COSFI-RIMAC: A Cooperative Short Frame Identifier Receiver Initiated MAC Protocol for Wireless Sensor Network

Lakhdar Goudjil1 · Fouzi Semchedine2

Accepted: 29 August 2022 / Published online: 12 September 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Power consumption is the most important factor to evaluate the performance of Wireless Sensor Networks. Most sensor network Medium Access Control (MAC) protocols operate based on a duty cycle mechanism. The asynchronous receiver-initiated MAC duty cycle protocols are popular due to their relatively higher energy efficiency. However, recent advances harnessing the benefits of cooperative communication have become one of the solutions of MAC duty cycle protocol. In this article, we improve the RI-MAC protocol by introducing a short frame identifier to notify the sender when the receiver wakes up. This resolution reduces idle listening, which increases energy performance. When the sender node receives a short frame identifier, it cooperates with neighboring senders, which minimizes collisions. Our protocol is called: a Cooperative Short Frame Identifier Receiver Initiated MAC protocol, COSFI-RIMAC is an asynchronous MAC protocol cooperative service cycle initiated by the receiver. The simulation result on the NS2 simulator shows that the COSFI-RIMAC mechanism reduces power consumption, produces minor latency, and increases the rate of packet delivery.

Keywords MAC · Receiver initiated · Cooperation · Energy saving · Idle listening · Collision

1 Introduction

Nowadays, Wireless Sensor Network (WSN) offers great promise for information capture and processing for different applications, such as animal tracking, precision farming, environmental monitoring, security and surveillance, military applications, smart buildings, health care, and so on. A WSN consists of several detection stations called sensor nodes. The power of each sensor node comes from a battery. A problem of such sensor devices is the limited lifetime because of the limitation of the battery capacity [1]. However, Energy

* Lakhdar Goudjil
lakhdar.goudjil@univ-setif.dz

1 Department of Computer Science, Faculty of Sciences, Ferhat Abbas University Setif 1, 19000 Setif, Algeria
2 Institute of Optics and Precision Mechanics (IOMP), Ferhat Abbas University Setif 1, 19000 Setif, Algeria
efficiency is essential to extend the life of the network. Media Access Protocols (MAC) is responsible for assembling data into frames and for disassembling frames to retrieve information [2]. The MAC sub-layer has widely been targeted by researchers to improve the performance of WSN. This is because the MAC layer coordinates directly with the radio of the node and can therefore play a vital role in managing energy usage [3]. The main objective of MAC protocol design is to minimize the energy waste due to idle listening, overhearing, packet overhead, collision, etc.

The basic idea of short duty cycle protocols is that sensor nodes can turn their radio on or off. In this way, we reduce the time during which a node is inactive or goes to hear unnecessary activity by putting the node on standby. For contention-based MAC protocols, the WSN hub typically accesses the media when it needs to pass data to each other [4]. Once a node wakes up, it listens to the channel for any activity before transmitting or receiving packets. If no packet is to be transmitted or received, the node returns to the standby state. A complete cycle consisting of a sleep period and a listening period is called a sleep/wake-up period. Indeed the ideal condition for low duty cycle protocols is when a node is on standby most of the time and only wakes up when it transmits or receives packets. Dynamic MAC protocols are contention-based protocols and adapt to topology changes. In contention-based MAC protocols, dynamic channel allocation is performed by contention algorithms such as CSMA, CSMA / CA; in which the communication resources are allocated dynamically, as a result, the design of a parallel transmission protocol must take into account not only the allocation of communication resources but also the process of exchanging control information.

1.1 Synchronous Low Duty Cycle MAC Protocols

Each node has two modes which are the wake and sleep nodes, synchronous duty cycling MAC protocols reduce the energy consumption by synchronizing the sensor node’s sleep and wake-up. Synchronous MAC protocols such as S-MAC [5] and T-MAC [6], require synchronization of sender and receiver clocks. Synchronous schemes can be based either on contention or on reserved time-slots. In both cases, a portion of the active state is used to synchronize all the nodes to a global sleep/wake-up schedule. Several protocols have been proposed for the same purpose, such as Speed-MAC [7], MC-LMAC [8], SEA-MAC [9]. Synchronous MAC protocols generate a large number of control messages to achieve synchronization before data exchange. Each transmitting node will create its way to the receiving node before transmission, which limits its usefulness when the duration of the node’s activity is very short, which also may increase the latency of data delivery. Likewise, the implementation of the synchronization is complex and difficult to be achieved when the number of sensor nodes is large.

