation in the closer range to be lower [30] and at longer range it grows higher and especially after the breakpoint in the opposite way. The behaviour of *Breakpoint-PathlossModel* method in MiXiM is shown in the following formula for the signal power loss:

$$P_{Loss}(d) = \begin{cases} 10 * \log_{10} \left(1 * 10^{PL_{01}/10} * d^{\alpha_1} \right), & d < d_b \\ 10 * \log_{10} \left(1 * 10^{PL_{02}/10} * \left(\frac{d}{d_b} \right)^{\alpha_2} \right), & d \geq d_b. \end{cases} \quad (5)$$

The following values enable simulating the red model in Figure 20. Adding 45 dB to this model enables archiving the same test result model in Figure 19. Therefore, this *BreakpointPathlossModel* method enables simulating the real distance attenuation model in WSN.

Logarithmical Shadowing Model. This model simulates physical effects such as reflection, diffraction, scattering, and absorption [20]. This shadowing let the signal fading and enables simulating that two nodes have from time to time, the good or bad signal between each other. The behaviour

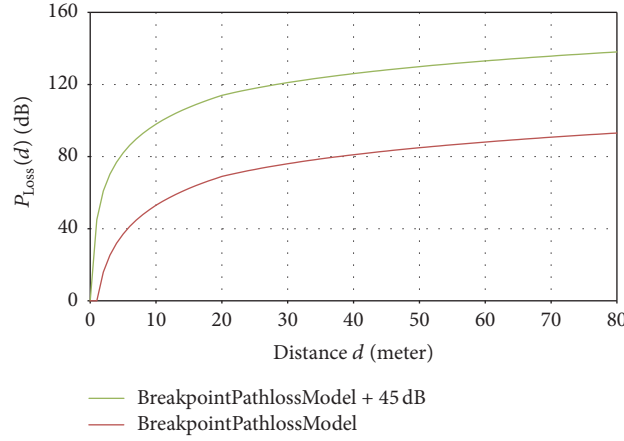


FIGURE 20: BreakpointPathlossModel.

of *LogNormalShadowing* method in MiXiM is using the following formula of the normal distribution [31]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} * e^{-\frac{1}{2}((x-\mu)/\sigma)^2}. \quad (6)$$

The model calculates a random x value from the normal distribution. The mean attenuation $\mu = 3.0$ dB and standard derivation $\sigma = 1.0$ dB are real measurement results [32] chosen.

These two scenarios show that the MiXiM framework provides the necessary simulation base for subsequent implementation.

5.2. Implementation

5.2.1. Extension of MiXiM's CSMA to CSMA/WSD. The MiXiM framework implements different MAC protocols already. The introduced new CSMA/WSD protocol is based on the original CSMA protocol. The idea for this new CSMA/WSD protocol is already described. The new protocol requires that only the direct neighbours can send acknowledgments in the case of an acknowledgment failure by the two direct participant nodes. This is the reason why the implementation uses additionally a *NeighbourList* to save the neighbours for each node. Each node learns about the neighbour during the transmissions and saves the information about its neighbours in this list. Taking advantage of this protocol requires that all nodes communicate before. This can be done by a simple broadcast packet exchange before the data transaction. The new implementation defines an additional *rxDirectAckTimer* method which is the deadline for Acks from directed nodes. The initial *rxAckTimer* method is used as the deadline for neighbour-Ack. Figure 21 shows the FSM of the CSMA/WSD protocol.

5.2.2. Implementation Test Framework. The implementation requires running different tests to prove the basic functionality of the implemented MAC protocol. The test case in Figure 22 uses the usual three-node setup which was already described in previous sections.

As Figure 23 shows, all three nodes use the same layer model. The MAC layer can be exchanged to enable a comparison between CSMA and CSMA/WSD by using a similar framework.

As the test application layer is responsible for creating the new packets, this was programmed to make the implementation based tests which create the necessary communication between these three nodes.

Figure 24 shows how the application layer performs. It first sends a broadcast from every node. This broadcast enables the nodes to learn from each other. After broadcasting, the only node[0] (or node A) sends one data packet to node[1] (or node B).

An implementation requires tests which are repeatable for tracing and debugging. The described analogue models previously have random signal quality behaviour which is not useful for implementation tests. To provide repeatability, a special *ScenarioModel* method for implementation scenarios is used. The success of the data packet from node[0] (Figure 23) and subsequent packets depends on this model. The connections between the nodes are viewed in Figure 25. The *ScenarioModel* method allows enabling or disabling each of these connections by setting a value in an array. This array has a length of nine and its indexes represent the connections in Figure 25. For example, if the transmission from nodes B to C should fail, the value in the array under the index five should be "0." In the case of success, the value should be "1."

5.2.3. Implementation Test. The implementation test framework let only node[0] send the data packet. Based on this, different constructive scenarios should be tested to prove and debug the implementation. The following Table 1 lists all possible constructive scenarios and compares both MAC protocols by using the same test parameters.

Both protocols are using an exponential backoff and in case the backoff increases, both protocols choose the latest time interval as a worst case. The unit backoff period is chosen as 40 ms which means that each time slot is 40 ms long. In case of an invalid connection, the BEB increase the number

