A Receiver Oriented MAC Protocol for Wireless Sensor Networks

Luca Campelli, Antonio Capone, Matteo Cesana
Dipartimento di Elettronica e Informazione
Politecnico di Milano, Milan, Italy
{campelli, capone, cesana}@elet.polimi.it

Eylem Ekici
Department of Electrical and Computer Engineering
Ohio State University, Columbus OH, United States
ekici@ece.osu.edu

Abstract

In this paper we propose SPARE MAC, a TDMA based medium access control (MAC) scheme for data diffusion in Wireless Sensor Networks (WSNs). The rationale behind SPARE MAC is to spare energy through limiting the impact of idle listening and traffic overhearing. To this extent, SPARE MAC implements a distributed scheduling solution which assigns to each sensor specific radio resources (i.e., time slots) for reception, summarized as Reception Schedules (RS), and spreads the information of the assigned RS to neighboring sensors. A transmitting sensor can consequently become active in correspondence of the RS of its intended receiver only.

We analyze the performance of SPARE MAC in terms of throughput, power consumption, and data delivery delay both through analytical models and through detailed simulations. Moreover, we compare the performance of SPARE MAC against SMAC.

1 Introduction

Wireless Sensor Networks (WSNs) are composed of small sized battery operated network devices geared with processing capabilities, wireless communication interfaces, and sensing functionalities. With diminishing cost of communication devices, WSNs have emerged as an ideal solutions to a large number of applications in both civilian and military scenarios where large network infrastructure is required [1]. Just to mention some of the fields where WSNs can be (or are already) deployed, WSNs can be used for detection and tracking, environmental monitoring, industrial process monitoring, and tactical systems. Obviously, target applications determine WSN capabilities and properties. Similarly, applications also determine the choice and design of communication protocols.

For example, if a WSN is used in inaccessible or hostile areas, the network deployment is done in a random manner (e.g., dropping sensors from air crafts), which implies that the network should be able to self configure to provide a minimal backbone infrastructure. On the other hand, WSNs have been proposed also for surveillance and monitoring applications where the actual position of the sensors can be planned a priori, in contrast to the paradigm of sensors regarded as "smart dust" only. WSNs for the support of multimedia traffic [2] and for the monitoring of underground soil [3] are good representatives of this latter class of applications. Furthermore, many WSNs feature a *convergecast* communication paradigm [4] according to which the information must be delivered to a specific device (cluster head, data center, base station, gateway, etc. . .).

Regardless of the application, in many cases, battery replacement is impractical or impossible in WSNs. Therefore, both the design of WSN hardware as well as communication protocols must be done to maximize energy efficiency. At MAC layer, energy efficiency can be achieved through minimization of idle listening, retransmissions, unwanted overhearing, and over-emitting.

In this paper, we propose an energy efficient data centric MAC scheme for data collection in WSNs characterized by low sensor mobility and low-to-moderate traffic. Our solution is well-suited for those target scenarios where the collection of sporadic data through multi-hop paths is required and very high energy efficiency is of paramount importance to extend network lifetime. Examples of such applications include environmental monitoring and underground sensing of the soil condition (humidity, density, movements, etc. . .).

Our proposed MAC protocol with Slot Periodic Assignment for Reception (SPARE MAC) limits the energy waste due to packet overhearing, packet over-emitting, and idle listening. SPARE MAC is a Time Division Multiple Access (TDMA) based scheme, which implements a distributed scheduling solution that assigns time slots to each sensor for data reception and shares such assignments with neighboring nodes. A transmitting sensor becomes active during the receiving period of its intended receiver only, limiting aforementioned problems of overhearing, over-emitting, and idle listening.

The paper has the following organization: Section 2 describes SPARE MAC operation mode, highlighting the procedures for acquiring and distributing the Reception Sched-

1-4244-1455-5/07/$25.00 ©IEEE 2007

ules (RS). In Section 3, we provide analytical models to evaluate SPARE MAC data delivery delay and power consumption. Section 4 is dedicated to the performance evaluation of SPARE MAC under selected network scenarios. Section 5 comments on the differences between our proposal and previously published related works, and Section 6 concludes the paper.

2 SPARE MAC Description

2.1 Rationale

SPARE MAC implements a dynamic Time Division Multiple Access (TDMA) scheme, where all the nodes are time-synchronized. Time synchronization can be achieved via low-power synchronization receivers [5, 6] or through other proposed methods [7, 8, 9]. Furthermore, radio resources are organized into periodical frames partitioned into time slots as shown in Figure 1.

To understand the operation of SPARE MAC, we introduce the basic concept of *Reception Schedule* (RS). An RS is defined as a time slot (or group of time slots) during which a sensor becomes active for receiving data. In other words, an RS is a portion of the slotted time frame during which a sensor periodically wakes up for data reception. Based on the concept of RS, the basic SPARE MAC philosophy can be summarized with the following two requirements:

- Each node is assigned an RS
- Each node knows the RS of all its potential receivers

SPARE MAC implements a distributed scheduling solution which assigns an RS to each node and spreads the information of the assigned RS to neighboring nodes. The rationale behind SPARE MAC is to spare energy through limiting the impact of unnecessary transmissions, idle listening and traffic overhearing, which are the main causes of energy waste in WSN. In fact, each node running SPARE MAC theoretically becomes active only:

- during the RS of the receiver if it has traffic to send,
- during its own RS (limited idle listening),
- when it actually receives the traffic destined to itself only (limited overhearing).

Collisions occur when multiple senders transmit data to the same receiver in the same time slot. SPARE MAC does not prevent collisions from happening, but it reacts to a collision event by adopting proper countermeasures.

The RS assignment mechanism outlined above must be supported by a signalling protocol handling both the actual RS reservation and the RS exchange among neighboring nodes. In the next sections, we present the implementation details of different components of SPARE MAC. We start off by describing the slotted frame structure (2.2), then we discuss the implementation of the signalling phase, which relies upon the Wake-up Reliable Reservation ALOHA protocol (2.3), and finally we describe the details of the data transfer procedures (2.4).

