

HEEMAC+: An Enhanced Harvested Energy-Efficient MAC Protocol for WSNs

Abdelmalek Bengheni*

University of Tiaret, LRIAS Laboratory, Computer Science Department,
Algeria

abdelmalek.bengheni@univ-tiaret.dz

Abstract. Wireless Sensor Networks (WSNs) require high energy efficiency to ensure long-term operation. While enhancing WSN performance overall, especially latency, an increase in energy economy is also required. Giving up some performance indicators, including throughput, in favor of energy efficiency is a popular approach for protocol design. Thus, an asynchronous MAC protocol called an enhanced Harvested Energy-Efficient MAC (HEEMAC+) is proposed in this paper to conserve energy inside a WSN with energy harvesters. Through the use of receiver-initiated schemes, HEEMAC+ uses beacon messages to reduce duty-cycle by utilizing the energy threshold in addition to the remaining energy that each sensor node in the WSN determines how long it will sleep for next. Moreover, HEEMAC+ uses the switching method to ensure load balancing in WSN. This allows the nodes to maximize their energy efficiency. According to simulation results performed using OMNeT++/MiXiM, HEEMAC+ outperforms both EEMEHSN, DSR, and DSP schemes.

Keywords. Duty-cycle, energy efficiency, HEE-MAC+, energy harvesting, load balancing, wireless sensor networks.

1 Introduction

The energy efficiency of wireless sensor networks (WSNs) is an important consideration due to the nature of their applications and the devices component them which operating on limited batteries. The MAC layer provides efficiency and reliability in this sort of network by managing the shared medium administration, buffer access to channel access techniques, and error management. In addition, and through management of sensor node communications, it

also makes a major contribution to a longer network lifetime. Nonetheless, before MAC protocols can be improved in terms of throughput, latency, and fairness, they must first be energy efficient. MAC protocols, for example, must be adjusted to the situation of little or the absence of network traffic. Current studies have concentrated on optimizing node energy savings by implementing state-of-the-art conservation techniques such as "Duty-cycling," which is utilized in WSN to save energy by letting each node switch between an active and a sleeping mode.

The asynchronous and synchronous duty-cycle MAC protocol variants are the two that might offer this technology [1, 2]. The energy required for the sensors to operate for a lengthy period of time is a barrier, though. Consequently, it is essential to use energy harvesters as an energy backup for the current energy management systems [3, 4]. Energy harvesting (EH) [5] is a novel technology for WSNs that aims to extend their lifespan by converting the gathered energy into electrical energy that the sensor nodes can promptly use. Actually, despite the potential for future energy harvesting increases, a number of MAC protocols, including ERIMAC [6] and ODMAC [7], have been proposed that are based on the "duty-cycle" and the EH technique. However, the majority of these protocols do not rely on any technical estimation to adjust the duty cycle of sensor nodes accordingly.

Furthermore, sensor nodes in a WSN can switch between the states of "ENO-Max" and "ENO" based on how these protocols function [8]. Whereas in the second state, the node's energy consumption is always higher than its energy

harvest, in the first state it is always equal to the quantity gathered. Hence, reducing node duration and consequently its duty-cycle might occasionally be necessary in order to minimize average packet latency, as well as the energy consumption brought on by packet collisions and channel congestion in addition to the amount of retransmissions inside the WSN.

This paper proposes an asynchronous MAC protocol called HEEMAC+ (an enhanced Harvested Energy-Efficient MAC) to save energy within a WSN with energy harvesters. With the help of the multiple energy thresholds technique, which is based on each node's current remaining energy quantity—which may increase over time due to its ability to harvest energy or the assessment of the possibility that the energy remaining is important in order to achieve the originally indicated goal—the HEEMAC+ is able to dynamically control over each node's duty cycle by estimating the next duration of each node's sleep. On the other hand, HEEMAC+ also works to secure load balancing in the WSN by using the method of switching nodes within it.

This paper's remaining sections are arranged as follows: The connected works are shown in the next section. The proposed energy-efficient MAC protocol for WSN with energy harvesters (HEEMAC+) is explained in Section 3 in detail. Section 4, on the other hand, covers the content relating to simulation, including the parameters and tools. Next, in Section 5, the HEEMAC+ evaluation is discussed in further depth. The paper is concluded in Section 6.

2 Related Works

The MAC techniques are widely used in Energy-Harvesting Wireless sensor networks (EH-WSNs) and have better performance than other systems [9]. They include the following: the Automatic Duty Cycle Modification for Lowest Power feature of the Adaptive Duty Cycle Algorithm (ADCA)[10] to greatly exploit network traffic, Receiver-Initiated MAC protocol based on harvested energy (ERIMAC) [6], On Demand MAC Protocol (ODMAC) [7]. The two approaches of duty-cycle scheduling are based on the Remaining Energy (DSR) and the Potential increase in Remaining

Energy (DSP) [12], A QoS Aware Energy-Efficient MAC Protocol (QAEE-MAC) [11], Dynamic Duty Cycle Scheduling Built on the Prospective Energy Increase (EDDSP) and Dynamic Duty Cycle Scheduling Based on Current Residual Energy (EDDSR) [13], an Energy-Harvested Management System (EH2M) [14], an Approach to EH-WSNs Using Multiple Thresholds (MTE-EHWSN) [15], and An Enhanced Energy Management System in EH-WSNs (EEM-EHWSN) [4].

In these protocols, all senders receive a beacon signal from the receiver instructing them to distribute data packets, which starts the connection. They also alter duty-cycle parameters and other settings using the energy they have acquired. In order to conserve energy inside a WSN with energy harvesters, an asynchronous MAC protocol called Harvested Energy Efficient MAC (HEEMAC) is developed in [17].

For instance, ADCA is a state-of-the-art energy-saving technology where the duty-cycle is precisely adjusted in accordance with traffic statistics, utilizing an IEEE 802.15.4 scheme driven by a bespoke mode. With regard to the operating method chosen for the beacon-equipped star network within the WSN, the suggested ADCA approach has been adjusted in order to increase the system energy efficiency, particularly for highly changeable traffic situations that fluctuate/varied over time. Based on the duty-cycle idea, the ODMAC for EH-WSNs may allow single duty cycles of nodes with varying power visibility. As a result, all nodes can increase their energy consumption and thus their performance until the energy used equals or surpasses the energy received.

