An Analytical Model for Wireless Sensor Networks with Sleeping Nodes

Carla-Fabiana Chiasserini and Michele Garetto

Abstract—We consider a wireless sensor network whose nodes may enter the so-called *sleep mode*, corresponding to low power consumption and reduced operational capabilities. We develop a Markov model of the network representing: 1) the behavior of a single sensor as well as the dynamics of the entire network, 2) the channel contention among sensors, and 3) the data routing through the network. We use this model to evaluate the system performance in terms of energy consumption, network capacity, and data delivery delay. Analytical results present a very good matching with simulation results for a large variety of system scenarios, showing the accuracy of our approach.

Index Terms—Wireless sensor networks, energy-efficiency, performance evaluation, modeling.

1 Introduction

A typical configuration of sensor networks consists of sensors working unattended and transmitting their observation values to some processing center, the so-called sink node. 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. Therefore, 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; in these cases, making good use of energy is a must. A widely employed energy-saving technique consists in placing idle nodes in *sleep mode*—an operational state characterized by low power consumption and reduced device capabilities. Indeed, in idle mode [1], sensors do not actually receive or transmit, however, 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) can be turned off. Clearly, a trade-off exists between node energy saving and 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: 1) the behavior of a single sensor as well as the dynamics of the entire network, 2) the channel contention among sensors, and 3) the data routing through the network. 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.

1.1 Our Results

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 varies. Note that we do not aim at optimizing any system metric in particular; rather, we want to investigate the existing trade-off between energy consumption and data delivery delay. Although our work mainly focuses on these two metrics, 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 attempt at developing an analytical model that specifically represents the sensor dynamics in sleep/active mode while taking into account channel contention and routing issues.

1.2 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.

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 works in [3], [4], [5], [6] propose wake-up scheduling schemes at the MAC layer which activate sleeping nodes when they need to transmit/receive, thus avoiding a degradation in network connectivity or quality

• *The authors are with the Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy.*
E-mail: {chiasserini, garetto}@polito.it.

Manuscript received 10 Dec. 2004; revised 22 July 2005; accepted 17 Feb. 2006; published online 16 Oct. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0332-1204.

of service provisioning. Also relevant to our work are the numerous network layer schemes that address the problem of data routing in the case where some network nodes are in sleep mode [7], [8].

With regard to analytical studies, results on the capacity of large stationary ad hoc networks are presented in [9] (note that sensor networks can be viewed as large ad hoc networks). The case of tree-like sensor networks is studied in [10], where the authors present optimal strategies for data distribution and data collection and analytically evaluate the time performance of their solution. The problem of data gathering in a sensor network with a sink is also addressed in [11], [12]. The work in [11] proposes a game-theoretic model of data routing where sensors trade off between reliability and communication cost. In [12], the goal is to minimize the transmission cost of transferring data along a routing tree rooted at the sink by applying source coding techniques to reduce information redundancy. An analytical approach to coverage and connectivity of sensor grids is introduced in [13]. Results on coverage and connectivity are derived under the assumption that sensors are unreliable and fail with a certain probability leading to random grid networks. In [14], [15], [16], [17], [18], [19], [20], just to name a few, the problem of field reconstruction in sensor networks is addressed. In particular, the works in [14], [15] present sleep scheduling schemes to maximize the network lifetime while meeting the application requirements on data latency and sensing coverage; the schemes in [16], [17], [18] instead prolong the network duration while guaranteeing that some targets of interest are covered. The studies in [19], [20] investigate the characteristics that a sensor network should have (e.g., node density, clustering) to provide a satisfactory reconstruction of the sensed field. Note that, in our work, we do not tackle the issue of sensing coverage but assume that the network has been designed in such a way that the sensor redundancy is sufficiently high to guarantee an acceptable field reconstruction.

The closest work to ours is in [21], where a large-scale sensor network is studied under the assumption that only a few nodes are active at a time so that the network can be disconnected. By making some simplifying assumptions (namely, the nodes broadcast any data they sense or receive to all their neighbors and interference is neglected), the authors derive bounds on data latency by using percolation theory.