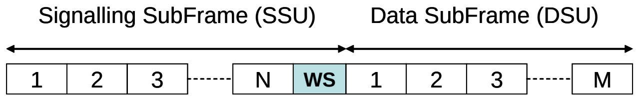

Figure 1. *Frame structure.*

2.2 Frame Structure and Signalling Support

SPARE MAC adopts the periodical frame structure shown in Figure 1. Each frame is divided into two subframes: a *Signalling SUbframe* (SSU) and a *Data SUbframe* (DSU) composed of *N* and *M* time slots, respectively. SSU is used for coordination purposes among neighboring nodes and for the exchange of all topological information needed to perform the scheduling of reception slots. Furthermore, the signalling part also includes a *Wake-up Slot* (WS) slot during which all stations are forced to stay active. Sensors willing to trigger signalling procedures send out a tone in the WS. Details on the usage of the WS and on the signalling procedures are given in Section 2.3. On the other hand, DSU contains data slots which are reserved for data reception. The dimension of the frame depends on the parameters *N* and *M*, which in turn depend on the topology of the network and on traffic requirements.

2.3 Wake-up Reliable Reservation ALOHA

The *Wake-up Reliable Reservation ALOHA* (WRR-ALOHA) protocol leverages the characteristics of the *Reliable Reservation ALOHA* protocol [13] with the concept of wake-up scheduling. The goals of the WRR-ALOHA protocol are the following:

- to handle the access to the network by new sensors
- to assign an RS to each sensor
- to let sensors exchange their RS assignment with one-hop neighbors

The first step for a sensor entering the network (after deployment and activation) is to acquire a slot in the SSU. To do so, the new sensor must gather information on the network topology and the current radio resource assignment from the already deployed sensors. To this end, the newcomer sends a tone in the wake-up slot, which is received by its neighboring nodes. Upon reception of this tone, neighboring nodes become active and send out a *Broadcast Signalling Packet* (BSP) in their own assigned slot during the following SSU. Each BSP contains, besides data and header information, a control field named *Frame Information* (FI). The FI is a vector with *N* entries specifying the status of each of the *N* slots in the previous SSU preceding the current transmission, as observed by the transmitting terminal itself. A slot is signalled as BUSY if a BSP has been correctly received from another terminal or transmitted by the terminal itself, otherwise is FREE. In the case of a BUSY slot, the identity of the transmitting terminal is reported.

Consequently, the FIs sent out by each sensor report the information on the activity of neighboring sensors as perceived in the previous SSU. Thus, a sensor receiving the FI from one of its neighbors gets aware of its neighbors'neighbors activity. Based on received FIs, the newcomer marks a slot in the SSU, say slot *k*, either as RESERVED or AVAILABLE according to the following:

Rule 1: *Slot k is RESERVED, if it is coded as BUSY in at least one of the FIs received; and AVAILABLE, otherwise.*

An AVAILABLE slot can be used by the newcomer. Upon accessing an AVAILABLE slot, the newcomer, say sensor *j*, will recognize in the next SSU the outcome of its access according to the following:

Rule 2 *The transmission is successful if the accessed slot is coded as "BUSY by sensor j" in all the received FIs; and failed, otherwise.*

If the access has been successful, the newcomer stops sending tones in the wake-up slot and goes to sleep. Otherwise, the newcomer attempts a new access in another AVAILABLE slot. Furthermore, it keeps sending tones in the wake-up slot until it has successfully acquired a slot in the SSU, thus forcing its neighbors to keep sending out their BSPs.

We emphasize here that the WRR-ALOHA protocol guarantees that any BSP transmission is fully one-hop reliable, i.e., all one-hop neighbors of the current transmitter correctly receives the transmitted BSP.

The very same BSP transmission procedure is used to assign RS to sensors and to exchange the RS assignments. In fact, each BSP, besides the FI and the transmitter's ID (see Figure 2), also contains information on the current RS, i.e., indicating the slots selected for reception by the BSP transmitter. Consequently, each sensor receiving BSPs from its neighbors can store its neighbors' RS, and choose an AVAILABLE RS for itself defined according to the following:

Rule 3: *An AVAILABLE RS must not overlap with any RS of one-hop neighbors.*

Rule 3 aims at avoiding those situations where two or more neighboring sensors[1] cannot communicate because they have chosen the very same slots for reception. We observe here that other policies of RS assignment can be implemented. For example, RS can be assigned also avoiding overlap with two-hop neighbors RSs.

Both the BSP and RS acquisition procedure outlined above are triggered whenever a new sensor wants to join the network. The RS acquisition and exchange is also triggered whenever any sensor needs to re-schedule its reception period.

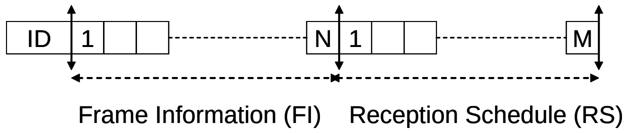

Figure 2. *Broadcast Signalling Packet format.*

Figure 3 shows the case of a new sensor entering the system (sensor 3). Sensor 3 waits for the next wake-up slot and sends out a tone which activates its neighbors (in frame 1). Neighbors 1 and 2 send out their BSPs in the time slot they have previously acquired (frame 2), further signalling their own RS (shaded slots in the DSU). The information in the BSPs helps the newcomer to choose its own broadcast slot in the SSU, and its own RS in the DSU. As for the broadcast slot, the newcomer randomly picks an AVAILABLE slot in the SSU according to ***Rule 1***, say slot 4 in the SSU. As for the RS choice, the newcomer randomly picks an AVAILABLE RS according to ***Rule 3***, say slot 1 in the DSU. The new sensor fires its own BSP in the following frame (frame 3), further indicating its own RS (shaded slot in the DSU). In frame 4, the new sensor checks the outcome of its access according to ***Rule 2***. Note that the newcomer keeps sending power in the WS to force all its neighbors to send their own BSPs until its slot acquisition has been acknowledged. After that, it goes to sleep.