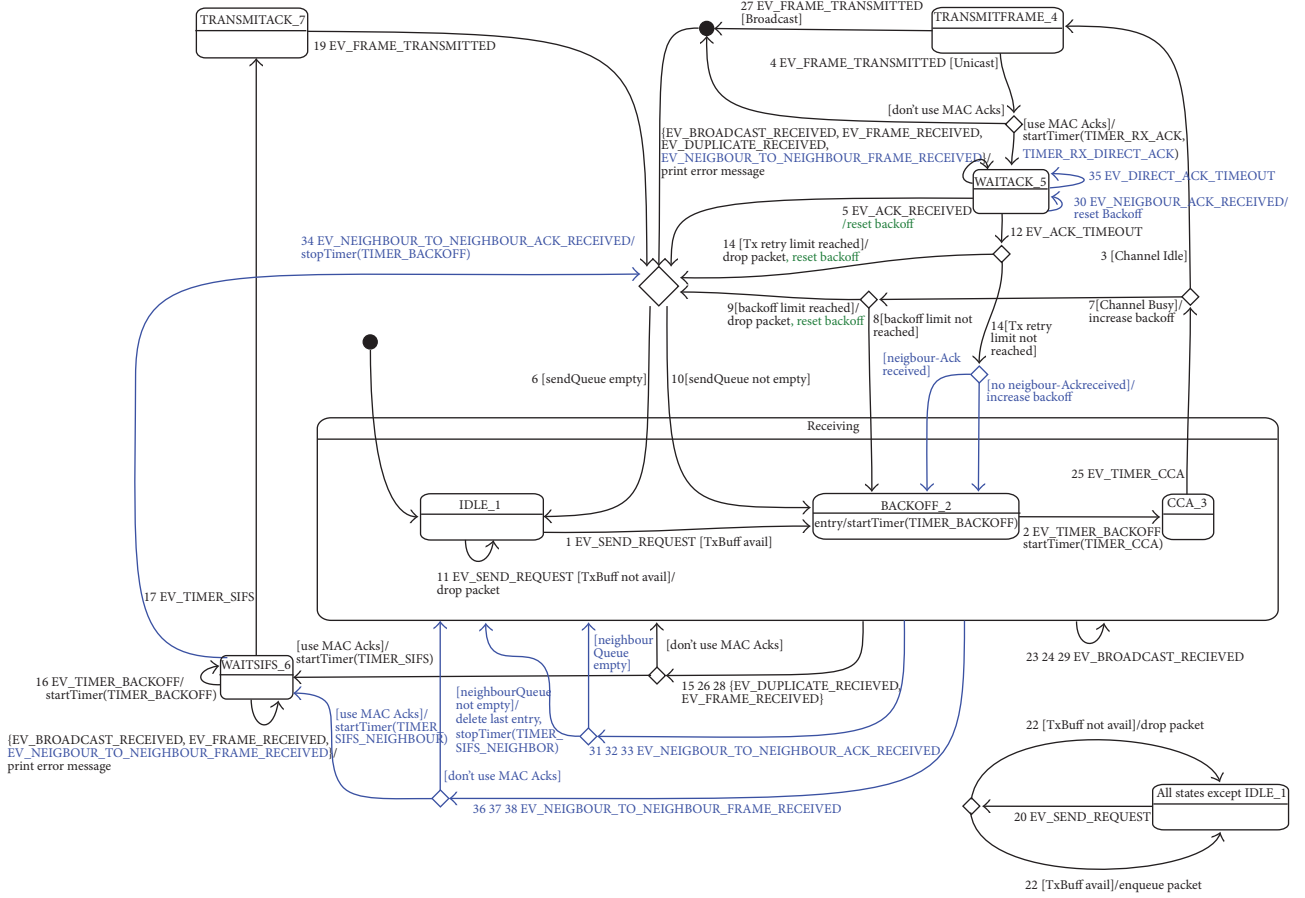


FIGURE 21: FSM of the CSMA/WSD protocol.

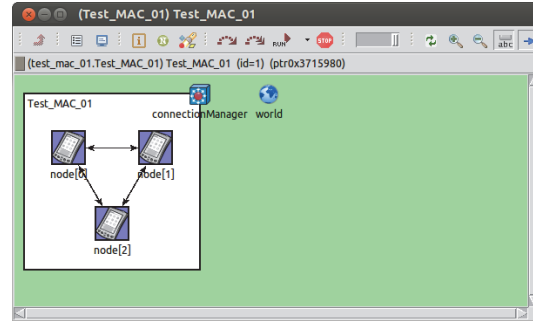


FIGURE 22: Implementation test scenario.

of time slots, and a MAC Error will occur after a max backoff time

$$\text{max backoff time} = \left(\sum_{i=1}^{10} (2^i - 1) + 6 * 1023 \right) \quad (7)$$

* 40 ms = 326.96 sec.

The first scenario shows nearly the same resolving time as calculated max backoff time. The reason for the inaccuracy is that the calculated time does not consider the additional delay for processing inside the nodes.

The results show some behaviour differences between both protocols. The recorded time is the real value excluding the time which is necessary for previous broadcasts.

In the first scenario, the data packet does not arrive at any other node. Node[0] will never receive an acknowledgment and repeat the transmission 16 times and drop the packet by the end.

It is visible that the CSMA/WSD need a little more time because it waits for an additional neighbour-Ack. But this is only 0.0156% more.

$$\frac{327.818 \text{ s}}{327.767 \text{ s}} * 100\% = 100.0156\%. \quad (8)$$

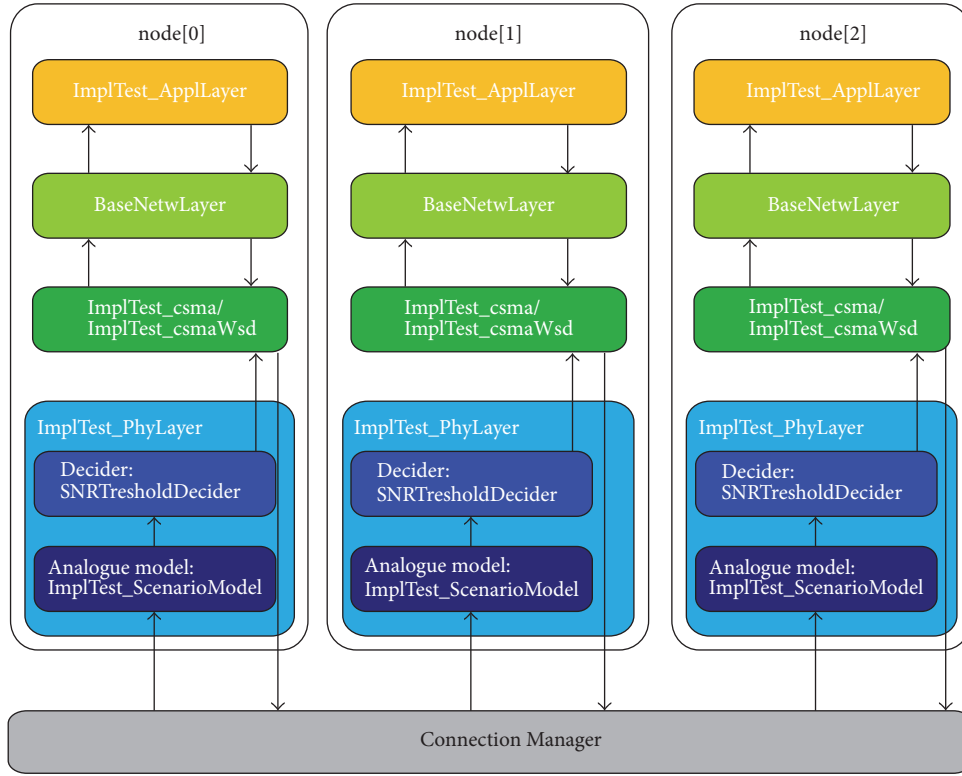


FIGURE 23: Implementation test framework.

