

# Modeling the Performance of Wireless Sensor Networks

C.-F. Chiasserini and M. Garetto

CERCOM - Dipartimento di Elettronica, Politecnico di Torino

Torino, Italy

Email: {chiasserini, garetto}@polito.it

**Abstract**—A critical issue in wireless sensor networks is represented by the limited availability of energy within network nodes; therefore making good use of energy is a must. A widely employed energy-saving technique is to place nodes in sleep mode, corresponding to a low-power consumption as well as to reduced operational capabilities. In this work, we develop a Markov model of a sensor network whose nodes may enter a sleep mode, and we use this model to investigate the system performance in terms of energy consumption, network capacity, and data delivery delay. Furthermore, the proposed model enables us to investigate the trade-offs existing between these performance metrics and the sensor dynamics in sleep/active mode. Analytical results present an excellent matching with simulation results for a large variety of system scenarios showing the accuracy of our approach.

## I. INTRODUCTION

Sensor networks are composed of a large number of sensing devices, which are equipped with limited computing and radio communication capabilities [1]. They operate in various kinds of fields, performing tasks such as environmental monitoring and surveillance. Although sensors may be mobile, they can be considered to be stationary after deployment. A typical network configuration consists of sensors working unattended and transmitting their observation values to some processing or control center, the so-called sink node, which serves as a user interface. Due to the limited transmission range, sensors that are far away from the sink deliver their data through *multihop* communications, i.e., using intermediate nodes as relays. In this case a sensor may be both a data source and a data router.

Most application scenarios for sensor networks involve battery-powered nodes with limited energy resources. Recharging or replacing the sensors battery may be inconvenient, or even impossible in harsh working environments. Thus, when a node exhausts its energy, it cannot help but ceases sensing and routing data, possibly degrading the coverage and connectivity level of the entire network. This implies that making good use of energy resources is a must in sensor networks.

Various solutions have been proposed to reduce the sensors energy expenditure. For instance, energy-efficient MAC layer schemes can be found in [2], [3]. Traffic routing and connectivity issues in sensor networks are addressed in [4], [5], [6], while energy-aware strategies for data dissemination and data collection appear in [7], [8], [9].

This work was supported by the Italian Ministry of University and Research through the VICOM and the PRIMO projects

From the energy saving viewpoint, a widely employed technique is to place nodes in a low-power operational mode, the so-called *sleep* mode, during idle periods [10]. In fact, in idle state sensors do not actually receive or transmit, nevertheless they consume a significant amount of power. In sleep mode, instead, some parts of the sensor circuitry (e.g., microprocessor, memory, radio frequency (RF) components) are turned off. As more circuitry components are switched off, the power consumption as well as the operational capabilities of the sensor decrease. Clearly, a trade-off exists between the node energy saving and the network performance in terms of throughput and data delivery delay.

In this work, we develop an analytical model which enables us to explore this trade-off and to investigate the network performance as the sensor dynamics in sleep/active mode vary.

We consider a sensor network with stationary nodes, all of them conveying the gathered information to the sink node through multihop communications. Each sensor is characterized by two operational states: *active* and *sleep*. In active state the node is fully working and is able to transmit/receive data, while in sleep state it cannot take part in the network activity; thus, the network topology changes as nodes enter/exit the sleep state. Through standard Markovian techniques, we construct a system model representing: (i) the behavior of a single sensor, (ii) the dynamics of the entire network, and (iii) the channel contention among interfering sensors. The solution of the system model is then obtained by means of a Fixed Point Approximation (FPA) procedure, and the model is validated via simulation.

By using our analytical model, we study the network performance in terms of capacity, data delivery delay and energy consumption, as the sensor dynamics in sleep/active mode change. Furthermore, we are able to derive the performance of the single sensor nodes as their distance from the sink vary. Although our work mainly focuses on energy consumption and data delay, the level of abstraction of the proposed model is such that it can be applied to investigate various aspects in the design of sensor networks.

To the best of our knowledge, this is the first analytical model that specifically represents the sensor dynamics in sleep/active mode, while taking into account channel contention and routing issues.

The remainder of the paper is organized as follows. Section II reviews some previous work on sensor networks. Section III introduces the network scenario under study and

the assumptions made while constructing our analytical model. In Section IV we present the Markov model and validate by simulation our assumptions; finally we introduce some performance metrics of interest. Section V shows the results obtained by solving the analytical model and compares them to simulation results. Section VI provides some conclusions and points out aspects that will be subject of future research.

## II. RELATED WORK

A large amount of research on sensor networks has been recently reported, ranging from studies on network capacity and signal processing techniques, to algorithms for traffic routing, topology management and channel access control.

From the energy consumption viewpoint, an effective technique is to place sensors in sleep mode during idle periods [10]. The benefits of using sleep modes at the MAC layer are presented in [2], where the authors describe the so-called PAMAS scheme that allows a node to turn off its RF apparatus when it overhears a packet that is not destined for it. The work in [3], [11], [12] propose wake-up scheduling schemes at the MAC layer which wake up sleeping nodes when they need to transmit/receive, thus avoiding a degradation in network connectivity or quality of service provisioning. Relevant to our work are also the numerous network layer schemes that address the problem of data routing in the case where some network nodes may be sleeping [5], [6].

With regard to analytical studies, results on the capacity of large stationary ad hoc networks are presented in [13] (note that sensor networks can be viewed as large ad hoc networks). In [13] two network scenarios are studied: one including arbitrarily located nodes and traffic patterns, the other one with randomly located nodes and traffic patterns. The case of tree-like sensor networks is studied in [9], where the authors present optimal strategies for data distribution and data collection, and analytically evaluate the time performance of their solution. An analytical approach to coverage and connectivity of sensor grids is introduced in [14]. The sensors are unreliable and fail with a certain probability leading to random grid networks. Results on coverage and connectivity are derived as functions of key parameters such as the number of nodes and their transmission radius. The results of the models discussed above and the techniques used there do not directly apply to our model, which focuses on randomly located sensors sending data to a unique destination and operating in sleep or active mode.

Finally, relevant to our work is the Markov model of the sensor sleep/active dynamics developed in [15]. The model predicts the sensor energy consumption; by acquiring this information for each sensor, a central controller constructs the network energy map representing the energy reserves available in the various parts of the system. Note that in [15] only the single node is represented by a Markov chain, while the network energy status is derived via simulation.

## III. SYSTEM DESCRIPTION AND ASSUMPTIONS

We consider a network composed of  $N$  stationary, identical sensor nodes. Sensors are uniformly distributed over a disk

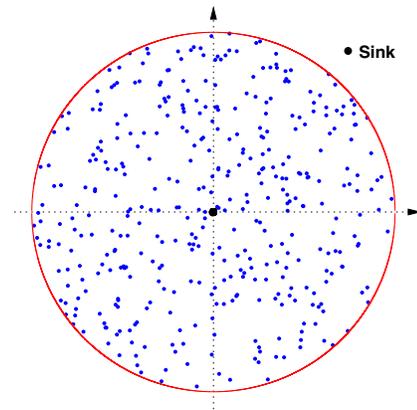


Fig. 1. Network topology of the reference scenario

of unit radius in the plane. The sink node collecting all information gathered by the sensors is located at the center of the disk. An example of network topology is shown in Figure 1 in the case of  $N = 400$ .

We assume that all nodes have a common maximum radio range  $r$  and are equipped with omnidirectional antennas. Nodes can choose an arbitrary transmit power level for each data transmission, provided that their transmission range does not exceed  $r$ . Also, we consider network topologies such that for any sensor there exists at least one path connecting the sensor to the sink.

The information sensed by a network node is organized into data units of fixed size, that can be stored at the sensor in a buffer of infinite capacity; the buffer is modeled as a centralized FIFO queue. Sensors cannot simultaneously transmit and receive; the time is divided into time slots of unit duration and the transmission/reception of each data unit takes one time slot. The wireless channel is assumed to be error-free, although our model could be easily extended to represent a channel error process.

Further assumptions on sensors behavior, traffic routing and channel access control are introduced below.

### A. Sensors Behavior

As highlighted in [1], [16], the main functions (and hence causes of energy consumption) in a sensor node are sensing, communication and data processing. Correspondingly, different operational states for a sensor can be identified.