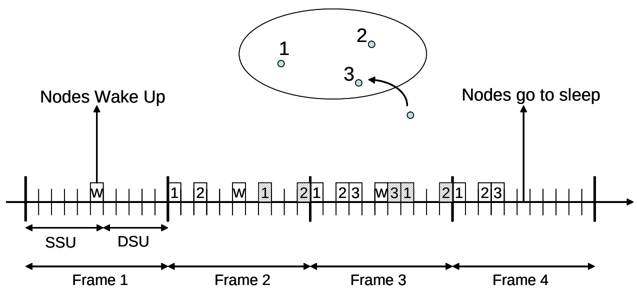

Figure 3. *BSP acquisition procedure triggered by the activation of a new sensor (sensor 3).*

[1] i.e., sensors within communication range

2.4 Data Transfer

Once a node has acquired a broadcast signalling slot according to the procedures explained in the previous section, it can access the radio resources of the DSU. To do so, it maintains a list of its one-hop neighbors with the the corresponding RS. If the node is a transmitter, i.e., it has data to transfer to a specific one-hop neighbor, it becomes active according to the neighbor's RS, transmits data and goes to sleep. On the receiver-side, each node wakes up according to its own RS to potentially receive data from its neighbors. We assume here that a node stays active throughout all its RS. Further optimizations in energy conservation are deferred to our future work.

Collisions may happen at a receiver when multiple sensors transmit on the same RS. To detect such collisions, receivers are required to switch on in the SSU following the DSU and to send out an explicit acknowledgement. A transmitter interprets the lack of an acknowledgment as the indication of collision. Collisions are resolved adopting a *Collision Resolution Algorithm* (CRA) based on binary exponential backoff. Each colliding station refrains from transmitting after a collision for a number of frames, i, computed as:

$$i = \begin{cases} random\,[1,\,2^k] & if\; k \le 10 \\ 2^{10} & otherwise, \end{cases} \qquad (1)$$

where k is the number of consecutive collisions experienced by the transmitted packet.

While SPARE MAC conserves energy by avoiding idle listening and overhearing, it does so at the expense of increased collision probability. At this point, we emphasize that SPARE MAC is efficient in those situations where data traffic is moderate-to-low on the average, which is the case in many WSN deployments. Under heavy traffic loads, the collision probability, hence the energy consumption, increases.

3 Protocol Analysis and Dimensioning Guidelines

In this section, we present an analytical model for evaluating the average delay experienced by a packet (Section 3.1) and the average consumed power (3.2) when assuming a single slot RS. Further on, we discuss on the configuration of the SPARE MAC parameters (3.3).

3.1 Delay Analysis

We assume that the total traffic received by a given node is a Poisson process with rate G *[packet/frame]*. Each transmitter sends packets in the first RS of the intended receiver without any initial delay. Within this framework, the total traffic on the channel, G^*, can be written as:

$$G^* = \frac{G}{1-P_c}, \qquad (2)$$

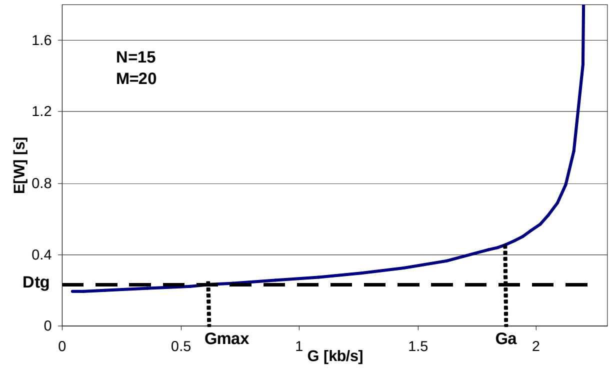

Figure 4. *One-hop delay (E[W]) versus offered traffic (G) when N=15 and M=20.*

where $P_c = 1 - e^{-G^*}$ is the probability for a packet to collide. The process of transmission/reception of a packet can be modelled as an M/G/1 queue with the service time distributed according to a random variable X with p.d.f. $f_X(x)$ (where m_X is the mean value and σ_x^2 the variance of X). We assume that X is an integer number representing the number of frames for a packet to be correctly received starting from its first transmission attempt.

Thus, the average waiting time $E[W]$, i.e., the average time an arriving packet has to wait before being received by the intended receiver is given by [10]:

$$E[W] = \left(\frac{\rho E[Z]}{1-\rho} + m_x - \frac{1}{2}\right) \times T_{frame}, \qquad (3)$$

where Z is the random variable representing the residual service time as seen by a packet entering the queue given that the server is busy, $\rho = m_X G$, and T_{frame} is the frame duration. The term $1/2$ is included to account for the average time between the packet arrival and the first RS which is half of a frame duration.

According to the renewal events theory, we can write:

$$E[Z] = \frac{m_X}{2} + \frac{\sigma_X^2}{2m_X}, \qquad (4)$$

By substituting Eq. (4) into Eq. (3), we can express the average waiting time as a function of mean value and variance of the service time, i.e., m_X and σ_X^2. Such quantities depend on traffic on the channel, which is driven by the specific collision resolution algorithms implemented by SPARE MAC. The derivation of closed-form expressions of m_X and σ_X^2 for a binary exponential backoff scheme is provided in the Appendix.

Furthermore, we observe that Eq. (3) defines a function which bounds the incoming traffic, G, to the one-hop delay, $E[W]$, in the case each sensor chooses a single-slot RS, $E[W] = f(G)$. In other words, it gives the observed delay if all traffic is transmitted using an RS made of a single slot. This result can be easily used to determine the minimum number of slots in the RS to match a given delay constraint. Suppose D_{tg} is the target average delay and $G_{max} = f^{-1}(D_{tg})$ the corresponding traffic value. If G_a is the actual offered traffic, the number of slots m to be used within the RS to match the constraint on the average delay can be computed as:

$$m = \lceil \frac{G_a}{G_{max}} \rceil \tag{5}$$

Consider the following example: Figure 4 shows the single-hop delay predicted by the model versus the channel traffic in the case $N = 15$ and $M = 20$. D_{tg} and G_{max} are respectively equal to 270 [ms] and 630 [bit/s]. Given the actual incoming traffic, $G_a = 1.8$ [kb/s], the number of slots m to be used within the RS can be computed using Eq. (5) as: $m = \lceil \frac{1800}{630} \rceil = 3$.

