

# Analysis and Design of a MAC Protocol for Wireless Sensor Networks with Periodic Monitoring Applications

Miguel A. Erazo, Yi Qian, Kejie Lu, and Domingo Rodríguez  
Department of Electrical and Computer Engineering  
University of Puerto Rico at Mayagüez  
Mayagüez, PR 00681

email: {miguel.erazo, yqian, lukejie, domingo}@ece.uprm.edu  
Phone: 787-458-4901, Fax: 787-833-3331

**Abstract** – In this paper, we present a new medium access control protocol (MAC) for wireless sensor networks with periodic monitoring applications. In particular, we design a listen-sleep time schedule for sensor nodes, and propose a path algorithm for route selection. We perform detailed analysis and simulations for the proposed protocol. Compared with some previous sensor network MAC protocols, we show that our new protocol can reduce energy consumption and packet delay, and at the same time, drastically reduce collisions.

**Keywords:** - wireless sensor networks, medium access control, energy efficiency, delay, collisions.

This research is supported by the National Science Foundation (NSF) under the grant CISE-CNS-0424546.

## I. INTRODUCTION

Wireless sensor networks consist of battery-operated sensor devices with computing, data processing, and communication components. Extensive research has been conducted in this area due to the wide range of applications of wireless sensor networks. Energy conservation is one of the most critical issues since batteries are the only energy source to power the sensor nodes.

Like in all shared-medium networks, medium access control (MAC) protocols enable the successful operation of the network. The MAC protocol in a wireless sensor network must achieve two goals. The first is the creation of the network infrastructure. Since thousands of sensor nodes are densely scattered in a sensor field, the MAC scheme must establish communication links for data transfer. This forms the basic infrastructure needed for wireless communication hop by hop and gives the sensor network self-organizing ability. The second objective is to fairly and efficiently share communication resources between sensor nodes.

Constraints on energy resources made researchers look for energy-efficient MAC protocols. Several MAC protocols for wireless sensor networks with an ultimate goal of increasing the network lifetime by conserving energy have been proposed in the literature.

Our goal in this paper is to develop a new MAC protocol for wireless sensor networks deployed for periodic monitoring, e.g., environmental monitoring. In our scheme, the Base Station (BS) starts and maintains synchronization. After nodes are synchronized, they only

wake up on specific times when a sample of a variable is taken.

The main contributions of this paper are design and analysis of the new MAC protocol for the wireless sensor networks. In particular:

- Design of a staggered listen-sleep time schedule where nodes only turn on their radios when a sample is taken with the objective of consuming the least possible amount of energy.
- Propose an algorithm to make sure that only one specific route (path) is active at a time. With this, we intend to lower delay, collisions and energy consumed for the sensor nodes.
- Comparison of the proposed new MAC performance with some of the previous MAC protocols in terms of energy consumption, delay and packet loss in simulations. We show that our new protocol reduces delay and collisions dramatically.

In the remainder of the paper, a brief survey of related works is presented in Section II. The description of our protocol is in Section III. Performance analysis and simulation results are presented in Section IV. Finally, the conclusions are discussed in Section V.

## II. RELATED WORK

Many protocols have been proposed for wireless sensor networks. Most of them aim to achieve low energy consumption in transmitting packets between nodes. These protocols also have the goals of low delay and minimum packet loss.

Erazo et al. [1] proposed SEA-MAC, a medium access control (MAC) protocol for wireless sensor networks (WSN) specialized for environmental monitoring applications. SEA-MAC focused on reducing energy consumption in environmental monitoring applications. Specifically, it reduces the duty cycle (DC) of nodes, lowering drastically idle listening.

Ye et al. [2] proposed S-MAC, a MAC protocol designed for WSN. S-MAC uses a few novel techniques to reduce energy consumption and support self-configuration. First, nodes form *virtual clusters* based on common sleep schedules to reduce control overhead and enable traffic-adaptive wake-up. Second, S-MAC uses in-

channel signaling to avoid overhearing unnecessary traffic. Finally, S-MAC applies *message passing* to reduce contention latency.

Ye et al. [3] introduced SCP-MAC, a protocol which uses Scheduled Channel Polling (SCP) to achieve more energy savings than those of protocols that use coordinated transmissions and listen periods. The contributions of SCP-MAC are the ultra low duty cycles it achieves and its capacity to adapt to variable traffic loads. Dam et al. [4] proposed T-MAC, a contention-based Medium Access Control protocol for wireless sensor networks. Lu et al. [5] proposed D-MAC, a protocol whose objective is to achieve very low latency.

