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

Lakhdar Goudjil (lakhdar.goudjil@univ-setif.dz)
Universite Ferhat Abbas Setif 1
Fouzi Semchedine
Universite Ferhat Abbas Setif 1

Research Article

Keywords: MAC, Receiver initiated, Cooperation, Energy saving, Idle listening, Collision

Posted Date: September 21st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-780298/v1

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

Lakhdar Goudjil[1],Fouzi Semchedine[2]

[1]Department of Computer Science,Faculty of Sciences,Ferhat Abbas University Setif 19000,Algeria

[2]Institute of Optics and Precision Mechanics (IOMP), Ferhat Abbas UniversitySetif 19000, Algeria

Corresponding AuthorEmail:lakhdar.goudjil@univ-setif.dz

ABSTRACT

Power consumption is the most important factor to evaluate the performance of Wireless Sensor Networks (WSNs). Most sensor network Medium Access Control (MAC) protocols operate on the basis of 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 has 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 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 have widely been targeted by the researchers for the purpose of improving the performance of WSN. This is because MAC layer coordinates directly with the radio of node and can therefore play a vital role for 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.

Extending the life of sensor nodes is still a difficult area of research in (WSN).The basic idea of short duty cycle protocols is that sensor nodes can turn their radio on or off.In this we may, 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 really 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 most 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 sensor node's sleep and wake-up. In fact, 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 their usefulness when the duration of 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 decreases 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 sendernode. 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 show 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.

There are two main reasons for focusing on receiver initiated MAC protocols: The first reason is idle listening at rest which represents the main source of power consumption in RI-MAC[13]. In RI-MAC (Fig. 1), senders have to listen unnecessarily for a large period of timeST_{idle} (a few seconds), while a data transaction ST_{data}(sender datatransaction) can take a few milliseconds(ms). The energy consumption of this idle listening makes the energy performance of RI-MAC protocol. So the objective is to minimize the idle listening at rest of the sending node which prevents an early awakening.

The second reason motivated in receiver initiated MAC protocols is the overheadcaused by the collision between the nodes, which is the reason for the communication delay in the sender-initiated protocols. So the goal is to reduce collisions between nodes, which minimize the energy associated with data retransmission.This paper presents a novel protocol based on RI-MAC protocol (COSFI-RIMAC). COSFI-RIMAC minimizes idle listening for the nodes (senders and receivers) we divide by small periods. Additionally, COSFI-RIMAC uses a collision avoidance technique based on the cooperation between the sending nodes.

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

2. RELATED WORK

Due to the drawbacks of sender-initiated protocols, researchers turned their attention to receiver-initiated approach. In this section, we first describe RI-MAC [13];which is the first asynchronous receiver-initiated MAC protocol. As seen in Figure 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 to the sleep mode.

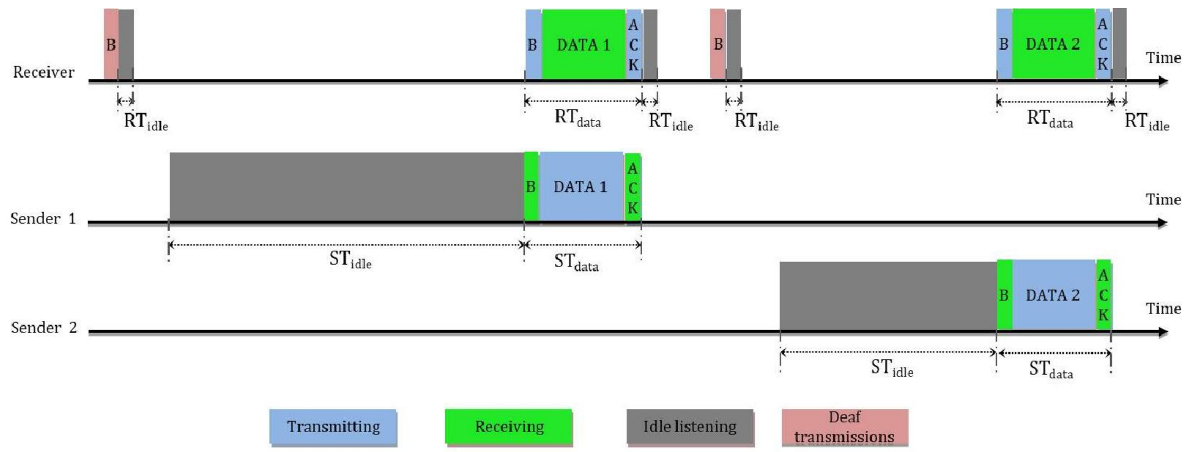


Figure.1 RI-MAC protocol operation process

Several works use cooperative communication in WSN. Among these different works, some proposals are based on the 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 save 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 which 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, 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 sender cannot get 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 sender gets beacon from receiver but 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 design an efficient tree technique to select an efficient helper node and resolve link asymmetry between sender and recipient, recipient to sender, or both. Also COASYM-MAC provides an algorithm for wake-up schedule for each node. COASYM-MAC works much better than a MAC protocol which manages asymmetric links: ASYM-MAC. EnRI-MAC [21], is a protocol based on RI-MAC. inEnRI-MAC, a 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 case of link asymmetry, other senders cannot receive the wake-up notification. In such a case, multiple senders transmit their data to the receiverwhich may increase 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 strob and then applied it on the sender side to reduce idle listening.

