On the Improvement of Receiver-Initiated MAC Protocols for WSNs by Applying Scheduling

Cristina Cano and David Malone
Hamilton Institute
National University of Ireland, Maynooth
Co. Kildare, Ireland
Email: {cristina.cano,david.malone}@nuim.ie

Boris Bellalta and Jaume Barceló
Dept. de Tecnologies de la Informació i Comunicacions (DTIC)
Universitat Pompeu Fabra
Barcelona, Spain
Email: {boris.bellalta,jaume.barcelo}@upf.edu

***Abstract*—The two main drawbacks of receiver-initiated Medium Access Control (MAC) protocols for Wireless Sensor Networks (WSNs) are that *i)* they require all nodes to send a beacon each time they wake up, and that *ii)* broadcast traffic is not efficiently supported. In this work, we propose addressing these limitations by extending receiver-initiated MAC protocols with scheduling, i.e, coordinating sensor nodes to wake up at nearly the same instant. Following this approach, only one sensor node in the neighborhood sends a beacon per wake-up period and, as all nodes are awake at the same time, broadcast transmissions are naturally supported. A distributed learning technique is used to establish the order of beacon transmissions. We present the protocol description and the time to convergence when a fully connected network is considered.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are formed by low-capability devices that obtain data from the environment and communicate it wirelessly to a central device, known as a sink. Energy conservation is a major issue in this kind of networks due to the fact that sensor devices are battery-operated [1]. The Medium Access Control (MAC) layer is mainly responsible for switching the transceiver, the most energy consuming component, between transmission, reception, idle and sleep modes. Among these, the sleep mode is the one with the lowest energy consumption. Therefore, the main goal is to keep the transceiver in sleep mode for the largest possible fraction of time. However, coordination between a receiver and a transmitter is needed as they both need to be awake at the same time to communicate.

A well-known approach to coordinate sensor nodes is preamble sampling, in which the sender transmits a long preamble before sending the data [1]. The receiver samples the channel periodically and remains awake to receive data if activity is detected. One extension of this approach is Scheduled Channel Polling (SCP)-MAC [2] that exploits synchronization information to wake up nodes at approximately the same instant to sample the channel. A tone is sent before the data transmission to account for the synchronization error.

Another approach is the receiver-initiated RI-MAC protocol [3]. In contrast to preamble sampling, RI-MAC requires the transmitter to wait until the receiver wakes up to start the transmission. After the reception of a beacon from the receiver, the sender can start sending data. This protocol provides good performance for moderate traffic loads and is interesting for newer transceivers, in which the power consumption in transmission mode is lower than that consumed in reception mode [4]. Beacons can also be used to coordinate sensor nodes by communicating useful information. The two most important drawbacks of this approach are the fact that each sensor must send a beacon each time it wakes up and the inefficient handling of broadcast traffic. To broadcast a packet in RI-MAC it is necessary to send a separate packet to each of the receivers as they do not wake up simultaneously.

In this work, a new protocol that addresses the main limitations of receiving-initiated approaches is presented. The proposed protocol applies scheduling to RI-MAC in a similar manner to SCP-MAC by making use of a distributed learning technique for collision-free operation of beacons. We expect the proposed protocol to reduce energy consumption compared to RI-MAC when broadcast traffic is considered. Also, energy gains are expected to be obtained compared to SCP-MAC in newer transceivers due to the reduced time to send a beacon vs. the time to sample the channel. The presented approach is a completely new paradigm for coordinating sensor nodes and it does not require a topology control protocol or a central controller, as it is completely decentralised.

II. RI-MAC WITH SCHEDULED WAKE-UP INSTANTS

In this section we describe the challenges involved in terms of synchronization, creation of the schedule and schedule adaptation, and how they are solved considering a fully connected network. Results of the time to convergence are also presented.

A. General Overview

We propose enhancing receiver-initiated protocols so that all nodes wake up at approximately the same instant. In each wake-up period (time at which nodes wake up per cycle), only one node transmits a beacon. One evident advantage of this scheme is the support of broadcast traffic. Having all nodes active at the same time removes the need to send repetitions of a message to all recipients. Moreover, the channel is expected to saturate more gradually as fewer beacons are sent. Fig. 1 shows the proposed approach compared to RI-MAC and SCP-MAC basic behaviors in a 3-node fully connected network.