Mainwaring et al. [6] provided an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements are developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management.

In this paper we propose a MAC protocol to achieve low energy consumption for periodic monitoring applications. Our protocol takes advantage of the fact that the traffic pattern is periodic to achieve low energy consumption levels. Furthermore, our approach focuses on reducing packet delay and collisions.

### III. PROTOCOL DESIGN OVERVIEW

#### A. Overview of Proposed Protocol

The objectives of the design of our new MAC protocol are the following: first, lower energy consumption by reducing duty cycle; second, reduce delay and collisions; and finally make synchronization simpler and unique in the whole network.

For the applications, we assume that it is desirable to sample physical variables periodically. In this way, we know in advance how many samples must be taken in an interval of time and the specific instants when they are taken. In our protocol, nodes periodically wake up when a sample is taken. In this way, nodes save a lot of energy in avoiding idle listening between two consecutive samples.

In a wide area large scale network it is likely that packets experience a huge delay, particularly when nodes are far away from the closest sink or Base Station (BS). For online monitoring, it is desirable the delay to be as low as possible. For that purpose, we use a staggered wakeup schedule as proposed in [5]. In that scheme, nodes on a path wake up sequentially to forward a packet to next hop, so sleep delay is reduced. Also, we did not use the RTS/CTS mechanism for transmitting packets whose destination is the BS. That reduces delay even more.

If the density of nodes within a network is high, collisions are very likely to occur. In order to reduce collisions, we propose an algorithm which computes time

schedules that sensor nodes follow in a way that collisions should not occur.

Synchronization for our protocol has been designed to be simple and robust. We intend to lower the frequency of synchronization losses in the network. Furthermore, the synchronization is unique for the whole network because only the Base Station starts and maintains synchronization while the other nodes only broadcast synchronization packets generated previously.

#### B. Time schedule

We can identify three main communication patterns in sensor network applications. The first involves local data exchange among nearby nodes. The second involves the dispatch of control and synchronization packets from the sink to nodes. The third and most significant traffic pattern in WSN is data gathering from sensor nodes to sink. We have proposed delay and collision reduction techniques for the third type of traffic pattern and created a separate active time slot for control packets.

Our protocol uses a schedule in which nodes only wake up when a sample from environment is taken. No periodic sleep/listen schedule will be necessary as it was proposed in protocols like S-MAC [2], T-MAC [4] or D-MAC [5]. Furthermore, it will not be necessary a short wake-up tone for senders to guarantee rendezvous as it was in SCP-MAC [3] because nodes know in advance when they have to turn on its radio to sample monitoring variables.

Time schedule of our protocol is shown in Figure 1. In our proposed time schedule we stagger the schedule to lower the delay in our network in similar way as proposed in [5]. Furthermore, we design our time schedule in such a way that all data gathered from sensor nodes be delivered to the sink in just one active period, i.e. a node would only wakeup once to receive and transmit packets destined to the sink.

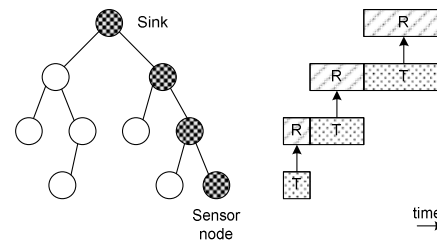


Fig. 1. Proposed time schedule for wireless sensor network

Figure 1 shows that as a node approaches the sink, its receiving and transmitting periods get larger. This happens because the closer a node is to the sink, the more data it receives from other nodes and the more packets it has to relay. In our approach we do not consider aggregation [7], which would reduce even more the receiving and transmitting intervals.

The time schedule of a node in our protocol is shown in Figure 2. Within a period (T), a node has a slot to receive

packets ( $t_{RX}$ ), another to transmit packets ( $t_{TX}$ ) and another to sleep ( $t_{SP}$ ). We denote the duration of an active slot as  $t_{LP}$  which is the sum of  $t_{RX}$  and  $t_{TX}$ .