3. Proposed COSFI-RIMAC protocol

In this article, we propose a new asynchronous receiver-initiated MAC duty cycle 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 major modifications of RI-MAC. First, in order to synchronize the senders and the receiver's nodes: the receiver node transmits short frame identifiers (SFI) containing more information on their wake-up time; second, our proposal: introduces cooperation (CO) between sender's nodes. 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 following:

• Minimize idle listening power consumption.

• Cooperationbetweensendernodes

• Reduce the energy consumption

• Reduce the number of transmissions,

• Reduce the number of collisions

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 containing 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.

Notation	Description
SFI	Short Frame Identifier
RI_{SFI}	Receiver Interval to send SFI
RWI_{SFI}	Receiver wakeup interval to send SFI
RT_{sleep}	Receiver Time sleep, dynamic sleep time between awakenings of a receiver to send SFI
$RTra_{sleep}$	RandomRT_{sleep}
RT_{sleep1}	RT sleep initial, the time RT between sending SFI_1 and SFI_2,
RI_{sleep}	ReceiverIntervalsleep,
AsynRI	AsynchReceiverInterval
Ω_{sleep}	The sleep duration between sending the last possible SFI and before the receiver wakes up;
RT_{idle}	Receiver Time idle ,Receiver idle listening to the channel for short periods
ST_{idle}	Sender Time idle
ST_{sleep}	Sender Time sleep ,dynamic time between idle of a sender
ST_{sleep1}	STsleep initial, the time ST between ST_{idle1} and $ST_{idle\,2}$,
SI_{idle}	SenderIntervalidle
SI_{sleep}	SenderIntervalsleep
AsynSI	AsynchSender Interval
$STra_{sleep}$	RandomST_{sleep}
$STCO_{sleep}$	Sender Time Cooperationsleep
SWI_{coop}	ReceiverWakeupIntervalCooperation
SW_{syn}	Sender Wake-up synch,
$AppR_{sleep}$	Appointement Receiversleep
$AppWI_{data}$	Appointement Wake-up Interval Data
$AppS_{sleep}$	Appointement Sender sleep
$SynWI_{data}$	Synchronous Wake-up Interval DATA
T_B	The time required to send a Beacon B
T_{data}	The time required to send a data packet
T_{ACK}	The time required to send ACK.

Table 1. Description of notations

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 Figures(2,3,4) 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 theappointment wake-up data transmission phases.