We highlight that the results of the models discussed above and the techniques used there do not directly apply to our model; indeed, our work focuses on randomly located sensors with an active/sleep behavior while taking into account channel contention and routing issues.

1.3 Paper Organization

The remainder of the paper is organized as follows: Section 2 describes the network scenario under study and the assumptions made while constructing our analytical model. In Section 3, we present the Markov model and validate our assumptions by simulation; then, we introduce some performance metrics of interest. Section 4 shows the results obtained by solving the analytical model and compares them to simulation results. Finally, Section 5 provides some conclusions.

2 System Description and Assumptions

We consider a network composed of ν stationary, identical sensor nodes. Sensors are uniformly distributed over a disk of unit radius in the plane and the sink node collecting all information gathered by the sensors is located at the center of the disk; however, different topologies could be considered as well. Here, we do not address the issue of field reconstruction at the sink node and assume that the number of sensors in the network is sufficiently large to ensure a successful reconstruction of the sensed field.

All nodes have a common maximum radio range r and are equipped with omnidirectional antennae; thus, any pair of nodes can communicate if they are within distance r from each other. Upon establishment of a link between two nodes, say i and j, i properly sets its transmission power level so that its transmission can be successfully received by j (no online power control is available, though[1]). Also, we consider network topologies such that, for any sensor, there exists at least one path connecting the sensor to the sink.[2]

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.

2.1 Sensors Behavior

As highlighted in [22], 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:[3] *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 transmit 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 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. An energy cost E^t is associated with each transition from sleep to active mode,

1. This is a fair assumption since typically sensor nodes are simple, low-cost devices.

2. That is, when all nodes are active, the considered network topologies are connected.

3. 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.

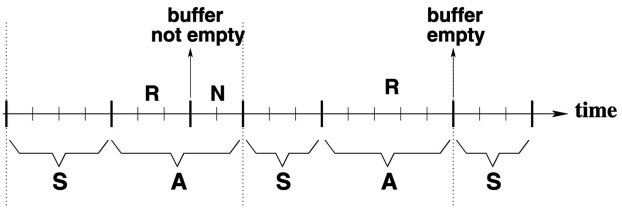

Fig. 1. Temporal evolution of the sensor state.

while the cost of passing from active to sleep mode is neglected [1]. We assume that E^t is twice the energy consumption per time slot in idle mode.[4]

Based on the above observations, we describe the temporal evolution of the sensor state in terms of cycles, as depicted in Fig. 1. Each cycle comprises a sleep phase (S) and an active phase (A). During phase S, the sensor is in sleep mode; 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 periods of sleep and activity, expressed in time slots, are modeled as random variables geometrically distributed with parameter q and p, respectively. Although such an assumption is made for tractability reasons, in Section 4, we show that our framework very well approximates the behavior of sensors going into periodic sleep mode, as is often the case in real systems [23], [3]. Also, in order to handle load variations in time [4], we assume that a sensor prolongs its phase A if its data buffer is not empty when it should enter the sleep phase; in this case, A is extended till all data units in the buffer are forwarded to other nodes. During this additional period of activity, the sensor does not accept relaying new data units nor generating 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 Bernoulli process with parameter g. In phase N, the sensor does not receive or generate data; it can only transmit the data units that are still in its buffer or be idle waiting for a transmission opportunity. In Fig. 1, it is highlighted that A coincides with R when, at the scheduled end time of A, the sensor buffer is empty.

We observe that introducing phase N 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. Finally, we 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.

2.2 Data Routing

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. Since we study 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; as a consequence, the overhead due to the routing scheme can be neglected.