1.2 Asynchronous Low Duty Cycle MAC Protocols

In contrast (to the precedent case), asynchronous duty cycling MAC protocols, do not require synchronization which may decrease energy consumption. However, asynchronous MAC protocols allow nodes to start their activity and inactivity periods independently. Asynchronous MAC protocols are distinguished by their short cycle time and their low communication cost. In the following, we describe the two main categories of asynchronous protocols: sender-initiated protocols (e.g. B-MAC [10], WiseMAC [11], and X-MAC [12]) and receiver-initiated protocols (e.g. RI-MAC [13], PW-MAC [14]
and EE-RI-MAC [15]). In sender-initiated MAC protocols, the transmission and data communication is decided and initiated by the sender node. Before sending data packets, a sender transmits a preamble to notify that there are data communications unaware of the state of the receiver (sleep or awake). Once a node wakes up during its regular period wake-up, it detects the preamble and will stay awake until it determines if it is the receiving node. This can require a large amount of energy from the transmitting node. The receiver-initiated MAC protocol shows better performance compared to the sender-initiated MAC protocols. In receiver-initiated protocols, receivers wake up independently to indicate their availability to receive data by sending a beacon. With this mechanism, the receiver beacon does not occupy the channel as long as the preambles in the sender-initiated protocols.

In this paper, we present a new protocol based on the RI-MAC protocol (COSFI-RIMAC), which is based on two main reasons: The first reason is the idle listening in receiver-initiated MAC protocols, which represents the main source of energy consumption. In Fig. 1, which represents the basic description of RI-MAC [13], senders have to listen unnecessarily for a large time $ST_{idle}$ (a few seconds), while a sender data transaction ($ST_{data}$) can take a few milliseconds (ms). The energy consumption of this idle listening affects the energy performance of the RI-MAC protocol. Our COSFI-RIMAC protocol minimizes the idle listening for the nodes (senders and receivers) which we divide by small periods thus avoiding early wake-up. The second reason is to minimize the overhead in receiver-initiated MAC protocols caused by a collision between nodes, which is the reason for communication delay in receiver-initiated protocols. Our COSFI-RIMAC proposal reduces collisions between nodes by using a collision avoidance technique based on cooperation between sending nodes, which minimizes the energy associated with data retransmission.

The rest of this article is organized as follows: Sect. 2 presents the related work. The COSFI-RIMAC protocol is described in Sect. 3. The evaluation and comparison of the performance of the proposed protocol with other protocols are presented in Sect. 4. Finally, the conclusion and the future scope of this work are presented in Sect. 5.
2 Related Work

Due to the drawbacks of sender-initiated protocols, researchers turned their attention to the receiver-initiated approach. In this section, we first describe RI-MAC [13]; which is the first asynchronous receiver-initiated MAC protocol. As seen in Fig. 1, in RI-MAC nodes sleep when not engaged in communication. Periodically, the receiver broadcasts a beacon after its wakeup, announcing that it is awake and ready to receive packets. Upon the beacon reception, the sender transmits its packet. The sender will enter the sleep mode after completing the data transmission. If the sender remains silent after a beacon, the receiver node will also enter into sleep mode.