To maximize latency and energy efficiency, the ERIMAC employs receiver-initiated and packet sequencing. In addition, a queuing technique is employed to manage the activity of a sensor node based on the quantity of EH observed in the surrounding environment. The QAEE-MAC selects the sender node by using the data priority technique. If the channel is accessible, a receiving node wakes up and broadcasts a beacon frame containing the allocated sender's address. Which sender node the receiving node selects depends on the priority of the data that a sender node sends.

A receiver node's energy condition affects both its waking and sleeping periods. As a result, the receiver node consumes the least amount of energy possible. DSR and DSP are the two dynamic duty cycle scheduling techniques that power harvesting WSNs can employ to minimize sensor node sleep latency while preserving sensor node power balance. By using the first way, the duty-cycle is just changed based on the current residual energy level. On the other hand, the second scheme exploits the potential increase in residual energy to sharply lower the duty-cycle when needed. Exponential Decision-Making serves as the foundation for both EDDSR and EDDSP, which are based mostly on the usage of residual energy as the main component. The primary MAC exponential decision technique, or EDDSR, dynamically modifies the duty cycle of every sensor node based exclusively on available energy. The other method, known as EDDSP, forecasts how a sensor node's residual energy will increase over time using an exponential decision. Wherever possible, it may actively reduce downtime to optimize energy efficiency. In order to reduce the duty-cycle, EH2M aims to utilize the remaining energy that is available. Every sensor node may anticipate its sleep interval depending on energy left and dynamically adjust its duty cycle with the help of EH2M. Additionally, using the proposed system (EH2M), each sensor node in a WSN may balance its energy harvesting capacity and consumption.

The EEMEHWSN uses an energy threshold strategy by using residual energy in receiver-initiated communications to regulate intervals of sleep and activity in order to minimize the duty-cycle and maintain balance between the energy employed and the possibility for collecting the energy of each sensor node within WSN. The duty cycle of each sensor node has been greatly and dramatically reduced by MTE-EHWSN, increasing the longevity and performance of WSN. Through the use of MTE-EHWSN, every sensor node may control energy collection and consumption, maintaining balance and dynamically adjusting its duty cycle by more fully and drastically determining its sleep interval based on the quantity of remaining energy. Moreover, MTE-EHWSN allows WSN to operate efficiently even with restricted energy harvesting.

By sending out beacon messages, HEEMAC utilizes receiver-initiated schemes, which allow each sensor node in the WSN to determine how long it will sleep for next in order to minimize its duty-cycle by utilizing the energy threshold in addition to utilizing the remaining energy, which may increase over time. Moreover, WSN can function effectively in low energy harvesting settings thanks to HEEMAC. The simulation's results demonstrate that HEEMAC uses less energy and has lower latency than EH2M. In summary, this section concludes with our new suggested HEEMAC+ scheme, which is appropriate for WSN and an improvement over HEEMAC. It also reflects other earlier schemes that were offered for fixed networks and offers a novel approach to lowering the communication energy usage in EH-WSNs.

To maintain network load balancing across nodes, HEEMAC+ employs a switching system method in conjunction with a dynamic duty cycle scheduling technique to dispatch packets from the beacon frame.

This work makes two main contributions: first, our scheme uses the remaining energy level to predict the duration of the node's next sleep period more accurately. Secondly, it ensures load balancing in WSN by using the same switching strategy used by EEMEHWSN, in which the corresponding node controls its switching mode (in relaying mode or sensing mode) during the wake-up period and by determining different future sleep periods among nodes within the network.

3 HEEMAC+: An Enhanced Harvested Energy- Efficient MAC Protocol for WSNs

In this section, we describe the operation of HEEMAC+ that represents an improvement of HEEMAC. The main communication mechanism used in HEEMAC+, followed by an explanation of the suggested energy threshold strategy that each sensor node uses to minimize energy consumption and determine how long to sleep based on remaining energy and in the switching technique used to controls its changing mode (in relaying mode or sensing mode) in WSN.

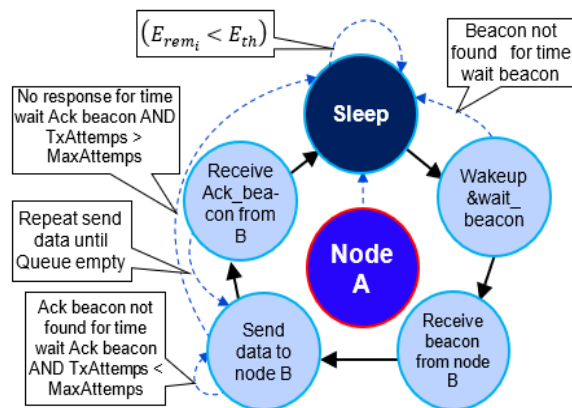


Fig. 1. A diagram of state transition for the transmitting and receiving processes of node A

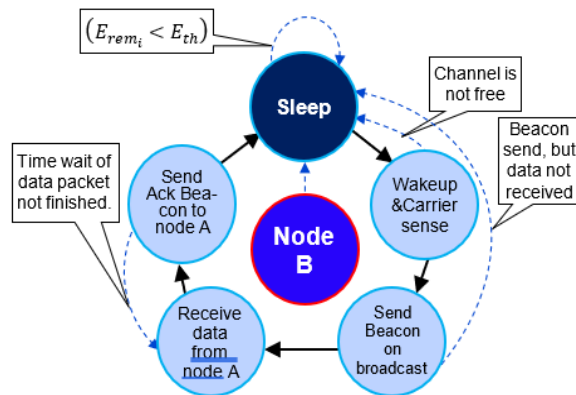


Fig. 2. A diagram of state transition for the transmitting and receiving processes of node B

This strategy optimizes duty-cycle and, as a result, minimizes operating time while maintaining balanced energy consumption across sensor nodes in WSNs that have the capacity to harvest energy.

3.1 Principal Communication Mechanism

HEEMAC+ usually employs the carrier sense multiple access / collision avoidance (CSMA/CA) protocol, in which a sender uses a carrier wave to keep the channel busy and a receiver listens to the wireless channel to avoid collisions, in order to minimize energy consumption via idle listening.