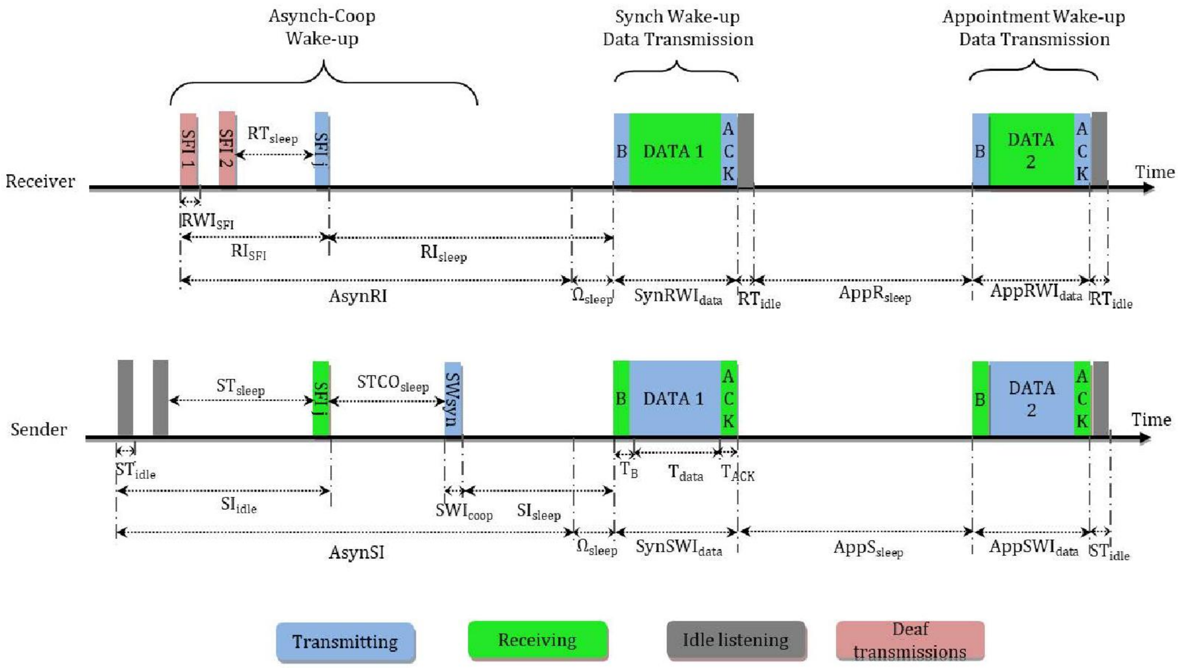


Figure 2. COSFI-RIMAC protocol operation process

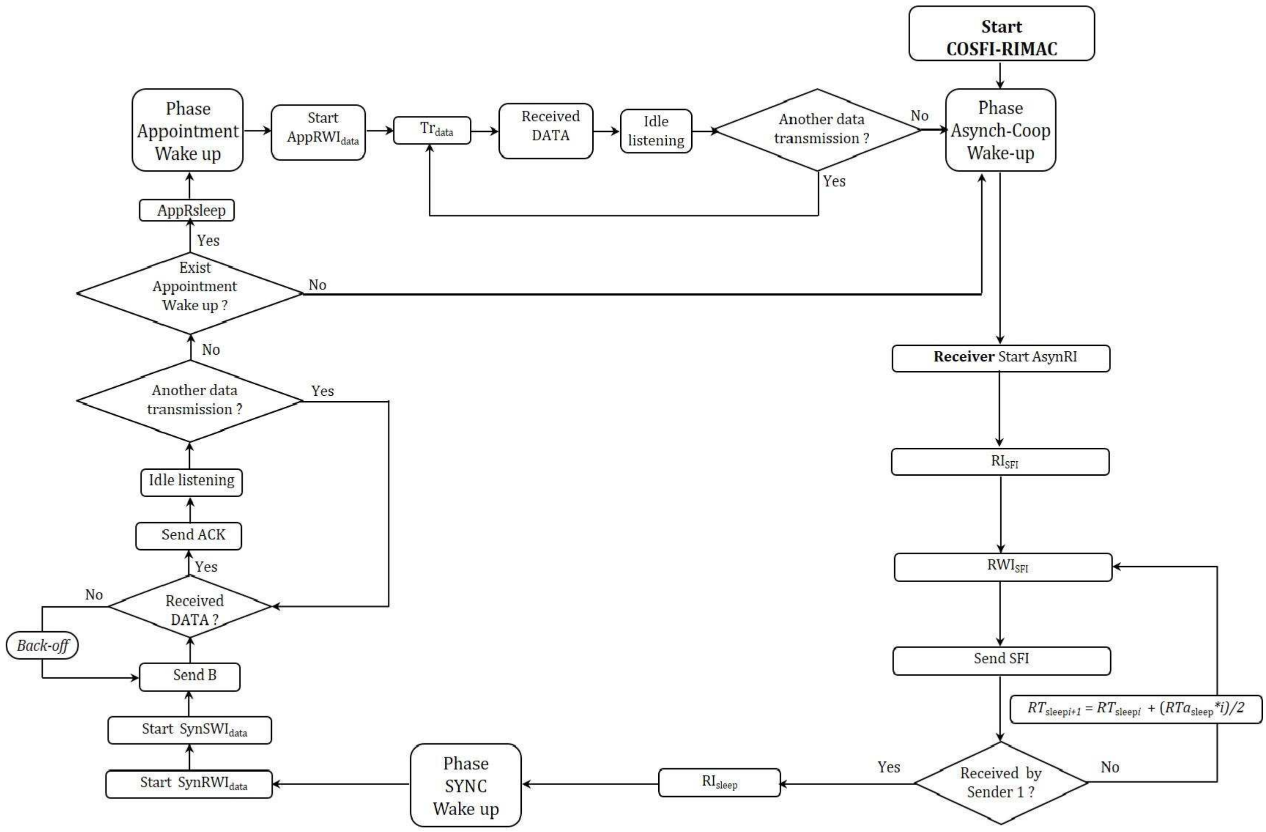


Figure 3.COSFI-RIMAC operating diagram, RECEIVER SIDE.