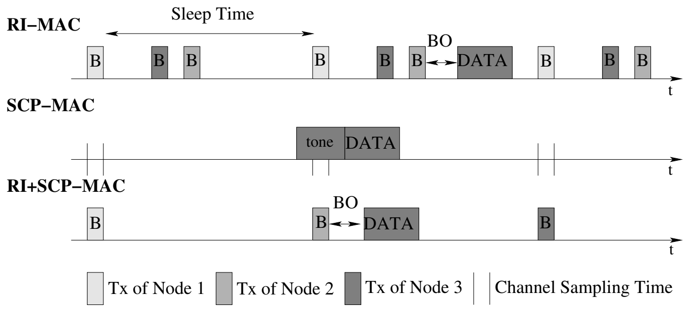

Fig. 1. Example of RI-MAC, SCP-MAC and RI+SCP-MAC behavior

As far as the authors know, this is the first work that proposes coordinating sensor nodes to wake-up at approximately the same instant considering receiver-initiated communication. In order to make the protocol to work, sensor nodes have to coordinate themselves to establish an order for beacon transmissions (only one node must transmit a beacon at a given wake-up period). A schedule of transmissions that is repeated in cycles can be defined, where the schedule length is the number of wake-up times per cycle. It seems reasonable that when a node has already transmitted a beacon in a given wake-up time without collision it continues transmitting in the same wake-up period to avoid further collisions. A node can randomly pick a wake-up period in the schedule and remain transmitting in the same instant in the next cycles if its transmission is successful. This idea is known as distributed learning for collision-free operation and has been studied for Wireless Local Area Networks (WLANs) in [5], [6] and [7].

However, the use of these techniques for convergence of beacon transmissions to collision-free operation is not straightforward. The main problem is related to the broadcast nature of beacons. In WLANs, these techniques are applied to data messages. Therefore, nodes can rely on the reception of the acknowledgement to decide whether the message has been correctly received and so if the same slot can be used in the next cycle. This is not possible with beacons as they are broadcast. Nevertheless, the lack of acknowledgements in broadcast messages can be addressed by requiring each node to include in each transmitted beacon the information of the wake-up periods in which a correct reception or a collision has been observed, as well as wake-up periods seen as empty.

Another issue relates to the adaptation of the schedule length. Previous work in [5] and [6] consider the schedule length substantially larger than the number of participating devices, since the presence of some empty slots does not imply a significant performance penalty. In the case addressed here, the length of the schedule should be close to the number of nodes in order to keep the guard time to account for clock drift small.

B. Synchronization

In order to coordinate sensor nodes to wake-up at approximately the same instant, a certain degree of synchronization is required. This requirement can be addressed by including synchronization information in beacons. Note that, a similar mechanism is adopted in SCP-MAC [2]. While in SCP-MAC sensor nodes send a tone before data transmission to account for synchronization errors, in the approach considered here, sensor nodes will wake-up during a guard time before the expected beacon reception to account for clock drift. To reduce the synchronization error coming from other sources, a scheme such as MAC time-stamping can be used to eliminate synchronization errors due to channel access, transmission and reception [8].

C. Creation of the Schedule

As already mentioned, a distributed learning technique for collision-free operation is adopted for establishing the schedule of beacon transmissions in order to make only one node to transmit the beacon at a given wake-up period. The protocol selected for that purpose is the Learning Zero Collision protocol (L-ZC) [7]. Using L-ZC, a node keeps track of the free slots in the schedule. After a successful transmission, a node continues transmitting in the same slot. However, after a collision, it changes to one of the slots seen as free in the last schedule with $(1-\gamma)$ probability and remains in the same slot with probability γ, where γ is a control parameter.

The selection of L-ZC is motivated by its fast convergence time. However, it relies on previous information of the occupied, collision and empty slots. Since we have chosen to include this information in the beacons, L-ZC is a good option for the problem addressed here. Beacons will then include the schedule length (that we redefine as the number of wake-up periods in a cycle) and the wake-up periods seen as free, occupied and the ones that resulted in a collision. This allows a node to both, know the available wake-up times of the schedule and realize whether its previous beacon transmission was correctly received. The adaptation of this protocol to schedule beacon transmissions in a fully connected network is as follows:

1) A node i will listen to the channel during CT_{p}, with C being the initial length of the schedule and T_{p} the period between wake-up instants. The node moves to step 2) if no beacon is received and to step 3) otherwise.
2) The node assumes that it is the first node in the network. Then, it selects a wake-up time in which to send the beacon uniformly in $\{1, 2, ..., C\}$. Move to step 4).
3) The node knows from the beacons received the current length of the schedule $C^{(i)}$ and the remaining idle wake-up periods $k_{\text{free}}^{(i)}$ (wake-up periods seen as free by all its neighboring nodes). Then, it selects with equal probability one of the $k_{\text{free}}^{(i)}$ idle wake-up instants and sends a beacon in it. Move to step 4)
4) For all beacons received in the current schedule, the node checks whether its beacon transmission has been correctly received. If all neighboring nodes confirm the correct reception, it selects the same wake-up period again in the next schedule. Otherwise, it selects the same wake-up time with probability γ and chooses one of

the remaining empty wake-up periods with probability $(1-\gamma)/k_{\text{free}}^{(i)}$.

