Scientific Research Publishing

Wireless Sensor Network, 2026, 18(1), 1-21
https://www.scirp.org/journal/wsn
ISSN Online: 1945-3086
ISSN Print: 1945-3078

SLOT-TIME Sensor MAC: A MAC Protocol for Energy Consumption in Wireless Sensor Networks

Mohamed Conte[1], Ibrahima Kalil Toure[1], Mohamed Dinah Bangoura[2], Renfa Li[3], Binko Mamady Toure[1]

[1]Computer Sciences Center, Gamal Abdel Nasser University of Conakry, Conakry, Republic of Guinea
[2]Department of Mathematics, Gamal Abdel Nasser University of Conakry, Conakry, Republic of Guinea
[3]College of Information Science and Engineering, Hunan University, Changsha, China

Email: mohamed.conte@uganc.edu.gn, ibrahima.k.toure@ci.edu.gn, mohamed.dinah@uganc.edu.gn, lirenfa@vip.sina.com, bmtoure@ci.edu.gn

How to cite this paper: Conte, M., Toure, I.K., Bangoura, M.D., Li, R. and Toure, B.M. (2026) SLOT-TIME Sensor MAC: A MAC Protocol for Energy Consumption in Wireless Sensor Networks. *Wireless Sensor Network*, **18**, 1-21.
https://doi.org/10.4236/wsn.2026.181001

Received: November 15, 2025
Accepted: January 27, 2026
Published: January 30, 2026

Abstract

This paper presents an analysis of energy consumption performance in wireless sensor networks (WSNs). One of the crucial problems for WSNs is designing an efficient medium access control (MAC) protocol to reduce energy waste when the wireless communication module of the sensor network is operating. The energy efficiency of the MAC protocol makes a strong impact on the lifetime of the sensor network. Energy efficiency remains one of the most critical challenges in Wireless Sensor Networks (WSNs) and Internet of Things (IoT) systems. Traditional MAC protocols such as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) suffer from high-energy consumption due to idle listening and collisions, while deterministic approaches like IEEE 802.15.4e TSCH provide reliability but lack adaptability. A novel hybrid MAC protocol that integrates Time Slotted Channel Hopping (TSCH), wake-up radio technology, and machine learning techniques to optimize energy consumption while maintaining high reliability and low latency. We propose a new algorithm called Slot-Time Sensor Network (SS-MAC) based on a hybrid MAC protocol that optimizes energy consumption. Simulation results demonstrate that the proposed approach reduces energy consumption by up to 75% compared to conventional protocols while improving packet delivery ratio and latency performance.

Keywords

Slot Time, Wireless Sensor Networks, Contention-Based MAC Protocol, Energy Consumption, TSCH, Wake-Up Radio, Machine Learning, Energy Efficiency

1. Introduction

The medium is a common feature for all nodes in WSNs; it requires a mechanism that manages access nodes to determine the right to issue each of them in the network. MAC protocols have been an active research area for more than 30 years [1], and research on medium access control in wireless sensor networks is very fertile.

Indeed, medium access protocols are very different depending on the transmission type. All nodes access the channel in competition; there is potential for collisions when more than two nodes access the channel simultaneously. There is a clear way to improve the MAC protocol management of communication time between nodes, which consumes the most energy. The rapid evolution of digital technologies has led to the emergence of the Internet of Things (IoT), a paradigm where billions of interconnected devices collect, exchange, and utilize data in real time [2]. Wireless Sensor Networks (WSNs) form the core infrastructure of these systems, enabling monitoring and control in diverse fields such as industry, healthcare, smart agriculture, and smart cities. These networks are composed of nodes constrained by energy, computing power, and memory. In this context, optimizing communications becomes a major challenge. The Medium Access Control (MAC) layer, responsible for managing access to the radio medium, plays a central role in the overall network performance. However, the requirements of modern IoT applications, particularly in terms of reliability, latency, and energy efficiency, exceed the capabilities of traditional MAC protocols [3]. This has led to the introduction of new standards and the exploration of innovative solutions. WSNs and IoT rely on devices with limited energy and computing power. The MAC layer plays a key role in optimizing communications by reducing collisions, latency, and energy consumption. MAC protocols can be classified as contention-based protocols (CSMA/CA), scheduled protocols (Time Division Multiple Access (TDMA), TSCH [4]), and hybrid protocols. The emergence of industrial and mission-critical applications has led to the evolution of standards toward more deterministic solutions. Several studies have argued that energy is conserved better under traditional protocols. Our work focuses on this paper, specifically on the contention-based medium access protocol in wireless sensor networks. We analyze specific MAC protocols for wireless sensor networks and propose a novel active-sleep algorithm, Slot-Time Sensor MAC (SS-MAC), to optimize energy consumption.

In wireless sensor networks, where the radio range is limited, there are some problems with using CSMA. These problems are named the hidden station and exposed station [5] [6].

Figure 1 illustrates: (a) C tries to transmit a frame to B when A transmits to B, a collision occurs. (b) B is a neighbor of A, and A is a neighbor of C, too. When C wants to send a data frame to D, collisions occur because A occupies the channel.

The rest of this paper is organized as follows. Section 2 is related work through contention-based MAC protocol; Section 3, we analyze and discuss related contention-based MAC protocols in wireless sensor networks; Section 4, we discuss the advantages in energy consumption; Section 5, we present the Slot-Time Sensor MAC protocol; Section 6, we show simulation experiments; finally, we conclude and present other subjects for future work.

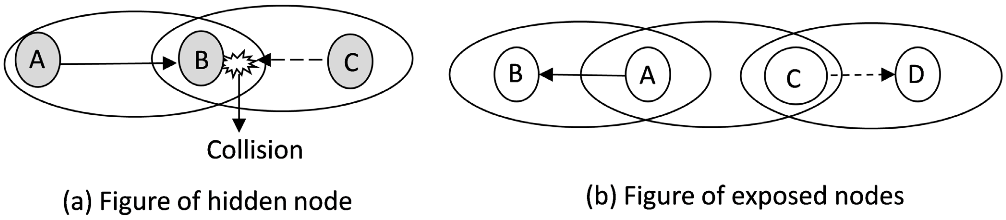

Figure 1. The hidden and exposed station problem.

2. Related Work

Contention-based protocol arose in the early wireless communication; in this protocol, given a transmit opportunity toward a receiver node, it can in principle be taken by any of its neighbors. If only one node tries its luck to send data, the packet goes through the channel, but if two nodes try to send the data through the channel, they have to compete with each other and unlucky cases; due to hidden-terminal situations, a collision can occur, which leads to wasted energy. The first contention-based MAC protocol is the Additive Links On-Line Hawaii Area (ALOHA). The main idea of the ALOHA protocol is that a node accesses the channel when it transmits data. If multiple nodes access at the same time, data frames collide and then cause retransmission. After a collision, nodes wait a random time before trying to transmit again. ALOHA provides short access and transmission delays; under light loads, the number of collisions increases, which decreases the throughput efficiency and increases transmission delay. Slotted ALOHA [7] divides time into time slots corresponding to data frames, and a node is allowed to start packet transmission only at the beginning of a slot. Nodes synchronize themselves, and they transmit data frames at the beginning of each slot. The collision occurs only when two nodes transmit simultaneously on the same slot. Slotted ALOHA generates better performance than conventional ALOHA. In short, the synchronization reduces the probability of collision.