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.[5] For the generic route ρ, the energy cost $e(\rho)$ is computed as follows: Given a node $i \in \rho$, we denote with $\sigma_\rho(i)$ the node immediately succeeding i on ρ (the route includes the source and the relays but not the sink). We have: $e(\rho) = \sum_{i\in\rho} E_{i,\sigma_\rho(i)} = \sum_{i\in\rho}\left(E^{(tx)}_{i,\sigma_\rho(i)} + E^{(rx)}_{\sigma_\rho(i)}\right)$, where $E_{i,\sigma_\rho(i)}$ is the energy cost for transferring a data unit from node i to its next-hop in route ρ, equal to the sum of the transmission energy spent by i $(E^{(tx)}_{i,\sigma_\rho(i)})$ and the reception energy consumed by $\sigma_\rho(i)$ $(E^{(rx)}_{\sigma_\rho(i)})$. As described in Section 2.1, $E^{(rx)}_{\sigma_\rho(i)}$ is due to the transceiver electronics $(E^{(ele)})$ and to processing functions $(E^{(proc)})$; while $E^{(tx)}_{i,\sigma_\rho(i)}$ 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 [24]. Thus, we rewrite $e(\rho)$ as

$$e(\rho) = \sum_{i\in\rho}\left[2\left(E^{(ele)} + E^{(proc)}\right) + d^{\eta}_{i,\sigma_\rho(i)}E^{(amp)}\right], \qquad (1)$$

where $E^{(amp)}$ is a constant value, $d_{i,\sigma_\rho(i)}$ is the distance between i and $\sigma_\rho(i)$ in the disk of unit radius, and η is the exponential decay factor that typically takes values between 2 and 4.

Our model is general enough to deal with different routing schemes; for concreteness, we consider the following strategy: A sensor wishing to transmit a data unit polls its next-hops giving priority to the routes associated with the lowest energy cost until either it finds a next-hop that is ready to receive or it has polled all of the next-hops listed in its routing table. Thus, a sensor dispatches its data units to the best next-hop among the available ones; if none of the stored next-hops is available, a sensor just defers its transmission.[6] We acknowledge that the selected strategy may favor energy saving over data delivery delay; however, as mentioned above, our model allows for the study of different strategies.

2.3 Channel Access

We assume that sensors employ a CSMA/CA mechanism with handshaking, as in the MACA scheme [25] (although,

4. Indeed, the transition cost from sleep to active state is typically very high.

5. The routing table might contain a smaller number of entries if the sensor has less than M neighbors.

6. That is, the fact that a next-hop is found unavailable to receive because it is in sleep mode does not cause its remotion from the sensor routing table.

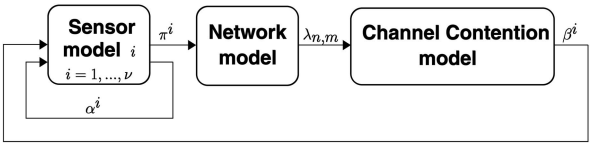

Fig. 2. Close loop used to obtain the global solution of the system.

other MAC protocols could be considered as well), and that the radio range of handshaking messages transmission is equal to r. Thus, a generic node i wishing to transmit to node j ($1 \leq i \leq \nu$, and $0 \leq j \leq \nu$ with 0 indicating the sink) attempts to access the channel if it senses the medium as idle. If so, 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 prevent it from receiving[7]), 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.[8] Note that, if a neighboring node of i, say k, accesses the channel by polling node l, i will detect the channel as busy only if it hears a reply from l; otherwise, after a timeout, i will consider the channel as idle.

We highlight that the transmission of each data unit is assumed to take one time slot, and that this time interval includes the polling phase. Although this is clearly a simplification, it can be justified by considering that, in sensor networks, energy consumption due to protocol overhead is often relatively insignificant when compared to other causes of energy consumption [4].

To conclude, 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.

3 System Model

In this section, we present our modeling approach to analyze the behavior of the sensor network described in Section 2. Our model consists of three building blocks that will be described and validated separately: 1) the *sensor model* (Section 3.1), 2) the *network model* (Section 3.2), and 3) the *channel contention model* (Section 3.3). 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 closed loop is depicted in Fig. 2; the fixed point procedure is explained in detail in Section 3.4. In Section 3.5, we describe the performance metrics that can be obtained by solving the proposed model.