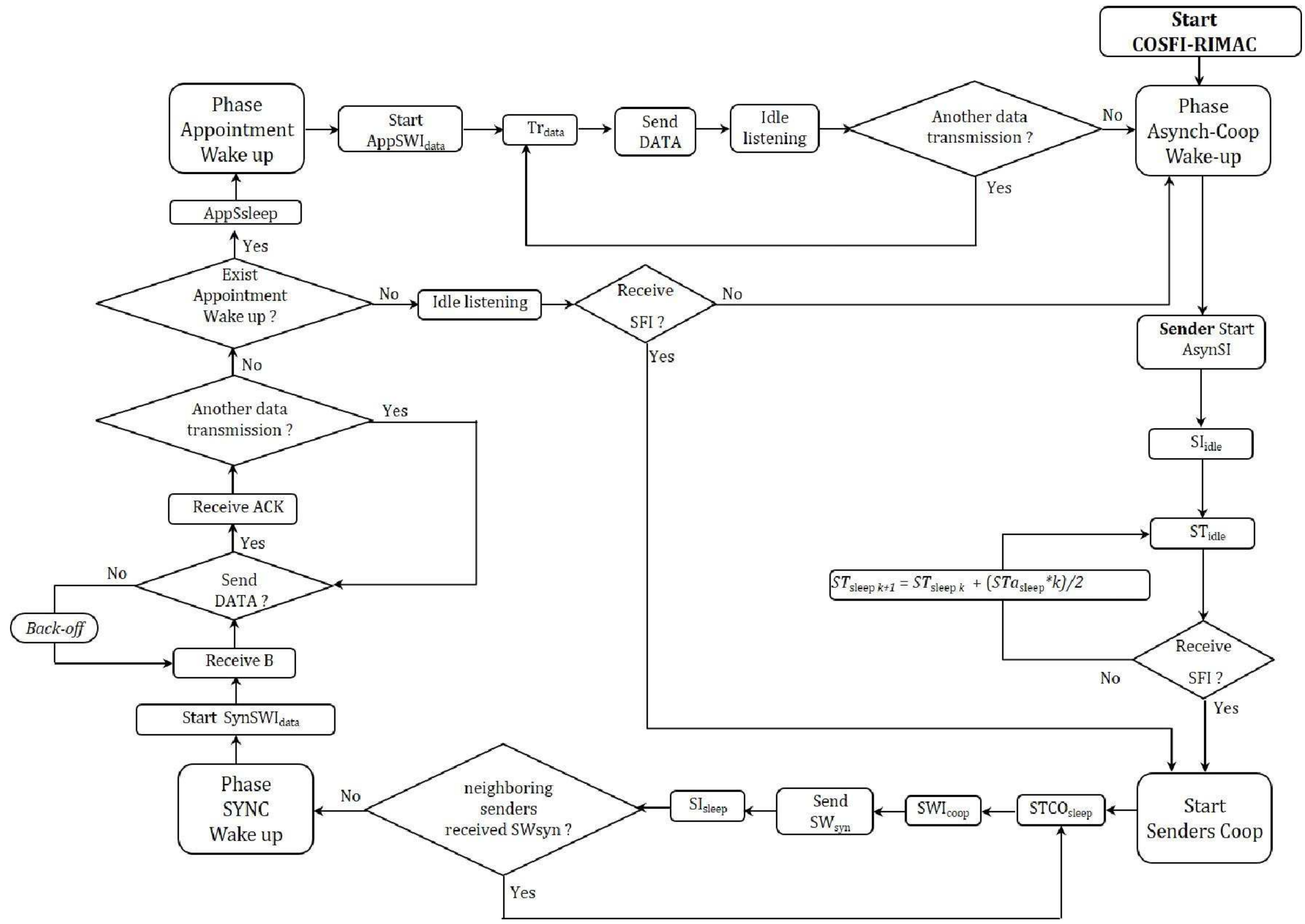


Figure 4.COSFI-RIMAC operating diagram, SENDER SIDE.

3.1.2 Phase Asynch-Coop Wake-up

In COSFI-RIMAC at the start, each node has its own 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 next SFI frame related to the previous TR_{sleep} interval time and calculated as shown in the following equation (1):

$$RT_{sleepj+1} = RT_{sleepj} + (RTa_{sleep}*j)/2 \qquad (1)$$

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 equations (2,3)

$$AsynRWI= (RWI_{SFI}) \qquad (2)$$

$$AsynRI_{sleep=} (RT_{sleep})+RI_{sleep}$$

$$AsynRI_{sleep}= (AsynRI_{SFI} - AsynRWI) + \Omega_{sleep} \qquad (3)$$

If a sender receives a SFI_j from the expected receiver, it will synchronize its data transmission wake-up. After the receiver goes into RI_{sleep} sleep mode in order to save power. When a node has a frame to send DATA, it listens to the channel for short time $ST_{idle}$and waits for an SFIj from a potential receiver, if it does not receive it returns to sleep mode in a dynamic interval STsleep calculatedas shown in Equation (4):

$$\begin{cases} k = 1 \text{ a max=n-1(4)} \\ ST_{\mathbf{sleep}k+1}= ST_{\mathbf{sleep}k} + (STa_{\mathbf{sleep}}*k)/2 \ ; \end{cases}$$