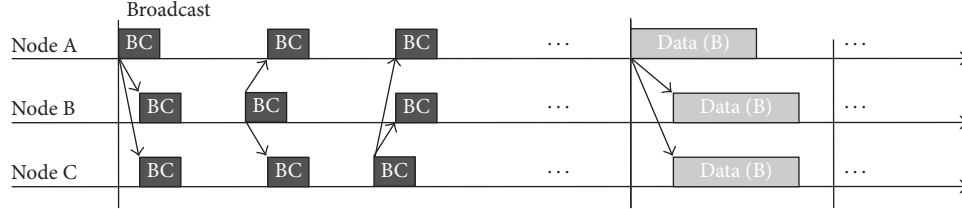


FIGURE 24: Implementation test application layer.

The big advantage of the CSMA/WSD is in scenario 7 which makes use of neighbour-Ack and reduces the resolving time by a good factor as below.

$$\frac{327.767 \text{ s}}{0.85815 \text{ s}} = 381.94. \quad (9)$$

The disadvantage of this protocol is shown in scenario 16. This scenario is a successful transmission between both nodes A and B. It triggers an additional neighbour-Ack which is not necessary and spends time as well. This overhead increases the time shortly by 24.33%.

$$\frac{0.05008 \text{ s}}{0.04028 \text{ s}} * 100\% = 124.33\%. \quad (10)$$

Based on all these scenarios, it shows that the usage of the neighbour for acknowledging increases the reliability and reduces time in cases where the backoff increases exponentially. In cases of successful transmission between direct participants, the time increases. The next section shows if this

protocol brings an advantage in a real communication scenario where the time for communication is chosen randomly, and the packets are arriving depending on the signal strength which varies from time to time.

5.3. Simulation. Similar to the implementation, the simulation requires a test environment as well. This environment creates a communication load and handles the success of the transmission. The new MAC protocol is designed to detect weak signals and to increase the throughput. The simulation requires throughput performance tests and an application layer which constantly sends new packets after previous packets have arrived successfully.

Figure 26 shows how the simulation application layer performs. It sends initial broadcast packets to enable the node to learn from each other. After all broadcasts are sent, a transmission of data packets is started by a simulation time of 10 seconds. As all nodes send first at the same time, a collision (red) will be caused, and each node triggers

TABLE 1: Scenarios to compare both MAC protocols.

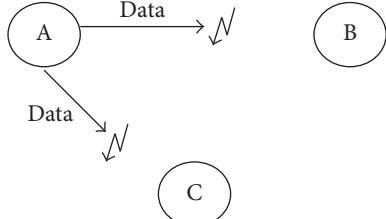
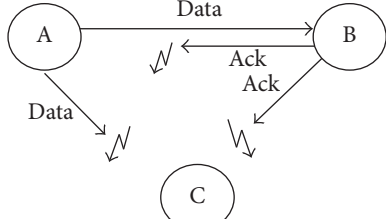
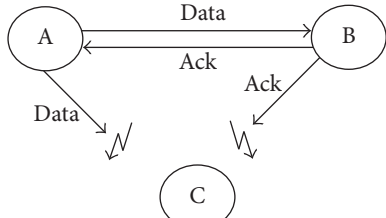
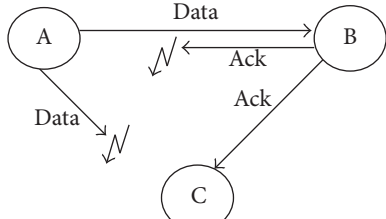
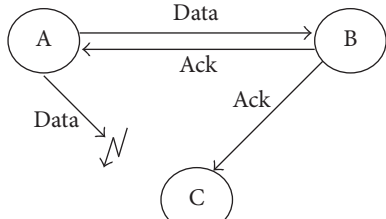
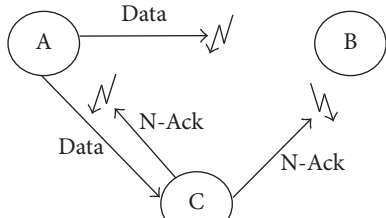
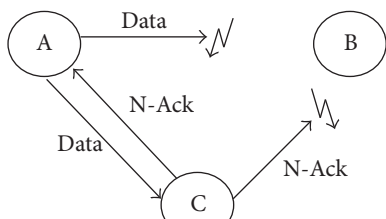
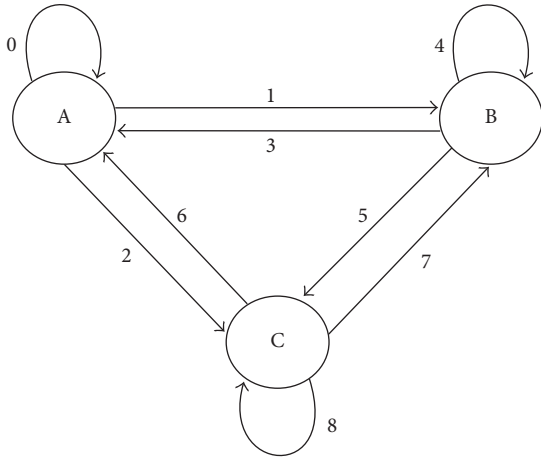
	Scenario	CSMA	CSMA/WSD
1		Result output (i) 1 transmission (ii) 16 retransmissions (iii) Backoff time increases exponential (iv) Resolving time: 327.767 sec	Result output (i) 1 transmission (ii) 16 retransmissions (iii) Backoff time increases exponential (iv) Resolving time: 327.818 sec
2		Result output was the same as in scenario 1	Result output was the same as in scenario 1
3		Result output (i) Successful transmission in one attempt (ii) Resolving time: 0.04028 sec	Result output (i) Successful transmission in one attempt (ii) Resolving time: 0.04028 sec
4		Result output was the same as in scenario 1	Result output was the same as in scenario 1
5		Result output was the same as in scenario 3	Result output was the same as in scenario 3
6		Result output was the same as in scenario 1	Result output was the same as in scenario 1
7		Result output was the same as in scenario 1	Result output (i) 16 retransmissions (ii) Backoff time stays constant (iii) Resolving time: 0.85815 sec

TABLE 1: Continued.

	Scenario	CSMA	CSMA/WSD
8	<pre> graph LR A((A)) -- Data --> B((B)) B -- N-Ack --> C((C)) C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 1
9	<pre> graph LR A((A)) -- Data --> B((B)) B -- N-Ack --> C((C)) C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 7
10	<pre> graph LR A((A)) -- Data --> B((B)) B -- Ack --> C((C)) C -- Data --> A B -- N-Ack --> C C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 1
11	<pre> graph LR A((A)) -- Data --> B((B)) B -- Ack --> C((C)) C -- Data --> A B -- N-Ack --> C C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 7
12	<pre> graph LR A((A)) -- Data --> B((B)) B -- Ack --> C((C)) C -- Data --> A B -- N-Ack --> C C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 1
13	<pre> graph LR A((A)) -- Data --> B((B)) B -- Ack --> C((C)) C -- Data --> A B -- N-Ack --> C C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 7
14	<pre> graph LR A((A)) -- Data --> B((B)) B -- Ack --> C((C)) C -- Data --> A </pre>	Result output was the same as in scenario 1	Result output was the same as in scenario 1

TABLE 1: Continued.