3.1 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,

7. The possibility to exploit the capture effect is not considered.

8. Alternatively, it can be assumed 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-hops until a timeout expires.

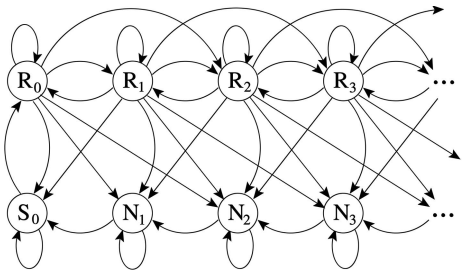

Fig. 3. Markov chain describing the sensor behavior.

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: 1) the cycle phase in which the sensor is in the current time slot (namely, S, R, or N) and 2) the number of data units stored in the sensor buffer, which can be any integer value ranging from 0 to ∞. The resulting Markov chain is shown in Fig. 3, where the different phases are indexed with the number of data units stored in the sensor buffer.

Let $\boldsymbol{P}$ 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:

1. the sensor sleep-active dynamics, determined by the input parameters p and q (introduced in Section 2.1),
2. the data unit generation process (in phase R only), in which we denote with g the probability that a data unit is generated by the sensor in a time slot,
3. the reception of data units from neighboring nodes (in phase R only), in which we indicate with α the probability that a data unit is received in a time slot, and
4. the data unit transmission (in phases R and N only), in which we denote with β the probability that a data unit is transmitted in a time slot.

While p, q, and g are input parameters to the model, α and β need to be estimated; for each sensor, we have $\alpha + \beta \leq 1$, since a node cannot transmit and receive simultaneously. For the sake of simplicity, we assume that p, q, and g have the same value for all sensors; however, the extension to the case where they are node dependent is straightforward. The parameters α and β, instead, depend on the particular node that is considered; later in the paper, when the interaction among several nodes is studied, we will add the sensor index as an apex to the α and β notations.

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.

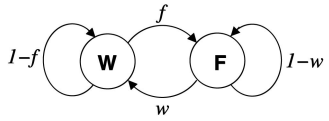

Fig. 4. DTMC model describing the behavior of the sensor next-hops.

TABLE 1
Transition Probabilities from Empty Buffer (for the Sake of Brevity, $l_0 = (1-\alpha)(1-g)$, $b_0 = g(1-\alpha) + \alpha(1-g)$)

Q_i	Q_j	$P(Q_i, Q_j)$
R_0^W	R_0^F	$f\ l_0\ (1-p)$
	S_0^F	$f\ l_0\ p$
	S_0^W	$(1-f)\ l_0\ p$
	R_1^W	$(1-f)\ b_0\ (1-p)$
	R_1^F	$f\ b_0\ (1-p)$
	S_1^F	$f\ b_0\ p$
	S_1^W	$(1-f)\ b_0\ p$
	R_2^W	$(1-f)\ g\ \alpha\ (1-p)$
	R_2^F	$f\ g\ \alpha\ (1-p)$
	S_2^F	$f\ g\ \alpha\ p$
	S_2^W	$(1-f)\ g\ \alpha\ p$
R_0^F	R_0^W	$w\ l_0\ (1-p)$
	S_0^W	$w\ l_0\ p$
	S_0^F	$(1-w)\ l_0\ p$
	R_1^F	$(1-w)\ b_0\ (1-p)$
	R_1^W	$w\ b_0\ (1-p)$
	S_1^W	$w\ b_0\ p$
	S_1^F	$(1-w)\ b_0\ p$
	R_2^F	$(1-w)\ g\ \alpha\ (1-p)$
	R_2^W	$w\ g\ \alpha\ (1-p)$
	S_2^W	$w\ g\ \alpha\ p$
	S_2^F	$(1-w)\ g\ \alpha\ p$
S_0^W	R_0^F	$f\ q$
	R_0^W	$(1-f)\ q$
	S_0^F	$f\ (1-q)$
S_0^F	R_0^W	$w\ q$
	R_0^F	$(1-w)\ q$
	S_0^W	$w\ (1-q)$