In the event that the channel is available, every receiver in the HEEMAC+ protocol periodically sends out a beacon to other receivers indicating that it is ready to accept incoming data packet transfers. All nodes that have packets waiting to be transmitted to the sink are keeping an eye on the channel, waiting for an appropriate beacon to appear. As soon as the beacon is detected, the data packet is transmitted. This technique completely removes the receiver node's idle listening. All that the receiver does is use energy to periodically emit a tiny beacon frame. On the other hand, the transmitter uses energy as it waits for a beacon. However, the energy lost at the transmitter is insignificant on low-traffic networks when transmitting queues are usually empty.

The main components of the HEEMAC+ communication process are shown in Figs. 1 and 2. Let us say that sensor node A wants to transmit a data packet to sensor node B. Listening to the channel, node A waits for a beacon when it wakes. Immediately upon the conclusion of its sleep phase, node B enters an active phase in which it endeavors to transmit a beacon broadcast. After a long period of listening, Node B first checks the channel. In the event that the channel is used for the full duration, Node B returns to sleep. If the channel is accessible, Node B transmits the broadcast beacon, starts a waiting timer, and waits for incoming data packets as in [6,16]. If no communications are received within the allocated time, Node B returns to sleep.

If Node A doesn't receive a beacon within the designated wait time, it will go back to sleep. If Node A does receive a beacon from B before transmitting a data packet, it will back off for a random number of slots of time determined from the interval $[0, CW - 1]$, where CW represents the contention window specified by the administrator.

If the channel is accessible and the back off has passed, node A begins transmitting a data packet to node B. Following an effective transmission termination, node A starts a waiting timer for the data packet acknowledgement beacon from node B. In the event that node B receives a data packet from node A, it will send an acknowledgement beacon to node A and start a waiting timer for any more data packets that node A or other transmitters may deliver.

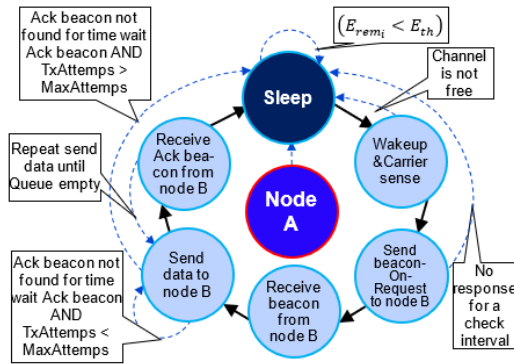


Fig. 3. A diagram of state transition for the transmitting and receiving processes of node A, where it first transmits a Beacon-On-Request

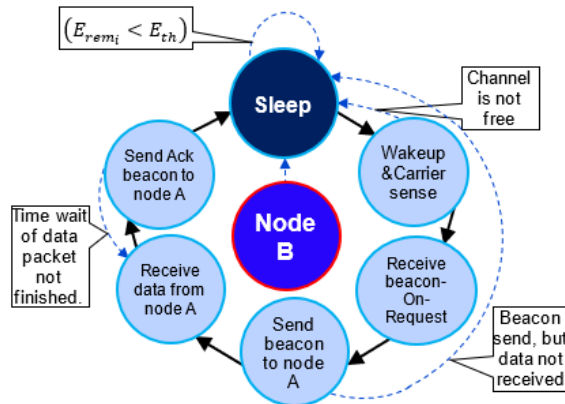


Fig. 4. A diagram of state transition for the node B receiving processes after receiving Beacon-On-Request

If no transmission is received from these nodes during the designated timeframe, Node B will go back to sleep; if a data packet is received within the designated wait time, it will transmit an acknowledgement beacon of the data packet to the appropriate node.

In order to prevent a collision, node A backs off for an arbitrary number of time slots determined from the interval $[0, BW - 1]$, where BW is the back-off window specified in the acknowledgement (Ack) beacon that is transmitted by node B as in [16].

If node A receives the acknowledgement (Ack) beacon from B and his queue is not empty, node

A will not transmit the data packet. TxAttempts is a counter used to determine the total number of transmissions completed, and MaxAttempts is the maximum number of transmission attempts per data packet by the transmitter in all figures.

Under HEEMAC+, if one of its receptor nodes is active, the transmitter node A may initiate communication when it awakens to send a data frame to the receiver B, as shown in [9]. As can be observed in Fig. 3, it uses CCA to check the channel before emitting a beacon known as "Beacon-on-Request". The sender A in this beacon specifies the recipient, B's Dst field address. As can be seen in Fig. 4, if B becomes active after a random duration longer than the BW specified in the beacon received from A, it generates a beacon in response. Thanks to this beacon that receiver B created at their request, sender A may deliver the awaited data frame immediately rather than waiting for the beacon from node B.

3.2 Energy Consumption Modeling

Assuming a network consisting of N nodes, we examine a particular kind of periodic traffic production in which nodes produce data at a predetermined frequency. All of the terms utilized in the energy model are shown in Table 1.

The total of the predicted energy consumption in each operation (carrier sensing (cs), environment sensing (es), Transmission(tx), Reception(rx)) is the estimated energy consumption in wake-up state for the time t by a node i in HEEMAC+ indicated by the following formulate:

$$E_{Consumed_i} = \sum_{j \in Operation} (T_j \times C_j) \times V. \quad (1)$$

Every wake-up interval, the transmitter node waits for a beacon from the receiver node and senses the surroundings to measure physical values, while the receiver node detects the carrier before transmitting a beacon. The sender node in the HEEMAC+ protocol may request to transmit a DATA frame by issuing a beacon after waking up and sensing the carrier:

$$T_{cs} \leq 2T_{acs}R_{data}, \quad (2)$$

$$T_{es} = T_{asense}R_{sense}. \quad (3)$$

Table 1. Terms used energy model

Term	Explanation
C_j (mA)	Current consumption of operation j
T_j	Time used by operation j
V	Voltage
R_{data}	Rate to RX/TX a data packet
R_{sense}	Rate to sense the environment
L_{beacon}	Beacon length
$L_{abeacon}$	Acknowledgement beacon length
$L_{beaconR}$	Beacon-on-Request length
L_{ack}	Acknowledgement length
L_{packet}	Data packet length
T_{cs}	Carrier sense time
T_{es}	Environment sense time
T_{acs}	Average carrier sense time
T_{asense}	Average time to sample data
$T_{(t/r)xb}$	time to Tx/Rx a byte
T_{sw}	Time for radio switching

The total amount of time needed to send a beacon, data packet, acknowledgement beacon, and W beacon-on-request, where $W \in \{0,1\}$, is the node's transmission time. An acknowledgement beacon is used to initiate the request for the next data transfer to this node as well as to acknowledge previously received data.