	Scenario	CSMA	CSMA/WSD
15		Result output was the same as in scenario 3	Result output was the same as in scenario 3
16		Result output was the same as in scenario 3	Result output (i) Successful transmission in one attempt (ii) Resolving time: 0.05008 sec (iii) Time for neighbour-Ack at the end required
17		Result output was the same as in scenario 3	Result output was the same as in scenario 16

FIGURE 25: Connections between nodes in the *ScenarioModel*.

the BEB algorithm to choose a random backoff time for retransmission. After that, each node sends to one of its neighbours and then it sends a data packet to its other neighbours as well (not visible in the figure). This is one sending period. After every node sent a packet to each

neighbour successfully, a new period for transmissions will be started. The number of period cycles in this layer is adjustable.

Both simulation scenarios use this application layer, and the number of periodic cycles in both simulations is adjusted to 50 so that every node sends 100 data packets to its neighbours. Following parameters in each layer are used in both simulation scenarios as in Table 2.

The whole size of the data packet is the sum of all the header lengths. The size of the Ack packet is only 24 bits. The MiXiM framework does not add any additional header to this Ack packet.

5.3.1. Simulation 1: Packet Loss Using PERModel. This simulation proves the throughput stability of the new MAC protocol in contrast to the CSMA protocol. The distance between the nodes and the signal quality are not crucial in this simulation.

The test framework in Figure 27 uses an analogue model which let a certain number of transmissions randomly fail. The percentage amount of random transmission failures is the PER. These can be adjusted from 0% up to 100%.

The first simulation result is shown in Figure 28 which compares the throughput in bits/sec over the PER value between two MAC protocols. The throughput is the sum of all node throughputs. Each node throughput is calculated by

$$\text{throughput} = \frac{\text{no. of received Acks} * \text{Ack size} + \text{no. of received data packets} * \text{data size}}{\text{whole transmission time}}. \quad (11)$$

TABLE 2: Simulation parameters.

Channel parameters	
Maximum transmission power	100 mW
Saturation	-90 dBm (minimum signal attenuation threshold)
Alpha	3.0 (minimum path loss coefficient)
Carrier frequency	2.412 GHz
Physical layer parameters	
Thermal noise	-77 dBm
Sensitivity	-90 dBm
Maximum transmission power	100 mW
Physical header length	0 bit
MAC layer parameters	
Transmission power	100 mW
Bitrate	15360 bit/sec
CCA detection time	0.0005 s
Unit backoff period	40 ms
Backoff method	exponential
MAC header length	24 bits
Ack packet length	40 bits
Ack wait duration (Ack timeout)	10 ms
Neighbour-Ack wait duration (neighbour-Ack timeout)	13 ms
SIFS time	192 μ s
Network layer parameters	
Network header length	32 bits
Application layer parameters	
Application header length	512 bits

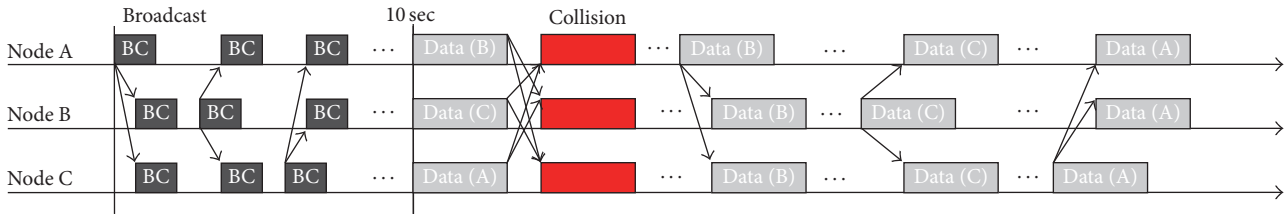


FIGURE 26: Simulation application layer behaviour.

The increasing PER value increases the number of unsuccessful data and Ack transmissions. These unsuccessful transmissions have to be retransmitted which will increase the time for the whole transmission and decrease the throughput in both protocols.

It is visible that the new CSMA/WSD protocol has less throughput with a PER value of 0%. The reason for this is the required time to listen for additional neighbour-Acks. If the PER value is higher than 20%, more transmissions fail and enable more impact for the usage of neighbour-Ack. These neighbour-Acks keep the backoff time small, and the whole transmission requires less time than the CSMA protocol. This increases the throughput significantly, and the CSMA/WSD appears more stable under higher PER values.

Figure 29 compares simulation results of mean backoff time (MBT) over the PER value between two MAC protocols.

The MBT is calculated as the average of each node's MBT which is calculated as

$$\text{mean backoff time node } [i] = \frac{\text{whole backoff time node } [i]}{\text{number of backoffs node } [i]} \quad (12)$$

The MBT increases with increasing PER value because more packets have to be retransmitted, and the BEB is triggered to increase the backoff time. The CSMA/WSD protocol has with 0% PER a low MBT to start with and it increases because the neighbour-Acks cause collisions as well. These collisions trigger the BEB algorithm more which increases the MBT. Further, with increasing PER value, the CSMA/WSD protocol assumes weak signals and by using neighbour-Acks it avoids