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

$$STCO_{sleep}= ST_{sleepj-1} + (STa_{sleep}*j) \qquad (5)$$

In this phase as indicated in algorithm 1, the COSFI-RIMAC protocol minimizes the idle listining of the sender nodes (AsynSIidle) which only becomes the sum of the ST_{idle}, as indicated in equations (6), which gives the majority of the senders time go into sleep mode $AsynSI_{sleep}$ as shown in equation (7).

$$AsynST_{idle}= (ST_{idle}) \quad (6)$$

$$AsynSI_{sleep}= (ST_{sleep}) + STCO_{sleep} + SI_{sleep} \quad (7)$$

A formal description of the Phase Asynch-Coop Wake-up is presented in Algorithm 1.

Algorithm 1 Asynch-Coop Wake-up

Begin of Asynch-Coop Wake-up receiver
Begin of ASYNC Wake up of receiver
Initialization : i=1 ; k=1 ; $ST_{sleep\ 1}$; $RT_{sleep\ 1}$
Sender launch Asynch wakeup interval AsynSI
Start of SIidle
Sender S1 listens to the channel for short time STidle
Receiver R1 launch Asynch wakeup interval AsynRI
Start of RISFI
Receiver R1 wakeup RWISFI
Receiver R1 send SFI1
If *SFI* $_1$ *not Received by Sender* ***then***
initial sleep $ST_{sleep\ 1}$ *between* ST_{idle1} *and* ST_{idle2}
initial sleep $Rt_{sleep\ 1}$ *between* $\mathbf{RWI}_{SFI\ 1}$ *and* $\mathbf{RWI}_{SFI\ 2}$
Endif
DO
$RT_{sleep\,i+1} = RT_{sleep\,i} + (RTa_{sleep}*i)/2$
$ST_{sleep\,k+1} = ST_{sleep\,k} + (STa_{sleep}*k)/2$
ReceiversendSFI$_i$
UNTIL *(Sender Receive SFI) OR (end of* RI_{SFI}*)*
sender S1 synchronizes their alarm clock with receiver R1
receiversleepRI_{sleep}
End of ASYNC Wake up of receiver
Begin Coop senders
*Calculate STCOsleep= STsleep j-1 + (STasleep*j)*
Sender S1 sleepSTCOsleep
while *(sender* S_i *receive Swsyn)and (not end of SISFI)* ***do***
Sender S_i*wake up* SWI_{COOP}
Sender Si Send Swsyni to other neighboring senders
Sender S i switches to SIsleep sleep mode,
End while
End coop senders
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 equation (8).

$$SynWI_{data}= T_B + T_{data} + T_{ACK} \quad (8)$$

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 $AppR_{sleep}$ state. If there is a Wake up appointment, the sender S1 goes into the $AppS_{sleep}$ 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 Wake 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****$_{data}$interval*
3. *wakeup of receiver R1 and sender S1*
4. *sender S1 wait in T_B for beacon B*
5. ***if*** *sender S1 receive beacon from receiver R1* ***then***
6. *sender S1 send 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 sendbeacon$B_{backoff}$*
11. *Sender S1 receive$B_{backoff}$*
12. *Sender S1 send a DATA to receiver R1*
13. ***end while***
14. ***if*** *data packet received from receiver R1* ***then***
15. *send ACK in T_{ACK} to sender S1*
16. ***End if***
17. *receiver waitsfor another* transmission of *sender S1*
18. ***while*** *sender S1 there is one data to be transmitted* ***do***
19. *Sender S1 send a DATA2 in T_{data} to receiver R1*
20. *Receiver R1 receive DATA2*
21. *Receiver R1 send ACK in T_{ACK} to sender S1*
22. ***end while***
23. *Receiver go to AppRsleep state*
24. ***if*** *not exist Appointment Wake up* ***then***
25. *Sender S1 waitsa SFI for another receiver R2*
26. ***if*** *Sender S1receive SFI for another receiver R2* ***then***
27. *Sender S1 synchronize leur réveil avec le récepteur R2*
28. *Sender S1 go to SYNC Wake up data transmission phase,*
29. ***else*** *sender S1 go to AppSsleep state*
30. ***End if***
31. ***End if***
32. ***End of endof 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.Sender sends the next DATA (if it exists) after each ACK.If there is a 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 ofAppointment Wake up data transmission***
2. *Launch of AppWI$_{data}$*
3. *sender S1 receive beacon from receiver*
4. *sender S1 send a DATA in T_{data} to receiver R2*
5. *receiver R1 receive DATA*
6. *receiver R1 send ACK*
7. *receiver wait for another data for sender S1*
8. *sender S1 receive ACK*
9. ***while*** *sender S1 there is data to be transmitted* ***do***
10. *Sender S1 send a DATA to receiver R1*
11. *receiver R1 receive DATA*
12. *receiver R1 send ACK to sender S1*
13. ***end while***