Consequently, if a transmitter possesses Npq data packets to transmit, it transmits a data packet and instantly switches its radio to listen status upon receiving each acknowledgement beacon; this procedure is continued until the transmitter's queue is empty. Eventually, with $(1 \leq Qp \leq Npq)$, the transmitter node can deliver Qp data packets to the identical recipient:

$$T_{tx} = R_{data}(L_{beacon} + QpL_{packet} + (Qp + W)L_{abeacon} + WL_{beaconR})T_{(t/r)xb} + (Qp + W + 1)T_{sw} \quad (4)$$

During the time where $(W \leq m \leq n)$ and $W \in \{0,1\}$, a node may receive a beacon, Qp data packets, m beacon-on-request, and $(Qp + W)$ acknowledgement beacons. But as soon as it receives a beacon request, it becomes a target

node for that particular beacon-on-request. It then transmits an acknowledgement beacon to the sender after acknowledging receipt of the packet:

$$T_{rx} = R_{data}(L_{beacon} + QpL_{packet} + (Qp + W)L_{abeacon} + mL_{beaconR})T_{(t/r)xb} + (Qp + W)T_{sw} \quad (5)$$

3.3 Remaining Energy Modeling

(E_{rem_i}) is the leftover energy for node i at instant t_c in HEEMAC+, and it is computed using the following equation:

$$E_{rem_i} = (E_{remp_i} - E_{consumed_i}) + E_{harvested_i} \quad (6)$$

In this case, E_{remp_i} represents the energy that node i had left over at the previous check instant t'_c , and $E_{harvested_i}$, $E_{consumed_i}$ justify the energy that was harvested and consumed by node i over the identical time period $(t_c - t'_c)$ respectively with $(t'_c < t_c)$.

3.4 Functioning Mechanism of the Recommended Energy Threshold Strategy in HEEMAC+

To accomplish the purpose of our HEEMAC+ protocol, we employed a type of wireless sensor node based on a "Harvesting-Storing-Spending" structure [5]. By inserting a fresh energy collection module, its energy is gathered as frequently as feasible and stored for use at a later time. This module's goal is to keep the WSN operational up to a hardware failure. Assuming that the WSN has n energy harvesting sensor nodes, each of which is designated by a number i , with at least one energy collector that transforms ambient energy into electric energy in order to power its component units where (E_{max}) is the maximum capacity of the collected energy storage device.

It also has a wireless transmitter for transmission, a memory for data storage, a CPU for processing, and a measuring micro-sensor. Additionally, three energy thresholds are used. The first, denoted by E_{th_s} , which each node will employ in the suggested switching strategy to change the way they operate (in relaying mode or sensing mode). The second, is E_{th} closer to E_{th_s} where $(E_{th} < E_{th_s})$, permits each node to maintain a balance between its capacity for energy

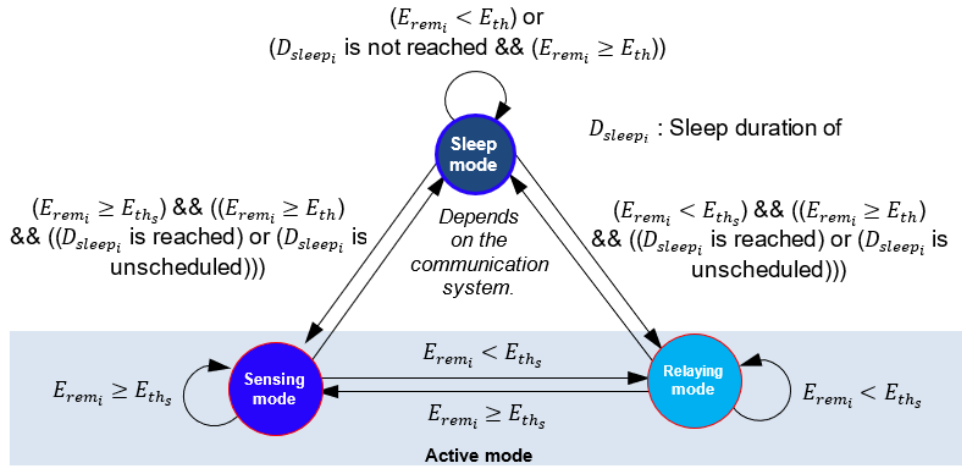


Fig. 5. A diagram of state transition for three possible modes of sensor node in HEEMAC+

gathering and its energy consumption, and the third, denoted by E_{cri} , enables each node to ensure that a certain power level is used during the sleep time where $(E_{th} > E_{cri})$.

3.4.1 Switching Strategy Used

As [4], this strategy's basic concept is as follows: when in the wake-up mode, every sensor node regularly determines how much energy is left in its battery. At the instant of check, the node will switch to relaying mode if this latter is less than the energy threshold E_{th_s} , and to sensing mode otherwise. During the lifetime of the WSN, the previously indicated stages are repeated as shown in Figure 5.

The definitions of these modes used in this strategy are as follows:

- *Mode of sensing*: The node gathers information (events) in this mode of sensing and relays it subsequently to the base station (sink node).
- *Mode of relaying*: The relay node exclusively binds the data of its neighboring nodes in this mode.
- *Mode of sleep*: The node radio interface is disabled in this mode.