For the sake of simplicity, we consider two major operational states<sup>1</sup>: *active* and *sleep*. The *sleep* state corresponds to the lowest value of the node power consumption; while being asleep, a node cannot interact with the external world. The *active* state includes three operational modes: *transmit*, *receive*, and *idle*. In the transmitting mode, energy is spent in the front-end amplifier, that supplies the power for the actual RF transmission, in the transceiver electronics and in the node processor implementing signal generation and processing functions. In the receiving mode, energy is consumed entirely

<sup>1</sup>In general, several sleep states could be defined considering that each sensor component may have different power states and various combinations of the components operational states are possible.

by the transceiver electronics and by processing functions, such as demodulation and decoding. In the idle state, a node typically listens to the wireless channel without actively receiving. In idle mode, energy expenditure is mainly due to processing activity, since the voltage controlled oscillator is functioning and all circuits are maintained ready to operate. Notice also that an energy cost  $E^t$  is associated with each transition from sleep to active mode, while the cost of passing from active to sleep mode can be neglected [10]. We assume that  $E^t$  is twice the energy consumption per time slot in idle mode<sup>2</sup>.

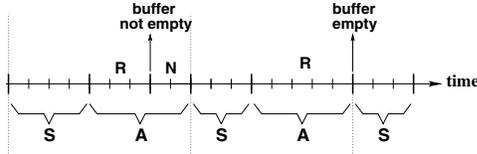


Fig. 2. Temporal evolution of the sensor state

Based on the above observations, we describe the temporal evolution of the sensor state in terms of cycles, as depicted in Figure 2. Each cycle comprises a sleep phase ( $S$ ) and an active phase ( $A$ ). During phase  $S$ , the sensor is in sleep mode; the duration of  $S$ , expressed in number of time slots, is assumed to be geometrically distributed with parameter  $q$ . When the sensor switches to the active mode, phase  $A$  begins and the sensor schedules a time instant in the future at which it will go back to sleep. The scheduled active period, expressed in time slots, is a random variable geometrically distributed with parameter  $p$ . However, at the time slot at which the sensor should go to sleep, its data buffer may not be empty. In this case we assume that phase  $A$  is prolonged till all data units are forwarded to other nodes. During this additional period of activity the sensor does not accept to relay new data units nor generates data on its own, in order to go back to sleep as soon as possible. The active phase can thus be divided into an initial phase ( $R$ ) and (possibly) a phase ( $N$ ). In phase  $R$  the sensor can receive and transmit; also it generates data units according to a Poisson process with rate equal to  $g$ . In phase  $N$  the sensor does not receive nor generate data; it can only transmit the data units that are still in its buffer or be idle waiting for a transmission opportunity. In Figure 2 it is highlighted that  $A$  coincides with  $R$  when at the scheduled end time of  $A$  the sensor buffer is empty.

We observe that the behavior described above allows sensors to simply adapt to traffic conditions and prevents network instability due to overload. However, this is not a critical assumption in constructing our analytical model, which could be easily modified to represent a different sensor behavior.

We also highlight that, although sensors can be in different operational states, they are always functioning. Indeed we assume a stationary network topology and the event that a sensor either runs out of energy or fails is not considered.

## B. Data Routing

In this work we consider a sensor network whose nodes have already performed the initialization procedures necessary to self configure the system. Therefore sensors have knowledge of their neighboring nodes, as well as of the possible routes to the sink. (for instance through a routing algorithm such as the one proposed in [17]). Since we consider a network of stationary nodes performing, for instance, environmental monitoring and surveillance, the routes and their conditions can be assumed to be either static or slowly changing.

We assume that sensors can communicate with the sink using multiple routes. Each sensor constructs its own routing table where it maintains up to  $M$  routes, each of which corresponds to a different next-hop node (hereinafter just called next-hop) and is associated with a certain energy cost. The routing table might contain a smaller number of entries if the sensor has less neighbors. For the generic route  $\rho$ , the energy cost  $e(\rho)$  is computed as follows. Given a node  $i \in \rho$ , we denote with  $\nu_\rho(i)$  the node immediately succeeding  $i$  on  $\rho$  (the route includes the source and the relays but not the sink). We have,

$$e(\rho) = \sum_{i \in \rho} E_{i, \nu_\rho(i)} = \sum_{i \in \rho} \left( E_{i, \nu_\rho(i)}^{(tx)} + E_{\nu_\rho(i)}^{(rx)} \right) \quad (1)$$

where  $E_{i, \nu_\rho(i)}$  is the energy cost for transferring a data unit from node  $i$  to its next-hop in route  $\rho$ , equal to the sum of the transmission energy spent by  $i$  ( $E_{i, \nu_\rho(i)}^{(tx)}$ ) and the reception energy consumed by  $\nu_\rho(i)$  ( $E_{\nu_\rho(i)}^{(rx)}$ ). As described in Section III-A,  $E_{\nu_\rho(i)}^{(rx)}$  is due to the transceiver electronics ( $E^{(ele)}$ ) and to processing functions ( $E^{(proc)}$ ); while  $E_{i, \nu_\rho(i)}^{(tx)}$  has to account for  $E^{(ele)}$ ,  $E^{(proc)}$ , as well as for the energy consumption due to the amplifier, that is assumed to be proportional to the squared distance between transmitter and receiver [18]. Thus, we re-write  $e(\rho)$  as,

$$e(\rho) = \sum_{i \in \rho} \left[ 2 \left( E^{(ele)} + E^{(proc)} \right) + d_{i, \nu_\rho(i)}^2 E^{(amp)} \right] \quad (2)$$

where  $E^{(amp)}$  is a constant value and  $d_{i, \nu_\rho(i)}$  is the distance between  $i$  and  $\nu_\rho(i)$  in the disk of unit radius.

When a sensor wants to transmit a data unit, it adopts the following routing strategy (although other strategies could be considered as well). The node polls its next-hops giving priority to the routes associated with the lowest energy cost, until it finds a next-hop that is ready to receive. Thus, a sensor always dispatches its data units to the best next-hop among the available ones.

## C. Channel Access

Consider a transmission over one hop and let nodes  $i$  and  $j$  ( $1 \leq i \leq N$ , and  $0 \leq j \leq N$  with 0 indicating the sink) be the transmitter and the receiver, respectively. The transmission is successful if [13]:

- 1) the distance between  $i$  and  $j$  is not greater than  $r$ ,

$$d_{i,j} \leq r \quad (3)$$

<sup>2</sup>Indeed, the transition cost from sleep to active state is typically very high.

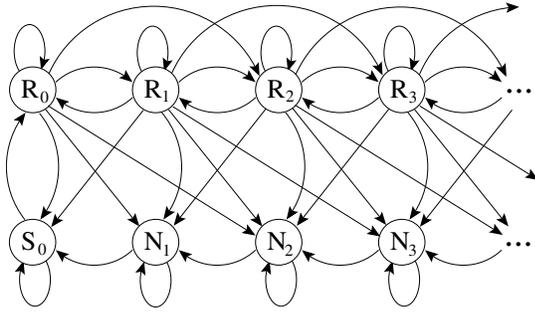


Fig. 3. Markov chain describing the sensor behavior

2) for every other node,  $k$ , simultaneously receiving,

$$d_{i,k} > r \quad (4)$$

3) for every other node,  $l$ , simultaneously transmitting,

$$d_{l,j} > r. \quad (5)$$

To avoid unsuccessful transmissions, we assume that sensors employ a CSMA/CA mechanism with handshaking, as in the MACA and MACAW schemes [19], [20] (although, other MAC protocols could be considered as well), and that the radio range of handshaking messages transmission is equal to  $r$ . If  $i$  wants to transmit to  $j$  and senses the channel as idle,  $i$  sends a transmission request to  $j$  and waits till either it receives a message indicating that  $j$  is ready to receive (i.e., it is active and there are not other simultaneous transmissions that could interfere), or a timeout expires. In the former case,  $i$  sends the data to  $j$ ; in the latter case,  $i$  will poll the following next-hop<sup>3</sup>. While  $i$  is looking for a next-hop that is ready to receive, data are buffered at the node waiting for transmission.

In a nutshell, our model accounts for channel contention, however data transmissions are collision-free. Moreover, since buffers are assumed to be of infinite capacity, data units are never lost while traveling through the network.