The ALOHA protocol does not account for other transmissions, which increases the number of collisions in the network; to solve this problem, the proposed Carry Sense Multiple Access (CSMA) [8]. In this protocol, first, the node is required to listen to the medium. If the medium is idle, the node starts transmission. If the medium is busy, the node waits and tries to transmit again. However, if the channel is free, the node starts transmission in a time slot with some probability Pr [9].

An extension of CSMA, Busy Tone Multiple Access (BTMA) [10], was proposed to avoid the problem of a hidden station. The idea of BTMA is to use two communication channels: one, transmitting data frames, and the second to indicate a node is in receiver state. When a node receives data on the first channel, it sends a busy signal on the second channel to neighboring nodes. Thus, the fact of using two communication channels makes BTMA complicated in terms of hardware design. To extend BTMA Multiple Access Collision Avoidance (MACA) proposed [11]. In this protocol, nodes use a single communication channel. To avoid the hidden station problem, MACA uses the control packets Request to Send (RTS) and Clear to Send (CTS); its idea is that the node exchanges control packets before transmitting the data frames. Once a node receives CTS after sending RTS, it begins to transmit immediately without using carrier sensing. When the neighboring nodes receive the control packets, they know there is a transmission in the medium and do not try to access the channel immediately. The MACA protocol works well where there is no transmission error. But, in a wireless network, transmission errors are unavoidable because of noise or obstacles in the environment. MACAW (MACA for Wireless LANs) [12] adds the ACK (Acknowledgement) packets to the end of each frame of data sent to ensure the best transmission in the network. MACAW improves MACA by increasing the throughput of the network.

With the rapid development of different techniques for medium access of wireless LANs, the IEEE 802.11 standard was developed [13]. This standard provides two MAC protocols, Distributed Coordination Function (DCF) and Point Coordination Function (PCF). The DCF mode is a medium access protocol purely distributed and based on MACAW, adding listening to a virtual channel. This is an evaluation of the transmission time when a node receives packets RTS, CTS, ACK, and others. To reduce collisions caused by hidden and exposed nodes in the network, IEEE 802.11 uses a four-way RTS/CTS/DATA/ACK exchange. A node that wishes to send a data packet first sends an RTS after the DCF-Inter Frame Space (DIFS) period. If the network is idle, the destination responds with CTS after the Short Inter Frame Space period (SIFS). The duration of SIFS is shorter than DIFS. This gives priority to the current transmission. In the case of transmission failure or after successfully transmitting, the node adopts a mechanism of Binary Exponential Backoff (BEB) to select a random backoff. This random number is drawn within [0, cw], where cw is the contention window. The sender then transmits the data packet and waits for an Acknowledgment (ACK) from the receiver. If a node overhears RTS or CTS, it knows the medium is busy for some time and avoids new transmissions or sending any CTS packets. To avoid this type of collision, this protocol uses a virtual carrier sensing (VCS) mechanism with a network allocation vector (NAV).

In the network, a node wants to start transmission. The sender node checks its NAV; if the NAV equals zero, continue with RTS transmission. If NAV is not equal to zero, then the sender checks the link status of other links (a link is a pair of sender and receiver) in the network that are under transmission at that time. If the sender node's link status is not the same as the other node's link status, the sender continues with RTS transmission. If the link status is the same, the sender waits for NAV to become zero. As shown in **Figure 2**, the node that needs to transmit a packet issues an RTS frame. When the destination receives the RTS frame, it will transmit a CTS frame after the SIFT interval, immediately following the reception of the RTS frame. The source node is allowed to transmit its packet if and only if it receives the CTS correctly. At the same time, all the other nodes will update the NAV based on the RTS from the source node and CTS from the destination node, which helps to overcome the hidden terminal problem.

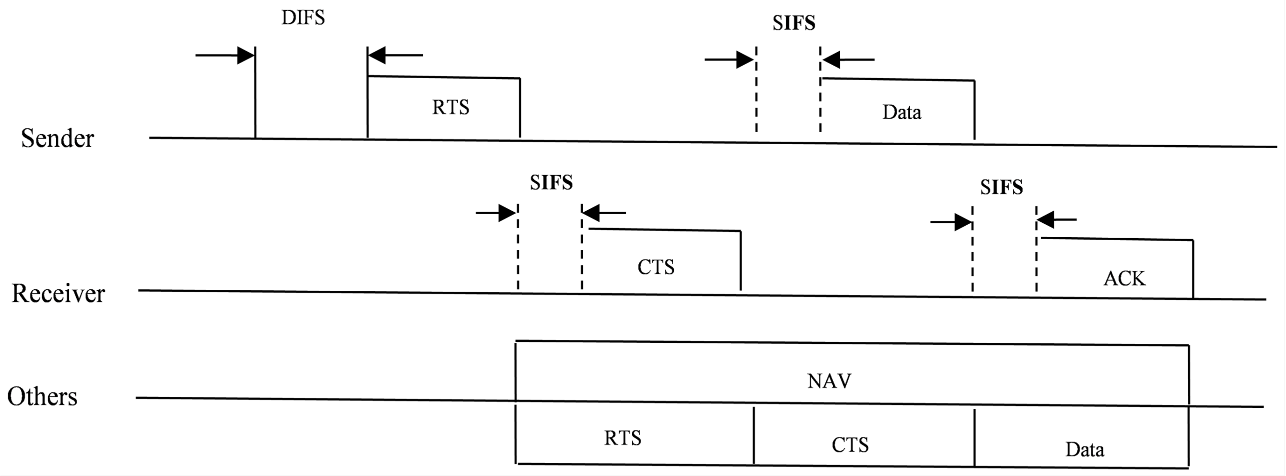

Figure 2. Structure of network allocation vector.

3. MAC Protocol Optimized Energy Consumption

In a wireless sensor network, the devices are very sensitive to energy consumption. We must find a way to minimize the level of energy consumption in order to have a longer lifespan. Before analyzing MAC protocols that optimize energy consumption, we first analyze the reasons for energy dissipation. Indeed, sensors consume their energy in two cases: when they communicate among themselves and when they process information locally. In these cases, the sensors consume more energy. Get to estimate the level of energy consumption. Limiting communication time is a solution to reduce energy consumption. Moreover, when sensors communicate, they are still wasting energy. We need to understand these reasons in order to reduce wastage. In [14] [15], some reasons for energy wastage were proposed: i) the collision case, when two nodes transmit data frames at the same time to one destination; this collision involves a retransmission of data frames and increases energy consumption, ii) the idle listening or (overhearing), where nodes listen to data frames that are not their destination, iii) the wireless channel is a noisy transmission medium, and there are often errors in transmission, then, using a control packet is an effective method to control the error frames. However, when a packet is empty, it consumes that wasted energy, and iv) the time when a node listens to the medium and waits for a possible transmission. When a node does not yet know when other nodes will send the data frame, it must always keep its transceiver on. Thus, if there is no transmission to him, it is a waste of energy.