TABLE 2
Transition Probabilities from Nonempty 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	$(1-w)\beta(1-g)p$	
	N_{i-1}^W	$w\beta(1-g)p$	$i \geq 2$
	N_{i-1}^F	$(1-w)\beta(1-g)p$	
R_i^W	R_i^F	$fl_0(1-p)$	$i \geq 1$
	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	$w\beta$	
	S_0^F	$(1-w)\beta$	$i = 1$
	S_0^W	$w\beta$	
N_i^W	N_i^F	f	$i \geq 1$

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 successfully receive provided that channel contention conditions allow it. We assume that the evolution of the state of each next-hop is independent of the others; this is a reasonable assumption as shown by the good agreement between the analytical and simulation results presented in the following. Transitions between W and F are modeled by the two-state DTMC shown in Fig. 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 Fig. 3. Tables 1 and 2 report the transition probabilities $P(s_o, s_d)$ from state s_o, listed in the first column, to the successor state s_d, listed in the second column. Self transitions are not reported in the tables due to the lack of space; however, they can be easily derived from the other transitions. In Table 2, the fourth column denotes the conditions that 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 buffered 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 β 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 $\boldsymbol{\pi} = \{\pi_s\}$, where s is a generic state of the model. By solving the sensor model, we obtain $\boldsymbol{\pi}$ and derive the following metrics:

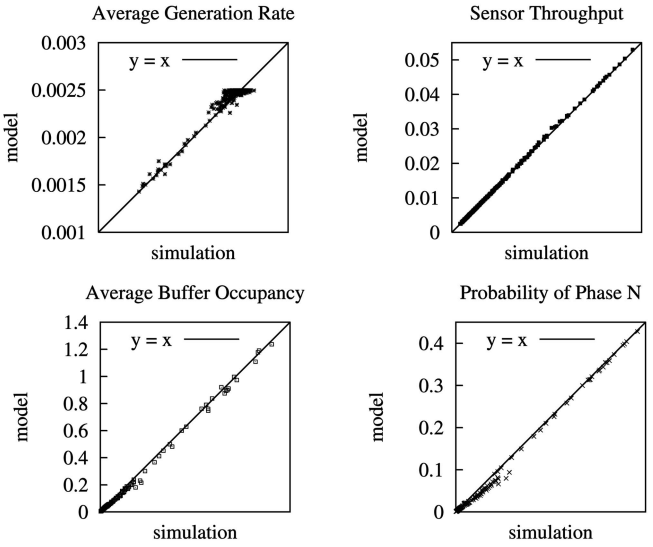

Fig. 5. Validation of the sensor model on the reference scenario.

- The average number, Λ, of data units generated in a time slot,

$$\Lambda = \sum_{i=0}^{\infty} \left(\pi_{R_i^F} + \pi_{R_i^W} \right) g. \tag{2}$$

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

$$T = \sum_{i=1}^{\infty} \left[\pi_{R_i^F} + \pi_{N_i^F} \right] \beta. \tag{3}$$

- The overall probabilities, π_R, π_S, and π_N, that a sensor is in the corresponding phases, R, S, and N.
- The average buffer occupancy,

$$\overline{B} = \sum_{k=1}^{\infty} \left[k \left(\pi_{R_k^F} + \pi_{R_k^W} + \pi_{N_k^F} + \pi_{N_k^W} \right) \right]. \tag{4}$$

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 α, β, 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 (2)-(4). We then verify whether the values of the above metrics match those obtained by simulation.

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

Fig. 5 shows four plots derived with generation rate $g = 0.005$,[9] 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 labels "model" and "simulation" on the plot axes refer to analytical and simulation results, respectively. The alignment of the points on the bisector $y = x$ proves the accuracy of the sensor model.

9. We highlight that $g = 0.005$ corresponds to a heavy network load condition; indeed, g is the generation rate of a single sensor node.