if there is another Appointment Wake up data transmission ***then***
go to Appointment Wake data transmission phase,
else *sender S1* ***waitsa SFI for another*** *receiver l,*
if sender S1 *receive SFI* ***then***
sender i synchronize their wake-up with the receiver R2
sender S1 go to SYNC Wake up data transmission phase
end if
else*go to Asynch-Coop Wake-up phase.*
Endif
endof Appointment Wake up data transmission

3.2 Cooperation and Collision settlement

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

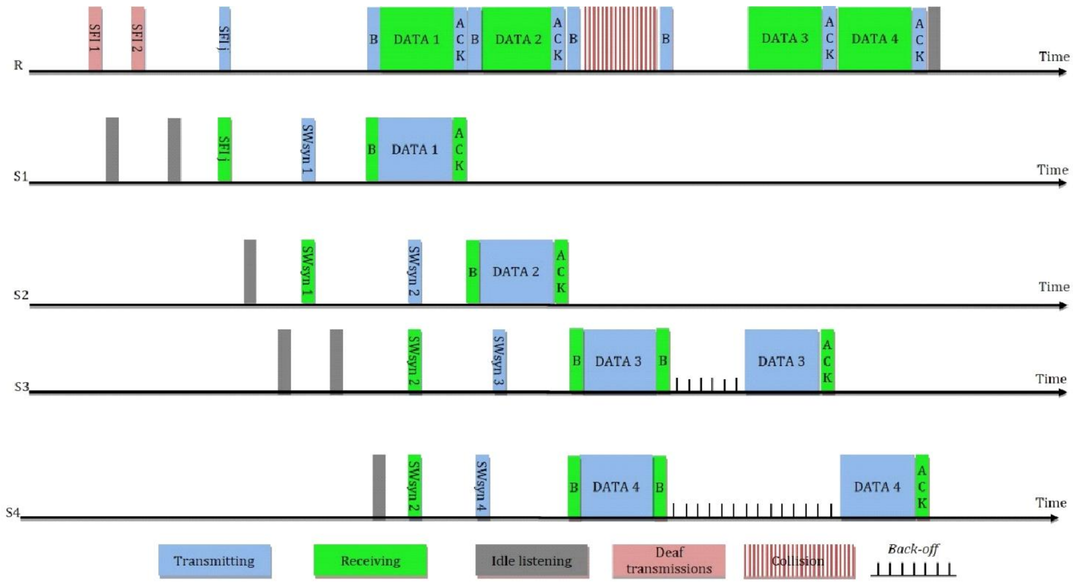


Figure 5. COSFI-RIMAC: Cooperation and solving the problem of collisions.

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 sleep state. The sender S2 is already in the SWI_{SFI} 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.