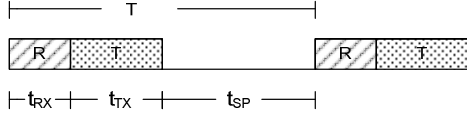


Fig. 2. Time schedule parameters

As stated above,  $t_{TX}$ ,  $t_{RX}$  and  $t_{SP}$  (schedule parameters) do not have to get the same value for all nodes within a route. Their values depend on the position of a node within a particular route. Figure 3 shows how we characterize a route within a network to determine the value of the schedule parameters.

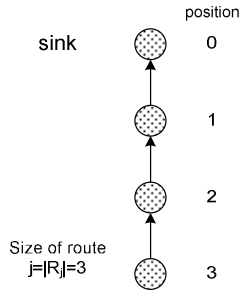


Fig. 3. Nodes are numbered according to its position from the sink

Values of schedule parameters for a particular node within a route are given by the following expressions:

$$t_{TX,i,j} = t_B L_{DATA} (|Rj| - i + 1) \quad (1)$$

$$t_{RX,i,j} = t_B L_{DATA} (|Rj| - i) \quad (2)$$

$$t_{SP,i,j} = T - t_{TX,i,j} - t_{RX,i,j} \quad (3)$$

Table I summarizes terms used in above expressions.

TABLE I  
List of symbols used for schedule parameters

Symbol	Meaning
$t_{TX,i,j}$	Duration of transmitting slot of node $i$ within route $j$
$t_{RX,i,j}$	Duration of receiving slot of node $i$ within route $j$
$t_{SP,i,j}$	Duration of sleeping slot of node $i$ within route $j$
$ Rj $	Size of route $j$
$t_B$	Time to Tx/Rx a byte
$L_{DATA}$	Data packet length

### C. Route partition

In order to lower delay even more and reduce collisions, we propose a mechanism that called *route partition*. Route partition is a mechanism through which only one route is active at a time. When only one route is active at a time and using the staggered time schedule described in B, our protocol nullifies the *hidden terminal* problem without using the RTS/CTS packet exchange or the virtual carrier sense [2]. Figure 4 shows an example of route partition.

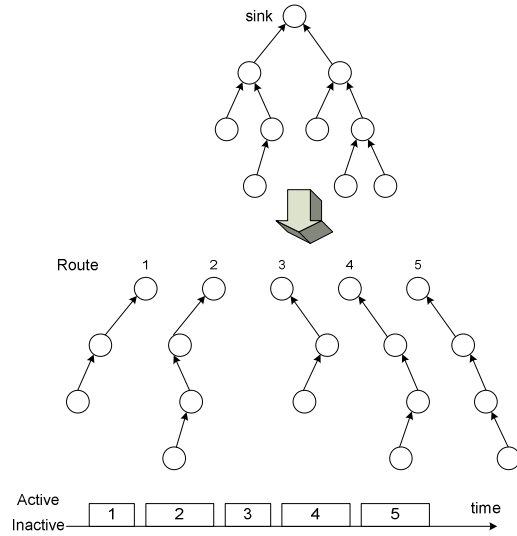


Fig. 4. Route partition mechanism

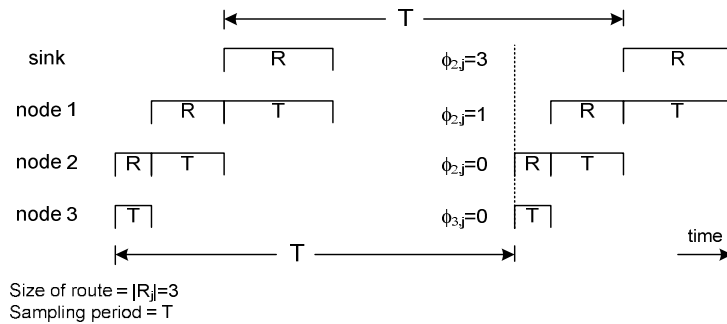


Fig. 5. Time offset in time schedule in a route of size 3

In our protocol, the sink computes the time schedule that every node in the network will follow. A node within the network can be involved in more than one route and consequently will follow more than one schedule.

Route partition has two states: *initialization* and *maintenance*. At the beginning of initialization state, sensor nodes dispatch route advertisement (RA) packets to the sink. As RA packets traverse the network, each packet is added the address of each node it goes through. In this way, each RA originated by a sensor node contains the complete route from the node to the sink. Once the sink has received RA packets from all sensor nodes, it runs our proposed algorithm to compute time schedules of all sensor nodes. Time schedules will be disseminated through schedule dissemination packets (SD). Upon

receiving SD packets, nodes will start to follow their respective time schedule. As part of the maintenance state, sensor nodes send RA packets and the sink broadcasts SD packets periodically to find possible changes.