3.2 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, recall that data units are never lost while traversing the network. Thus, given the average generation rate Λ^i of the generic sensor i, the total arrival rate at the sink, that is, the network capacity C, is given by: $C = \sum_{i=1}^{\nu} \Lambda^i$.

Then, our goal is to derive the total traffic forwarded by a sensor, denoted by T^i, which includes both the data units that are locally generated and the data units that are received from other network nodes. Given the average generation rates Λ^is, this can be done by solving the system of flow balance equations:

$$\boldsymbol{T} = \boldsymbol{T}\boldsymbol{R} + \boldsymbol{\Lambda}, \tag{5}$$

where $\boldsymbol{T}$ and $\boldsymbol{\Lambda}$ are row vectors stacking T^is and Λ^is, respectively, and $\boldsymbol{R}$ 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 $\boldsymbol{R}$, 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 2. The simplest approach is to consider only the stationary probabilities of the next-hops states and to assume that the next-hops state are independent; furthermore, we neglect the impact of channel contention. Following this approach, the transition probability $R(i,j)$ can be computed as

$$R(i,j) = K \left(\prod_{m \in N_{i,j}} (\pi_S^m + \pi_N^m) \right) \pi_R^j, \tag{6}$$

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 js 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.

To validate our approach, we take from simulation the average generation rates Λ^is and the state probabilities of the network nodes and compute the transition matrix $\boldsymbol{R}$ using (6). Then, we derive the T^is by solving (5) and compare them to those obtained by simulation. As shown in Fig. 6 for a network load equal to 0.6, our analytical results (labeled by "model") are very close to those derived by simulation (labeled by "simulation"). Note that each point in the plot stands for an element of vector $\boldsymbol{T}$: The closer the

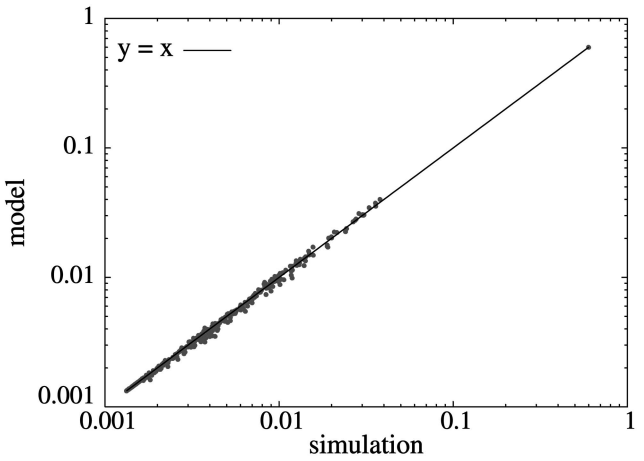

Fig. 6. Comparison of the total traffic rate, T^i, of each sensor between simulation and analysis.

point to the bisector, the better the agreement between analysis and simulation.

3.3 Channel Contention Model

The purpose of the channel contention model is to compute, for the generic node i $(1 \leq i \leq \nu)$, the parameter β^i to be used into the sensor model presented in Section 3.1. The method used to estimate the other parameters α^i, f^i, and w^i needed to solve the sensor model[10] will be described in Section 3.4.

Recall that β^i has been defined as the probability that node i transmits a data unit in a time slot given that its buffer is not empty and at least one of its next-hops is in phase R at the beginning of the slot. If there were no contention on the wireless channel, β^i would be equal to 1 $(1 \leq i \leq \nu)$. We model channel contention as described in Section 2.3; thus, the computation of β^i requires a careful investigation of channel contention between node i and other nearby sensors wishing to transmit. In order to explain our approach, consider the set of nodes shown in Fig. 7. The transmission range of three nodes, {A, F, H}, is represented by a circle. Assume that we want to estimate the parameter β^A; notice that A has two next-hops, B and C. We need to find all transmissions that could potentially compete 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 note that transmissions like (D, E) and (H, C) compete with A's transmission since receivers E and C 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) compete with A since they prevent A's next-hops from receiving. 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*, while 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*.