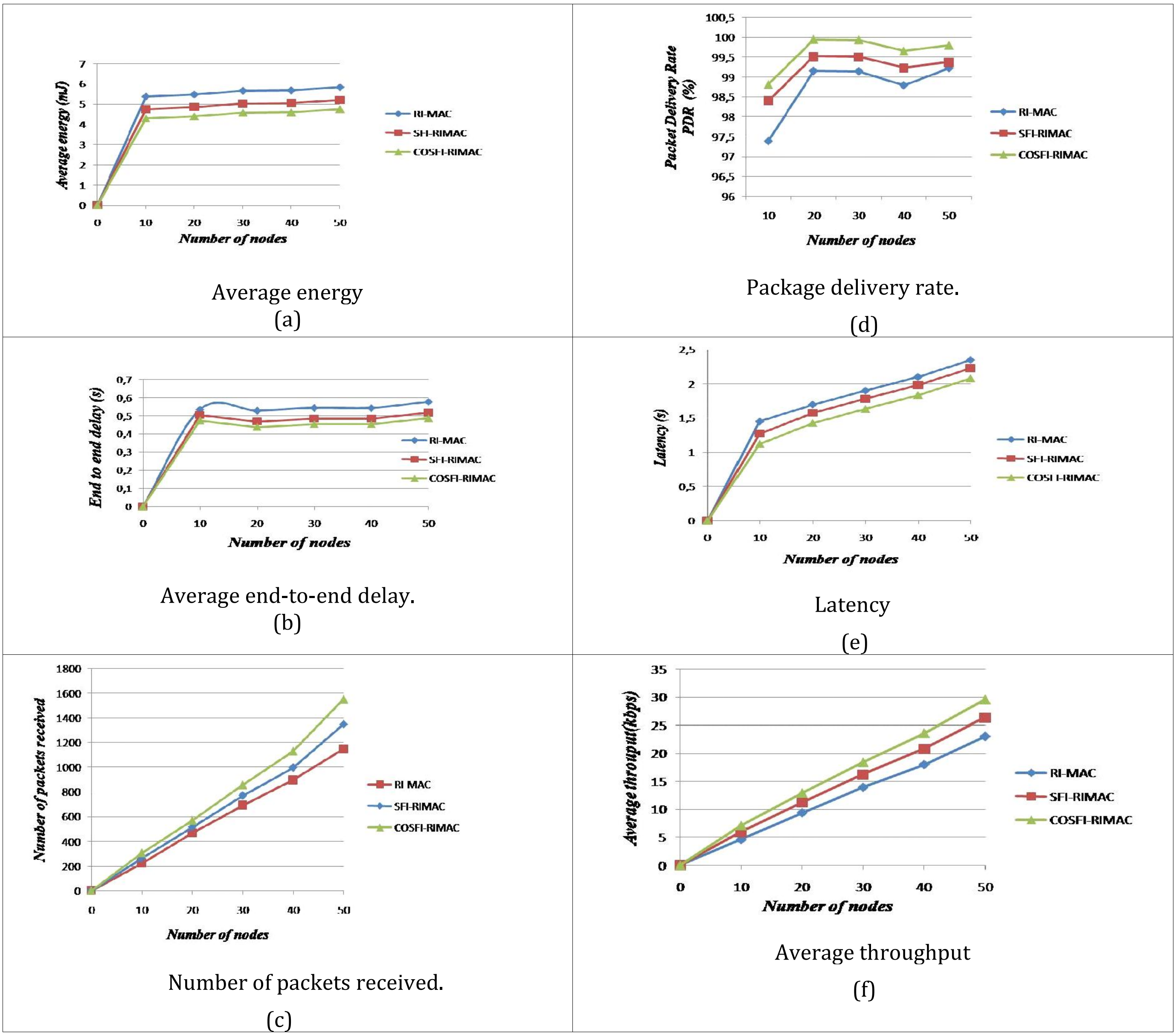
Simulation Parameter	
parameter	**Value**
MAC Protocol	RIMAC,SFI-RIMAC,COSFI-RIMAC
Node deployment aria	550 m X 550 m
Simulation Time	200s
Number of nodes	10,20,30,40,50
package size	2000 Byte
SFI size	12 Byte
B size	12 Byte

ACK size	12 Byte
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	ST_{idle}=0.5 ms
Random Sender Time sleep	$STra_{sleep}$Random[0.05s , 0.1s]
$RTra_{sleep}$	RT_{sleep}Random[0.05s , 0.1s]
Sender Time sleep initial	$ST_{sleep1=}$0.3s
Receiver Time sleep initial	$RT_{sleep1=}$0.2s
The sleep duration Ω	Ω=0 ,5 s

Table 2.COSFI-RIMAC parameters for simulation

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 6 (a) 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 reduce the power consumption compared to RI-MAC, as it reduces idle listening time and minimizes the number of collusions.



Average energy (a)

Package delivery rate. (d)

Average end-to-end delay. (b)

Latency (e)

Number of packets received. (c)

Average throughput (f)

Figure 6. Performance from different MAC protocols (RI-MAC, SFI-RIMAC, COSFI-RIMAC)

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 6 (b) 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 also improves COSFI-RIMAC.

Figure 6 (c) 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 6 (d) 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 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 6 (e) 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 increases 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 6 (f) 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 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 an innovative MAC protocol that uses the duty cycle mechanism to minimize the alternation between active / standby mode which consumes more power.Thus, it exploits the advantages of receiver initiated asynchronous protocols, which are much more efficient especially when network traffic is intense and / or subject to interference. COSFI-RIMAC minimizes the list of inactivity of sending nodes by breaking it down into small periods. In our protocol, the receivers make small wake-ups in the asynchronous wake-up phase, to synchronize the period of data transmission with the sending nodes. This synchronization and cooperation between transmitters minimizes the number of collisions.

We implemented COSFI-RIMAC in the NS2 simulator, the simulation results show that our proposal minimizes power consumption, average throughput, average end-to-end delays and latency, COSFI-RIMAC allows to transfer more than data and increases the packet delivery rate. Future work includes improving our protocol to support cooperation between receiving nodes.