3.4.2 The Suggested Method for Determining the Sleep Duration

The formulas listed below are used to calculate sleep duration of node i noted by (D_{sleep_i}) to lowers its duty-cycle depending on its remaining energy (E_{rem_i}) :

– After computing only D_{sdt_i} :

$$D_{sleep_i}(E_{rem_i}) = \left(\frac{E_{max} - E_{rem_i}}{E_{max} - E_{th}} \right) \times D_{sdt_i}. \quad (7)$$

– After computing D_{csdt_i} and D_{sdt_i}

$$D_{sleep_i}(E_{rem_i}) = \left(\frac{E_{max} - E_{rem_i}}{E_{max} - E_{th}} \right) \times D_{sdt_i} + \delta(E_{rem_i}) \times \theta. \quad (8)$$

with:

$$\delta(E_{rem_i}) = \frac{(E_{rem_i} - E_{max}) * (E_{rem_i} - E_{th})}{E_{th} - E_{cri}}, \quad (9)$$

$$\theta = \left(\frac{D_{csdt_i}}{E_{max} - E_{cri}} - \frac{D_{sdt_i}}{E_{max} - E_{th}} \right), \quad (10)$$

$$D_{sdt_i} = \min(D_{sdt_i}, 1), \quad (11)$$

$$D_{csdt_i} = \min(D_{csdt_i}, 1), \quad (12)$$

with:

D_{csdt_i} : The critical sleep duration required by node i before its remaining energy is equal to or greater than the critical energy threshold E_{cri} .

D_{sdt_i} : The duration sleep required by node i to before its remaining energy is equal to or greater than the energy threshold E_{th} .

E_{rem_i} : Remaining energy of node i .

$\delta(E_{rem_i}) * \theta$: The excess sleep time will be used in the next sleep duration computed by node i when it's crucial sleep duration (D_{csdt_i}) is computed.

HEEMAC+ uses two types of duty cycles for node activity, which are necessary to implement the suggested power threshold method and to assist them improve energy consumption, which are as follows: The first is stable and used only at the start, when the supervisor sets the active and sleep durations in full. The second is dynamic and employed for the remainder of the HEEMAC+ work it determines when the remaining energy is lower than E_{cri} or E_{th} based on the rate of energy gathering. Frequently, the nodes with a higher power reserve sleep for a shorter duration, wake up later, and receive and send better packets.

Furthermore, this technique reduces collisions caused by nearby nodes that awaken concurrently and vie for the channel. Node sleep durations are purposefully calculated using HEEMAC+ to ensure that they are not comparable to a group of nodes in terms of energy efficiency. As a result, in order to save as much energy as possible while maintaining a guaranteed power level, a node with limited remaining energy opts for a prolonged sleep period. This section describes how to estimate the sleep time (D_{sleep_i}) of a node i using HEEMAC+:

- A node i checks his remaining energy (E_{rem_i}) right before going to sleep:
 - If $((E_{rem_i} > E_{cri}) \text{ and } (E_{rem_i} < E_{th}))$, node i only calculates the sleep period (D_{sdt_i}) using equation (11) Up to the point that its remaining energy exceeds the E_{th} .
 - If $(E_{rem_i} < E_{cri})$, node i first computes the critical sleep duration (D_{csdt_i}) using equation (12) if its remaining energy

(E_{rem_i}) reaches E_{cri} , then it computes its second sleep duration (D_{sdt_i}) using equation (11) when its E_{rem_i} reaches E_{th} .

- Steps (1), (2), and (3) are executed, successively, by node i while the ($E_{rem_i} < E_{th}$); Otherwise, it computes its sleep duration (D_{sleep_i}) using equation (7) if its (D_{sdt_i}) is only computed, and equation (8) if both (D_{sdt_i}) and (D_{csdt_i}) are computed.

4 Simulation

We utilized the MiXiM simulation framework (version 2.3) in conjunction with the OMNeT++ (version 4.6) simulator to implement HEEMAC+ [18, 19]. To avoid potential overheads, the WSN is built with MiXiM with a grid topology, and the different parameters are defined in the omnetpp.ini file.

These parameters are displayed in Table 2 and include the total number of nodes, destination node, traffic rate, traffic param (the maximum time interval expressed in seconds between two packet transmissions), network type, application type, playground size, battery parameters (number, capacity, and number of devices), and parameters related to energy harvesting, which include harvesting type, harvesting param (the maximum time interval expressed in seconds between two harvested energy moments), and harvesting rate. A variety of scenarios are used and put into practice, ranging from 16 deployed nodes in a 400 m × 400 m area to 100 deployed sensor nodes in an 1100 m × 1100 m area.

In these scenarios, the distance between two neighbors in the grid is 100 m for 1000 simulated seconds, and all source sensor nodes are able to send their periodic traffic to the sink via multi-hop transmissions every 1.5 or 2.0 seconds. With harvesting for each sensor node randomly set between [0mW...50mW], the energy harvesting pace is periodic (every 1.5 seconds), as most ambient energy sources have distinct properties that vary depending on the various environmental conditions.

Table 2. Simulation parameters

Parameter	Value
Number of nodes	From 16 to 100
Topology	Grid
Simulation time	1000s
Playground Size (x,y)	(400m, 400m), (600m, 600m), (800m, 800m), (950m, 950m), (1100m,1100m)
Initial remaining energy E_{rem_i}	50 J
Energy threshold E_{ths}	35 J
Energy threshold E_{th}	30 J
Critical energy threshold E_{cri}	15 J
MaxAttempts	2
Traffic Type	Periodic
Traffic Param	1.5s, 2.0s
Number of Packets sent	1000
Destination Node	From 15 to 99
Mac Type	HEEMAC+Layer, EEMEHWSNLayer, DSRLayer, DSPLayer
Queue Size	10 packets
Battery capacity	100 J
Harvesting Type	Periodic
Harvesting Param	1.5 s
Harvesting rate randomly	[0mW...50mW]

Each sensor node has an initial energy of 50 J and a maximum energy value of 100 J. The values of the first, second, and critical energy thresholds, E_{ths} , E_{th} , and E_{cri} , are set to 35 J, 30 J, and 15 J, respectively.

5 Performance Evaluation

We evaluated our novel protocol, HEEMAC+, against three schemes, EEMEHWSN, DSR, and DSP, which employ a dynamic duty-cycle, similar to HEEMAC but do not more thoroughly and radically, through comprehensive simulations utilizing the OMNeT++ simulator with MiXiM framework. The DSR and DSP do not ensure load balancing in the network for which the maximum

wake-up interval of sensor nodes I_{dc}^{max} is set to 1 second. However, our protocol uses switching and sleep period calculation to assure load balance. The following performance metrics were employed to assess the performances:

Packet Delivery Ratio: the ratio of packets sent from the sources to packets received at the destination. This must be increased.

Mean Packet Latency: The average time needed by each packet to get to its target node is calculated. It is best to reduce this.