#### IV. SYSTEM MODEL

In this section we present our modeling approach to analyze the behavior of the sensor network described in Section III. Our model consists of three building blocks that will be described and validated separately: (i) the *sensor model* (Section IV-A), (ii) the *network model* (Section IV-B) and (iii) the *interference model* (Section IV-C). The overall solution is obtained by means of a Fixed Point Approximation (FPA) procedure in which the three blocks interact by exchanging various parameters along a closed loop till a final equilibrium is reached. The fixed point procedure will be explained in detail in Section IV-D. In Section IV-E, we describe the performance metrics that can be obtained by solving the proposed model.

<sup>3</sup>Note that we could also assume that  $i$  sends only one poll message and its next-hops reply after time intervals of different duration so as to avoid collisions. The response delays should be set according to the order of the associated routes in  $i$ 's routing table. In this case  $i$  will just wait to receive a response from one of its next-hop until a timeout expires.

#### A. Sensor Model

We study the behavior of a single sensor by developing a discrete-time Markov chain (DTMC) model, in which the time is slotted according to the data unit transmission time, i.e., the time interval necessary to transmit a data unit including the overhead required by the MAC layer. Although the DTMCs describing the individual node behavior are solved independently of each other, the sensor model incorporates the dynamics resulting from the interactions between the sensor and its neighbors, as will be explained later in this section.

As a first step, let us introduce the DTMC of a sensor neglecting the operational state of its neighbors. The state of this simplified DTMC is defined by: (i) the cycle phase in which the sensor is in the current time slot (namely,  $S$ ,  $R$ , or  $N$ ), and (ii) the number of data units stored in the sensor buffer, which can be any integer value ranging from 0 to  $\infty$ . The resulting Markov chain is shown in Figure 3, where the different phases are indexed with the number of data units stored in the sensor buffer.

Let  $\mathbf{P}$  be the transition matrix, whose element  $P(s_o, s_d)$  denotes the probability that the chain moves in one time slot from source state  $s_o$  to destination state  $s_d$ . In deriving the probabilities  $P(s_o, s_d)$ 's, the following dynamics have to be taken into account:

- The sensor sleep-active dynamics, determined by the input parameters  $p$  and  $q$  (introduced in Section III-A);
- The data unit generation process (in phase  $R$  only); we denote with  $g$  the probability that a data unit is generated by the sensor in a time slot;
- The reception of data units from neighboring nodes (in phase  $R$  only); we indicate with  $\alpha$  the probability that a data unit is received in a time slot;
- The data unit transmission (in phase  $R$  and  $N$  only); we denote with  $\beta$  the probability that a data unit is transmitted in a time slot. Notice that  $\beta$  accounts for the channel contention, i.e., it would be equal to 1 if there were no contention on the wireless medium.

While  $p$ ,  $q$  and  $g$  are input parameters to the model,  $\alpha$  and  $\beta$  need to be estimated. Also, since a node cannot transmit and receive simultaneously, we have:  $\alpha + \beta \leq 1$ .

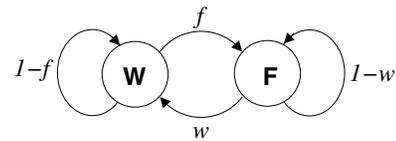


Fig. 4. DTMC model describing the behavior of the sensor next-hops

Next, we include in the above DTMC the model of the sleep/active dynamics of the sensor next-hops. To this end, we introduce a further state variable which can take two values: *Wait*, denoted by  $W$ , and *Forwarding*, denoted by  $F$ .  $W$  corresponds to all next-hops being unable to receive because they are in phases  $S$  or  $N$ .  $F$  represents the case where at least one next-hop is in phase  $R$  and, thus, it can receive provided that interference conditions allow it. We assume that the evolution of the state of each next-hop is

TABLE I

TRANSITION PROBABILITIES FROM NON-EMPTY BUFFER (FOR THE SAKE OF BREVITY,  $l_0 = (1 - \alpha)(1 - g)$ ,  $l = \beta g + (1 - \alpha - \beta)(1 - g)$ ,  $b_0 = g(1 - \alpha) + \alpha(1 - g)$ ,  $b = g(1 - \alpha - \beta) + \alpha(1 - g)$ )

$s_o$	$s_d$	$P(s_o, s_d)$	Condition		
$R_i^F$	$R_i^W$	$wl(1 - p)$	$i \geq 1$		
	$N_i^W$	$wlp$			
	$N_i^F$	$(1 - w)lp$			
	$R_{i+1}^F$	$(1 - w)b(1 - p)$			
	$R_{i+1}^W$	$wb(1 - p)$			
	$N_{i+1}^W$	$wbp$			
	$N_{i+1}^F$	$(1 - w)bp$			
	$R_{i+2}^F$	$(1 - w)g\alpha(1 - p)$			
	$R_{i+2}^W$	$wg\alpha(1 - p)$			
	$N_{i+2}^W$	$wg\alpha p$			
	$N_{i+2}^F$	$(1 - w)g\alpha p$			
	$R_{i-1}^F$	$(1 - w)\beta(1 - g)(1 - p)$			
	$R_{i-1}^W$	$w\beta(1 - g)(1 - p)$			
	$S_0^W$	$w\beta(1 - g)p$		$i = 1$	
$S_0^F$	$(1 - w)\beta(1 - g)p$	$i \geq 2$			
$N_{i-1}^W$	$w\beta(1 - g)p$				
$N_{i-1}^F$	$(1 - w)\beta(1 - g)p$	$i \geq 1$			
$R_i^W$	$R_i^F$		$fl_0(1 - p)$		
	$N_i^F$		$fl_0p$		
	$N_i^W$		$(1 - f)l_0p$		
	$R_{i+1}^W$		$(1 - f)b_0(1 - p)$		
	$R_{i+1}^F$		$fb_0(1 - p)$		
	$N_{i+1}^F$		$fb_0p$		
	$N_{i+1}^W$		$(1 - f)b_0p$		
	$R_{i+2}^W$		$(1 - f)g\alpha(1 - p)$		
	$R_{i+2}^F$		$fg\alpha(1 - p)$		
	$N_{i+2}^F$		$fg\alpha p$		
	$N_{i+2}^W$		$(1 - f)g\alpha p$		
	$N_i^F$		$N_i^W$	$w(1 - \beta)$	$i \geq 1$
			$N_{i-1}^F$	$(1 - w)\beta$	$i \geq 2$
		$N_{i-1}^W$	$w\beta$	$i = 1$	
$S_0^F$		$(1 - w)\beta$			
$S_0^W$		$w\beta$			
$N_i^W$		$f$	$i \geq 1$		

independent of the others. Transitions between  $W$  and  $F$  are modeled by the two-state DTMC shown in Figure 4, where the transition probabilities  $f$  and  $w$  are additional parameters to be estimated.

The diagram of the complete DTMC model describing the joint evolution of the sensor and the state of its next-hops is not shown here; however the state space can be obtained by duplicating the states of the simplified DTMC model depicted in Figure 3. Table I reports the transition probabilities  $P(s_o, s_d)$  from state  $s_o$  where the buffer is not empty, to the successor state  $s_d$ . The remaining transitions are not listed in the Table due to the lack of space; however they can be easily derived following the same rational. In the Table, the first two columns list states  $s_o$  and  $s_d$ , respectively; the fourth column denotes the conditions which state  $s_o$  has to satisfy in order to admit the transition reported in the third column. To represent the states of the complete DTMC we use the same notation as for the simplified model, adding a superscript  $W$  or  $F$  to represent the state of the next-hops.

With regard to the complete DTMC model, we make the following remarks.

- In states denoted by apex  $W$  transmissions are not possible (i.e., the number of buffer data units cannot be decremented), because all of the next-hops are in phases  $S$  or  $N$ ; transmissions can occur only in states denoted by apex  $F$ .
- The probability  $\beta$  to transmit a data unit in a time slot is now conditioned on the fact that the sensor buffer is not empty and *at least one next-hop is in phase  $R$* .
- Since we assume an infinite buffer capacity, the DTMC has an infinite number of states. This allows us to efficiently compute the stationary distribution using a matrix geometric technique. However, the extension to the case of a finite buffer size would be straightforward.