Several works use cooperative communication in WSN. Among these different works, some proposals are based on asynchronous receiver-initiated communication. In this work, we discuss some asynchronous receiver-initiated cooperative protocols. OC-MAC [16] improves RI-MAC by exploiting the cooperation between active senders to reduce their latency, especially in a network where the nodes have many neighbors which saves more energy. In OC-MAC, active neighboring senders are allowed to exchange their frames with each other while waiting for the receiver to wake up. Thus, after having designated a relay that waits for the receivers to be available and to have transmitted their frames to it, the other senders can go into sleep mode, which reduces the energy consumption of the senders. ASYM-MAC [17, 18] works over RI-MAC designed for asynchronous low-power duty-cycled. ASYM-MAC combines sender and receiver-initiated MAC protocols. By default, the system operates on receiver-initiated mode (R-mode), when the link is determined to be asymmetric; Asym-MAC switches its mode to the sender-initiated protocol (T-mode). When the sender cannot get a beacon from its receiver, it needs to stay in the listening state for time ‘t’ until the beacon packet arrives, and then switches to sender initiated mode where it sends its data followed by a long preamble. However, when the sender gets a beacon from the receiver but the sender cannot send data due to link asymmetry, this protocol performs very poorly. COASYM-MAC [19, 20] is a cooperative asynchronous duty cycle MAC protocol for WSNs proposed to minimize the harmful effects of asymmetric links and to meet sufficient quality of service requirements. It designs an efficient tree technique to select an efficient helper node and resolve link asymmetry between sender and recipient, recipient to sender, or both. In addition, COASYM-MAC provides an algorithm for the wake-up schedule for each node. COASYM-MAC works much better than a MAC protocol that manages asymmetric links: ASYM-MAC. EnRI-MAC [21], is a protocol based on RI-MAC. In EnRI-MAC, cooperation between senders is performed by sending a notification message by each sender before receiving the recipient’s tag, to inform other senders of the same intended recipient that they have data to send. Despite obtaining beacons from the intended receiver, other nodes refrain from sending data in the current cycle. In the case of link asymmetry, other senders cannot receive the wake-up notification. In such a case, multiple senders transmit their data to the receiver that may increase the packet collision rate. RIVER-MAC [22] is a receiver-initiated asynchronous duty cycle MAC protocol. It offers two major changes to RI-MAC: Rendezvous CCA to reduce idle listening, and beacon flow-based collision resolution to reduce conflict between receivers. The first major improvement is a Rendezvous based on the CCA. They adapted this mechanism in the CCA strobe and then applied it on the sender side to reduce idle listening.
3 Proposed COSFI-RIMAC Protocol

In this article, we propose a new cooperative asynchronous receiver-initiated MAC protocol called Cooperative Short Frame Identifier Receiver Initiated MAC protocol (COSFI-RIMAC). Our main contribution is to reduce collision and idle listening, and thus energy consumption. Our scheme is based on RI-MAC. We offer two improvements: First, the sender’s nodes listen to the channel independently during a short period and wait for a Short Frame Identifier (SFI) from a potential receiver, if it does not receive it goes back to sleep mode in a dynamic interval which minimizes idle listening enormously. The second improvement, introducing cooperation between the sender’s nodes, which minimizes collisions. This protocol also introduces a dynamic adaptation mechanism of the service cycle, which aims to converge the waking and waking hours to data traffic conditions.

The objectives of the COSFI-RIMAC protocol are mentioned as follows:

- Minimize idle listening power consumption.
- Cooperation between sender nodes.
- Reduce the number of collisions.
- Reduce the number of transmissions,
- Reduce the energy consumption.

3.1 Protocol Design

The COSFI-RIMAC protocol is a contention, asynchronous, duty cycle, and receiver-initiated MAC protocol is based on the following main axes:

- The duty cycle is not fixed, but dynamic.
- The receiver does not immediately go into sleep mode after an ACK packet but rather waits for another.
- Potential DATA packet from a sender.
- The SFI frame contains more information to inform the sender when the receiver wakes up.

Now we describe our proposed protocol COSFI-RIMAC in detail. Table 1 describes the notations used in this paper.

3.1.1 Protocol Operations Diagrams

COSFI-RIMAC improves RI-MAC by reducing the energy wastage of the sender when it waits for a receiver to wake up. As seen in Figs. 2, 3 and 4 the COSFI-RIMAC, has three phases: Asynchronous cooperation wake-up (Asynch Coop-Wake-up), synchronous wake-up data transmission (Synch Wake-up Data Transmission), and the appointment wake-up data transmission phases.
3.1.2 Phase Asynch-Coop Wake-Up

In COSFI-RIMAC at the start, each node has its duty cycle, during the asynchronous wake-up period AsynRI$_{SFI}$. The receiver makes small RWI$_{SFI}$ wakes between dynamic RT$_{sleep}$ intervals to send short SFI frames, each SFI contains the information: wake-up time of Receiver and Receiver Address.