To address these different modes of energy consumption, several solutions have been proposed as follows: Considering the first proposal in the contention-based MAC protocol that reduces energy consumption, S-MAC [16] uses three novel techniques to reduce energy consumption and support self-configuration. When listening on an idle channel, nodes periodically sleep. Neighboring nodes form virtual clusters to auto-synchronize on sleep schedules. Power Aware Multi Access protocol (PAMAS) was proposed [17] for ad-hoc networks. The protocol combines the busy tone solution and RTS/CTS handshake, similar to the MACA [18]. Indeed, inspired by MACA, using two communication channels, one for control packets and the other for data frames. After exchanging control packets, the nodes begin to communicate using the transmission channel. At the same time, it sends a busy signal on the second channel. When neighbor nodes receive a busy signal, they know there is a transmission and do not transmit their data frames. The solution of PAMAS also reduces collisions compared to MACA. In addition, when nodes receive a busy signal, they know that they cannot receive or transmit data frames; they will turn off their transceiver to reduce energy consumption. This is an advantage of PAMAS compared to MACA. The fact that using two communication channels limits its application in sensor networks, since most sensors use one communication channel for reasons of cost and design of sensors. The main purpose of this work was to reduce the time to listen to non-active sensors. Depending on the operation mode, these works are divided into two parts: the synchronized MAC protocols and preamble sampling MAC protocols. We will further analyze these two sub-families of the MAC protocol in the following section.

3.1. Synchronized MAC Protocols

Noting that, in a sensor network, fairness and latency are less important than energy consumption, the authors proposed S-MAC (Sensor-MAC) [19]. The main idea of S-MAC is to divide the transceiver operating time into two parts: active and sleep. When the transceiver turns active, the sensor is ready to transmit or receive data, whereas when the transceiver is put in a sleeping state, the sensor cannot transmit or receive data. As the purpose of S-MAC is to reduce energy consumption, it must extend the sleep time of the transceiver to reduce the energy consumption of sensors. Yet when the sensors are in sleep state or periodically active, there could be a problem, as the sensors operate independently; a sensor can be active while its neighbors are in sleep state. If the active sensor transmits data frames to a sensor that is asleep, this will cause the problem called "sending deaf" (deaf listening) because the asleep sensor cannot receive data. The solution of S-MAC is that nodes must synchronize with each other to get into the active state and sleep state at the same time. Synchronization is achieved through the SYNC packet. Nodes periodically send a SYNC packet to synchronize them and to know the time they have the right to transmit, and they must turn off to save energy. When a node participates in a network, it searches the network for a SYNC Packet if there is already a synchronization cycle. If this is the case, this node adapts its duty cycle to the existing cycle. Otherwise, this node creates its own cycle and broadcasts the SYNC packet to all neighboring nodes.

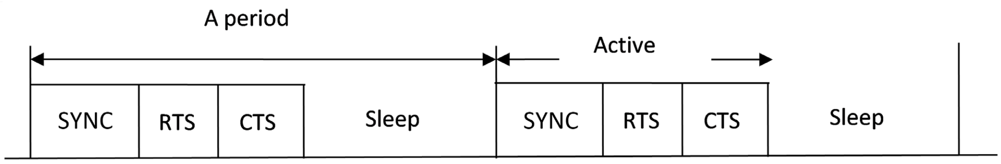

Figure 3. Periodically active and sleep in S-MAC.

Indeed, the active time interval is the time during which the nodes synchronize and transmit control packets. Figure 3 illustrates the composition of these intervals in time. Nodes that have data to transmit or will receive exchange control packets, RTS/CTS. Once these control packets are exchanged, the transmitter and receiver remain active throughout this schedule. Nodes that do not have data to transmit or receive turn off their transceivers at the scheduled end.

This method allows the sensors to avoid the problem of no idle listening, but it increases latency because the nodes cannot transmit at the beginning of each period. After receiving the data, the receiver cannot forward data frames immediately, as the next node in the state is already asleep. It must wait until the beginning of the next period to access the channel. An adaptive listening approach is proposed to reduce this latency. When a node listens to an RTS/CTS, if the transmitter/receiver is active, it will activate its transceiver. Otherwise, it goes to sleep only during the scheduled transmission between transmitter and receiver, and then it wakes up just at the end of this transmission to see if it is the receiver of a future transmitter. However, these extensions are still limited to two jumps because the nodes cannot report the remote nodes that need to be active for the entire multi-hop transmission. However, with some sensor network applications, this latency is acceptable. S-MAC saves energy, and thus a sensor network can extend its life for a few years. A major drawback of S-MAC is that the proportion of active and sleep time is fixed, which prevents the sensors from adapting to different traffic levels.

To overcome the drawbacks of S-MAC, T. van Dam and K. Langendoen proposed T-MAC (Timeout-MAC) [14] [15]. In T-MAC, the nodes synchronize to bring their sleeping and active transceivers periodically as S-MAC. However, T-MAC adjusts the time interval active at different levels of network traffic. This time interval is no longer fixed by the application but varies with network traffic. If traffic is heavy, the sensors remain active longer in order to transmit more data. On the contrary, if the traffic is low, the sensors remain active for the shortest time to save energy. Indeed, the sensors decide to go to sleep when they are not active in the network for a time called TA (Active Time). These activities involve the receipt of a frame or collision, and TA is a threshold long enough to ensure that there is no activity in the network.

3.2. Optimizing MAC Protocols

Energy efficiency remains one of the most critical challenges in Wireless Sensor Networks (WSNs) and Internet of Things (IoT) systems. Traditional MAC protocols, such as CSMA/CA (Flexible but energy-efficient), suffer from high-energy consumption due to idle listening and collisions, while deterministic approaches like IEEE 802.15.4e TSCH (reliable but rigid) provide reliability but lack adaptability. A novel hybrid MAC protocol that integrates Time Slotted Channel Hopping (TSCH), wake-up radio technology [20], and machine learning techniques (based on adaptive duty-cycling) [21] to optimize energy consumption while maintaining high reliability and low latency.

4. Proposed Slot Time Sensor MAC Protocol (SS-MAC)

SS-MAC protocol uses active and sleep time-based S-MAC; during the pick-out period, the node sender that wants to send a data frame chooses its slot time using RTS/CTS with the node receiver, and data transmission does not occur directly after CTS. However, during the transmission period, each node chooses a slot time to transmit a data frame, allowing it to sleep throughout the transmission in which it does not participate. Thus, we will avoid the frequent transition between different modes. In the pick-out period, we use the CSMA access method. This choice is motivated by the fact that computations are easier with this variant. CSMA with exponential backoff (as in the IEEE 820.11 standard) can provide lower delays when the channel is highly utilized. Nodes wishing to transmit must reserve slot time using RTS/CTS. Other nodes in the network register their status in each slot according to the activity in this period. If the node is receiving an RTS, then it records that it is the receiver for the slot reserved by the sender.