Let us denote the stationary distribution of the complete DTMC by  $\pi = \{\pi_s\}$ , where  $s$  is a generic state of the model. By solving the sensor model, we obtain  $\pi$  and derive the following metrics:

- the average number of data units generated in a time slot  $\Lambda_E$ ,

$$\Lambda_E = \sum_{i=0}^{\infty} \left( \pi_{R_i^F} + \pi_{R_i^W} \right) g \quad (6)$$

- the sensor throughput  $T$ , defined as the average number of data units forwarded by the sensor in a time slot,

$$T = \sum_{i=1}^{\infty} \left[ \pi_{R_i^F}(1 - \alpha) + \pi_{N_i^F} \right] \beta \quad (7)$$

- the overall probabilities  $\pi_R$ ,  $\pi_S$ ,  $\pi_N$  that a sensor is in the corresponding phases  $R$ ,  $S$ ,  $N$
- the average buffer occupancy,

$$\bar{B} = \sum_{k=1}^{\infty} \left[ k \left( \pi_{R_k^F} + \pi_{R_k^W} + \pi_{N_k^F} + \pi_{N_k^W} \right) \right]. \quad (8)$$

(We will add the sensor index as an apex to the notation of the above metrics when they refer to a particular node.)

We validate our sensor model by computing the unknown parameters  $\alpha$ ,  $\beta$ ,  $w$  and  $f$  by simulation. These values are used in the sensor model to derive the stationary distribution of the DTMC, which, on its turn, is used to compute (6)-(8). We then verify whether the values of the above metrics match those obtained by simulation. The validation procedure is carried out by executing a sufficiently long simulation run on the reference scenario shown in Figure 1. The data generation, the data routing and the channel access scheme are as described in Section III.

Results prove to be very accurate under a variety of parameter settings. Here, as an example, we present the results obtained by taking  $r = 0.25$ ,  $N = 400$ ,  $M = 3$ , and  $p = q = 0.1$  for all sensors. The same simulation scenario will be used to validate the other building blocks of our model.

Figure 5 shows four plots derived with generation rate  $g = 0.005$  (a heavy load condition), comparing some of the above metrics derived through the sensor model with those measured through simulation. Each point represents the value attained for a particular sensor. The alignment of the points on the bisector  $y = x$  proves the accuracy of the sensor model.

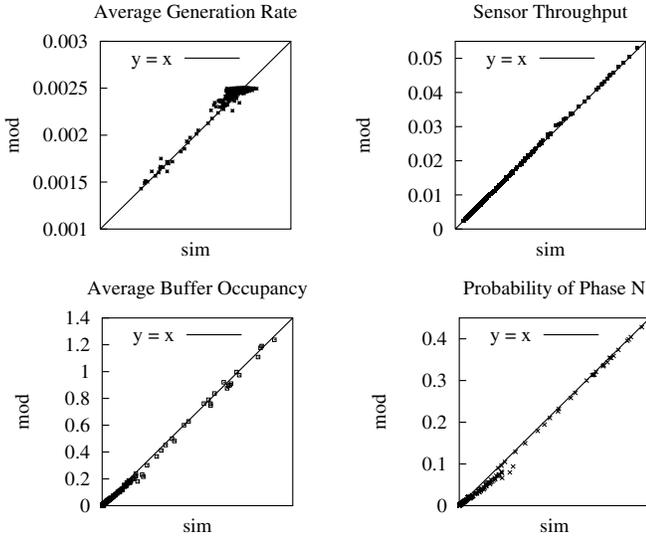


Fig. 5. Validation of the sensor model on the reference scenario

### B. Network Model

We now introduce our approach to modeling the sensor network. The sensor network can be regarded as an open queueing network in which each queue corresponds to the buffer of a sensor, and the external arrival rate to each queue corresponds to the data unit generation rate at the sensor.

First of all, remind that data units are never lost while traversing the network. Thus, given the average generation rate  $\Lambda_E^i$  of the generic sensor  $i$ , the total arrival rate at the sink, that is the network capacity  $C$ , is given by,

$$C = \sum_{i=1}^N \Lambda_E^i. \quad (9)$$

Then our goal is to derive the internal arrival rate at each sensor  $\Lambda_I^i$ , given the average generation rates  $\Lambda_E^i$ 's. This can be done by solving the system of flow balance equations:

$$\Lambda_I = \Lambda_I \mathbf{R} + \Lambda_E \quad (10)$$

where  $\Lambda_I$  and  $\Lambda_E$  are row vectors stacking the rates  $\Lambda_I^i$ 's and  $\Lambda_E^i$ 's, respectively, and  $\mathbf{R}$  is the (unknown) matrix of transition probabilities between the queues of the network. Element  $R(i, j)$  represents the fraction of outgoing traffic of sensor  $i$  that is sent to its next-hop  $j$ . In order to compute  $\mathbf{R}$ , one has to account for the routing policy chosen by the sensor, as well as the effect of the sleep/active dynamics of the next-hops and the contention on the wireless channel. In our case the routing policy is a strict priority for the best available next-hop, as described in Section III. The simplest approach is to consider only the stationary probabilities of the next-hops state, and to assume that the next-hops state are independent. Following this approach the transition probability  $R(i, j)$  can be computed as,

$$R(i, j) = K \left( \prod_{m \in N_{i,j}} (\pi_S^m + \pi_N^m) \right) \pi_R^j \quad (11)$$

where  $N_{i,j}$  is the set of next-hops that have higher priority than  $j$  in the routing table of  $i$ , and  $K$  is a normalization factor such that the sum of  $R(i, j)$  over all  $j$ 's is equal to one. This expression means that a data unit is forwarded to a given sensor  $j$  if and only if  $j$  can receive while all next-hops with higher priority cannot.

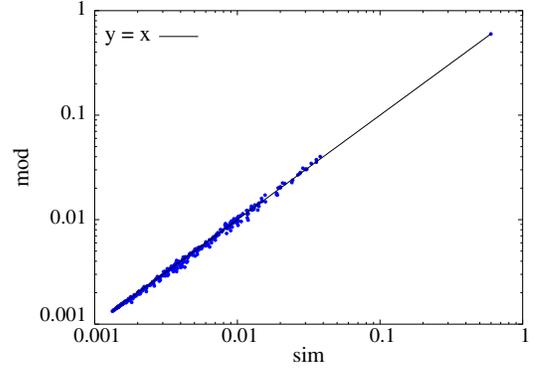


Fig. 6. Comparison of arrival rates at each sensor between simulation and analysis

To validate our approach, we take from simulation the average generation rates  $\Lambda_E^i$ 's and the state probabilities of the network nodes, and compute the transition matrix  $\mathbf{R}$  using (11). Then, we derive the arrival rates  $\Lambda_I^i$ 's solving (10) and compare them to the arrival rates obtained by simulation. As shown in Figure 6 for a network load equal to 0.6, our analytical results are very close to those derived by simulation (each point in the plot stands for an element of vector  $\Lambda_I$ ).

Note that by solving the network model we can also obtain the expected throughput of the generic sensor  $i$  as,

$$T^i = \Lambda_I^i + \Lambda_E^i. \quad (12)$$

### C. Interference model

The purpose of the interference model is to compute for each node the parameter  $\beta$  to be used into the sensor model presented in Section IV-A. The method used to estimate the parameters  $\alpha$ ,  $f$ , and  $w$  needed to solve the sensor model will be described in Section IV-D.

We remind that  $\beta$  has been defined as the probability to transmit a data unit in a time slot given that the buffer is not empty and at least one next-hop is in phase  $R$  at the beginning of the slot. If there were no contention on the wireless channel,  $\beta$  would be equal to 1. As described in Section III, a node transmission attempt is successful if the conditions expressed as in (3)–(5) are satisfied. The computation of  $\beta$  thus requires a careful investigation of the interference produced by other sensors trying to transmit in proximity of the node for which we want to estimate  $\beta$ . In order to explain our approach, consider the set of nodes shown in Figure 7. The transmission range of three nodes,  $\{A, F, H\}$ , is represented by a circle. Assume that we want to estimate the parameter  $\beta$  of node A, which has two next-hops, B and C. We need to find all transmissions that could potentially interfere with the transmission of A to its next-hops. Let  $(X, Y)$  denote the transmission from the generic node X to the generic node Y. We notice that transmissions

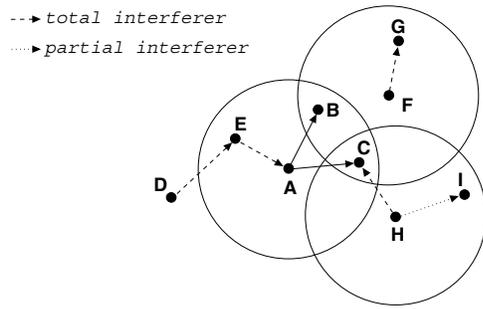


Fig. 7. Example of channel contention and hindered transmissions

like (D,E) and (H,C) violate condition (4) since the receivers are within the radio range of A; a special case is given by the transmissions whose receiver is A itself (e.g., (E,A)). Instead, transmissions like (F,G) and (H,I) meet condition (4) and violate condition (5) since the transmitters interfere with A's next-hops. In addition, we observe that transmissions as (D,E), (E,A), (H,C) and (F,G) totally inhibit A's transmission, thus we call them *total interferers*. Instead, transmissions like (H,I) do not necessarily prevent A from sending data (e.g., (A,B) could take place), thus we call them *partial interferers*. We highlight that transmissions violating (4) or where A is the receiver are always *total interferers*.