The sending of the next SFI frame is related to the previous TR$_{sleep}$ interval time and calculated as shown in the following Eq. (1):

$$RT_{sleep,j+1} = RT_{sleep,j} + (RT_{a sleep} * j)/2$$ (1)

In the Asynch Wake-up phase, the receivers do perform a small AsynRWI wake-up, then they remain in AsynRI$_{sleep}$ sleep mode most of the time as indicated in Eqs. (2, 3)
AsynRWI = (RWISFI)

AsynRI_{sleep} = (RT_{sleep}) + RI_{sleep}
If a sender receives an SFIj from the expected receiver, it will synchronize its data transmission wake-up. After the receiver goes into RIsleep sleep mode to save power. When a node has a frame to send DATA, it listens to the channel for short time STidle and waits for an SFIj from a potential receiver, if it does not receive it returns to sleep mode in a dynamic interval STsleep calculated as shown in Eq. (4):

$$ST_{sleep_{k+1}} = ST_{sleep_k} + (STa_{sleep} * k)/2;$$

The transmitting node makes a small SWICOOP wake-up to cooperate with their neighboring senders by sending a SWsyn frame after the STCOsleep interval calculate as shown in Eq. (5), after which it also goes into SIsleep sleep mode, to save energy.

$$STCO_{sleep} = ST_{sleep_{k-1}} + (STa_{sleep} * j)$$

In this phase as indicated in algorithm 1, the COSFI-RIMAC protocol minimizes the idle listening of the sender nodes (AsynSIdle) which only becomes the sum of the STidle- as indicated in Eq. (6), which gives the majority of the senders time to go into sleep mode AsynSIsleep as shown in Eq. (7).

$$AsynST_{idle} = (ST_{idle})$$
A formal description of the Phase Asynch-Coop Wake-up is presented in Algorithm 1.

Algorithm 1 Asynch-Coop Wake-up

1. Begin of Asynch-Coop Wake-up receiver
2. Begin of ASYNC Wake up of the receiver
3. Initialization: \( i=1; \ k=1; \ ST_{\text{sleep}} ; RT_{\text{sleep}} \)
4. Sender launch Asynch wakeup interval \( AsynSI \)
5. Sender startSI_{idle} interval
6. Sender SI listens to the channel for short time \( ST_{\text{idle}} \)
7. Receiver RI launch Asynch wakeup interval \( AsynRI \)
8. Receiver launch \( RI_{SFI} \) Interval
9. Receiver RI wakeup \( RWI_{SFI} \)
10. Receiver RI send SFI
11. If \( SFI \) is not Received by Sender SI then
12. Initial sleep Sender \( ST_{\text{sleep}} \) between \( ST_{idle1} \) and \( ST_{idle2} \)
13. Initial sleep Receiver \( RT_{\text{sleep}} \) between \( RWI_{SFI1} \) and \( RWI_{SFI2} \)
14. Endif
15. DO
16. \( RT_{\text{sleep}i+1} = RT_{\text{sleep}i} + (RTa_{\text{sleep}*i})/2 \)
17. \( ST_{\text{sleep}k+1} = ST_{\text{sleep}k} + (STa_{\text{sleep}*k})/2 \)
18. Receiver sends SFI
19. UNTIL (Sender Receive SFI) OR (end of \( RI_{SFI} \))
20. Sender SI synchronizes their awakewith receiver RI
21. The receivergoes to sleepRI_{sleep} interval
22. End of ASYNC Wake up of the receiver
23. Begin Coop senders
24. Calculate \( STCO_{\text{sleep}} = ST_{\text{sleep}} j-1 + (STa_{\text{sleep}*j}) \)
25. Sender SI goes to sleepSTCO_{\text{sleep}}
26. while (sender \( S_i \) receive Swsyn )and (not the end of \( SI_{SFI} \)) do
27. Sender \( S_i \) wake up SW_{COOP}
28. Sender \( S_i \) Send Swsyni to other neighboring senders
29. Sender \( S_i \) switches to \( SI_{sleep} \) sleep mode
30. End while
31. End coop senders
32. End of Asynch-Coop Wake-up receiver