DECLARATIONS

- Funding (information that explains whether and by whom the research was supported): 'Not applicable' for that section.
- Conflicts of interest/Competing interests (include appropriate disclosures): 'Not applicable' for that section.
- Availability of data and material (data transparency): 'Not applicable' for that section.
- Code availability (software application or custom code): 'Not applicable' for that section.
- Authors' contributions (optional: please review the submission guidelines from the journal whether statements are mandatory) : 'Not applicable' for that section.

REFERENCES

[1] Su,M, Chowdhury A, Hossain."Diferent Energy Saving Schemes in Wireless Sensor Networks: A Survey" Springer Science+Business Media, LLC, part of Springer Nature ,2020.

[2] Kochhar, A., et al. "Protocols for wireless sensor networks: A survey". Journal of Telecommunications and Information Technology, 7,18, 2018.

[3] D. Chen and Z. Tao, "An adaptive polling interval and short preamble media access control protocol for wirelesssensor networks," Frontiers of Computer Science in China, vol. 5, no. 3, 2011.

[4] Ajmi N, Helali A, Lorenz P, Mghaieth R. SPEECH-MAC: Special purpose energyefficient contention-based hybrid MAC protocol for WSN and Zigbeenetwork.Int J Commun Syst.e4637, 2020.

[5] Ye, W., Heidemann, J., &Estrin, D.“An energy-efficient MAC protocol for wireless sensor networks,” In INFO COM 2002. Twenty-FirstAnnual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, (Vol. 3, pp. 1567-1576),2002.

[6] Du, S., Saha, A. K., & Johnson, D. B.“RMAC: A routingenhanced duty-cycle MAC protocol for wireless sensor networks,” InIEEEINFOCOM 2007-26th IEEE International Conference on Computer Communications, (pp. 1478-1486), May 2007.

[7] Choi, L., Lee, S.H., Jun, J.A.: SPEED-MAC: Speedy and energy efficient data delivery MACprotocol for real-time sensor network applications. In: Proceeding of the International Conference on Communications (ICC), pp. 1–6, May 2010.

[8] Ozlem, D.I., Lodewijk, V.H., Pierre, J., Paul, H.: MC-LMAC: A multi-channel MAC protocolfor wireless sensor networks. In: Ad Hoc Networks, vol. 9. Elsevier ,2011.
[9] Zhao, Y.Z., Miao, C.Y., Ma, M.: An energy-efficient self-adaptive duty cycle MAC protocolfor traffic-dynamic wireless sensor networks. In: Wireless Personal Communications, pp.1287–1315,2012.
[10] J. Polastre, J. Hill, and et al., "Versatile low power media access for wireless sensor networks," in ACM SenSys'04, pp. 95–107, 2004.
[11] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in Algorithmic Aspects of Wireless Sensor Networks, pp. 18–31, Springer, 2004.
[12] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in ACM SenSys'06, pp. 307–320, ACM, 2006.
[13] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in Proc. ACM SenSys, 2008, pp. 1–14.
[14] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in IEEE INFOCOM'11, pp. 1305–1313, 2011.
[15] Yong, Y., Chow, C., Kanesan, J., & Ishii, H. EE-RI-MAC: An energy-efficient receiver-initiated asynchronous duty cycle MAC protocol for dynamic trafc loads in wireless sensor networks. International Journal of Physical Sciences, 6(11), 2633–2643, 2011.
[16] X. Wang, X. Zhang, G. Chen, and Q. Zhang. Opportunistic cooperation in low duty cycle wireless sensor networks. In IEEE ICC'10, pages 1–5,May 2010.
[17] Won, M.; Park, T.; Son, S.H. Asym-MAC: A MAC Protocol for Low-Power Duty-Cycled Wireless Sensor Networks with Asymmetric Links. IEEE Commun. Lett. 18, 809–812, 2014.
[18] Pegatoquet, A.; Le, T.N.; Magno, M. A Wake-Up Radio-Based MAC Protocol for Autonomous Wireless Sensor Networks. IEEE/ACM Trans. Netw., 27 56–70,2019.
[19] Hasan, M.M.; Karmaker, A.; Moni, S.S.; Alam, M.S. 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, 20–22, December 2016.
[20] Md.Hasan, M. M., Karmaker, A., Alam, M. S., and Craig, A., Minimizing the Adverse Effects of Asymmetric Links: A Novel Cooperative Asynchronous MAC Protocol for Wireless Sensor Networks. MDPI Sensors, May 2019.
[21] Lee, J.; Kim, S. EnRI-MAC: An enhanced receiver-initiated MAC protocol for various traffic types in wireless sensor networks. Wirel. Netw.1–10, 2018.
[22] M. L. Wymore, and D. Qiao. 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) 1, 860-869,2019.