To estimate  $\beta$  for the generic sensor  $i$  we proceed as follows. First we compute for each node  $n$  ( $1 \leq n \leq N$ ) the probability  $I^i(n)$  that a transmission in which  $n$  is involved as either transmitter or receiver, totally inhibits  $i$ 's transmission (*total interferer*). Our approach is based on the knowledge of the average transmission rates  $\lambda_{n,m}$  between  $n$  and its generic receiver  $m$ . We write,

$$I^i(n) = \sum_{m=1}^N \lambda_{m,n} \mathbf{1}_{\{d(n,i) \leq r\}} + \sum_{m=0}^N \lambda_{n,m} \mathbf{1}_{\{d(m,i) > r\}} V^i(n) C^i(n) \quad (13)$$

where  $m = 0$  denotes the sink and  $\mathbf{1}_{\{\cdot\}}$  is the indicator function. The first summation on the right hand side accounts for the transmissions violating (4) or destined to  $i$ ; while the second summation accounts for the transmissions that meet (4) but violate (5). The term  $V^i(n)$  is equal to 1 if there exists at least one next-hop of  $i$  within the transmission range of  $n$ , with  $n$  being different from  $i$ :

$$V^i(n) = \begin{cases} 1 & \exists k \in H^i : d(n,k) \leq r, n \neq i \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where  $H^i$  is the set of next-hops of  $i$ . The term  $C^i(n)$  is equal to 1 if  $n$ 's transmission is a *total interferer*; otherwise it accounts for a *partial interferer* considering that this becomes a *total interferer* if the next-hops of  $i$  outside the transmission range of  $n$  are also unable to receive because they are in phases  $S$  or  $N$ . Hence,

$$C^i(n) = \begin{cases} \prod_k (\pi_S^k + \pi_N^k) \forall k \in H^i : d(n,k) > r & \text{if any} \\ 1 & \text{else.} \end{cases} \quad (15)$$

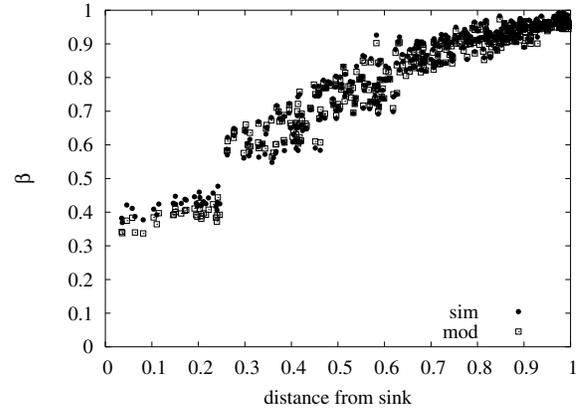


Fig. 8. Estimation of  $\beta$  using conditioned transmission rates obtained from simulation, for the various network nodes

Then,  $\beta^i$  is estimated as follows:

$$\beta^i = \prod_{n=1}^N [1 - I^i(n)]. \quad (16)$$

To validate our estimate of  $\beta^i$ , we take from simulation all transmission rates  $\lambda_{n,m}$ 's. Since  $\beta^i$  is a transmission probability conditioned on the fact that the sensor buffer is not empty and at least one next-hop is available, the correct values of  $\lambda_{n,m}$  to be used should also be conditioned on this fact. For a network load equal to 0.6, we obtained from simulation the conditioned transmission rates, and using (13) and (16) we computed the parameter  $\beta^i$  for each sensor. Results are shown in Figure 8 as a function of the distance from the sink, and present an accurate matching with simulation results proving that our approach to estimating  $\beta^i$  as in (16) is correct.

Unfortunately, the conditioned transmission rates seem to be hard to be evaluated analytically. Thus we resorted to the unconditioned rates  $\lambda_{n,m}$ 's provided by the network model, and slightly refined the interference model in order to account for the neglected correlation between the  $\lambda_{n,m}$ 's and the state of the sensor for which  $\beta^i$  is computed. Our approach is briefly described in the rest of this section.

For each sensor  $i$  whose distance from the sink is smaller than  $r$ , we define the set of nodes  $A_i$  whose transmission range covers all of the next-hops of  $i$ . We compute the average probability  $t_{A_i}$  that a node in this set is ready to transmit a packet as,

$$t_{A_i} = \frac{1}{N_{A_i}} \sum_{k \in A_i} (\pi_N^k + \pi_R^k - \pi_{R_0}^k) \quad (17)$$

where  $N_{A_i}$  is the cardinality of set  $A_i$ . Then, we consider that node  $i$  will be able to transmit only if it gets control of the channel before every other node in  $A_i$ , assuming that nodes are equally likely to seize the channel at the beginning of a time slot, and their probability to be ready to transmit are independent. We therefore obtain a refined estimate of  $\beta^i$  as,

$$\beta^i = \sum_{k=0}^{N_{A_i}} \frac{1}{k+1} \binom{N_{A_i}}{k} t_{A_i}^k (1 - t_{A_i})^{N_{A_i} - k} \prod_{n \notin A_i} [1 - I^i(n)] \quad (18)$$

For each sensor  $i$  whose distance from the sink is greater than  $r$ ,  $\beta$  is estimated using (16), that provides an accurate solution for these nodes even if we plug in the unconditioned transmissions rates  $\lambda_{n,m}$ . Doing so we obtain the values of  $\beta^i$  labeled in Figure 9 as “mod” and compare them with simulation results (labeled in the plot as “sim”). The plot shows that our estimation of  $\beta$  is quite accurate. In fact, even if our approach tends to overestimate  $\beta$  for nodes very close to the sink (namely, for node distance from the sink shorter than 0.1), the probability that a data unit is received from neighboring nodes by such sensors is usually quite small, so that the impact on the overall solution is marginal. Figure 10 proves this statement showing the probability that a data unit is received from neighboring nodes in a time slot (parameter  $\alpha$  of the sensor model) as a function of the node distance from the sink, according to simulation.

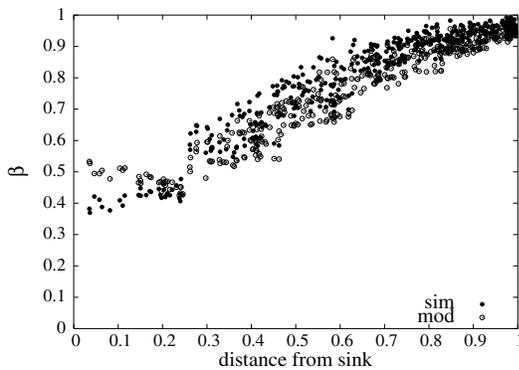


Fig. 9. Estimation of  $\beta$  using unconditioned transmission rates computed by the model, for the various network nodes

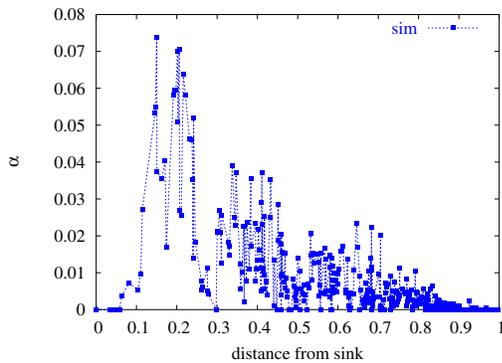


Fig. 10. Probability that a data unit is received from neighboring nodes in a time slot (parameter  $\alpha$  of the sensor model), for the various network nodes

#### D. Fixed Point Approximation

The three building blocks of the model described in Sections IV-A, IV-B and IV-C can be combined together to obtain a global system solution which does not require to get any parameter values from simulation. This is done by using an FPA, based on the close loop depicted in Figure 11.