3.2 Energy Analysis

Energy consumption prediction plays an important role in designing solutions for WSNs. Here we present an average power consumption prediction procedure, which depends on the process representing the status of the sensors within the frames. The energy status of a sensor within one slot can be transmitting (tx), receiving (rx), sleeping (sl) or idle (id). The total energy consumed by a sensor within a frame is equal to the sum of the energy consumed in each slot of the frame.

Assume that the traffic generated by a given sensor towards a specific recipient and the total received traffic are distributed as Poisson point processes with parameters G_t and G_r respectively, measured in *[packets/frame]*. The outgoing traffic to a receiver, G_t^*, and the incoming traffic, G_r^*, are related to the respective offered traffic through Equation (2), i.e.,

$$G_r^* = \frac{G_r}{e^{-G_r^*}}, \quad G_t^* = \frac{G_t}{e^{-G_r^*}}. \tag{6}$$

If we consider a target sensor, the total energy consumed within a frame depends on whether the sensor will be receiving in a frame and/or transmitting to one or more recipients. Under the Poisson assumption for the incoming and outgoing traffic, we can define the following probabilities:

- probability for a sensor to be transmitting towards i recipients in the same DSU:

$$p_{tx}(i) = \binom{M-1}{i}(1 - e^{-G_t^*})^i e^{-G_t^*(M-1-i)} \tag{7}$$

  with $(0 \leq i \leq M-1)$.

- probability for a sensor to be receiving from $j \in \{0,1\}$ transmitters in the same DSU:

$$p_{rx}(0) = e^{-G_r^*} \quad p_{rx}(1) = 1 - p_{rx}(0) \tag{8}$$

- the joint probability for a sensor to be transmitting to i recipients and receiving from j transmitters in the same DSU:

$$p \, \langle tx(i) \, , \, rx(j) \rangle \; = \; p_{tx}(i) \; p_{rx}(j) \tag{9}$$

The average consumed energy per frame, E_f, comes from a weighted average of the energy consumed in transmitting, receiving, idle, and sleeping states. Thus, we can write:

$$E_f = \sum_{j=0}^{1} \sum_{i=0}^{M} p \, \langle tx(i) \, , \, rx(j) \rangle \; E(i,j), \tag{10}$$

where $E(i,j)$ is the energy consumed in a frame when transmitting to i recipients and receiving from j transmitters. Consequently, we can write:

$$\begin{aligned} E(i,j) = \quad & ie_s^{rx} + je_s^{tx} + (N-i-j)e_s^{sl} + \\ & + e_w^{id} + (M-i-j)e_d^{sl} + ie_d^{tx} + je_d^{rx} \end{aligned} \tag{11}$$

where e_x^y is the energy consumed in state y ($y \in$ [tx, rx, idle, sleep]) in a slot of type x ($x \in$ [signalling (s), data (d), wake-up (w)]). Finally, the average consumed power per frame, P_f, can be calculated as:

$$P_f = \frac{E_f}{T_{frame}}. \tag{12}$$

3.3 Frame Dimensioning

Dimensioning a SPARE MAC frame deals with finding optimum values for M and N considering the trade-off between capacity, energy consumption, and delivery delay. The value N is strictly bounded by the need of assigning a unique BSP to all sensors in two-hop clusters. Hence, N mainly depends on the topology of the specific WSN. In the network topology shown in Figure 5, the biggest two-hop neighborhood contains $S = 7$ sensors. Therefore, a minimum $N = 7$ is required.

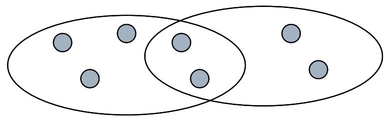

Figure 5. *Cluster clouds network topology.*

On the other hand, the value M is strictly related to the type of application traffic to be delivered. Roughly speaking, setting high M decreases the actual data rate offered at reception and increases the average delivery delay. On the other hand, a large M decreases the average energy consumption. Once N is set, the range in which M can be chosen is given by an interval [M_{min}, M_{max}], where M_{min} is strictly related to the network topology, and can be defined as the dimension of the biggest one-hop neighborhood in

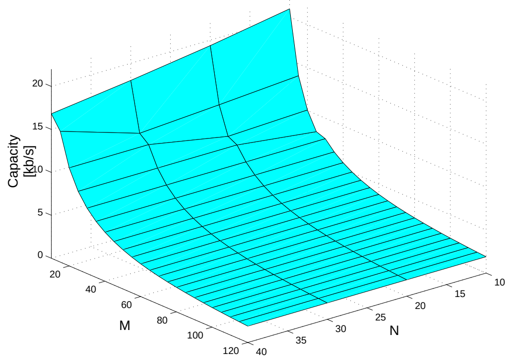

Figure 6. *Reception capacity versus N and M in the case of single slot RS. Data packets L = 512 bytes.*

the network. Referring again to Figure 5, the cardinality of the biggest one-hop cluster (the one on the left side) is equal to 5, thus $M_{min} = 5$. If M was equal to 4 two sensors would be forced to choose the same RS, thus they could not communicate.

On the other hand, M_{max} depends on the bandwidth requirement of the application. Having in mind the dimensioning guidelines of N and M, the data rate, R, of the channel obtained by assigning a single RS can be calculated as:

$$R = \frac{B_{RS}}{NT_{BSP}(N) + MT_{DP} + T_w} \qquad (13)$$

where B_{RS} is the RS dimension (in bits) and $T_{BSP}(N)$, T_{DP} and T_w, are the BSP duration, the RS duration (in seconds), and the wake up slot duration, respectively. Figure 6 shows the behavior of the reception capacity as a functions of the number of slots in the DSU, M, and the number of slot in the SSU, N.

4 Numerical Results

To test the performance of SPARE MAC, we conducted a simulation analysis in two different WSN topologies shown in Figure 7. In the first topology (Fig. 7.a), each sensor is within the transmission range of all the other $S-1$ sensors in the network, thus representing a fully connected network scenario. The second topology (Fig. 7.b) represents the case of *convergecast* traffic going from leaf sensors to a data sink.