Due to the staggered time schedule that we proposed in our protocol, nodes start their active periods at different times depending on their position within a route. As shown in Figure 1, nodes that are farther from the sink start transmitting earlier than those closer to the sink. We denote the offset in time between the start of transmission of furthest node from the sink in route  $j$  and the start of transmission of node  $i$  within the same route as  $\phi_{i,j}$ .

Figure 5 depicts the concept of offset within a route.

TABLE II  
Time schedule algorithm in sensor nodes

```

Receive (packet)
  if(packet.destination_address == myAddress)
    i=1
    while(rj[i] ≠ myAddress)
      i++
      myPositionfromSink = i
      tRX.myAddressj = tB L_DATA (sizeof(rj[])) - myPositionfromSink
      tTX.myAddressj = tB L_DATA (sizeof(rj[])) - myPositionfromSink + 1
      SamplingPeriod = Packet.T
       $\phi_{myPositionfromSink,j} = \text{compute\_my\_Offset}(rj[], \text{packet}, \phi_{1,j})$ 
      timeToInitiateActivePeriod = packet.TS - rxDelay + ( packet. $\phi_{1,j} - (\phi_{myPositionfromSink,j} - 1) * \text{sizeof}(rj[]) + (\phi_{myPositionfromSink,j} / 2) * (\phi_{myPositionfromSink,j} + 1) - 1) * (tB L_DATA)$ 
      if(I am not the last node in route)
        packet.sourceAddress = myAddress
        UnicastRDpacket (packet, sizeof(packet))
    return
  
```

TABLE III  
Path computing algorithm in the sink

```

Create used //object of a data structure
Create subpaths //object of a data structure
for i=1 to N //for all original paths
  (if used.find (Ri)!){ //path has not been used
    subpaths = findSubpaths(Ri, R) //find subpaths
    Ri.add(Ri) //route is added to R'
    used.add(Ri) //path has been used and now can be discarded for future use
    for j=1 to sizeof(subpaths)
      used.add(subpaths .returnNextSubpath) //discard subpaths for future use
  }
}

findSubpaths(Ri, R){
  Create subpaths
  for j=1 to N
    if(|Ri.Rj| < |Ri|) //possible subpath
      for k=1 to |Ri.Rj|
        equalElements = 0
        for l=1 to |Ri.Rj| //compare all elements of Ri with all those of Rj
          if rij == rli
            equalElements+=1
        if(equalElements==0)
          goto label 1
        subpaths.insert(Ri.Rj)
      label 1
  return subpaths
}
  
```

The sink, when broadcasting SD packets for a particular route, includes the sampling period (T), the time when a node starts to follow the time schedule of a particular route (TS), and the route size  $|R_j|$ . Then, every sensor node computes its own offset  $\phi_{i,j}$  according to the following expressions:

$$\phi_{i,j} = \phi_{1,j} - (i-1)|R_j| + \frac{i(i+1)}{2} - 1 \quad (4)$$

$$\phi_{1,j} = \frac{|R_j|^2}{2} - \frac{3|R_j|}{2} + 1 \quad (5)$$

Upon receiving SD packets, sensor nodes follow the algorithm proposed in Table II whereby time schedule can be computed for each route a sensor node belongs to. In Table II,  $r_j$  denotes a vector in which every element  $r_j[i]$  is the address of the sensor node that is  $i$  hops away from sink in route  $j$ .

As stated in previous paragraphs, the sink receives RA packets from all sensor nodes within the network. Consequently, after some time interval, the sink should have  $N$  packets. Then, the sink builds up a matrix of routes called  $R$  in which every column represents the path from a sensor node to the sink. A particular element of  $R$ ,  $r_{i,j}$  is the address of a node that is  $i-1$  nodes away from the sink that belongs to route  $j$ .

The necessary assumptions for the algorithm to partition routes, which will be run in the sink, are:

- Sensor nodes are fixed without mobility.
- Route to the sink are durable and are established by a suitable routing protocol.
- The sink is the only destination of data packets.