The procedure starts with the solution of the DTMC representing the individual sensor behavior for each sensor  $i$

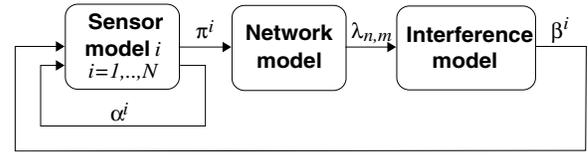


Fig. 11. Close loop used to obtain the global solution of the system

( $1 \leq i \leq N$ ), from which we obtain the stationary distribution probabilities  $\pi^i$ 's<sup>4</sup>. Then we run the network model and derive the data rates  $\lambda_{n,m}$ 's for each pair of nodes in the network, as well as the expected throughput for each sensor (i.e.,  $T^i$  as in (12)). The data rates  $\lambda_{n,m}$ 's are given as inputs to the interference model to estimate the parameter  $\beta^i$  for each sensor. On their turn, the  $\beta^i$ 's are given as input to the sensor models, thus closing the loop.

Within each sensor model, given the value of  $\beta^i$  and employing a numerical technique, we derive the unknown parameter  $\alpha^i$ .  $\alpha^i$  is estimated so that the sensor throughput given by (7) approximates the value previously predicted by the network model using (12). In Figure 11 this procedure is highlighted by the inner loop around the block of the sensor model. We point out that obtaining a precise estimate of  $\alpha^i$  inside the inner loop is not worthwhile, since the target value of sensor throughput is updated by the exterior loop, thus we decided to limit the number of iterations in the inner loop to 3.

Furthermore, to solve the sensor model, we need to estimate parameters  $w^i$  and  $f^i$  of the DTMC describing the behavior of the next-hops (see Section IV-A and Figure 4). We compute the stationary probability of state  $W$  for sensor  $i$  as follows:

$$\pi_w^i = \prod_{k \in H^i} (\pi_S^k + \pi_N^k) \quad (19)$$

using the most recent estimate of the stationary probabilities of the sensor next-hops. The transition probability  $f^i$  is estimated as

$$f^i = 1 - \prod_{k \in H^i} \left( 1 - p \frac{\pi_R^k}{\pi_S^k + \pi_N^k} \right). \quad (20)$$

where  $p \frac{\pi_R^k}{\pi_S^k + \pi_N^k}$  approximates the transition probability of sensor  $k$  from the aggregate state including phases  $S$  and  $N$ , to phase  $R$ . It is then straightforward to derive the other unknown transition probability:  $w^i = f^i \frac{\pi_w^i}{1 - \pi_w^i}$ . Once we have solved numerically the DTMC of each sensor [21], we can compute all metrics of interest described at the end of Section IV-A, and in particular a new estimate of the generation rates  $\Lambda_E^i$ 's (using (6)) to be plugged again into the network model. The overall procedure is repeated until convergence on the parameter estimates is reached. We use as stopping criterion the worst relative error among all sensors for two successive estimates of the sensor throughput.

We highlight that the complexity of the numerical method used to solve the sensor model reduces to the solution of

<sup>4</sup>At the very first iteration of the FPA procedure, we solve the DTMC for each sensor assuming that only the considered node generates data; thus, we obtain:  $\Lambda_E^i = g \frac{q}{p+q}$ . Note that the parameters  $g$ ,  $p$  and  $q$  could be specialized for each node.

a linear system of equations of dimension  $4L$ , where  $L$  is small (using  $L = 5$  is enough to obtain a solution very close to the exact solution). Furthermore, we observe that the FPA procedure exhibits good convergence properties. In fact, less than 10 iterations are usually required to have the worst relative error fall below a threshold of  $10^{-4}$ .

### E. Performance metrics

Many interesting performance metrics can be derived from the solution of our model. The detailed behavior of each individual sensor in the network is obtained from the sensor model described in Section IV-A. The network capacity  $C$  is simply the arrival rate of data units at the sink, which is computed by the network model. The average transfer delay  $\bar{D}$ , that is the average number of time slots required to deliver a data unit from a source node to the sink, follows from the application of Little's formula to the whole network, and it is given by

$$\bar{D} = \frac{\sum_{i=1}^N \bar{B}^i}{C}. \quad (21)$$

The network energy consumption per time slot  $\bar{E}$  can be divided into three contributions. The first one is the sum of the energy consumption at each node due to the operational state of the sensor, and it is given by

$$\sum_{i=1}^N \left[ \pi_S^i E_s + (\pi_N^i + \pi_R^i) E^{(proc)} \right] \quad (22)$$

where  $E_s$  and  $E^{(proc)}$  are the values of energy consumption in sleep mode and in idle mode, respectively (see Section III-A). The other two contributions are (i) the energy required to transmit and receive data units, and (ii) the energy spent during transitions from sleep to active state. They are given by,

$$\sum_{i=1}^N \left[ T^i \sum_{j \in H^i} (E_{i,j} R(i,j) + \pi_{S_0}^i q E^t) \right]. \quad (23)$$

It is also possible to compute the entire distribution of the transfer delay of data units from a given source to the sink, using a technique that we briefly describe in the rest of this section.

We build an additional Markov Chain representing the current location of one individual data unit, generated at a given source, while traversing the network towards the sink. When the data unit is stored into a sensor node, we distinguish five different states shown in the diagram of Figure 12, which represents only a portion of a much larger Markov chain comprising  $5N$  states, plus one state representing the arrival of the data unit at the sink. States labeled as  $Q_W$  and  $Q_F$  are used when the data unit is enqueued into the buffer after other data units waiting for transmission. The subscripts  $W$  and  $F$  have the same meaning described in the sensor model, representing two states in which the next-hops of the current sensor can be. When the data unit comes at the head of the queue, it is ready to be transmitted to another sensor, and it transits to state  $Z_F^5$ .

<sup>5</sup>If the data unit was previously enqueued, it comes at the head of the queue only when a service has been completed, which implies that at least one next-hop is ready to receive.

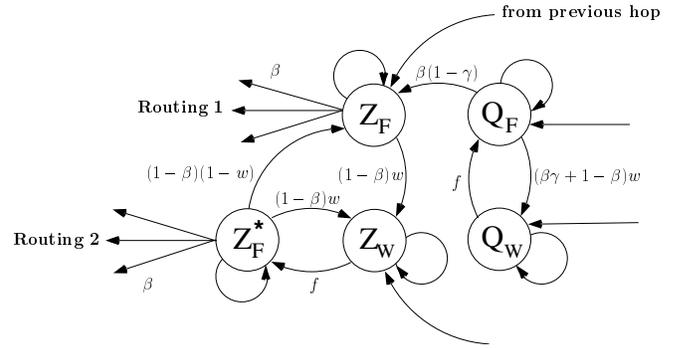


Fig. 12. Portion of the DTMC used to compute the transfer delay distribution representing the states related to the same node

In state  $Z_W$  the data unit is ready to be transmitted, but all next-hops are not available, so it has to wait for one of them to wake up again. When this happens, a transition occurs to state  $Z_F^*$ , which specializes  $Z_F$  during the initial time slot in which one of the next-hops becomes available again. This is done because the routing of the data unit to one of the next-hops is different between states  $Z_F$  and  $Z_F^*$ : from state  $Z_F$  we use the routing probabilities given by (11). In state  $Z_F^*$  we refine these probabilities using the information that at least one next-hop has just become available from a condition in which all of them were not available. In this case the routing probabilities are expressed by

$$R^*(i,j) = K \left[ \prod_{k \in N_{i,j}} \left( 1 - \frac{p\pi_R^k}{\pi_S^k + \pi_N^k} \right) \right] \frac{p\pi_R^j}{\pi_S^j + \pi_N^j}. \quad (24)$$

Transition probabilities among the states of Figure 12 are reported in the diagram, except for self-transitions that can be derived from the others. In the Figure,  $\gamma$  is the parameter of the geometric decay that characterizes the queue length distribution of a sensor, and can be computed from the analysis of the DTMC representing the detailed behavior of a sensor. Finally, notice that the arrival of a data unit at a sensor can occur in any of the states  $Q_F$ ,  $Q_W$ ,  $Z_F$  or  $Z_W$ , with probabilities derived from the stationary probabilities computed by the detailed sensor model.