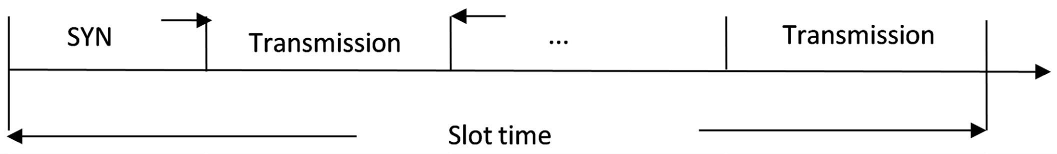

Figure 4. Contents of slot time.

The sender and receiver's neighboring nodes record their sleep state in the slot designated in the RTS or CTS. Thus, at the end of the pick-out period, each node will have its status in each slot during a transmission period. In that period, nodes use TDMA to access the channel, as shown in **Figure 4**. The sender transmits its data in the slot it reserved, and the receiver remains awake during the slot it is receiving until it receives the data and answers with an ACK. Other nodes that are neither sender nor receiver sleep. In this schema, each node sleeps for some time, and then wakes up and listens to see if any other node wants to talk to it [15]. During sleep, the node turns off its radio and sets a timer to wake itself later.

We consider a multi-hop wireless sensor network, in which nodes communicate with their neighbors at one hop, and data can be relayed over several hops to reach its destination. The nodes are assumed to be mostly static, which is consistent with environmental monitoring applications, although minor positional variations could be introduced in some simulations to test the protocol's robustness.

To improve energy efficiency and reduce contention, the network is structured into clusters formed in a distributed manner based on radio proximity (neighborhood) or simple criteria such as identifier or residual energy. Each cluster elects a cluster head. The cluster head plays a central role in the network's operation: it coordinates access to the medium, for example, allocating TDMA slots within the cluster, collecting data transmitted by member nodes, and potentially aggregating it to reduce traffic volume. It can also serve as an inter-group relay, thus participating in the multi-hop transmission of data to the base station. This model makes it possible to reconcile distributed operation with an efficient hierarchical organization, adapted to the energy and scalability constraints of sensor networks.

4.1. Performance Evaluation

We now formalize our intuition about the shape of the SIFS backoff probability distribution. Suppose all nodes pick a slot $r \in [1, cw]$ using a probability Pr. We say slot r is silent if no node chooses that slot, and there is no collision if more than one node chooses that slot. A node also wins in slot r if it is the only one to choose slot r, and all others choose later slots. Pr is calculated from a truncated increasing geometric distribution.

$$\mathrm{Pr} = \frac{(1-\alpha)\alpha^{cw}}{1-\alpha^{cw}} \times \alpha^{r} \quad \text{for} \quad r = 1, \cdots, cw \tag{1}$$

where α $(0 < \alpha < 1)$ is a distribution parameter which can be interpreted as the expected fraction of the nodes to be successful in transmitting their data in a given cw (Contention Window). In addition, cw is the contention window in this range of α, Pr increases with r, so the later slots have a higher probability, as shown in **Figure 5**.

Let N_i be the number of nodes willing to transmit in th stage. Assume that at the initial stage, $N_{i=1} = N$ nodes are willing to transmit. The main idea of the proposed protocol is to choose X_i, a size of cw at th stage, that would allow α fraction of nodes to transmit without collision. In the protocol, we assume that the cluster head (CH) knows the number of nodes in its cluster before starting the data collection process. This number can be determined when clusters are formed after the network deployment [17]. CH initiates the date collection cycle with a beacon containing the value of N. The node calculates the corresponding cw size with the help of this N and α. All nodes then start their first data transmission attempt with the cw of the size X_1. Being within the single hop, each contending node can detect the collision taking place in the cluster. The number of ACK packets transmitted by CH represents successful transmissions, and thus, nodes can determine the number of collided nodes that will be contending in the next stage. Since in each stage, α fraction of the nodes are allowed to transmit, it may take a large number of stages to transfer data from all nodes successfully. This situation can be prevented by using a fixed-size window when the number of contending nodes falls below a threshold.

To implement the protocol, we need to devise a formula to calculate X_i for any given values of α and N_i. Let N_i be contending in a *cw* of size X_i. The probability of choosing a slot by an individual node is $1/X_i$. Therefore, the probability of k nodes, out of N_i contending nodes, choosing a particular slot is

$$P(k)=\binom{N_i}{k}\left(\frac{1}{X_i}\right)^k\left(1-\frac{1}{X_i}\right)^{N_i-k} \tag{2}$$

Here, $k=0, k=1, k>1$ represent empty slots, successful transmission, and collision, respectively. Therefore,

$$P_e=P(k=0)=\left(1-\frac{1}{X_i}\right)^{N_i} \tag{3}$$

$$P_s=P(k=1)=\left(\frac{1}{X_i}\right)^k\left(1-\frac{1}{X_i}\right)^{N_i-1} \tag{4}$$

$$P_c=P(k>1)=\left(1-P_s-P_e\right) \tag{5}$$

where P_s, P_e, and P_c are respectively the probabilities when only one node chooses this slot with success, and the probability that no node chooses this slot and then more than one node chooses this slot.

Now, the expected number of nodes L_i that will be successful in transmitting their packets is

$$L_i=X_i\times P_s=N_i\left(1-\frac{1}{X_i}\right)^{N_i-1} \tag{6}$$

For our proposed protocol, we choose a window size X_i such that α_{N_i} node will be able to transmit successfully. Therefore, we have,

$$\alpha_{N_i}=N_i\left(1-\frac{1}{X_i}\right)^{N_i-1} \tag{7}$$

$$X_i=\frac{1}{1-\alpha^{\frac{1}{N_i-1}}} \tag{8}$$

For our proposed scheme, we choose an initial *cw* of 18, which corresponds to $\alpha = 0.67$. In this case, we also have one collision. Since it is a larger window, the collision probability has further decreased. In the second stage, the window is much smaller than the initial *cw*. This ensures a reduction in unnecessary free slots in the later stage without increasing collision probability.

4.2. Energy Consumption

A primary consumption of energy in a Wireless network is time spent for listening, but not receiving packets (idle energy). Span and S-MAC address this issue and are compatible with CSMA or SIFT simply by modulating the uniform backoff distribution of those protocols. Packets received by an overhearing node are another energy drain. Again, no changes to the above power saving protocols are required to take advantage of our novel backoff distributions. Finally, collisions have been shown to be costly in terms of energy, so by minimizing the number of collisions, our distribution reduces the energy consumption of any CSMA-based MAC even further.