The performance of SPARE MAC in each of these topologies are assessed using three performance metrics: the end-to-end throughput, the end-to-end delivery delay, and the consumed power. All the results presented in this section have been obtained using ns2.29 [12], running 50 simulations for each network configuration and averaging the results. The measured confidence index for all collected statistics is below 5% in 98% of all cases. Table 1 summarizes the standard setting of the simulation parameters.

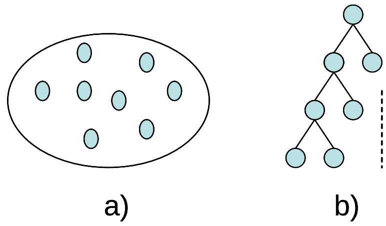

Figure 7. *Test topologies: fully connected cluster topology (a), tree topology (b)*

Table 1. *Standard setting of the simulation parameters.*

Parameter	Value
Simulation Run Length	1000 s
Bandwidth	250 kb/s
Data Slot	560 byte
Signalling Slot	50 byte
Wake-up Slot	9 byte
Packet Length	512 byte
TX Power	24 mW
RX Power	13.5 mW
Idle Power	13.5 mW
Sleep Power	5 μW

First, we validate the proposed energy and delay models by comparing them with simulation results in Section 4.1. In Section 4.2, we compare the performances of SPARE MAC and SMAC [14] in the *cluster* topology, whereas Section 4.3 studies the performance of SPARE MAC in a multi-hop network scenario where data traffic converges to a sink node.

4.1 Model Validation

Figure 8 shows the comparison between the average delay measured in the simulations and predicted by the model in Section 3.1 in the cluster topology with 10 sensors. As clear from the figure, the delay predicted by the model has the same tendency as the simulated one. The model slightly underestimates the actual delay mainly due to the simplified assumptions on the collision probability.

Regarding the power consumption model, Figure 9 compares the simulated values of the consumed power against the one predicted by the model in Section 3.2 in the same one-hop cluster topology as above. Here, the predicted and measured results match closely for the tested traffic loads.

The good match between simulation and analytical model has been observed also for different number of sensors within the cluster. The corresponding results are not reported here for the sake of brevity.

4.2 SPARE MAC vs SMAC

We also compared the performance of our proposed SPARE MAC protocol with SMAC, which is one of the

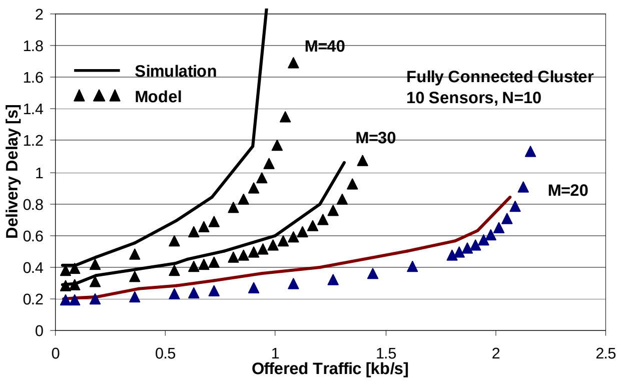

Figure 8. *Average delivery delay versus the offered traffic in the cluster topology with 10 sensors (RS=1). Delay model validation.*

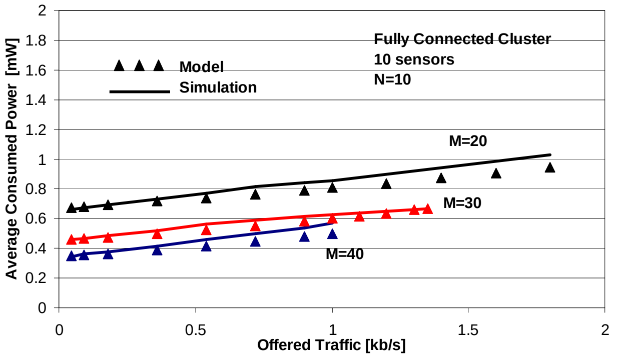

Figure 9. *Average power consumed versus the offered traffic in the cluster topology with 10 sensors (RS=1). Power consumption model validation.*

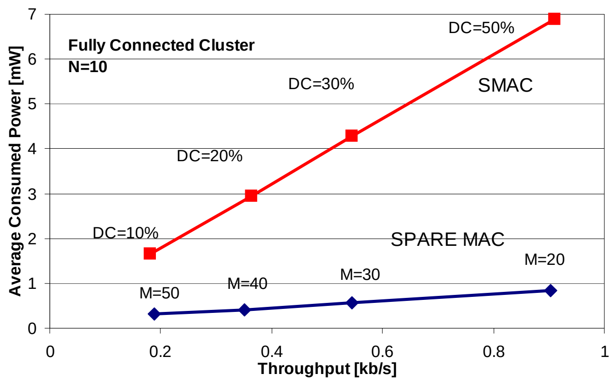

Figure 10. *Average consumed power versus the achieved throughput in the cluster topology (RS=1, N=10). Comparison between SMAC and SPARE MAC.*

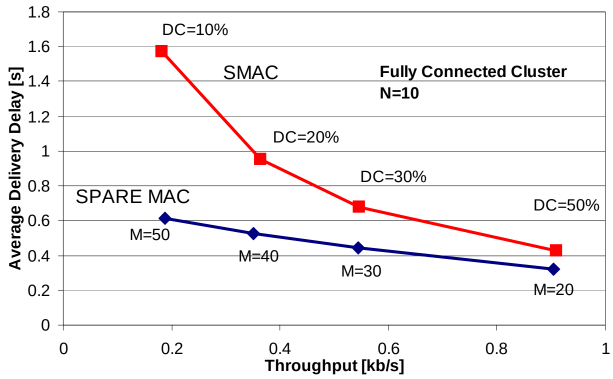

Figure 11. *Average delivery delay versus the achieved throughput in the cluster topology (RS=1, N=10). Comparison between SMAC and SPARE MAC.*

most widely adopted MAC protocols. SMAC implements an access scheme based on an evolution of the IEEE 802.11b Distributed Coordination Function to account for energy efficiency. Under SMAC, the radios of the sensors are switched on and off periodically according to specific activity schedules. Sensors exchange their own schedule with other sensors through a synchronization procedure. The main parameters of SMAC are the *sync interval*, i.e., the interval between two consecutive packets carrying the activity schedules, and the *duty cycle*, that is, the percentage of the time each sensors is active with respect to the overall cycle (sleep + active). In the simulations of SMAC, we adopted the same bandwidth and packet lengths reported in Table 1.