3.1.3 Phase Synch Wake-Up Data Transmission

This phase consists of the synchronization between nodes (transmitter, receiver) for the transmission of data. In this synchronous communication cycle \( SynWI_{data} \) the sender
node which receives an SFI wakes up (a synchronized wake-up) simultaneously with the receiver node, this time the sender starts sending this data immediately, as soon as the receiver sends the short Beacon signal indicating their awake; The time required to send the data as shown in Eq. (8).

\[
\text{SynWI}_{\text{data}} = T_B + T_{\text{data}} + T_{\text{ACK}}
\]  

(8)

Once the receiver receives the data, it sends an ACK indicating that the data was received successfully. After the transmitted beacon, the receiver R1 waits for another transmission from the sender S1 and then goes into the AppR\text{sleep} state. If there is a Wake-up appointment, the sender S1 goes into the AppS\text{sleep} state, otherwise, it waits for an SFI from another receiver R2 to synchronize its wake-up and goes into SYNC Wake up data transmission phase.

A formal description of the Phase SYNC Wakes up data transmission is presented in Algorithm 2.
Algorithm 2  SYNC Wake up data transmission

1. Begin of SYNC Wake up data transmission
2. Launch of the SynWI_{data}\text{interval}
3. Wakeup of receiver R1 and sender S1
4. Sender S1 wait in T_{B}\text{for beacon B}
5. if sender S1 receives a beacon B from receiver R1 then
6.  Sender S1 sends a DATA in T_{data} to receiver R1
7.  End if
8. while collision do
9.  Receiver R1 Calculate random Back off
10. Receiver R1 send beacon B_{backoff}
11. Sender S1 receive B_{backoff}
12. Sender S1 sends a DATA after back off period to receiver R1
13. end while
14. if a data packet is received from receiver R1 then
15.  Send ACK in T_{ACK} to sender S1
16. end if
17. the receiver R1 waits for another transmission of sender S1
18. while sender S1 there is another data to transmit do
19.  Sender S1 sends a DATA_{i} in T_{data} to receiver R1
20.  Receiver R1 receive DATA_{i}
21.  Receiver R1 sends ACK in T_{ACK} to sender S1
22. end while
23. The receiver goes to AppSleep state
24. if not exist Appointment Wake up then
25.  Sender S1 waits for an SF1 for another receiver R2
26.  if Sender S1 receive SF1 for another receiver R2 then
27.  Sender S1 synchronize their awakewith the receiver R2
28.  Sender S1 go to SYNC Wake up data transmission phase,
29. else sender S1 goes to AppSleep state
30. end if
31. end if
32. End of end of SYNC Wake up data transmission

3.1.4 Phase Appointment Wake-Up Data Transmission

In the previous phase, the response ACK indicates the next wake-up of nodes as appointment Wake-up. In this phase, the sender S1 sends to DATA after the reception of a beacon from the receiver R1. The sender sends the next DATA (if it exists) after each ACK. If there is the next communication, the nodes pass another appointment Wake-up. Otherwise, the
sender S1 waits for an SFI for another receiver R2 to synchronize its wake-up and go to the SYNC wake-up data transmission phase, otherwise go to the Asynch-Coop wake-up phase.

A formal description of the Phase Appointment Wake-up data transmission is presented in Algorithm 3.