To obtain the distribution of the transfer delay of a data unit from a given source to the sink, we study the transient behavior of the complete Markov chain described in this section starting from the initial condition in which the data unit is stored at the source. The Markov chain has an absorbing state that is the state in which the data unit arrives at the sink, so as time goes to infinity the probability of this state grows from zero to one. Such probability is also the cumulative distribution of the transfer delay of the data unit. From the cumulative distribution we easily obtain the probability density function (pdf) of the data delivery delay.

## V. RESULTS

In this section we present a collection of results obtained exploring the parameter space of the network scenario described in Section III. Analytical predictions derived from the

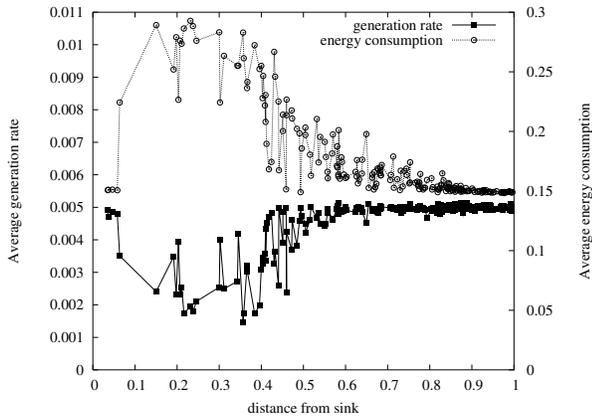


Fig. 13. Unfairness among the network nodes

global system solution presented in Section IV-D are compared against detailed simulations of the same system.

We set the system parameters as follows:  $r = 0.25$ ,  $E^{(amp)} = 0.057$  mJ/slot,  $E^{(ele)} = E^{(proc)} = 0.24$  mJ/slot,  $E^{(sleep)} = 300$  nJ/slot, and  $E^t = 0.48$  mJ. Moreover, unless differently specified, we assume that all sensors generate data and we set the number of nodes to  $N = 400$ , the maximum number of routes available to each sensor to  $M = 6$ , and the sleep/active transition probabilities to  $p = q = 0.1$ . Note that, assuming that a node never enters phase  $N$ , having  $p = q$  corresponds to the case where a sensor spends an equal amount of time in sleep and in active state. Several results are derived under different traffic load conditions. To clearly express the considered values of traffic load, we define a theoretical network load as:  $G = gNq/(p + q)$ , where  $g$  is the sensor generation rate and  $p$  and  $q$  are the sleep/active transition rates. Note that  $G$  represents the sum of all nodes generation rates as if they were in isolation, and only includes parameters that are in input to the system model.

First of all, we show an important phenomenon that is observed when the network load  $G$  is close to 1. Multipoint-to-point communications suffer from the well known problem of data implosion at the destination [7]. Solving this problem was not the scope of this work; thus, in Section III we simply adopted an architectural solution that allows nodes to adapt to traffic conditions avoiding network instability for any value of  $G$ . A drawback of this approach is that nodes closer to the sink generate less data than those far away from the sink, consuming also a larger amount of energy. This results into unfairness among the network nodes, as shown in Figure 13 where the average generation rate and energy consumption are plotted vs. the node distance from the sink, for  $G = 1$  and a particular topology with  $N = 200$ .

Since the maximum theoretical value of network capacity cannot exceed 1 (the sink cannot receive more than one data unit per time slot), it seems reasonable to limit the network load  $G$  to the interval  $(0, 1]$ . Having fixed the value of  $G$  to 1, we investigate what are the actual network capacity  $C$  and the average data delivery delay  $\bar{D}$  that we can obtain for different values of the system parameters. Table II shows the results of this study comparing analytical predictions (in brackets) and

TABLE II

NETWORK CAPACITY AND MEAN DATA DELIVERY DELAY  
(AVERAGING THE RESULTS OF SEVERAL TOPOLOGIES)

	N = 200		N = 400	
	50% S - 50% A	80% S - 20% A	50% S - 50% A	80% S - 20% A
Network capacity ( $C$ )	0,867 (0,873)	0,720 (0,811)	0,968 (0,969)	0,920 (0,930)
Average delay ( $\bar{D}$ )	144,3 (168,0)	393,9 (236,5)	41,5 (43,9)	129,7 (128,8)

simulation results averaged over several different topologies. In all of these experiments  $p=0.1$ ;  $S$  and  $A$  represent the percentage of nodes in sleep and active state, respectively. These results provide a useful indication on the quality of service degradation that we incur when we try to maximize the network capacity. We observe that the network performance is strongly affected by the average number of active nodes in the network, which depends on both the number of deployed sensors ( $N$ ) and the sleep/active dynamics. The model captures quite well the behavior observed by simulation, the major discrepancies appearing when the average number of active nodes is very small and, hence, some of the nodes around the sink are heavily congested.

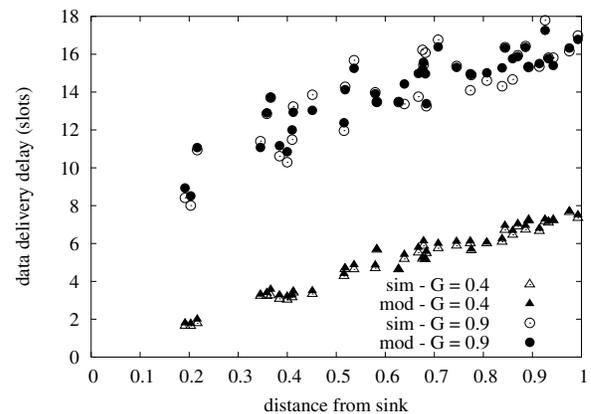


Fig. 14. Average data unit delivery delay vs. the sensor distance from the sink, for varying traffic load conditions. Analytical and simulation results are compared

Next, we present some results obtained by considering the network topology shown in Figure 1 with  $N = 400$ .

Figure 14 shows the average data unit delivery delay expressed in time slots, as a function of the sensor distance from the sink, for  $G = 0.4, 0.9$ . The analytical results (labeled by “mod” in this and in the following plots) closely match the simulation results (labeled by “sim”). The average delay significantly increases as the distance from the sink grows, and as the network load increases. However, once we fix  $G$ , there may be some nodes experiencing a smaller delivery delay than other nodes that are closer to the sink. This is due to the specific considered network topology. Also, we point out that in this case the main contribution to the delivery delay is given by the time spent by the data units in the sensor buffers; in fact we observed that the average number of hops between the sensors and the sink is equal to 3.8 (remind that a one-hop

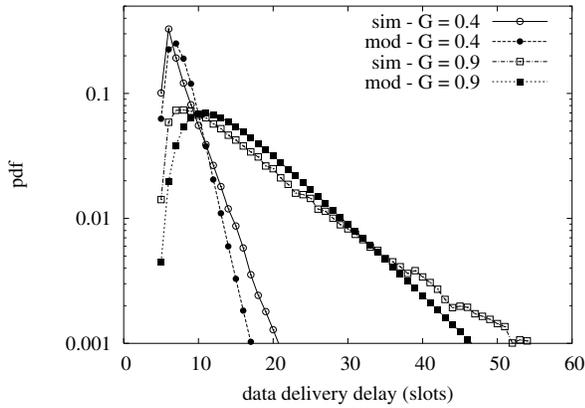


Fig. 15. Probability density function (pdf) of the data delivery delay for the farthest source from the sink, under load conditions. Analytical and simulation results are compared

transmission is completed in one slot).

Figure 15 shows the probability density function (pdf) of the data unit delivery delay expressed in time slots. The delay pdf refers to the farthest source node from the sink. The plot shows the good agreement between the delay distributions resulting from the analytical and simulation studies when  $G = 0.4, 0.9$ . Note that, in order to obtain reliable simulation estimates of the delay distribution, we had to limit the number of traffic sources to 40 by randomly selecting them out of the 400 nodes. In fact, while rare events are accurately predicted by our analytical model, they can be hardly observed via simulation.