To estimate β^i for the generic sensor i, we proceed as follows: First, we compute for each node n $(1 \leq n \leq \nu)$ the

10. Recall that we now add the sensor index as an apex when we refer to a particular node.

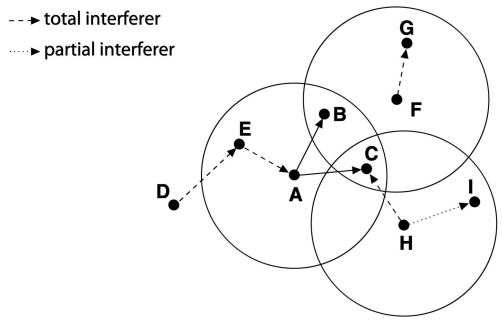

Fig. 7. Example of channel contention and hindered transmissions.

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, where $\lambda_{n,m} = T^n R(n,m)$. We write

$$I^i(n) = \sum_{m=1, m\neq i}^{\nu} \lambda_{m,n} 1_{\{d_{(n,i)} \leq r\}} + \sum_{m=0}^{\nu} \lambda_{n,m} 1_{\{d_{(m,i)} > r\}} V^i(n) C^i(n), \tag{7}$$

where $m = 0$ denotes the sink and $1_{\{\cdot\}}$ is the indicator function. Observe that, considering the example above, the first summation on the right hand side of (7) accounts for transmissions like (D, E) and (H, C) or (E, A), while the second summation accounts for transmissions like (F, G) and (H, I). 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:

$$V^i(n) = \begin{cases} 1 & \exists k \in H^i : d_{(n,k)} \leq r \ , \ n \neq i \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

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,

$$C^i(n) = \begin{cases} \prod_k (\pi_S^k + \pi_N^k) \ \forall k \in H^i : d_{(n,k)} > r & \text{if any} \\ 1 & \text{else.} \end{cases} \tag{9}$$

Then, β^i is estimated as follows:

$$\beta^i = \prod_{n=1}^{\nu} \left[1 - I^i(n)\right]. \tag{10}$$

To validate our estimate of β^i, we take from simulation all transmission rates $\lambda_{n,m}$s. Since β^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 (7) and (10), we computed the parameter β^i for each sensor. Analytical results (labeled by "mod") and simulation results (labeled by "sim") are shown in Fig. 8 as functions of the distance from the sink; they present a good

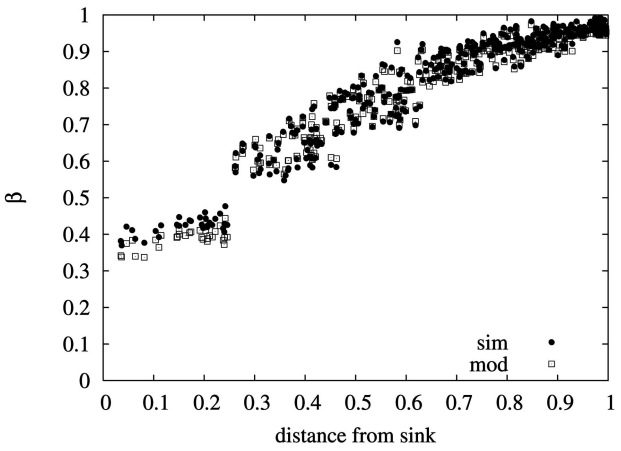

Fig. 8. Estimation of β^i using conditioned transmission rates obtained from simulation for the various network nodes $(1 \leq i \leq \nu)$.