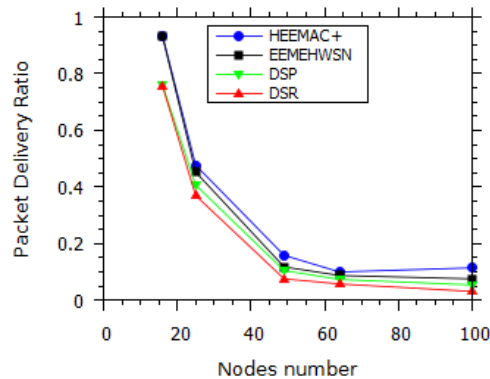
Throughput: The overall transmission rate that ought to be optimized (received bytes/simulation time).

Average Energy Consumption per node: It is calculated by dividing the total average energy used by all WSN nodes by their entire number of nodes, which need to be reduced.

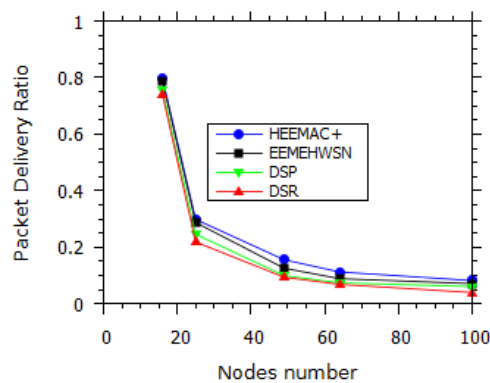
We present the following findings from the simulation tests conducted for our system HEEMAC+, the DSP, DSR, and EEMEHWSN schemes:

The simulation results are shown in Figure 6 for the delivery ratio measure with traffic parameter values of 1.5 and 2.0 seconds and a maximum number of packets transmitted set to 1000, with a network size ranging from 16 to 100 nodes. The findings show that when the number of nodes increases, the packet delivery ratio falls. Additionally, the results demonstrate that HEEMAC+, by dynamically and more profoundly adjusting the duty-cycles based on the remaining energy, which can increase over time, is able to produce a higher packet delivery ratio than EEMEHWSN, DSP, and DSR, which use the dynamic duty-cycle for various EH rates.

The sleep times in HEEMAC+ are deliberately chosen, making it unlikely that they will be the same for a neighborhood. Additionally, there is a decrease in sleep latency than DSR and DSP. As a result, there are fewer packet collisions and retransmissions. This is because, by calculating each node's appropriate sleep duration based on its remaining energy and switching technique usage, HEEMAC+ supports load balancing in WSN more than DSP and DSR, hence minimizing the likelihood of collisions.



(a) Traffic param = 1.5 seconds



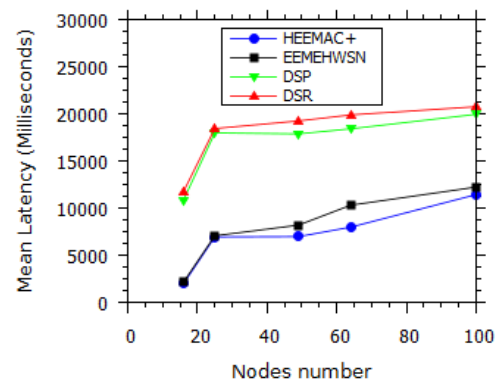
(b) Traffic param = 2.0 seconds

Fig. 6. Packet Delivery Ratio

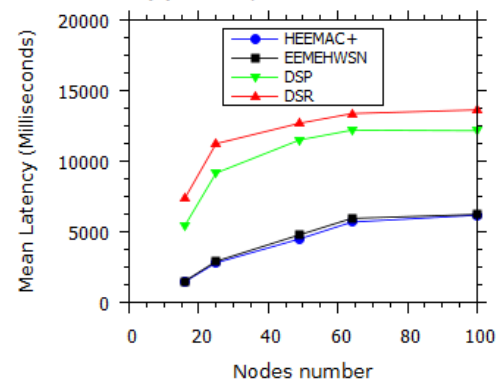
For instance, as can be seen in figure 6(a), HEEMAC+ performs better than EEMEHWSN, DSP, and DSR in terms of packet delivery ratio, with values ranging roughly between 0.32% and 25.74%, 18.69% and 34.07%, 18.99% and 52.36%, respectively, when the traffic parameter is set to 1.5 seconds and the node count increases from 16 to 49.

When the traffic parameter is set to 1.5 and the node count is 49, EEMEHWSN exhibits a higher packet delivery ratio (about 25.74%, 34.07%, and 52.37% in comparison to EEMEHWSN, DSP, and DSR, respectively) in Figure 6(a). Consequently, it is evident that HEEMAC+ outperforms EEMEHWSN, DSR, and DSP in terms of delivery ratio.

The findings for the mean latency measure are shown in Figure 7. We find that as the number of nodes rises, so does the mean latency. The findings demonstrate that because EEMEHWSN leverages the load balancing mechanism in WSN,



(a) Traffic param = 1.5 seconds



(b) Traffic param = 2.0 seconds

Fig. 7. Mean Latency

it introduces a lower mean latency than DSP and DSR. In contrast, EEMEHWSN, DSP, and DSR all have higher mean latency than HEEMAC+.

The first is that our scheme uses switching to ensure load balancing in WSN, which reduces the likelihood of more collisions, and the second is that nodes with high energy levels are favored to be active due to their low sleep period duration, which is calculated based on two equations (7) or (8) more accurately than EEMEHWSN.

Compared to EEMEHWSN, DSP, and DSR, there is a lower likelihood that data packets will be buffered, lowering the mean latency. Furthermore, load balancing is not taken into account only the sleep latency is lowered for DSR and DSP. In contrast to EEMEHWSN, DSP, and DSR, respectively, figure 7(a) for HEEMAC+ demonstrates a gain in terms of reduced mean latency of about 14.83%, 60.86%, and 63.66% when the traffic parameter = 1.5s and the nodes number = 49.