We denote a path  $j$  as  $P^j$  which is a set of all node addresses belonging to a column  $j$  of  $R$ . In this way,  $P^j = \{r_{i,j} | 1 \leq i \leq |R_j|, (\forall j)\}$ . A path  $P^k$  will be called a subpath of  $P^j$  if  $P^k \subset P^j$ . In this case we say that  $P^j$  contains  $P^k$  and only  $P^j$  should be broadcasted in a SD packet. Table III shows the proposed algorithm to detect subpaths and eliminate them for future broadcast in SD packets. Finally, the algorithm will output a matrix  $R'$  which will be used by the sink to broadcast all remaining useful routes in SD packets.

#### D. Energy analysis

We consider a network of  $j$  paths which corresponds to the number of columns of matrix  $R'$ , which is the output of the algorithm described in Table III. We compute the energy consumed in the whole network as the sum of the energy consumed in every path. The expression for energy consumption is:

$$E_T = \sum_{j=1}^{N'} \sum_{i=0}^{|R_j|+1} E_{i,j} \quad (6)$$

In expression (6),  $E_{i,j}$  is the energy consumed by node  $i$  in path  $j$ . The energy spent in each node is the energy consumed in transmitting, receiving, and sleeping. Each term can be expressed as the average power in that state multiplied by the time the node is on that state. Table IV summarizes the terms we use. The expression for the energy consumed in the whole network in a time interval  $T$  is:

$$\begin{aligned} E_{i,j} &= E_{TX,i,j} + E_{RX,i,j} + E_{SP,i,j} + E_{IL} \\ &= P_{TX} t_{TX,i,j} + P_{RX} t_{RX,i,j} + P_{SP} t_{SP,i,j} + P_L t_{ID,i,j} \\ E_{i,j} &= P_{TX} t_B (L_{ACK} + L_{DATA,i,j}) + \\ &P_{RX} t_B (L_{ACK} + L_{DATA,i-1,j}) + \\ &P_{SP} t_{SP,i,j} + P_L t_{IL,i,j} \end{aligned} \quad (7)$$

TABLE IV  
List of symbols used in energy analysis

Symbol	Meaning
$P_{tx}$	Power in transmitting
$P_{rx}$	Power in receiving
$P_{listen}$	Power in listening
$P_{sleep}$	Power in sleeping
$t_B$	Time to Tx/Rx a byte
$T_{data}$	Data packet period
$T_{sync}$	Synchronization packet period
$r_{data}$	Data packet rate ( $1/T_{data}$ )
$r_{sync}$	Synchronization packet rate ( $1/T_{sync}$ )
$L_{data}$	Data packet length
$L_{sync}$	Synchronization packet length
$t_{tx}$	Average time a node is transmitting
$t_{rx}$	Average time a node is receiving
$t_{cs}$	Average time a node is on carrier sense
$t_{sleep}$	Average time a node is sleeping

Replacing equations (1) and (2) into (7) we get:

$$\begin{aligned} E_{i,j} &= P_{TX} t_B (L_{ACK} + L_{DATA}(|R_j| - i + 1)) + \\ &P_{RX} t_B (L_{ACK} + L_{DATA}(|R_j| - i)) + \\ &P_{SP} t_{SP,i,j} + P_L t_{ID,i,j} \end{aligned} \quad (8)$$

Equation (8) gives the energy consumption in any sensor node. We also have the energy consumption for the sink and for nodes that are most distant from the sink in every path:

$$E_{\text{sink}} = P_{TX} L_{ACK} t_B + P_L t_{ID,i,j} + P_{RX} t_B L_{DATA} |R_j| \quad (9)$$

$$E_{|R_j|} = P_{RX} L_{ACK} t_B + P_L t_{ID,i,j} + P_{TX} t_B L_{DATA} \quad (10)$$

The sleeping time ( $t_{SP,i,j}$ ) in sensor nodes is not constant and depends on the position of the nodes within a path and the number of paths a sensor node belongs to within the network. However,  $P_{SL}$  is significantly less than  $P_{RX}$  or  $P_{TX}$  and consequently we consider  $E_{SP,i,j}$  negligible for our

purposes. Replacing equations (8), (9) and (10) into (6) we get the following expression.

$$E_T = \sum_{j=1}^{N'} \left( \left( \sum_{i=1}^{|Rj|-1} E_{i,j} \right) + E_{|Rj|} + E_{\text{sink}} \right) \\ = \sum_{j=1}^{N'} \left( \begin{aligned} &P_{TX} t_B (L_{ACK} |Rj| + L_{DATA} |Rj|^2) + \\ &P_{RX} t_B (L_{ACK} |Rj| + L_{DATA} |Rj|^2) + \\ &E_{IL} (|Rj| + 1) - \\ &t_B L_{DATA} (P_{TX} + P_{RX}) (|Rj| - 1) \end{aligned} \right) \quad (11)$$