Data transmissions can still happen even if there is no beacon transmission or a collision of beacons occurs in the current wake-up period. Thus, it seems reasonable to assume that nodes are able to know their neighboring nodes by overhearing their data transmissions. Therefore, if all beacon transmissions collide (then, no feedback on previously transmitted beacons is received), a node can assume that it shares the wake-up period with another neighbor. Thus, in this case, the sensor node is still able to change the wake-up period in the same way as if a beacon not acknowledging its transmission was received.

D. Setting γ

The control parameter γ has a clear impact on the time to reach convergence. It was shown in [7] that for L-ZC this parameter has its optimal value at $\gamma = 1/(C-N+2)$, with N being the number of nodes in the network. In the case addressed here, sensor nodes are relying on the information from all their neighbors and not only on their view of the schedule. Therefore, changes in the free wake-up periods are not updated immediately by all sensor nodes. This has a small impact on convergence time and on the optimal value of γ. However, $\gamma = 1/(C-N+2)$ has been found to be a good value for this parameter also in the problem addressed here.

E. Schedule Adaptation

As previously stated, having a longer schedule than the one needed to accommodate all nodes results in an increased energy consumption mainly due to the increased guard time to account for clock drift. It also has an impact on the beacon length, as having more wake-up periods implies that extra, non-useful information has to be included in beacons.

Assuming a fully connected network, unitary schedule length adaptation as proposed in [7] is possible. Since all nodes have the same view of the schedule no schedule conflicts can occur. Sensor nodes will therefore, increase the schedule length by one after observing that all wake-up periods are occupied and reduce it in one when there are at least two empty wake-up periods. Since nodes include the schedule length in their beacons, all nodes are aware of a change in the schedule length of a neighbor and can, therefore, also trigger the schedule adaptation procedure.

F. Time to Convergence

A custom simulator has been used to evaluate the convergence procedure when all nodes join the network at the same time. Fig. 2 shows the average values and 95% confidence intervals of the number of schedules to convergence obtained from 1000 simulation runs with $\gamma = 1/(C-N+2)$. We observe that the number of schedules to converge is low (less than 11), even for challenging scenarios in which 60 nodes compete for 60 wake-up periods.

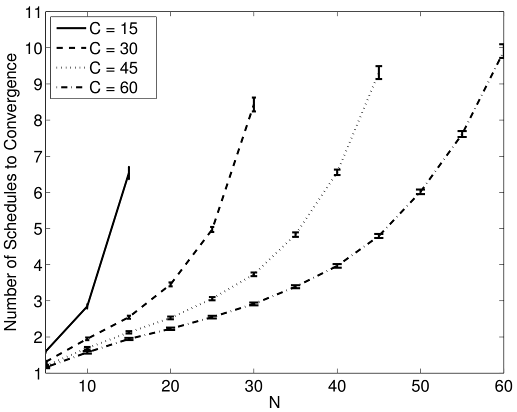

Fig. 2. Number of schedules to converge for different values of N and C

III. CONCLUSIONS

We propose to enhancing receiver-initiated MAC approaches for WSNs by applying scheduling with the final goal of reducing energy consumption, especially when broadcast traffic is considered. A new protocol to coordinate beacon transmissions in a fully connected network, as well as results of the time to convergence, have been presented. Further work to extend the protocol to multi-hop networks is needed as several challenges, in terms of creation of the schedule and optimal parameter configuration, appear when not all nodes are in mutual coverage range. To avoid collisions of beacons in a multi-hop network, nodes must select a wake-up period not used in the 2-hop neighborhood. Moreover, optimal parameter values will depend on the conditions observed by each node.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Government under Projects TEC2008-06055 (Plan Nacional I+D) and CSD2008-00010 (Consolider - Ingenio Program), by the Catalan Government (SGR2009#00617) and by the Science Foundation Ireland grant 08/SRC/I1403.

REFERENCES

[1] I. Akyildiz and M. Vuran, “Wireless Sensor Networks,” 2010.

[2] W. Ye, F. Silva, and J. Heidemann, “Ultra-low Duty Cycle MAC with Scheduled Channel Polling,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, 2006, p. 334.

[3] Y. Sun, O. Gurewitz, and D. Johnson, “RI-MAC: A Receiver-initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks,” in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (Sensys ’08)*, 2008, pp. 1–14.

[4] H. Cao, K. W. Parker, and A. Arora, “O-mac: a receiver centric power management protocol,” in *Proceedings of the 14th IEEE International Conference on Network Protocols, 2006. ICNP’06.* IEEE, 2006, pp. 311–320.

[5] J. Lee and J. Walrand, “Design and Analysis of an Asynchronous Zero Collision MAC Protocol,” *Arxiv preprint arXiv:0806.3542*, 2008.

[6] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, “Learning-BEB: Avoiding Collisions in WLAN,” in *Eunice Summer School*, 2008.

[7] M. Fang, D. Malone, K. Duffy, and D. Leith, “Decentralised Learning MACs for Collision-free Access in WLANs,” *Wireless Networks*, 2012.

[8] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys ’04)*. ACM, 2004, pp. 39–49.