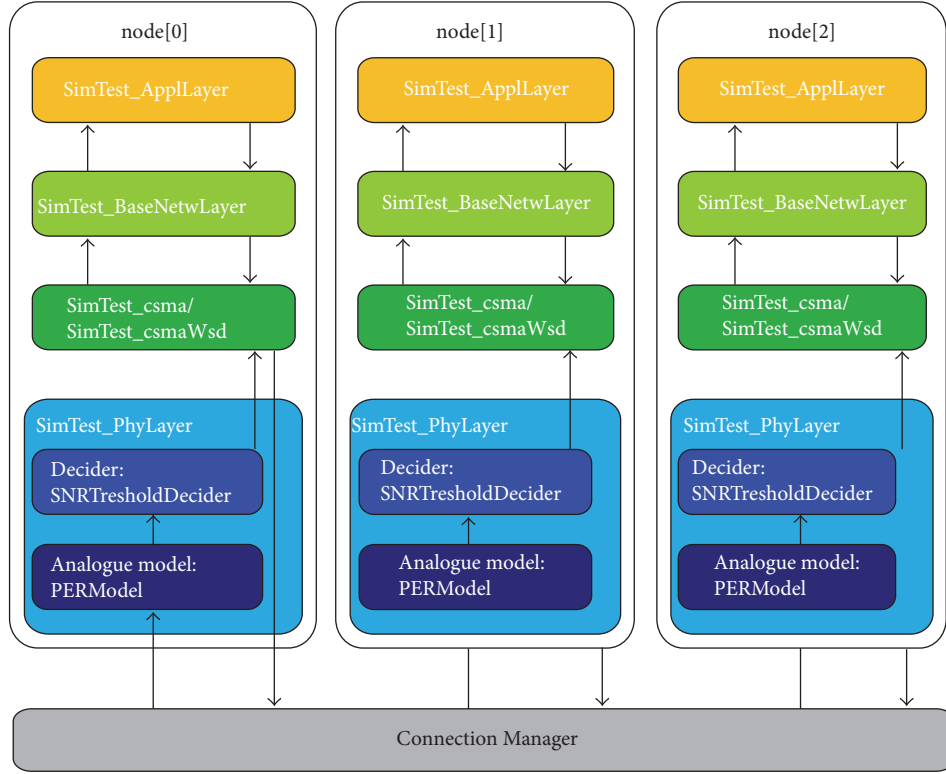


FIGURE 27: Simulation 1 test framework.

triggering the BEB which keeps the MBT lower especially with a PER higher than 30%.

The average delay is the average time which is required to transmit a packet from one node to the other node which is shown in Figure 30 and is calculated as

$$\text{average delay} = \frac{\text{whole transmission time}}{\text{average (no. of received Acks on each node) + average (no. of received data packets on each node)}} \quad (13)$$

The increasing PER value requires packet retransmission which increases the whole transmission time and therefore the average delay. As the CSMA/WSD has a higher throughput by using neighbour-Acks, it also has a small average delay with PER values above 30%.

Figure 31 shows the average number of collisions per 100 received data packets and is calculated as follows:

$$\begin{aligned} \text{no. of collisions} &= \text{no. of collisions in node [0]} \\ &+ \text{no. of collisions in node [1]} \\ &+ \text{no. of collisions in node [2]}. \end{aligned} \quad (14)$$

The number of collisions decreases with a higher PER value because the high packet loss causes less probability for collisions. It is visible that the new MAC protocol has more collisions because it transmits additional neighbour-Ack which increases the probability of collisions. The protocol has the biggest throughput advantage within a PER value

of 30% to 70% (Figure 28) and therefore also has a bigger collision rate.

5.3.2. Simulation 2: Packet Loss Using Analogue Models. This simulation compares the throughput stability on increasing distance between the new CSMA/WSD and the CSMA protocol. Figure 32 shows the structure of this simulation where the throughput and other values are measured under stepwise increasing distance d .

As before the test framework in Figure 33 uses two analogue models. These models simulate the real world signal behaviour under different node distances and are already described before.

The chosen values as physical parameters as discussed before are typical for nodes which use a ZigBee transceiver. Therefore, the maximum distance between these nodes is 75 meters. It is expected that a distance higher than 75 m will influence the measured values such as throughput and MBT.

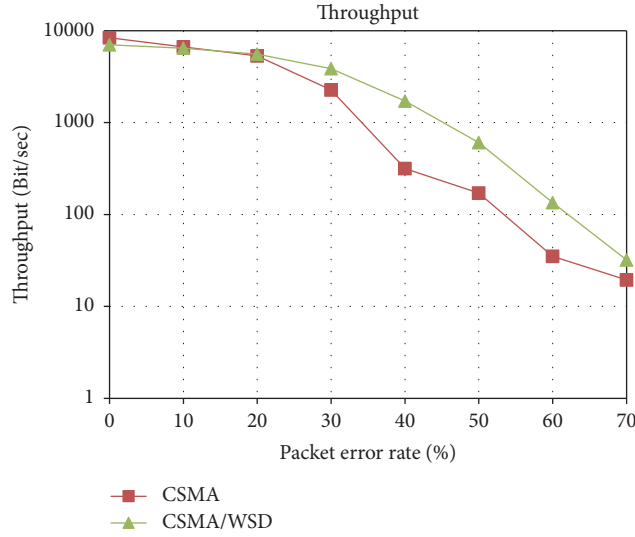


FIGURE 28: Simulation 1 throughput diagram.

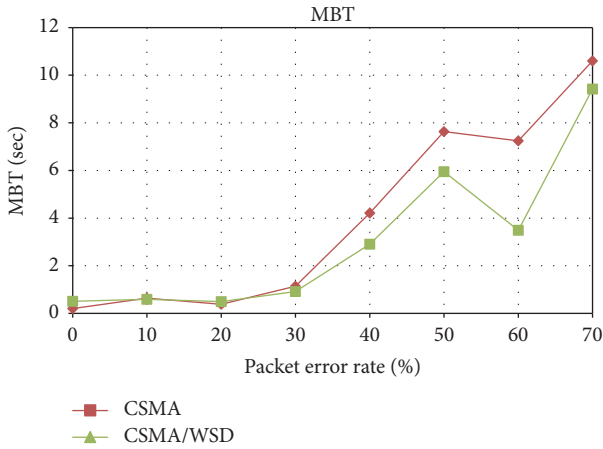


FIGURE 29: Simulation 1 mean backoff time.

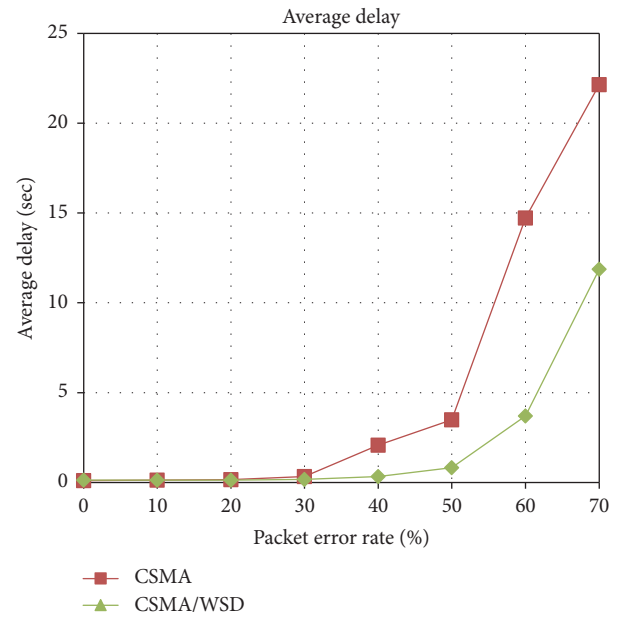


FIGURE 30: Simulation 1 average delay.