### Algorithm 3 Appointment Wake-up data transmission

1. **Begin of appointment Wake-up data transmission**
2. Receiver LaunchAppRWI\text{dataInterval}
3. Receiver LaunchAppRWI\text{dataInterval}
4. Sender S1 receives a beacon from the receiver R1
5. **Sender S1 sends a DATA in T\text{data} to receiver R1**
6. Receiver R1 receives DATA
7. Receiver R1 sends ACK
8. Receiver wait for another data for sender S1
9. Sender S1 receives ACK
10. **while sender S1 there is data to be transmitted to R1 do**
11. Sender S1 sends a DATA to receiver R1
12. Receiver R1 receives DATA
13. Receiver R1 sends ACK to sender S1
14. **end while**
15. **if** there is another Appointment Wake-up data transmission **then**
16. **go to Appointment Wake-up data transmission phase,**
17. **else** sender S1 waits for an SFI for another receiver Rj
18. **if** sender S1 receive SFI **then**
19. sender S1 synchronize their wake-up with the receiver Rj
20. **end if**
21. **else** go to Asynch-Coop Wake-up phase.
22. **Endif**
23. **end of Appointment Wake-up data transmission**

### 3.2 Cooperation and Collision Settlement

The RI-MAC protocol is more vulnerable to collisions because the senders immediately transmit the data after receiving a Beacon to the same receiver, which is also a source of wasted energy. As seen in Fig. 5, the scenario below shows the collision avoidance technique used by our COSFI-RIMAC protocol.

When the sender node S1 receives an SFI, it makes a small SWICOOP wake-up to cooperate with their neighbors after SWsleep by sending a SWsyn 1 frame which contains the information of the SFI and wake-up time and SynWIdata of S1 and it goes into a sleep state. The sender S2 is already in the SWISFI period, in this case, the S2 receives SWsyn 1, it synchronizes its data transmission after S1, in the same cooperation mechanism S2 informs their neighbors. In our example, the RI-MAC protocol case, all the sending nodes (S1, S2, S3, and
S4) perform a back-off, on the other hand in the COSFI-RIMAC only S3 and S4 which perform a back-off which minimizes the number of collisions and therefore data retransmissions, which reduces power consumption and increases the delivery rate of data packets.

### 4 Performance Evaluation of COSFI-RIMAC

#### 4.1 Simulation Environment

In this section, we evaluate and compare our SFI-RIMAC proposals (our proposal without cooperation) and our final COSFI-RIMAC protocol (SFI-RIMAC plus cooperation) with the RI-MAC. These simulations are carried out using the NS-2 network simulator by performing different scenarios. Table 2 illustrates the simulation parameters for COSFI-RIMAC. The simulation lasted 200 seconds, during which 10, 20, 30, 40, and 50 nodes were randomly deployed in an area of 550 square meters.

#### 4.2 Result and Discussion

Energy consumption is often presented in joules/bits and is calculated as the ratio of the total energy consumed by the total number of data bits transmitted. The smaller this value, the more efficient the protocol will be. Figure 6a shows the results of power consumption by the number of nodes. When the number of nodes increases, the power consumption increases in COSFI-RIMAC, SFI-RIMAC, and RI-MAC as the number of retransmission increases. In addition, our proposal has reduced the power consumption compared to RI-MAC, as it reduces idle listening time and minimizes the number of collisions.

The average end-to-end delay is a measure of the delay between the sending of the message by the source node and its receipt by the receiving node. Figure 6b shows...
the end-to-end average delay results. Compared to RI-MAC, synchronization between
sender and receiver minimizes average end-to-end delays in SFI-RIMAC and COSFI-
RIMAC protocols, and cooperation between senders improves COSFI-RIMAC. Figure 6c represents the number of packets received; our SFI-RIMAC and COSFI-RIMAC proposals minimize the number of collisions which gives more data transformation than in RIMAC protocol. Packet Delivery Rate (PDR) is the percentage of successfully received packets divided by the number of data packets sent. Figure 6d shows the results for the packet delivery rate for the COSFI-RIMAC, SFI-RIMAC, and RI-MAC protocols. In general, the simulation results show that the SFI-RIMAC protocol has a higher packet delivery rate than RI-MAC due to the insertion of the SFI short frame identifier. Also, our proposed mechanism COSFI-RIMAC increases the rate of packet delivery.