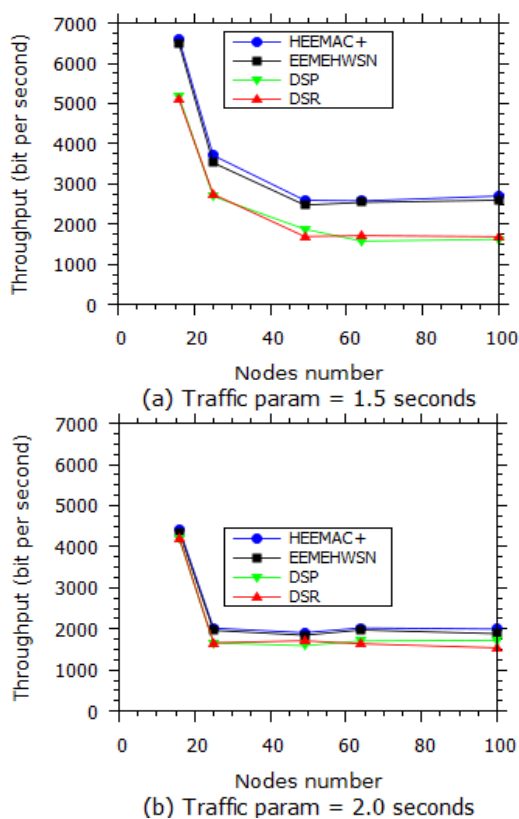


Fig. 8. Throughput

Similarly, when the traffic parameter is set to 2.0 s and the nodes count is 49, figure 7(b) illustrates an increase in reduced mean latency of around 6.11%, 60.81%, and 64.44%.

The throughput values produced by different numbers of nodes, expressed in bits per second (bps), are shown in Figure 8. The outcome shows that as the number of nodes increases, the throughput drops. Additionally, the results demonstrate that HEEMAC+ can provide greater throughput compared to EEMEHWSN, DSP, and DSR because it uses a method of switching and dynamically selects sensor nodes' sleep times, they are not likely to be the same and depend on the different amount energy harvested.

It also allows for duty cycles dynamic adjustment. In comparison to EEMEHWSN, DSP, and DSR, this reduces the maximum number of retransmissions and collisions by ensuring load balancing in the WSN. Because a successful transmission requires more time the more

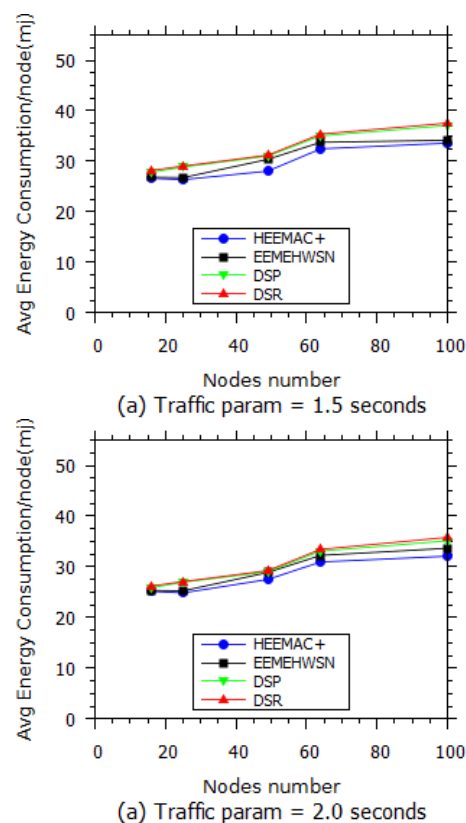


Fig. 9. Average energy consumption per node

accidents that occur. For instance, figure 8(a) shows that, when the nodes number is equal to 100 and the traffic parameter is set to 1.5s, HEEMAC+ produced substantial performance advantages (about 3.85, 38.84%, and 37.46%) in throughput when compared to EEMEHWSN, DSP, and DSR, respectively.

Subsequently, when the nodes number is 100 and the traffic parameter is adjusted to 2.0s, figure 8(b) shows notable performance advantages (about 5.65%, 13.56%, and 22.92%) in comparison with EEMEHWSN, DSP and DSR respectively.

Figure 9 shows that for different EH rates, HEEMAC+ performs better in terms of energy performance than EEMEHWSN, DSP, and DSR schemes. This is because the HEEMAC+ uses the switching method to ensure load balancing in WSNs that are unsecured by DSP and DSR. Each sensor node can more accurately adjust its duty cycle dynamically, and it is more deeply than

the EEMHWSN based on remaining energy that may increase over time.

The HEEMAC+ deliberately selects sleep periods that are unlikely to be the same for a neighborhood in the WSN for the sensor nodes, thereby minimizing collision and retransmissions, which are thought to be some of the main reasons for rising power consumption and frequent failures during data transmission within the network.

Figure 9(a) illustrates, for instance, that when the traffic parameter is set to 1.5 seconds and the nodes number is equal to 100, HEEMAC+ yields notable performance benefits (about 1.61, 9.39%, and 10.54%) in terms of energy savings when compared to EEMHWSN, DSP, and DSR for comparison.

Second, with the nodes count set to 100 and the traffic parameter set to 2.0s, figure 9(b) shows notable performance advantages (about 4.61%, 8.60%, and 10.34%) in comparison with EEMHWSN, DSP, and DSR respectively.

Consequently, lowering duty-cycle and load balancing is crucial to preventing energy consumption brought on by collisions and high node activity periods in order to improve a WSN's performance in terms of delivery ratio, mean latency, throughput, and energy efficiency in the EH.

In summary, based on the above-mentioned scenario results, we can conclude that the DSP system is more effective because it decreases the duty-cycle more dramatically than DSR on its foundation of residual energy, where the sleep latency decreased without considering load balancing, but it is no superior to EEMHWSN and HEEMAC+.

EEMHWSN has a greater average delivery ratio and lower average mean latency than DSP and DSR schemes because it employs a dynamic duty cycle with a determined sleeping interval, as well as a switching approach, ensuring load balancing in WSN.

As a result, HEEMAC+ is more efficient than EEMHWSN because it cuts the duty cycle more profoundly based on the remaining energy. So, when compared to the EEMHWSN, DSP, and DSR systems, HEEMAC+ displays excellent results.

6 Conclusion

Power is an essential component for WSN. The two main components are power preservation and delivery in the necessary shape, and an effort is made to rebuild the power that is used by replenishing the sensor node with power from an outside source, like solar cells. In specifically, this article aims to minimize the energy consumption of wireless sensor networks by lowering the duty-cycle of individual sensor nodes and, therefore, their operating duration, and also through the use of the switching process. For this reason, an asynchronous MAC protocol called HEEMAC+ (An Enhanced collected Energy Efficient MAC) was developed specifically for this kind of network.