All presented values in diagrams are similarly calculated as in simulation 1.

The first result is the throughput which is shown in Figure 34. The throughput decreases with increasing distance because the signal becomes too weak to receive the packet. As a result, more packets are dropped which have to be retransmitted. This increases the whole transmission time and decreases the throughput. Until the distance of 76 meters, both protocols have nearly the same throughput. At a distance of 78 meters the packet loss by weak signal increases. The new MAC protocol determines these weak signals and uses neighbour-Acks to keep the whole transmission time small. The distance of 80 meters causes too many packet losses which cannot be resolved by a retransmission because both protocols become incapable of transmission.

The next result is the MBT which is shown in Figure 35. This increases with increasing distances because dropped packets have to be retransmitted. Similar as the throughput, the new protocol performs better and has a smaller MBT above 76 meters.

As the throughput decreases, the average delay (Figure 36) increases with higher distance. It is visible that the new protocol provides better results with a distance above 76 meters.

Similar to the previous simulation, the number of collisions (Figure 37) is higher between 72 and 80 meters. This is exactly that area where the new protocol provides more advantage over the original CSMA by using the WSD.

5.3.3. Summary of Simulations. Two different scenarios are simulated, and the results are reviewed. The first simulation proves both protocols' behaviour under different PER values. The packet transmission failures which are caused by PER value can be caused by different physical sources

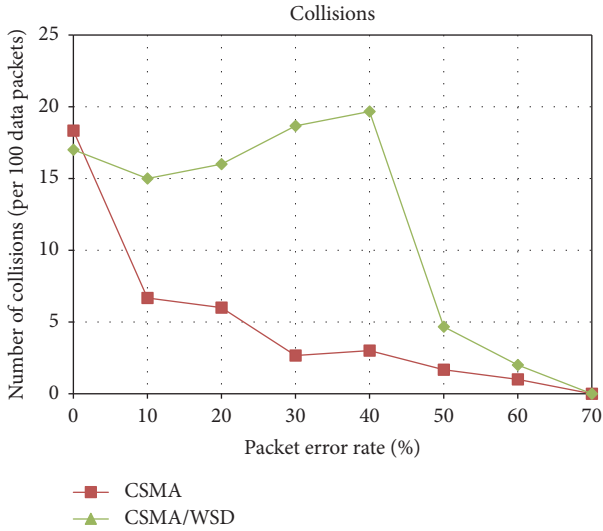


FIGURE 31: Simulation 1 number of collisions.

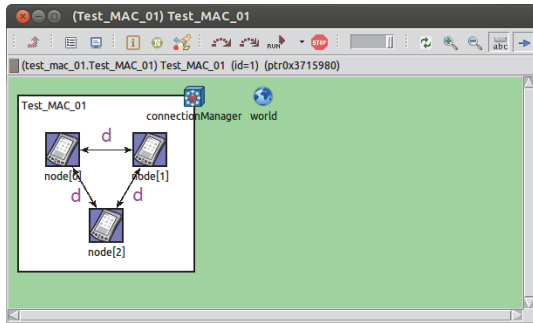


FIGURE 32: Simulation 2 scenario.

of disturbance. It is visible that the new protocol has more throughput stability with increasing PER values. Other values such as MBT and average delay are also better with higher PER values.

The second simulation considers the signal strength under increasing distance for both protocols. Here the thermal noise and signal attenuation are considered to create a possible real world scenario. The new protocol at distances over 76 meters shows better performance than the CSMA protocol with less mean backoff time and average delay. The number of collisions increases in both simulations but the new protocol shows better average result within 80 meters. The results show that this protocol has good performance and acts more stably in critical environments which suffer under signal variations and cause packet loss.

The energy consumption of the new protocol is not directly tracked. But the decrease in the whole transmission time and average delay implies that the nodes spend less time in active mode. This indicates indirectly less energy consumption and a high network lifetime. Some of the other observations are as follows:

- (i) Collision: this protocol causes more collision by usage of neighbour-Ack. The first simulations show that the mean backoff time decreases and causes less

transmission time. Only a retransmission causes an additional usage of the transmitter and therefore more energy consumption.

- (ii) Overhearing: the nodes record neighbour packets the whole time to provide a neighbour-Ack if a direct Ack was not transmitted. These packets cause extra energy consumption.
- (iii) Packet overhead: neighbour-Acks are additional control packets and contain no useful data; sending and receiving of these packets are the waste of energy.
- (iv) Idle listening: as this is the major source of energy waste in many WSN applications, it requires that the new protocol puts the nodes as soon as possible into sleep mode. If two nodes have a packet exchange, a third node listening in CSMA/WSN can help. To keep this node in listening mode is indirect idle listening and an additional waste of energy.
- (v) Invalid sending: if a node switches too early into the sleep mode and does not receive the last transmission, the result of the loss diagnosis in CSMA/WSD is a weak signal. A neighbour node will trigger the transmitting node to retransmit. It is more likely that all these retransmissions will not be received by the sleeping node. Using a BEB algorithm will increase the probability that, after a longer period, the sleeping node will switch to active mode and will receive one of these retransmissions.

This protocol can be implemented as a base to other proposed MAC protocols which reduces some of the sources of energy wastage. For example, using the CSMA/WSD protocol as a base for WiseMAC offers more throughput stability by using neighbour-Ack and decreases idle listening by usage active/sleep periods of the WiseMAC protocol.

6. Conclusion

The aim of this work was to create a new MAC protocol in WSN which will overcome the disadvantages of existing protocols. Therefore, different proposed MAC protocols were studied, and their advantages and disadvantages were summarised. The majority of these protocols are based on the CSMA protocol. The improvement of this protocol could improve all other protocols. Therefore, CSMA was the basis of the new MAC protocol. The newly proposed MAC protocol called CSMA/WSD is a contention-based protocol which should allow more throughput by performing a loss diagnosis.

The implementation of this protocol required an additional study of different simulation tools. OMNeT++ is eventually chosen and using the MiXiM simulation framework, it provides the necessary support and a module library for WSN simulations. The CSMA/WSD protocol was implemented by extending MiXiM's basic CSMA protocol. The finite-state machine (FSM) in the CSMA protocol was extended and proved with different function test scenarios.