Choosing a hybrid design that combines CSMA for the reservation phase and TDMA for data transmission allows us to leverage the complementary advantages of both approaches. CSMA offers high flexibility and low coordination costs, making it particularly well-suited to the dynamic phases of initial medium access and resource negotiation. Contention protocols such as CSMA are efficient for variable traffic but suffer from collisions and energy overhead as the load increases. Conversely, using TDMA slots for transmission eliminates collisions and ensures deterministic channel access, significantly improving energy efficiency by reducing retransmissions and unnecessary listening. The authors emphasize that TDMA approaches are particularly well-suited to energy-constrained sensor networks, as they allow nodes to enter sleep mode outside of their designated slots. This hybrid approach is preferable to using S-MAC alone, which relies primarily on fixed listening and sleeping cycles combined with contention access. While S-MAC reduces passive listening, it introduces increased latency and remains susceptible to collisions under varying traffic. The hybrid design shows that purely contention-based protocols do not adapt effectively to traffic variations and lead to energy inefficiencies. Thus, the CSMA/TDMA combination reconciles adaptability (CSMA) with energy efficiency and reliability (TDMA), offering better overall performance than S-MAC in the scenario considered.

5. Simulation Result

Finding the value of a cw that ensures a smaller number of collisions while minimizing unnecessary slot time wastage depends on the right choice of a, which is clearly a function of N. As shown in **Figure 6**, SS-MAC outperforms CSMA in terms of wastage and throughput.

We have performed the simulation in Java and averaged the results over 1000 runs. As physical layer properties do not contribute to cw calculation, we have considered ideal channel conditions, such as BER, in any event-driven simulator, such as NS-2, OPNET, and so on. In our simulation, we have used the same values of the parameters as used in [15]. More specifically, in the simulation setting, we have assumed a data packet of size 1000 bits. The time needed to transmit the packet consists of RST, CTS, DATA packet, ACK, and their corresponding IFS waiting times. The performance metric of our protocol is the time wastage in one cycle of data collection. Time wasted can be calculated as the difference between the total time needed in a single cycle of data collection and the time spent in effective data transmission. In our protocol, we need to fix some size cw when the number of remaining nodes falls below some threshold. In our simulation, we used a fixed cw of size 10 when the number of nodes is less than 4, this corresponds to a probability of collision of 10^{-2}, which can be considered negligible for a small value of $N = 3$.

The relation for the use of BEB in IEEE 802.11 is that a cw of size power of 2 facilitates arithmetic computations at the nodes. More specifically, for a given N, the nodes calculate a from the coefficients, determine cw using Equation (8), and round it to the next power of 2. It can be noted that the size of cw has been fixed to $2^4 = 16$ when the contending nodes fall below 3. As shown in **Figure 7**, although the performance of this modified scheme is inferior to the proposed scheme, it outperforms SS-MAC significantly.

We now use the optimal distribution to compare the fundamental performance limits of our protocol, with respect to the latency of the first transmission, when N stations come backlogged simultaneously in a previously quiescent network. Consider first an arbitrary non-persistent CSMA protocol that picks one of k slots for transmission. Let T_{slot} and T_{packet} be the time duration for a slot and a packet transmission (including any necessary inter-frame spacing), respectively. Define the latency $L_{\text{SS-MAC}}(N)$ to be the respective delay for a successful transmission when there are N contenders.

If there is a collision, then the delay is at least T_{packet}, so

$$L_{\text{CSMA}}(N) \geq \left(1-\tau(N)\right)T_{\text{packet}}, \tag{9}$$

where j is the slot selection distribution used by the protocol. Since

$\tau(N) \leq f(N)$, we get

$$L_{\text{CSMA}}(N) \geq \left(1-f_k(N)\right)T_{\text{packet}} \tag{10}$$

This is the general lower bound, so it is weak (see **Figure 8**). One way to strengthen the lower bound of Equation (10) is to specify the distribution P. For the popular uniform distribution,

$$\tau(N) = N\frac{1}{k}\left(1-\frac{1}{k}\right)^{N-1} + N\frac{1}{k}\left(1-\frac{2}{k}\right)^{N-1} + \cdots + N\frac{1}{k}\left(1-\frac{K-1}{k}\right)^{N-1} < \frac{N}{k}\left(\mathrm{e}^{\frac{N}{k}} + \cdots + \mathrm{e}^{(k-1)\frac{N}{k}}\right) < \frac{N}{k}\frac{1}{\mathrm{e}^{\frac{N}{k}}-1} \tag{11}$$

So, inequality Equation (9) now gives a lower bound on latency for the uniform distribution

$$L_{\text{ss-MAC}}(N) \geq \left(1-\frac{N}{k\left(\mathrm{e}^{\frac{N}{K}}-1\right)}\right)T_{\text{packet}} \tag{12}$$

This lower bound is much stronger than Equation (10), as illustrated in **Figure 7**.

Simulation Configuration

The simulator used is a custom Java implementation. The simulated topology consists of 1-100 nodes (default: 20) distributed according to a random topology over an 800 × 600 area. The traffic model is random, with a packet generation rate set to 0.05 packets/second per node. The radio parameters include a transmission range of 150 m, a radio data rate of kb/s, a transmit power of mW, and a receive power of mW. The energy model considers an initial energy of 10,000 Joules per node, as well as the costs associated with the following states: transmit 1.5 J/ticks, receive 1.0 J/ticks, idle listening 0.5 J/ticks, and sleep 0.01 J/ticks. The synchronization cost is set to 20 J/ticks per synchronization cycle. The basic parameters used for the protocols are as follows: for S-MAC, the cycle period is 100 ticks, with a sleep/activity ratio of 50%, a contention window of 70 ms, and a synchronization interval of a specific value. For CSMA, the minimum and maximum contention window sizes, respectively, are 16 and 1024, with a backoff time of 2 ticks and a channel listening mechanism before transmission. For LMAC, the frame is divided into 32 slots, with each node selecting a single transmission slot, a slot duration of 15 ms, and a frame period of 480 ms.

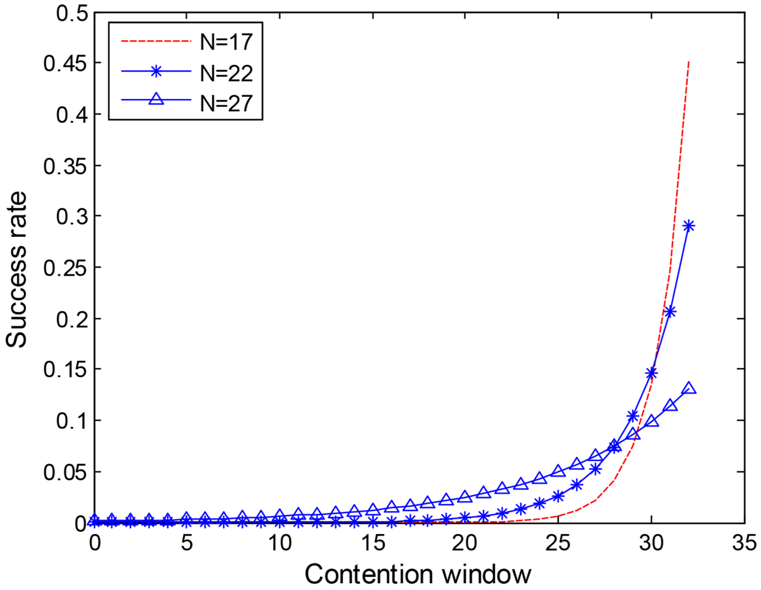

Figure 5. Illustrates the best value of the success rate follows the backoff probability distribution trend with the number of contenders.