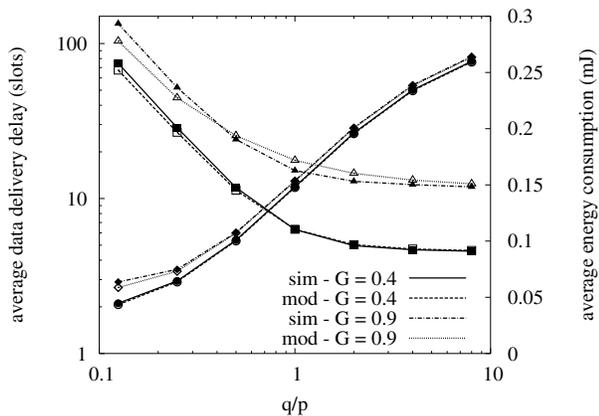


Fig. 16. Trade-off between average data unit delivery delay and average network energy consumption vs.  $q/p$ . Analytical and simulation results are compared for varying load conditions. Triangular and square markers indicate the delay curves, while circles and rhombi denote the energy curves

Figure 16 presents the trade-off between the average network energy consumption and data unit delivery delay, as a function of  $q/p$ . The average delay is obtained through (21), while the average energy consumption is computed using (22) and (23). We set  $p = 0.1$  and  $M = 3$ ; results are presented for two different values of network load, namely  $G = 0.4$  and  $0.9$ . Notice that the average number of active sensors in the networks at a given time slot is strictly related to the value of  $q/p$ . For instance, in the plot  $q/p = 1$  means that on average an equal number of nodes are in sleep and active

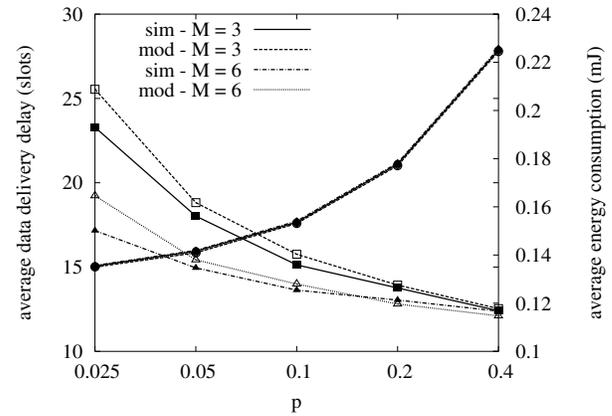


Fig. 17. Trade-off between average data unit delivery delay and average network energy consumption, as a function of the sleep/active transition rates ( $p$  and  $q$  with  $q = p$ ). Analytical and simulation results are compared for different values of the maximum number of available routes ( $M$ )

mode, and the fraction of active sensors grows with increasing values of  $q/p$ . In the plot we use a logarithmic scale for the values of delivery delay and of the abscissa. For low values of  $q/p$  we obtain a small energy expenditure at the expense of a very large delay in data delivery; instead, for values of  $q/p$  greater than 1, the energy consumption increases but the delivery delay is much smaller. Interestingly,  $q/p$  has a greater impact on the delivery delay than on the energy consumption. For example, as  $q/p$  passes from 0.2 to 2, the delay becomes 8 times smaller, while the energy consumption grows by a factor of 4. As for the impact of  $G$ , we observe that the load conditions are relevant to the delay performance, while do not significantly affect the overall energy consumption of the network. In fact, the nodes' energy consumption due to data transmission/reception is much smaller than the total energy expenditure in idle mode; thus the impact of  $G$  is small.

Finally, Figure 17 shows another interesting trade-off between the average data unit delivery delay and the average network energy consumption for  $G = 0.9$ , and  $M = 3, 6$ . The trade-off is presented as a function of the transition rate  $p$  and taking  $p = q$ , in order to study the network performance as the frequency with which sensors pass from sleep to active mode (and vice-versa) varies. As  $p$  increases, the transition frequency grows. First consider  $M = 3$ . We observe that for large values of  $p$  nodes are highly dynamic thus leading to a small delivery delay. However, the more frequent the state transitions, the higher the energy expenditure because of the transition energy cost. On the contrary, when the sensors dynamics are slow (i.e., low values of  $p$ ), we obtain large average delivery delays. We would like to mention that in this case we observed a significant increase also in the variance of the delivery delay. Next consider  $M = 6$ . As expected, the effect on the energy consumption of increasing the number of available routes is negligible. More interestingly, the impact of  $p$  on the delivery delay is very much mitigated by the fact that several routes are now available. In fact, a sensor can poll more next-hops thus increasing its probabilities to forward a data unit through the network, even when the system dynamics are slow.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we considered a sensor network where nodes send their data to a sink node by using multihop transmissions. To save energy, sensors alternate between two operational modes: sleep and active mode. While in sleep mode sensors consume lower power, their functional capabilities are also reduced. We developed an analytical model which enables us to investigate the trade-offs existing between energy saving and system performance, as the sensors dynamics in sleep/active mode vary. We were able to analytically derive several performance metrics, among which the distribution of the data delivery delay. By comparing analytical and simulation results we validated our model and showed the good accuracy of the proposed approach.

To the best of our knowledge, this is the first analytical model that specifically represents the sensor dynamics in sleep/active mode, while taking into account channel contention and routing issues.

The model could be easily modified to take into account some aspects that have not been addressed in this work and that can be interesting subject of future research. For instance, a model of the error process over the wireless channel can be included and some of the assumptions that we made while developing the analytical model, such as those on infinite buffer capacity or on the data generation process at the network nodes, can be modified. Furthermore, we point out that the model can be extended to describe various aspects in the design of sensor networks, such as data aggregation or back-pressure traffic mechanisms. Finally, cluster-based network architectures as well as the case where the network topology varies because some of the sensors run out of energy and die, could be studied.

## REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Mag.*, Aug. 2002, pp. 102–114.
- [2] S. Singh and C. S. Raghavendra, "PAMAS: Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks," *ACM Computer Communication Review*, July 1998, pp. 5–26.
- [3] W. Ye, J. Heidemann, and D. Estrin, "An Energy Efficient MAC Protocol for Wireless Sensor Networks," *IEEE Infocom*, New York, NY, June 2002.
- [4] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Direct Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *IEEE/ACM MobiCom 2000*, Boston, MA, Aug. 2000.
- [5] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing for Wireless Ad Hoc Networks Using Partial Information," *IEEE Personal Comm.*, Feb. 2001.
- [6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *IEEE/ACM MobiCom*, Rome, Italy, July 2001.
- [7] J. Kulik, W. Rabiner Heinzelman, and H. Balakrishnan, "Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks," *ACM/IEEE MobiCom '99*, Seattle, WA, Aug. 1999.
- [8] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *ACM/IEEE MobiCom 2002*, Atlanta, GA, Sep. 2002.
- [9] C. Florens and R. McEliece, "Packet Distribution Algorithms for Sensor Networks," *IEEE Infocom*, San Francisco, CA, Mar. 2003.
- [10] A. Sinha and A. P. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks," *IEEE Design and Test of Computers Magazine*, Vol. 18, No. 2, Mar.-Apr. 2001, pp. 62–74.
- [11] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Topology Management for Sensor Networks: Exploiting Latency and Density," *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.
- [12] R. Zheng, J. Hou, L. Sha, "Asynchronous Wakeup for Power Management in Ad Hoc Networks," *MobiHoc 2003*, Annapolis, MD, June 2003.
- [13] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. on Information Theory*, vol. 46, Mar. 2000.
- [14] S. Shakkottai, R. Srikant, and N. B. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," *IEEE INFOCOM*, San Francisco, CA, Apr. 2003.
- [15] A. F. Mini, B. Nath, and A. A. F. Loureiro, "A Probabilistic Approach to Predict the Energy Consumption in Wireless Sensor Networks," *4th Workshop de Comunicacao sem Fio e Computao Mvel*, So Paulo, Brazil, Oct. 2002.
- [16] A. Ephremides, "Energy Concerns in Wireless Networks," *IEEE Wireless Communications*, Aug. 2002.
- [17] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Imielinski and Korth, Eds., vol. 353, Kluwer Academic Publishers, 1996.
- [18] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *33rd International Conference on System Sciences (HICSS '00)*, Jan. 2000.
- [19] P. Karn, "MACA: A New Channel Access Method for Packet Radio," *9th Computer Networking Conference*, Sep. 1990, pp. 134–140.
- [20] V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *SIGCOMM '94 Conference on Communications, Architectures, Protocols and Applications*, Aug. 1994, pp. 215–225.
- [21] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, 1994.