Taking average on both sides we finally get:

$$E[E_T] = N \left( L_{ACK} \overline{|Rj|} (P_{TX} + P_{RX}) + \right. \\ \left. L_{DATA} t_B (\sigma_{|Rj|}^2 + \overline{|Rj|}^2 - \overline{|Rj|} + 1) (P_{RX} + P_{TX}) + \right. \\ \left. E_L (|Rj|^2 + 1) \right) \quad (12)$$

#### E. The synchronization

The sink is the only node that can start and maintain synchronization while the other nodes only disseminate synchronization in a multihop environment. This type of synchronization saves energy because every node only follows one global time schedule. The sink dispatches synchronization packets at a synchronization rate  $r_{\text{SYNC}}$ . To synchronize, sensor nodes do not follow the staggered time schedule proposed to cope with third type of communication traffic described briefly in subsection B. Instead, nodes set their time offset to 0 and disseminate synchronization packets from the sink to the furthest node. Once sensor nodes have received the synchronization packet and have successfully relayed it to the next node within a path, they go to sleep. Figure 6 shows the proposed time schedule for synchronization.

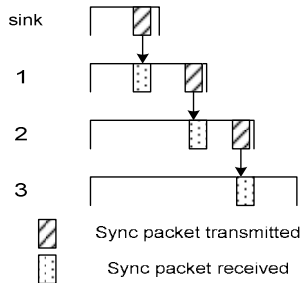


Fig. 6. Synchronization time schedule

## IV. EXPERIMENTAL RESULTS

### A. Results of proposed protocol

We have tested our protocol by simulations using the network simulator ns-2 [8]. In the simulation setup, we use 10 nodes, where the sink is placed in the middle of the

configurations and other 9 nodes are placed around the sink in a random location. We use AODV routing protocol [9]. Figure 7 shows our simulation configuration.

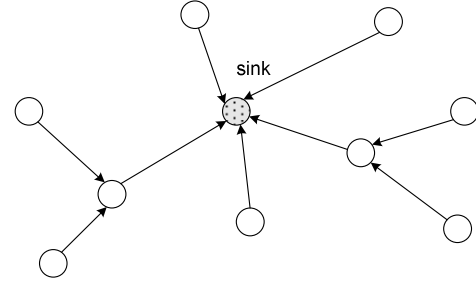


Fig. 7. Simulation configuration

Figures 8, 9 and 10 show the results of simulation for our proposed protocol in terms of energy consumption, packet delay and packet loss rate. The transmission interval is the time between transmissions of sensor nodes.

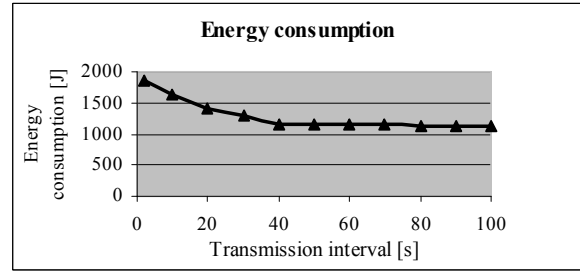


Fig. 8. Energy consumption

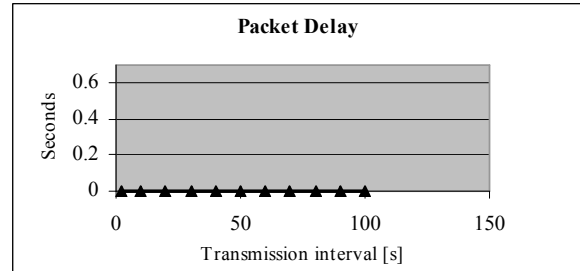


Fig. 9. Packet delay

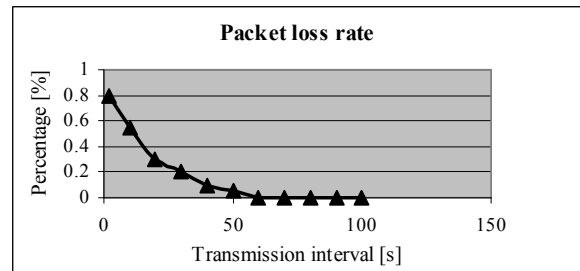


Fig. 10. Packet loss rate

As expected, the energy consumption increases as the transmission interval decreases, as shown in Figure 8. Figure 9 shows that the delay of data packets in our protocol remains practically constant for different values of the transmission interval. Finally, Figure 10 shows that

a low percentage of data packets are actually dropped for transmission intervals less than 50 seconds. However, the packet loss rate is never more than 1%.