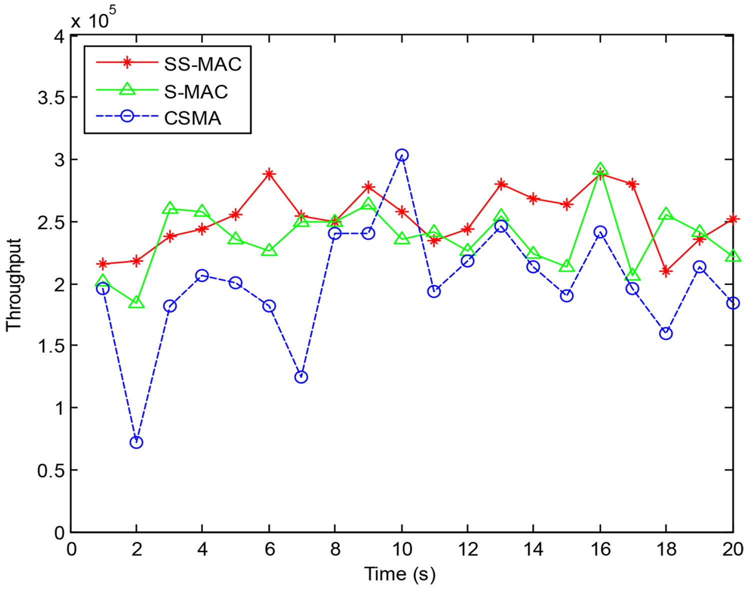

Figure 6. Illustrates the comparison of throughput between CSMA, S-MAC, and SS-MAC.

We used the NS-3 simulator to evaluate energy consumption optimization by comparing different MAC protocols. NS-3 allows us to measure the performance of each protocol and compare our SS-MAC protocol to four others, namely CSMA, S-MAC, L-MAC, and TSC-Hybrid. The metrics used were total energy consumption, average throughput, average latency, and packet delivery rate (PDR).

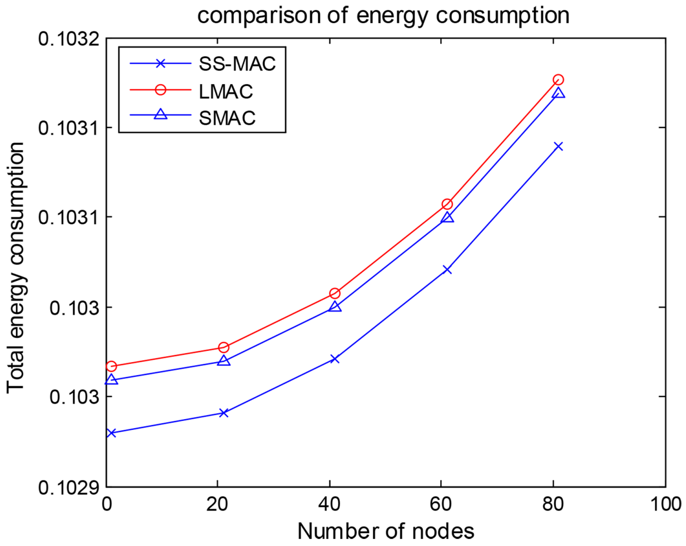

Figure 7. Illustrates the energy consumption of nodes over the slot time.

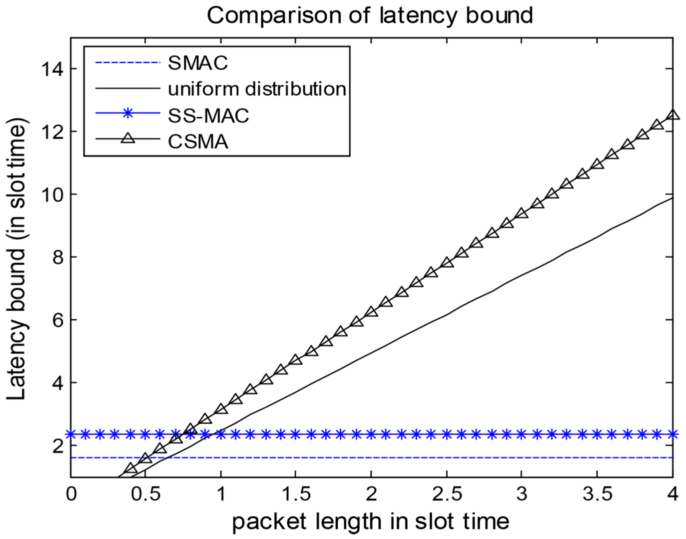

Figure 8. The graph compares the lower bounds in Equation (10) and Equation (12). The transmission latency of the first events. SS-MAC compared to LMAC and SMAC. With large node transmissions, SS-MAC performs better in comparison to LMAC and SMAC.

The performance of SS-MAC is compared to four reference configurations: CSMA, S-MAC, LMAC, and hybrid TSCH. The metrics used are total power consumption, average throughput, average latency, and packet delivery rate. To ensure a fair comparison, all five protocols are simulated with the same topology, number of nodes, radio model, packet generation rate, and channel conditions. The metrics used in Table 1 and Table 2 present the results as follows:

Table 1. Gain table SS-MAC vs CSMAC, S-MAC, LMAC, and TSCH-Hybrid.

Reference	Gain Energy SS-MAC	Gain Throughput SS-MAC	Gain Latency SS-MAC
vs CSMA	46%	30%	40%
vs S-MAC	20%	40%	55%
vs LMAC	21%	9%	18%
vs TSCH Hybrid	12%	−6%	−20%

Table 2. Table of average energy consumption, throughput, latency, and packet delivery rate.

Protocol	Average Energy Consumption (J)	Average Throughput (kb/s)	Average Latency (s)	Average PDR
CSMA	140	75	0.30	0.85
S-MAC	105	70	0.40	0.88
LMAC	95	90	0.22	0.92
TSCH Hybrid	85	105	0.15	0.96
SS-MAC	75	98	0.18	0.94

Figure 9 shows the throughput evolution as a function of the number of nodes for three MAC protocols: TSCH_hybrid, SS-MAC, and CSMAC. The results show that the throughput gradually decreases as the number of nodes increases, which is explained by the increase in channel contention, collisions, and transmission delays in the network. However, the TSCH_hybrid protocol maintains the best performance across all the scenarios studied. Its throughput increases from approximately 220 kbps for 10 nodes to 170 kbps for 50 nodes, demonstrating good adaptability to increased network load. Our protocol SS-MAC exhibits intermediate performance, with a throughput varying from approximately 200 kbps to 130 kbps as the number of nodes increases. In contrast, CSMAC shows the most significant performance degradation, with its throughput dropping from approximately 180 kbps to less than 70 kbps for 50 nodes. These results demonstrate that the TSCH hybrid approach allows for better management of radio resources and a reduction in collisions, thus ensuring higher throughput and better network scalability compared with conventional protocols. Figure 10 illustrates the evolution of energy consumption as a function of the number of nodes for the SS-MAC, TSCH_hybrid, and CSMAC protocols. The results indicate that energy consumption increases with network density for all three protocols studied. This increase is due to the multiplication of control exchanges, data transmissions, and channel listening periods. The SS-MAC protocol exhibits the lowest energy consumption, increasing from approximately 60 J to 85 J when the number of nodes increases from 10 to 50. The TSCH_hybrid protocol shows slightly higher consumption, between 65 J and 100 J, but maintains a relatively controlled energy level considering the gains achieved in terms of throughput. Conversely, CSMAC is the most energy-intensive protocol. Its consumption increases sharply, from approximately 80 J for 10 nodes to nearly 150 J for 50 nodes. This significant increase is mainly due to collisions, retransmissions, and prolonged periods of channel monitoring. The analysis thus highlights that TSCH_hybrid offers a good compromise between energy efficiency and network performance, while SS-MAC prioritizes energy savings, and CSMAC becomes less suitable for large networks due to its high energy consumption.