Its basis is collected energy and it allows for increased energy efficiency in WSNs. Similar to EEMHWSN, but with more success and dependability, HEEMAC+ may lower the nodes duty-cycle as each node's remaining energy gradually grows. It also employs a switching technique for each sensor node to switch between the sensing mode and relaying mode based on their remaining energy, which can be increased over time to ensure load balancing in WSN. Finally, it ensures a balance between their energy harvesting ability and consumption of it. The proposed protocol is designed to allow each node to have a variable sleep duration based on the different rates at which they gather energy in their local area. Simulations are carried out to assess the performance of the proposed system against the DSP, DSR, and EEMHWSN methods. Simulation results showed that HEEMAC+ reduces energy consumption and packet delay while improving throughput and packet delivery ratio.

References

1. **Afroz, F., Braun, R. (2020).** Energy-Efficient MAC Protocols for Wireless Sensor Networks: A Survey. *International Journal of Sensor Networks*, Vol. 32, No. 3, pp. 150–173. DOI: 10.1504/ijsn.2020.105563.
2. **Bengheni, A., Didi, F. (2019).** Evaluating the Performance of Asynchronous MAC Protocols

- for Wireless Sensor Networks. *International Journal of Wireless and Mobile Computing*, Vol. 17, No. 3, pp. 245–257. DOI: 10.1504/ijwmc.2019.102254.
3. **Barat, A., Prabuchandran, K.J., Bhatnagar, S. (2024).** Energy Management in a Cooperative Energy Harvesting Wireless Sensor Network. *IEEE Communications Letters*, Vol. 28, No. 1, pp. 243–247. DOI: 10.1109/lcomm.2023.3335143.
 4. **Bengheni, A., Didi, F., Bambrik, I. (2019).** EEM-EHWSN: Enhanced Energy Management Scheme in Energy Harvesting Wireless Sensor Networks. *Wireless Networks*, Vol. 25, pp. 3029–3046. DOI: 10.1007/s11276-018-1701-8.
 5. **Sudevalayam, S., Kulkarni, P. (2010).** Energy Harvesting Sensor Nodes: Survey and Implications. *IEEE Communications Surveys & Tutorials*, pp. 1–19. DOI: 10.1109/surv.2011.060710.00094.
 6. **Nguyen, K., Nguyen, V.-H., Le, D.-D. (2014).** ERI-MAC: An Energy-Harvested Receiver-Initiated MAC Protocol for Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, Vol. 10, No. 5, pp. 1–8. DOI: 10.1155/2014/514169.
 7. **Fafoutis, X., Dragoni, N. (2011).** ODMAC: An On-Demand MAC Protocol for Energy Harvesting – Wireless Sensor Networks. *ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pp. 49–56. DOI: 10.1145/2069063.2069072.
 8. **Husnain, H., Sherazi, R., Grieco, L.A. (2018).** A Comprehensive Review on Energy Harvesting MAC Protocols in WSNs: Challenges and Tradeoffs. *Ad Hoc Networks*. DOI: 10.1016/j.adhoc.2018.01.007.
 9. **Fafoutis, X., Di Mauro, A., Orfanidis, C. (2015).** Energy-Efficient Medium Access Control for Energy Harvesting Communications. *IEEE Transactions on Consumer Electronics*, Vol. 61, No. 4, pp. 402–410. DOI: 10.1109/tce.2015.7389793.
 10. **Rasouli, H., Kavian, Y.S., Rashvand, H.F. (2014).** ADCA: Adaptive Duty Cycle Algorithm for Energy Efficient IEEE 802.15.4 Beacon-Enabled Wireless Sensor Networks. *IEEE Sensors Journal*, Vol. 14, No. 11, pp. 3893–3902. DOI: 10.1109/jsen.2014.2353574.
 11. **Kim, S.C., Jeon, J.H., Park, H.J. (2012).** QoS-Aware Energy-Efficient (QAEE) MAC Protocol for Energy Harvesting Wireless Sensor Networks. *Convergence and Hybrid Information Technology*, pp. 41–48. DOI: 10.1007/978-3-642-32645-5_6.
 12. **Yoo, H., Shim, M., Kim, D. (2012).** Dynamic Duty-cycle Scheduling Schemes for Energy-Harvesting Wireless Sensor Networks. *IEEE Communications Letters*, Vol. 16, No. 2, pp. 202–204. DOI: 10.1109/lcomm.2011.121311.112111.
 13. **Varghese, J., Jose, A.L. (2014).** Dynamic Duty-Cycled MAC for Wireless Sensor Networks with Energy Harvesters. *International Conference on Circuits, Communication, Control and Computing*, pp. 156–160. DOI: 10.1109/cimca.2014.7057780.
 14. **Bengheni, A., Didi, F., Bambrik, I. (2017).** Energy-Harvested Management Mechanism for Wireless Sensor Networks. *5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, pp. 1–4. DOI: 10.1109/icee-b.2017.8192022.
 15. **Bengheni, A. (2022).** A Multi-Threshold Energy Approach for Energy Harvesting WSN. *2nd International Conference on New Technologies of Information and Communication (NTIC)*, pp. 1–5. DOI: 10.1109/ntic55069.2022.10100485.
 16. **Sun, Y., Gurewitz, O., Johnson, D.B. (2008).** RI-MAC: A Receiver Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. DOI: 10.1145/1460412.1460414.
 17. **Bengheni, A. (2023).** Energy-Efficient Asynchronous MAC Protocol for WSNs with Energy Harvesters. *IEEE 11th International Conference on Systems and Control (ICSC)*, pp. 641–646. DOI: 10.1109/icsc58660.2023.10449734.

- 18. Köpke, A., Swigulski, M., Wessel, K. (2008).** Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision. Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications. Networks and Systems & Workshops (SIMUTools), pp. 1–8. DOI: 10.4108/icst.simutools2008.3031.

- 19. MiXiM Documentation. (2022).** MiXiM Documentation, <http://mixim.sourceforge.net>.

*Article received on 20/06/2025; accepted on 03/07/2026.
Corresponding author is Abdelmalek Bengheni.