### B. Performance comparison

Figures 11, 12 and 13 show the results of our protocol performance comparing with S-MAC.

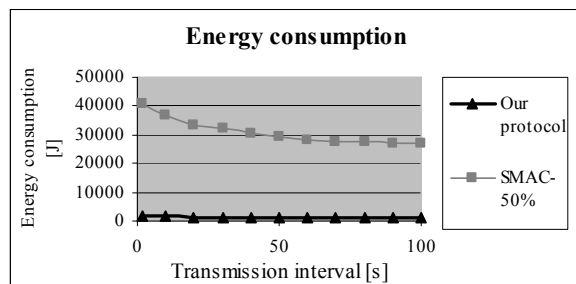


Fig. 11. Energy performance comparison

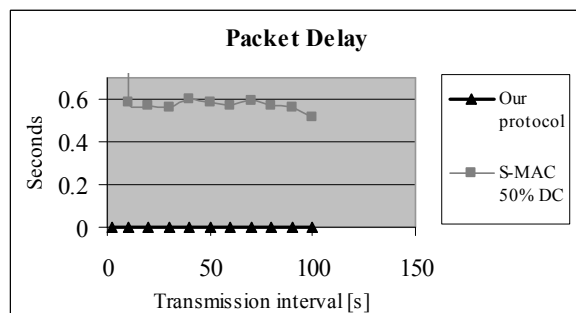


Fig. 12. Packet delay comparison

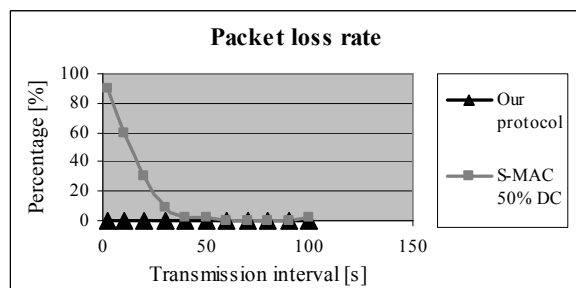


Fig. 13. Packet loss rate performance comparison

The results show that our protocol outperforms S-MAC, in terms of energy consumption. Furthermore, our protocol exhibits constant packet delay and outperforms S-MAC. Finally, the packet loss rate of our protocol increases slightly as transmission interval decreases, while the packet loss rate of S-MAC increases dramatically.

## V. CONCLUSIONS

In this paper we have proposed a new MAC protocol to reduce energy consumption, delay and collisions for periodic monitoring applications. Our protocol reduces drastically the energy consumption by making nodes to wake up only when a sample is taken. In this way, idle

listening is reduced. Also, synchronization has been made unique in the whole network in order to save energy by making nodes to follow only one source of synchronization. Simulation results show that our protocol achieves better energy performance, packet delay and loss than those of S-MAC.

## REFERENCES

- [1] M. Erazo, Y. Qian, "SEA-MAC: Simple Energy Aware MAC Protocol for Wireless Sensor Networks for Environmental Monitoring", Proceedings of ISWPC'2007, San Juan, PR, February 5-7, 2007.
- [2] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, Volume: 12, Issue: 3, Pages:493 - 506, June 2004.
- [3] W. Ye, F. Silva, J. Heidemann, "Ultra-Low Duty Cycle MAC with scheduled Channel Polling", ACM SenSys 2006, November, 2006.
- [4] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks", Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, pages 171-180, Los Angeles, California, USA, November 2003.
- [5] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks", Proceedings of IPDPS'2004, April 2004.
- [6] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", First ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA, September 28, 2002.
- [7] Z. Li, and B. Li, "Improving throughput in multihop wireless networks", IEEE Transactions on Vehicular Technology, Vol.55, No.3, pp.762-773, May 2006.
- [8] The ns-2 simulator: <http://www.isi.edu/nsnam/ns/>
- [9] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.