The simulation result (NS-3) comparing our SS-MAC protocol to CSMAC, TSCH-Hybrid are shown in Figures 9-11.

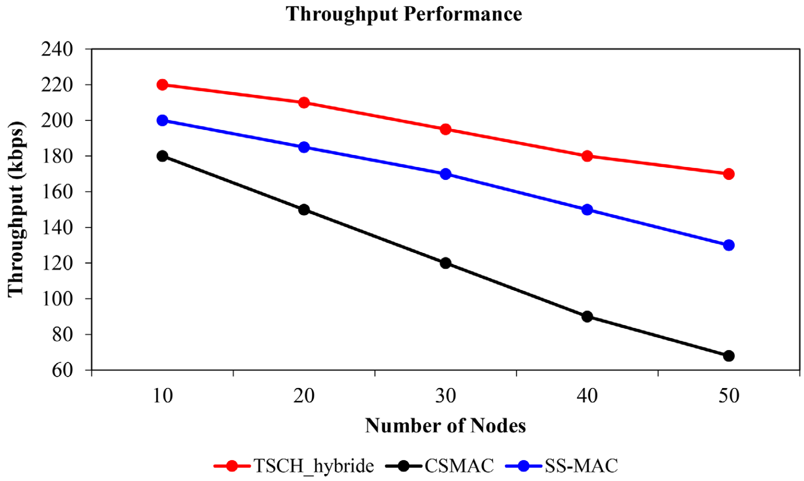

Figure 9. Illustrates the comparison of throughput between CSMAC, TSCH-Hybrid, and SS-MAC.

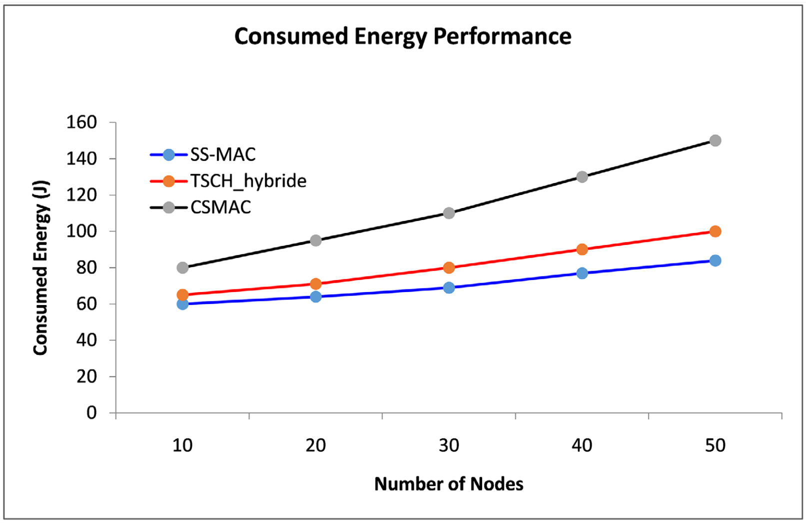

Figure 10. Illustrates the comparison of consumption energy between CSMAC, TSCH-Hybrid, and SS-MAC.

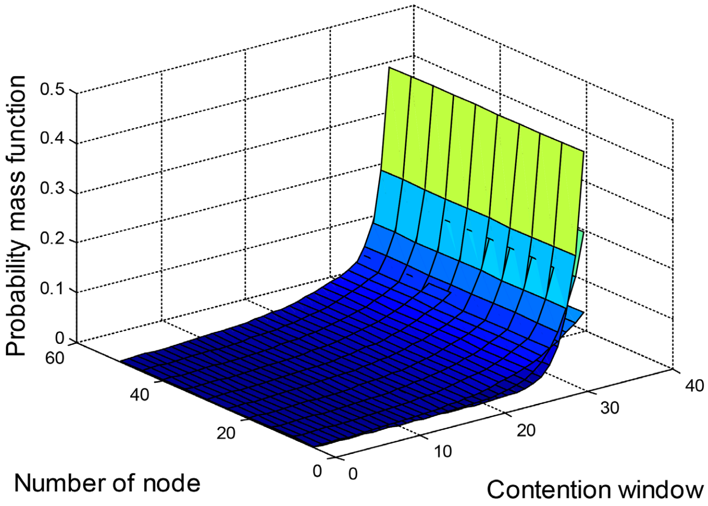

Figure 11. Represent a distribution probability, which allows contenders to choose a slot time with success on contention windows. According to Equation (1), we fixed the contention window size at 32; we also varied r from 0 to max cw and N values.

6. Conclusion

Instead of a workload consisting of a backoff, we have studied the MAC protocol when N nodes become simultaneously backlogged at some point in time. This workload is important at some point in time. This workload is important in the context of an event-driven and energy consumption in a wireless sensor network, when N sensors simultaneously sense an event of interest from the outside world. We have studied the backoff distribution probability for S-MAC using the CSMA protocol, in which each node chooses a contention slot with probability P. We have compared S-MAC and TSCH-Hybrid, our protocol SS-MAC, and, based on simulation results, our protocol optimizes energy consumption. Despite the observed gains, several limitations must be considered. First, the synchronization overhead introduces a significant energy and time cost, particularly in dynamic or low-density networks where resynchronization phases can become frequent. Second, the performance of Slot-Times-MAC (SS-MAC) is highly dependent on the accuracy of cluster size estimation: an incorrect estimate can lead to poor slot allocation, thereby reducing overall efficiency. Finally, under suboptimal channel conditions (interference, high losses) or irregular traffic, the protocol's performance can degrade, particularly in terms of latency and channel utilization. These factors suggest that additional adaptive mechanisms may be necessary to ensure the protocol's robustness in more realistic environments.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Abramson, N. (1985) Development of the ALOHANET. *IEEE Transactions on Infor-*

mation Theory, **31**, 119-123. https://doi.org/10.1109/tit.1985.1057021

[2] Naparstek, O. and Cohen, K. (2019) Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access. *IEEE Transactions on Wireless Communications*, **18**, 310-323. https://doi.org/10.1109/twc.2018.2879433

[3] De Guglielmo, D., Al Nahas, B., Duquennoy, S., Voigt, T. and Anastasi, G. (2017) Analysis and Experimental Evaluation of IEEE 802.15.4e TSCH CSMA-CA Algorithm. *IEEE Transactions on Vehicular Technology*, **66**, 1573-1588. https://doi.org/10.1109/tvt.2016.2553176

[4] Sahoo, P., Pattanaik, S. and Wu, S. (2017) A Novel IEEE 802.15.4e DSME MAC for Wireless Sensor Networks. *Sensors*, **17**, Article No. 168. https://doi.org/10.3390/s17010168

[5] Roberts, L.G. (1975) ALOHA Packet System with and without Slots and Capture. *ACM SIGCOMM Computer Communication Review*, **5**, 28-42. https://doi.org/10.1145/1024916.1024920

[6] Garcia-Luna-Aceves, J. (1997) Collision Avoidance and Resolution Multiple Access; First-Access Protocols. *Proceedings of ICC'97-Internal Conference on Communication*, Montréal, 8-12 June 1997, 699-703.