The simulations show that the CSMA/WSD protocol provides more throughput stability under increasing packet

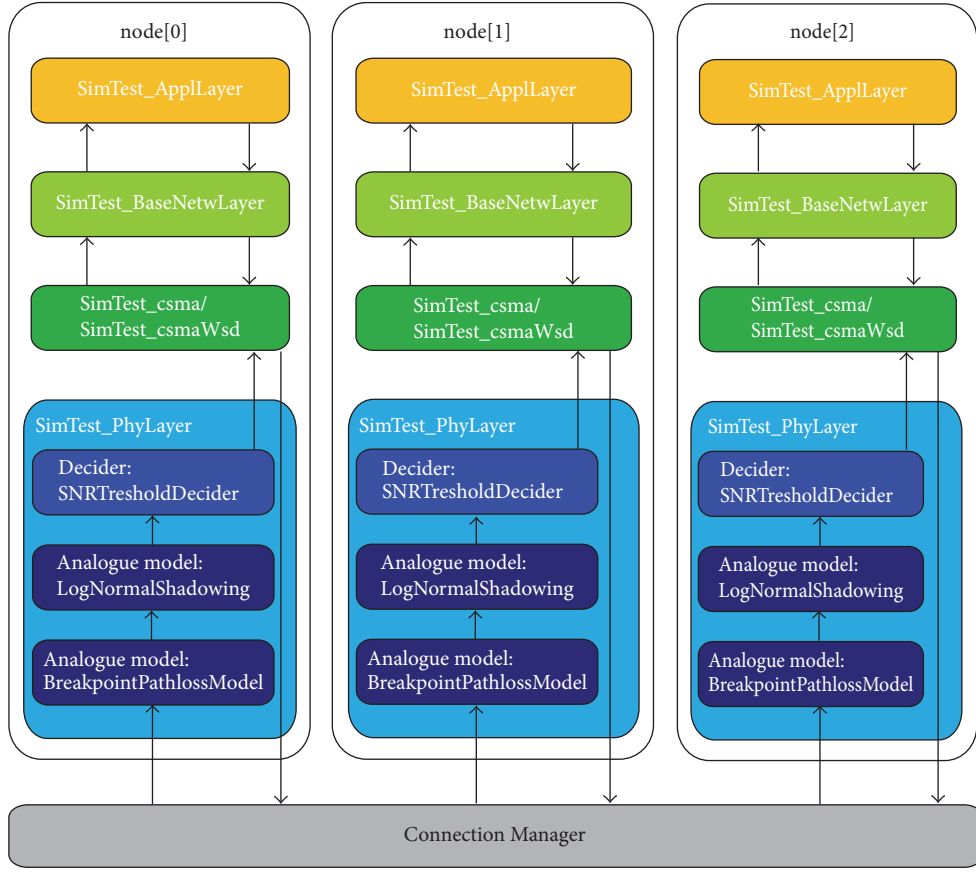


FIGURE 33: Simulation 2 test framework.

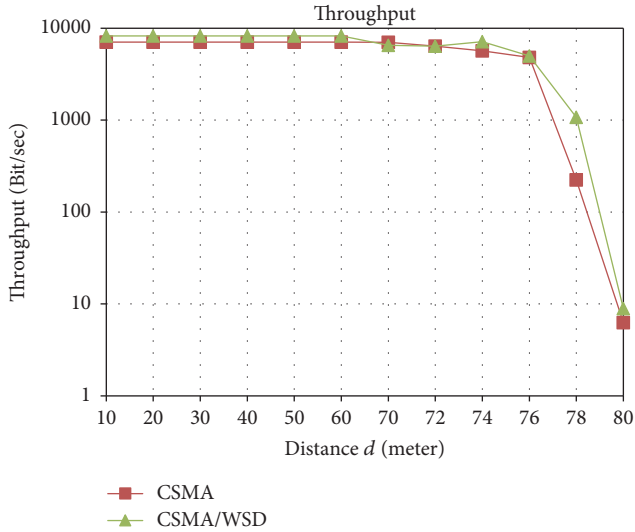


FIGURE 34: Simulation 2: throughput diagram.

error rate. The throughput under increasing distance between nodes is also better. Additionally, the average delay and mean backoff are lower as well. The usage of this protocol in critical environments where external circumstances cause a weak signal, noise, or packet loss is preferred. Furthermore, this protocol can divide between collisions from direct nodes

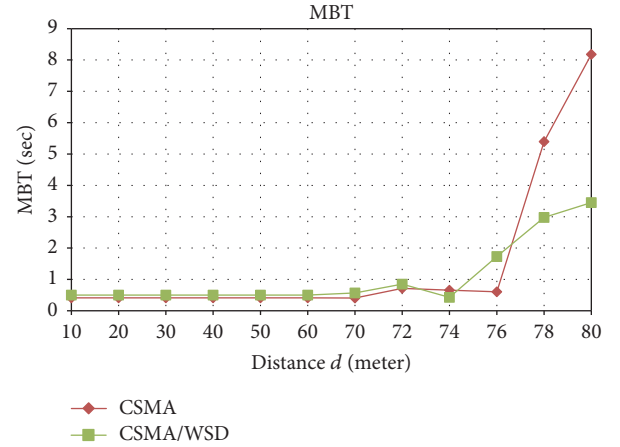


FIGURE 35: Simulation 2: mean backoff time.

and weak signals. This enables predicting the behaviour of the network more and usage in deterministic critical areas. The extension of other CSMA-based protocols such as WiseMAC would extend its capability and enable having both more throughput stability and deterministic behaviour. The development of this new CSMA/WSD protocol is the first step to upgrading the basic CSMA MAC protocol for usage in critical environments.

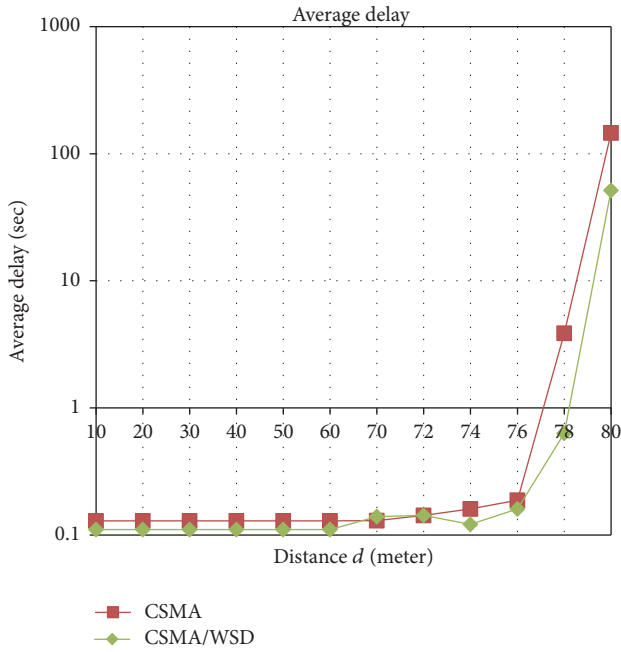


FIGURE 36: Simulation 2: average delay.