Figures 10 and 11 compare the behavior of SPARE MAC and SMAC in the cluster topology by reporting the average consumed power and the average delivery delay versus the achieved throughput. Under single-hop cluster topology, every sensor generates Poisson traffic with equal probability to all its neighbors. For each throughput value, we consider those configurations with the highest energy efficiency, i.e., with the lowest duty cycle in SMAC, and with longest frame (that is, highest M) in SPARE-MAC. From Figure 10, it is clear that SPARE MAC is able to support a given throughput level consuming much less power than SMAC. The power consumption increases linearly with respect to the achieved throughput in both cases. However, the SMAC curve is much steeper than the one of SPARE MAC.

As for the delivery delay, SPARE MAC always provides faster delivery with respect to SMAC with a gain ranging from 100 ms at high throughput to 1 s to lower throughput values.

4.3 Convergecast Applications

The network topologies tested so far are uniform with respect to the traffic, where all sensors have the same bandwidth requirements. However, WSNs devoted to monitoring and data gathering applications are often deployed and/or organized into hierarchical structures like the one in the tree topology of Figure 7. Here, each leaf generates the same traffic amount (Poisson distributed) towards the Sink. Therefore, the total amount of traffic at each level of the tree is different. The traffic bottleneck is represented by the sensors (two in this case) 1 hop away from the sink. To this end, the bandwidth assigned to each sensor must depend on the location of the sensor itself with respect to the traffic and the network topology. As seen in the previous sections, SPARE MAC allows to differentiate the bandwidth assigned in reception to different sensors by increasing/decreasing the number of slots in the correspondent RS.

Intuitively, the dimension of the RS depends on the throughput requirements, the end-to-end delay requirements of the application, and on the overall energy efficiency of the network. The problem of finding the optimal dimensioning can be formulated as the problem of assigning the minimum number of slots for reception throughout the network while ensuring a bounded average end-to-end delay and the requested throughput. In a homogeneous tree topology, the problem can be formally stated as follows:

$$\min \sum_{i=1}^{n} m_i a_i \qquad (14)$$

s.t.

$$\sum_{i=1}^{n} T_i\left(\frac{G_i}{m_i}\right) \leq T_{bound}, \qquad (15)$$

where n is the number of levels in the tree structure, a_i is the number of sensors in each level, m_i the number of slots in the RS at level i, and $T_i(\frac{G_i}{m_i})$ the delay experienced by traffic traversing level i. Such delay depends on the overall traffic entering level i, G_i, and the number of slots assigned to level i. Since the delay-traffic dependency is non-linear (see Eq. (3)), so is the constraint (15). It can be shown that the problem can be reduced to an MILP (Mixed Integer Linear Programming) formulation [11]. In the following we provide a simple heuristic to determine m_i. Such heuristic exploits the delay model and the configuration guidelines provided in Section 3.1.

Constraint (15) on the end-to-end delay can be split into several constraints of single-hop delays; namely, if we assume to evenly split the delay among different hops, i.e., $T_i = \frac{T_{bound}}{n}$, we can easily dimension the single-hop delay, using the methodology provided in Section 3.1. Given the actual traffic G_{leaves} generated by the leaves and the one-hop delay bound, $\frac{T_{bound}}{n}$ and consequently the corresponding maximum traffic G_{max}, the number of slots in the RS, m_i is given by Eq. (5), i.e., $m_i = \lceil \frac{G_i}{G_{max}} \rceil$, where $G_i = G_{leaves} 2^i$.

In other words, each triplet $\langle G_{leaves}, T_{bound}, n \rangle$ induces a scheduling assignment $\mathbf{m} = [m_1, m_2, \ldots m_n]$ for the tree topology of Figure 7.b.

Table 2 reports the scheduling assignment for the three levels of sensors in the tree topology of Figure 7.b, m_i, in the case the average end-to-end delay must be bounded at

Table 2. *RS dimensions in a three-level tree topology with a end-to-end delay target $D_{tg} = 810$ [ms] when varying the throughput. N=15, M=20.*

		Traffic offered by Leaves [kb/s]					
		0.125	***0.25***	***0.325***	***0.5***	***0.625***	***0.7***
RS	$\mathbf{m_1}$	2	4	5	7	8	9
	$\mathbf{m_2}$	1	2	3	4	4	5
	$\mathbf{m_3}$	1	1	2	2	2	3

$D_{tg} = 810$ [ms], and consequently the per-hop delay $T_i \leq 270$ [ms]. Level 1 represents the sink.

Figure 12 reports the end-to-end delay versus the achieved throughput when varying the dimension of the RS according to the dimensioning criteria provided by Table 2. As clear form the figure, the end-to-end delay is kept below the upper bound (810 [ms]) regardless the throughput by adjusting the dimensions of the RS. Obviously, when the throughput increases, the total dimension of the RS, i.e., $m_{tot} = \sum_{k=1}^{3} m_i$, increases forcing the sensors to operate with higher duty cycles; the effect is a higher power consumption throughout the network as represented in Figure 13.