Latency is the time it takes for a data packet to travel from a node to the sink. Figure 6e shows a comparison of the latency results between our proposals SFI-RIMAC, COSFI-RIMAC, and the RI-MAC protocol. Typically, latency increases as the number of nodes increase for all protocols. However, due to the use of all three transmission phases (Asynch-Coop Wake-up, Synch Wake-up Data Transmission, Appointment Wake-up Data Transmission), the proposed COSFI-RIMAC protocol exhibited the least latency compared to RI-MAC. The average throughput determines the number of packets transmitted from a source to a destination node per unit time. To express it, we use the size of the packets in bits, thus the result is in bits/seconds. Figure 6f represents the comparison of the average flow between our proposals SFI-RIMAC, COSFI-RIMAC, and RI-MAC protocol. The average throughput in the three protocols

| Parameter                  | Value                      |
|----------------------------|----------------------------|
| MAC protocol               | RIMAC, SFI-RIMAC, COSFI-RIMAC |
| Node deployment area       | 550 m × 550 m              |
| Simulation time            | 200 s                      |
| Number of nodes            | 10, 20, 30, 40, 50         |
| Package size               | 2000 Byte                  |
| SFI size                   | 12 Byte                    |
| B size                     | 12 Byte                    |
| ACK size                   | 12 c                       |
| Transmission energy        | 0.0312 w                   |
| Receiving energy           | 0.0222w                    |
| The time to send B         | T_B = 1.0 ms               |
| The time to send ACK       | T_ACK = 1.0 ms             |
| The time to send SFI       | RWI_{SFI} = 1.0 ms         |
| The time to send a data    | T_{data} = 2.5 ms          |
| Sender time idle listening | S_{idle} = 0.5 ms          |
| Random sender time sleep   | S_{tra sleep} = random [0.05 s, 0.1 s] |
| Random receiver time sleep | R_{tra sleep} = random [0.05 s, 0.1 s] |
| Sender time sleep initially| S_{sleep1} = 0.3 s         |
| Receiver time sleep initially| R_{sleep1} = 0.2 s         |
| The sleep duration Ω       | Ω = 0.5 s                  |
increases linearly with the increase in the number of nodes, the minimization of collisions in our proposals (SFI-RIMAC and COSFI-RIMAC) reduces the average throughput compared to the RI-MAC protocol.

5 Conclusion

In this paper, we have proposed a high-performance, collision-minimizing and energy-efficient cooperative receiver-initiated MAC protocol, called COSFI-RIMAC, for wireless sensor networks. COSFI-RIMAC dynamically minimizes node wakeups and
cooperates between neighboring senders to synchronize sleep/wakeup periods that reduce idle listening and minimize the number of collisions.

Simulation results show that COSFI-RIMAC significantly improves energy consumption, average throughput, average end-to-end delay, and latency. In addition, COSFI-RIMAC allows more data to be transferred and increases the packet delivery rate. Future work includes improving our protocol to support cooperation between receiving nodes.