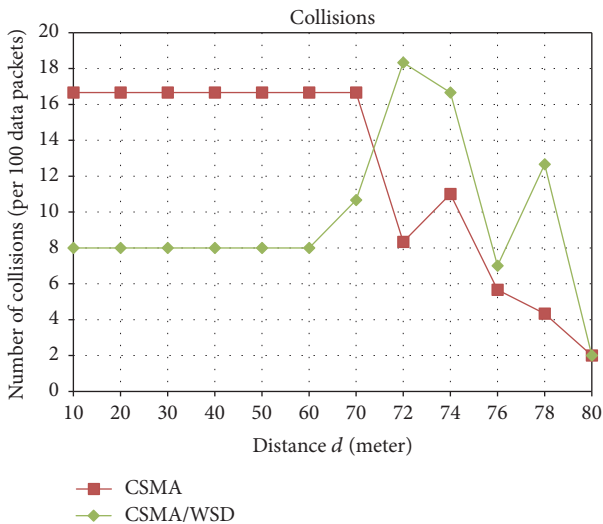


FIGURE 37: Simulation 2: number of collisions.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution," *Computer*, vol. 37, no. 8, pp. 62–70, 2004.
- [2] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile Ad Hoc and wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1448–1459, 2014.
- [3] A. A. A. Alkhatib and G. S. Baicher, "MAC layer overview for wireless sensor networks," in *Proceedings of the International Conference on Computer Networks and Communication Systems (CNCs '12)*, Kuala Lumpur, Malaysia, April 2012.
- [4] F. Yu, *A Survey of Wireless Sensor Network Simulation Tools*, 2011, <http://www.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html>.
- [5] A. Rajandekar and B. Sikdar, "A survey of MAC layer issues and protocols for machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175–186, 2015.
- [6] R. Yadav, S. Varma, and N. Malaviya, "A survey of MAC protocols for wireless sensor networks," *UbiCC Journal*, vol. 4, no. 3, pp. 827–833, 2009.
- [7] V. Cionca, T. Newe, and V. Dădărlat, "TDMA protocol requirements for wireless sensor networks," in *2nd International Conference on Sensor Technologies and Applications, SENSORCOMM 2008*, pp. 30–35, fra, August 2008.
- [8] T. Bernard and H. Fouchal, "A low energy consumption MAC protocol for WSN," in *Proceedings of the IEEE International Conference on Communications (ICC'12)*, pp. 533–537, Ottawa, Canada, June 2012.
- [9] C. Li, Y. Xu, C. Xu, Z. An, B. Diao, and X. Li, "DTMAC: a delay-tolerant MAC protocol for underwater wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4137–4146, 2016.
- [10] C. Buratti and R. Verdone, "L-CSMA: a MAC protocol for multihop linear wireless (sensor) networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 251–265, 2016.
- [11] T. Zheng, M. Gidlund, and J. Åkerberg, "WirArb: a new MAC protocol for time critical industrial wireless sensor network applications," *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2127–2139, 2016.
- [12] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: a priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 824–835, 2014.
- [13] F. Peng and M. Cui, "An energy-efficient mobility-supporting MAC protocol in wireless sensor networks," *Journal of Communications and Networks*, vol. 17, no. 2, pp. 203–209, 2015.
- [14] P. Huang, C. Wang, and L. Xiao, "RC-MAC: a receiver-centric MAC protocol for event-driven wireless sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 1149–1161, 2015.
- [15] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567–1576, IEEE, 2002.
- [16] H. Feng, L. Ma, and S. Leng, "A low overhead wireless sensor networks MAC protocol," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICET '10)*, pp. V4-128–V4-131, Shandong, China, 2010.
- [17] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 171–180, Los Angeles, Calif, USA, November 2003.
- [18] I. Demirkol, C. Ersoy, and F. Alagöz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.
- [19] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, pp. 3091–3098, Santa Fe, NM, USA, April 2004.

- [20] A. Koepke, M. Swigulski, K. Wessel et al., "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops*, pp. 71:1–71:8, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Marseille, France, 2008.
- [21] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, no. 1, pp. 63–78, 2006.
- [22] G. Dai, C. Miao, K. Ying, K. Wang, and Q. Chen, "An energy efficient MAC protocol for linear WSNs," *Chinese Journal of Electronics*, vol. 24, no. 4, pp. 725–728, 2015.
- [23] H. R. Shaukat and F. Hashim, "MWSN modelling using OMNET++ simulator," in *Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 597–602, Langkawi, Malaysia, 2014.
- [24] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: separating collision from weak signal," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1409–1417, IEEE, Phoenix, Ariz, USA, April 2008.
- [25] D. Hamdan, O.-E. Aktouf, I. Parissis, A. Hijazi, M. Sarkis, and B. E. Hassan, "SMART service for fault diagnosis in wireless sensor networks," in *Proceedings of the 6th International Conference on Next Generation Mobile Applications, Services, and Technologies (NGMAST '12)*, pp. 211–216, Paris, France, September 2012.
- [26] Texas Instruments Inc, "CC2520 Development Kit User's Guide," 2007, <http://www.ti.com/lit/ug/swru138/swru138.pdf>.
- [27] M. Petrova, J. Riihijärvi, P. Mähönen, and S. Laella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 487–492, April 2006.
- [28] S. X. Xu, Q. Wang, and Z. Chen, "Attenuation model of antenna signal with barriers in wireless sensor network," in *Proceedings of the 1st National Conference for Engineering Sciences (FNES '12)*, pp. 2812–2815, November 2012.
- [29] Texas Instruments Inc, *CC2430 Data Sheet (Rev. 2.1) SWRS036F*, 2013, <http://www.ti.com/lit/ds/symlink/cc2430.pdf>.
- [30] D. B. Faria, *Modelling Signal Attenuation in IEEE 802.11 Wireless Lans*, vol. 1, Computer Science Department, Stanford University, 2005.
- [31] R. S. Wilson, "A Derivation of the Normal Distribution," 2013, <http://www.sonoma.edu/users/w/wilsonst/Papers/Normal/default.html>.
- [32] P. Moravek, D. Komosny, M. Simek, M. Jelinek, D. Girbau, and A. Lazaro, "Signal propagation and distance estimation in wireless sensor networks," in *Proceedings of the 33rd International Conference on Telecommunications and Signal Processing*, pp. 189–194, 2010.