[7] Ma, R.T.B., Misra, V. and Rubenstein, D. (2009) An Analysis of Generalized Slotted-Aloha Protocols. *IEEE/ACM Transactions on Networking*, **17**, 936-949. https://doi.org/10.1109/tnet.2008.925633

[8] Kleinrock, L. and Tobagi, F. (1975) Packet Switching in Radio Channels: Part I—Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. *IEEE Transactions on Communications*, **23**, 1400-1416. https://doi.org/10.1109/tcom.1975.1092768

[9] Ye, W., Heidemann, J. and Estrin, D. (2004) Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, **12**, 493-506. https://doi.org/10.1109/tnet.2004.828953

[10] Tobagi, F.A. and Kleinrock, L. (1975) Packet Switching in Radio Channels: Part II—The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, **23**, 1417-1433. https://doi.org/10.1109/tcom.1975.1092767

[11] Haas, Z.J. and Deng, J. (2002) Dual Busy Tone Multiple Access (DBTMA)—A Multiple Access Control Scheme for Ad Hoc Networks. *IEEE Transactions on Communications*, **50**, 975-985. https://doi.org/10.1109/tcomm.2002.1010617

[12] Bharghavan, V., Demers, A., Shenker, S. and Zhang, L. (1994) MACAW: A Media Access Protocol for Wireless LANs. *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, London, 31 August-2 September 1994, 212-225. https://doi.org/10.1145/190314.190334

[13] IEEE 802.11 Working Group (1999) IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11. http://standards.ieee.org/getieee802/802.11.html

[14] van Dam, T. and Langendoen, K. (2003) An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Proceedings of the* 1*st International Conference on Embedded Networked Sensor Systems*, Los Angeles, 5-7 November 2003, 171-180. https://doi.org/10.1145/958491.958512

[15] Ye, W., Heidemann, J. and Estrin, D. (2002) An Energy-Efficient MAC Protocol for Wireless Sensor Networks. 21*st Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, 23-27 June 2002, 1567-1576.

[16] Bandyopadhyay, S. and Coyle, E.J. (2004) Minimizing Communication Costs in Hierarchically-Clustered Networks of Wireless Sensors. *Computer Networks*, **44**, 1-16. https://doi.org/10.1016/s1389-1286(03)00320-7

[17] Singh, S. and Raghavendra, C.S. (1998) PAMAS—Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. *ACM SIGCOMM Computer Communication Review*, **28**, 5-26. https://doi.org/10.1145/293927.293928

[18] Karthikena, A., Shankar, T., Srividhya, V. and Paravalika, P.V.L. (2012) Energy Efficient MAC Protocol for Energy Latency Tradeoff in Wireless Sensor Network Traffic. 2012 *IEEE International Conference on Computational Intelligence and Computing Research*, Coimbatore, 18-20 December 2012, 1-5. https://doi.org/10.1109/ICCIC.2012.6510240

[19] Halkes, G.P., van Dam, T. and Langendoen, K.G. (2005) Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks. *Mobile Networks and Applications*, **10**, 783-791. https://doi.org/10.1007/s11036-005-3371-x

[20] Rostami, S., Lagen, S., Costa, M., Valkama, M. and Dini, P. (2020) Wake-Up Radio Based Access in 5G under Delay Constraints: Modeling and Optimization. *IEEE Transactions on Communications*, **68**, 1044-1057. https://doi.org/10.1109/TCOMM.2019.2954389

[21] Sarang, S., Stojanović, G.M., Drieberg, M., Stankovski, S., Bingi, K. and Jeoti, V. (2023) Machine Learning Prediction Based Adaptive Duty Cycle MAC Protocol for Solar Energy Harvesting Wireless Sensor Networks. *IEEE Access*, **11**, 17536-17554. https://doi.org/10.1109/access.2023.3246108

Appendices

A.1. Pseudo Code

The pseudo code of our algorithm:

```
repeat
    if carrier sense clear and backlogged PickOut
        r Є [1, cw]
        wait r contention slots
        if carrier sense clear
            then transmit
        end if
    end if
end repeat
if transmission failure then
    if silence in the first K-1 slot then
        N ← N⁄2
        else N← N+1
    end if
end if.
```

A.2. Pseudo Code of Our Algorithm with Hybrid MAC/CSMA/RTS-CTS/TDMA and Sleep-Wake Scheduling

```
Initialization:
    cw ← cw_min
    N ← N_initial
    TDMA_Table ← empty
    Node_State ← SLEEP or ACTIVE according to schedule

repeat

    Phase 1: Wake-up and Channel Monitoring
    if wake_up_time reached then
        Node_State ← ACTIVE
        listen to channel during synchronization period
    end if

    if Node_State = ACTIVE and queue not empty then

        Phase 2: Channel Access through Contention
        if carrier sense clear then
            randomly choose r ∈ [1, cw]
            wait r contention slots

            if carrier sense clear then

                Phase 3: RTS/CTS Reservation
                send RTS to receiver
                wait for CTS during CTS_timeout

                if CTS received then

                    Phase 4: TDMA Slot Allocation
                    request or confirm TDMA slot
                    if TDMA slot available then
                        allocate TDMA_slot to node
                        broadcast or update TDMA_Table

                        Phase 5: Scheduled Data Transmission
                        wait until beginning of TDMA_slot
                        transmit data
                        wait for ACK

                        if ACK received then
                            transmission successful
                            cw ← cw_min
                            release or keep TDMA_slot depending on
traffic demand
                        else
                            transmission failed
                            cw ← min(2 × cw, cw_max)
                        end if

                    else
                        postpone transmission
                        cw ← min(2 × cw, cw_max)
                    end if
```

```
                else
                    reservation failed
                    cw ← min(2 × cw, cw_max)
                end if
            end if
        end if
    end if

    Phase 6: Sleep/Wake Scheduling
    if no packet to transmit
       and no upcoming allocated TDMA_slot
       and no expected reception then
            Node_State ← SLEEP
            sleep until next scheduled wake_up_time
    end if

until network termination or algorithm stop


Failure Handling:
if transmission failure then
    if silence detected in the first K-1 slots then
        N ← N / 2
    else
        N ← N + 1
    end if
end if
```