**Funding** Not applicable.

**Data Availability** Not applicable.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**References**

1. Chowdhury, S. M., & Hossain, A. (2020). Different energy saving schemes in wireless sensor networks: A survey. Springer.
2. Kochhar, A., et al. (2018). Protocols for wireless sensor networks: A survey. *Journal of Telecommunications and Information Technology, 7*, 18.
3. Chen, D., & Tao, Z. (2011). An adaptive polling interval and short preamble media access control protocol for wireless sensor networks. *Frontiers of Computer Science in China, 5*(3), 300–307.
4. Ajmi, N., Helali, A., Lorenz, P., & Mghaieth, R. (2020). SPEECH-MAC: Special purpose energy-efficient contention-based hybrid MAC protocol for WSN and Zigbee network. *International Journal of Communication Systems, 34*, e4637.
5. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFO COM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings* (Vol. 3, pp. 1567–1576). IEEE.
6. Du, S., Saha, A. K., & Johnson, D. B. (2007). RMAC: A routing enhanced duty-cycle MAC protocol for wireless sensor networks. In *IEEE Infocom 2007–26th IEEE international conference on computer communications* (pp. 1478–1486).
7. Choi, L., Lee, S. H., & Jun, J. A. (2010) SPEED-MAC: Speedy and energy-efficient data delivery MAC protocol for real-time sensor network applications. In *Proceeding of the international conference on communications (ICC)* (pp. 1–6).
8. Ozlem, D. I., Lodewijk, V. H., Pierre, J., & Paul, H. (2011) MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. In *Ad Hoc networks* (Vol. 9). Elsevier.
9. Zhao, Y.Z., Miao, C.Y., & Ma, M. (2012). An energy-efficient self-adaptive duty cycle MAC protocol for traffic-dynamic wireless sensor networks. In *Wireless personal communications* (pp. 1287–1315).
10. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *ACM SenSys ’04* (pp. 95–107).
11. El-Hoiydi, A., & Decotignie, J. -D. (2004) WiseMAC: An ultra-low-power MAC protocol for multi-hop wireless sensor networks. In *Algorithmic aspects of wireless sensor networks* (pp. 18–31). Springer.
12. Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *ACM SenSys ’06* (pp. 307–320). ACM.
13. 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. In *Proceedings of ACM SenSys* (pp. 1–14).
14. Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011) PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *IEEE INFOCOM’11* (pp. 1305–1313).
15. Yong, Y., Chow, C., Kanesan, J., & Ishii, H. (2011). EE-RI-MAC: An energy-efficient receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *International Journal of Physical Sciences, 6*(11), 2633–2643.
16. Wang, X., Zhang, X., Chen, G., & Zhang, Q. (2010). Opportunistic cooperation in low duty cycle wireless sensor networks. In *IEEE ICC’10* (pp. 1–5).
17. Won, M., Park, T., & Son, S. H. (2014). Asym-MAC: A MAC protocol for low-power duty-cycled wireless sensor networks with asymmetric links. *IEEE Communications Letters, 18*, 809–812.
18. Pegatoquet, A., Le, T. N., & Magno, M. (2019). A wake-up radio-based MAC protocol for autonomous wireless sensor networks. *IEEE/ACM Trans. Netw.*, 27, 56–70.

19. Hasan, M. M., Karmaker, A., Moni, S. S., & Alam, M. S. (2016). COASYM-MAC: A cooperative asymmetric MAC protocol for wireless sensor network. In *Proceedings of the 9th international conference on electrical and computer engineering (ICECE)*, Dhaka, Bangladesh.

20. Md Hasan, M. M., Karmaker, A., Alam, M. S., & Craig, A. (2019). Minimizing the adverse effects of asymmetric links: A novel cooperative asynchronous MAC protocol for wireless sensor networks. *MDPI Sensors*.

21. Lee, J., & Kim, S. (2018). EnRI-MAC: An enhanced receiver-initiated MAC protocol for various traffic types in wireless sensor networks. *Wireless Network* 1–10.

22. Wymore, M. L., & Qiao, D. (2019). RIVER-MAC: A receiver-initiated asynchronously duty-cycled MAC protocol for the internet of things. In *IEEE 43rd annual computer software and applications conference (COMPSAC)* (Vol. 1, pp. 860–869).

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

**Lakhdar Goudjil** is currently a research professor at the University of Sétif, Algeria. He obtained his Magister in 2009 in artificial intelligence under title at the Doctoral School of lest—Pole Annaba—(Algeria). He works in the field of routing aspects, MAC protocols, Medical IOT, security and quality of service in wireless sensor networks. He is also interested in the problem Optimizing MAC mechanisms for energy management in wireless sensor networks. Currently, he is a member in the research project “Fault Tolerance and Security Management in Medical IoT”.

**Fouzi Semchedine** is currently a Full Professor at the University of Sétif, Algeria. He received his PH.D in 2011 in computer science from the University of Béja’a, (Algeria). He works in the area of routing, security and quality of service aspects in wireless sensor networks, vehicular networks and hybrid sensor and vehicular networks. He interests also to the localization problem and the compression of data in wireless networks. Actually, he is a head of the research project “Security and Reliability of Wireless Networks (Ad hoc and Sensors)”.