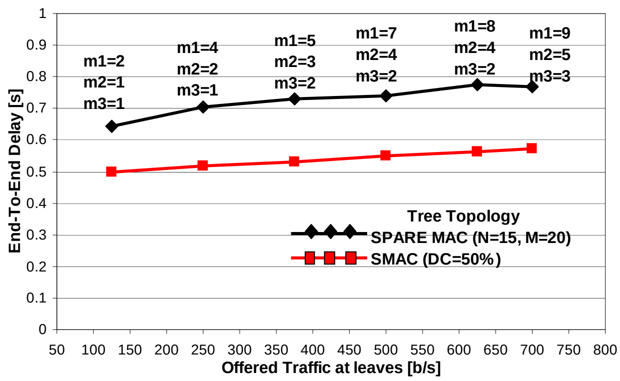

Figure 12. *Average end-to-end delivery delay versus the achieved throughput at the sink in the* tree *topology.*

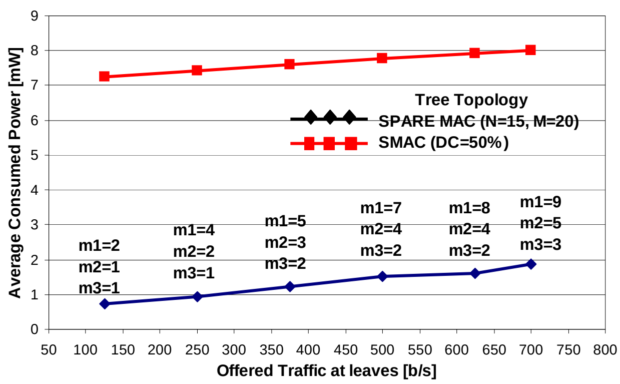

Figure 13. *Average power consumption versus the achieved throughput at the sink in the* tree *topology.*

The two figures report also the corresponding curves obtained under SMAC with adaptive listening, when bounding the single-hop average delivery delay at the very same value as above. We observe here that SMAC allows to define just one value for the Duty Cycle (DC) throughout the network, thus the duty cycle of all the sensors depends on the traffic at the sink only [2](DC=50% in the case of the figure). On the other hand, under SPARE MAC, the activity of each sensor is optimized on the local traffic. As clear from Figure 12, SMAC provides lower end-to-end delay at low-moderate traffic, but, on the other hand, it is forced to consume much more power than SPARE MAC (Figure 13).

5 Related Works

In the following we compare qualitatively SPARE MAC against the most common MAC solutions for WSNs. This analysis is not exhaustive since the number of MAC schemes for WSNs proposed in the literature is very large, but it rather aims at highlighting the novel points of SPARE MAC and those network scenarios where it can help.

MAC schemes for WSNs can be categorized as contention based and contention free protocols. Contention based schemes resort to some type of random access mechanism (ALOHA, CSMA, etc…), allowing potential collisions on packet transmission to happen, and trying to recover from these collisions through proper repetition techniques. The SMAC protocol [14], whose performance we compared against SPARE MAC, belongs to this category. The strategy adopted by SMAC follows the one proposed in PAMAS [15], with the difference that, while PAMAS requires an extra radio interface devoted to transmit control information, SMAC uses in-band signalling. SMAC lets the sensors exchange their activity schedules to minimize energy consumption due to overhearing and idle listening. Collisions are avoided and resolved through classical CSMA-CA schemes. Variants to SMAC have been proposed to let the activity cycle duration be adapted to traffic conditions [16].

Lu *et al.* propose the DMAC protocol [17] for data gathering in tree based network topologies. Similarly to SMAC, DMAC adopts periodic sleep-activity cycles with a CSMA-CA approach, but it further introduces the concept of activity cycles scheduling in order to reduce the latency of transmissions. In other words, sensors know where they are positioned on the tree topology and organize their duty cycle so that the transmission time of sensors belonging to tree level i corresponds to the reception period of sensors of level i-1. In this sense, DMAC shares with SPARE MAC the concept of coordination between sender and receiver, however, the DMAC is contention based and is mainly focused on latency reduction, rather than energy efficiency.

[2]over-dimensioning the duty cycles of leaf sensors

On the other side, collision free protocols often rely upon dynamic TDMA slotted structures, assume slot synchronization among sensors and implements some kind of slot scheduling to assign radio resources to sensors. Valuable examples of TDMA based access schemes are protocols NAMA [19] and TRAMA [18], which share the same design approach. In fact, they both implement a signalling phase among two-hop neighbors to eliminate hidden-node collisions and schedule collision-free transmission. However, NAMA does not address energy efficiency, whilst TRAMA lets the nodes go to sleep if they are not engaged in transmissions/receptions. Different from SPARE MAC, both TRAMA and NAMA schedule transmission at the sender side, i.e., the sender distributes the information on the radio resource it is going to use. Further on, the schedule distribution procedure in TRAMA and NAMA is implemented with a contention based approach, which can lead to potential collisions on scheduling packets. On the other side, the reception schedule distribution in SPARE MAC exploits a fully reliable collision free broadcast channel.

TDMA-W MAC [20] is a particular type of TDMA-based MAC protocol which adopts wake-up slots. Each sensor is assigned a couple of slots within a time frame: a wake-up slot where the sensor is actually receiving and a data slot. Whenever a sensors has data to deliver to an intended receiver it fires a control packet in the wake-up slot of the intended receiver signalling its own ID. The intended receiver will then switch on again in the next frame in correspondence of the data slot of the transmitter. Different from SPARE MAC, TDMA-W MAC assigns data slots to transmitters. On the other side, SPARE MAC implements a receiver-oriented slot assignment and the wake-up slot is devoted to trigger the signalling phase for the distribution of the reception schedules.

6 Concluding Remarks

In this paper we proposed SPARE MAC, an energy efficient data centric MAC scheme for data collection in WSNs. SPARE MAC belongs to the category of TDMA based schemes but, different from other TDMA based approaches, the scheduling algorithm is receiver oriented. In fact, under SPARE MAC, each sensor is assigned radio resources for data reception and exchanges such assignment with its neighbors. Transmitting sensors can consequently become active in the receiving period of their intended receivers only, limiting overhearing, unnecessary transmissions, and idle listening. Further on, the SPARE MAC provides a reliable one hop broadcast channel which can be used both for signalling purpose and for the support of broadcast oriented traffic (routing updates, reconfiguration messages, etc…).

We have proposed analytical models for SPARE MAC, validated against simulation results in terms of data delivery delay and energy efficiency. We have tested through simulation SPARE MAC performance in terms of achieved throughput, energy efficiency and data delivery delay in one-hop cluster topologies with uniformly distributed traffic, and in tree-like topologies for convergecast applications. We have shown that SPARE MAC outperforms SMAC in all tested scenarios.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A Survey on Sensor Networks, *IEEE Communications Magazine*, vol. 40, no. 8, August 2002, page(s): 102–116.

[2] I. F. Akyildiz, T. Melodia and K. R. Chowdhury, A Survey on Wireless Multimedia Sensor Networks, *Computer Networks Journal (Elsevier)*, vol. 51, no. 4, March 2007, page(s): 921–960.

[3] I. F. Akyildiz, E. Stuntebeck, Wireless Underground Sensor Networks: Research Challenges, *Ad Hoc Networks Journal*, vol. 4, July 2006, page(s): 669–686.

[4] I. Demirkol, C. Ersoy, F. Alagoz, MAC Protocols for Wireless Sensor Networks: A Survey, *IEEE Communication Magazine*, vol. 44, no. 4, April 2006, page(s): 115–121.

[5] A. Rowe, R. Mangharam, R. Rajkumar, RT-Link: A Time-Synchronized Link Protocol for Energy Constrained Multi-Hop Wireless Networks, *Proc. of IEEE SECON 2006*, page(s): 402–411.

[6] R. Mangharam, A. Rowe, R. Rajkumar, FireFly: A Time-synchronized Scalable Real-time Sensor Network Platform, *Proc. of IEEE SECON 2006* (poster).

[7] J. Elson, D. Estrin, Time Synchronization for Wireless Sensor Networks, *Proc. of IEEE IPDPS 2001*, page(s): 1965–1970.

[8] I. Rhee, A. Warrier, M. Aia, J. Min, ZMAC: a Hybrid MAC for Wireless Sensor Networks, *Proc. of ACM SenSys 2005*, page(s): 90–101.

[9] S. Ganeriwal, R. Kumar, M. Srivastava, Timing-sync protocol for sensor networks, *Proc. of ACM SenSys 2003*, page(s): 138–149.

[10] L. Kleinrock, *Queuing Systems, Volume I: Theory*, Wiley Interscience, 1975.

[11] L. Campelli, A. Capone, M. Cesana, A MAC Solution for Wireless Sensor Networks based on Slot Periodic Assignment for Reception, Politecnico di Milano, Department of Electronics and Information, Technical Report 2007.38, March 2007.

[12] Network Simulator 2, website: `http://www.isi.edu/nsnam/ns/`.

[13] F. Borgonovo, A. Capone, M. Cesana and L. Fratta, ADHOC MAC: new MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services, *ACM Wireless Networks*, vol.10, no. 4, page(s): 359–366, July 2004.

[14] W. Ye, J. Heidemann and D. Estrin, Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks, *ACM/IEEE Transaction on Networking*, vol. 12, no. 3, page(s): 493–506, June 2004.

[15] S. Singh and C. S. Raghavendra, PAMAS: Power Aware multi-access protocol with signalling for ad hoc networks, *ACM Computer Communication Review*, vol. 28, no. 3, page(s): 5–26, July 1998.

[16] T. van Dam , K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, *Proc. of ACM SenSys 2003*, page(s): 171–180.

[17] S. Bandyopadhyay and E. Coyle, An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks, *Proc. of INFOCOM 2003*, vol. 3, page(s): 1713–1723.

[18] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, Energy-efficient, collision-free medium access control for wireless sensor networks, *ACM Wireless Networks*, Feb. 2006, vol. 12, no. 1, page(s): 63–78.

[19] L. Bao, J. J. Garcia-Luna-Aceves, A New Approach to channel access scheduling for ad hoc networks, *Proc. of ACM MOOBICOM 2001*, page(s): 210–221.

[20] Z. Chen, A. Khokhar, Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks, *Proc. of IEEE SECON 2004*, page(s): 335–341.

Appendix: Service Time Mean Value and Variance

Under the assumptions of constant collision probability, p_c, independent on the actual backoff stage, m_X can be expressed as:

$$m_X = 1 - p_c + \sum_{i=1}^{\infty} p_c^i (1-p_c)[m_{X(i)} + 1], \qquad (16)$$

where $m_{X(i)}$ is the mean value of the integer random variable representing the service time at backoff stage i. Such variable is uniformly distributed in the interval $[1, 2^i]$, thus we can write:

$$m_{X(i)} = \sum_{k=1}^{i} \frac{2^k + 1}{2} \qquad (17)$$

Combining Eq. (16) with Eq. (17) and solving the geometrical series we find:

$$m_X = 1 - p_c + \frac{2p_c(1-p_c)}{1-2p_c} + \frac{p_c}{2(1-p_c)}. \qquad (18)$$

Similarly we can write:

$$\sigma_X^2 = m_{X^2} - m_X^2 \qquad (19)$$

where m_{X^2} is the second moment of r.v. X, and can be expressed as:

$$m_{X^2} = 1 - p_c + \sum_{i=1}^{\infty} p_c^i (1-p_c)[m_{X^2(i)} + 1] \qquad (20)$$

being $m_{X^2(i)}$ the second moment of X at backoff stage i. We can write:

$$m_{X^2(i)} = \sum_{k=1}^{i} \left(\frac{2^k+1}{2}\right)^2 + \sum_{k=1}^{i} \frac{2^{2k}-1}{12}, \qquad (21)$$

Combining Eq. (20) with Eq. (21) and solving the geometrical series we find:

$$\begin{aligned} m_{X^2} = & \; (1-p_c)\Big(1 + \tfrac{1}{9}\tfrac{4p_c}{1-4p_c} - \tfrac{1}{9}\tfrac{p_c}{1-p_c} + \\ & +2\tfrac{p_c}{1-2p_c} + \tfrac{1}{6}\tfrac{p_c}{(1-p_c)^2}\Big). \end{aligned} \qquad (22)$$

Finally, Eq. (22) and Eq. (18) can be substituted into Eq. (19) to find σ_X^2.