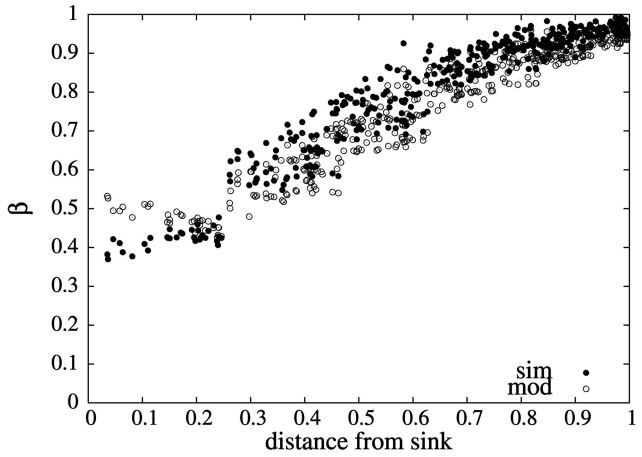

Fig. 9. Estimation of β^i using unconditioned transmission rates computed by the model for the various network nodes $(1 \leq i \leq \nu)$.

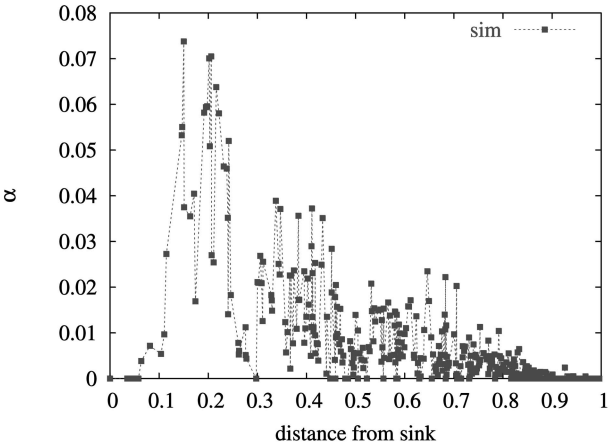

Fig. 10. Probability that a data unit is received from neighboring nodes in a time slot (parameter α^i of the sensor model) for the various network nodes $(1 \leq i \leq \nu)$.

agreement proving that our approach to estimating β^i as in (10) is correct.

Unfortunately, the conditioned transmission rates seem to be hard to evaluate analytically. Thus, we resorted to the unconditioned rates $\lambda_{n,m}$s provided by the network model, and slightly refined the channel contention model for the nodes within the sink radio range in order to account for the neglected correlation between the $\lambda_{n,m}$s and the state of the sensor for which β^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

$$t_{A_i} = \frac{1}{N_{A_i}} \sum_{k \in A_i} (\pi_N^k + \pi_R^k - \pi_{R_0}^k), \tag{11}$$

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 β^i as

$$\beta^i = \sum_{k=0}^{N_{A_i}} \frac{1}{k+1} \binom{N_{A_i}}{k} t_{A_i}^k (1 - t_{A_i})^{N_{A_i}-k} \prod_{n \notin A_i} [1 - I^i(n)]. \tag{12}$$

For each sensor i whose distance from the sink is greater than r, β^i is estimated using (10) which provides an accurate solution for these nodes even if we plug in the unconditioned transmissions rates $\lambda_{n,m}$. In doing so, we obtain the values of β^i labeled in Fig. 9 as "mod" and compare them with the simulation results already presented in Fig. 8 (labeled in the plot as "sim"). The plot shows that our estimation of β^i is quite accurate. In fact, even if our approach tends to overestimate β^i 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. Fig. 10 proves this statement showing the probability that a data unit is received from neighboring nodes in a time slot (parameter α^i of the sensor model) as a function of the node distance from the sink, according to simulation.

3.4 Fixed Point Approximation

The three building blocks of the model described in Sections 3.1, 3.2, and 3.3 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 Fig. 2.

The procedure starts with the solution of the DTMC representing the individual behavior for each sensor i $(1 \leq i \leq \nu)$ from which we obtain the stationary distribution probabilities π^is.[11] Then, we run the network model and derive the data rates $\lambda_{n,m}$s between each pair of nodes in the network, as well as the expected throughput for each sensor (i.e., T as in (5)). The data rates $\lambda_{n,m}$s are given as inputs to the channel contention model to estimate the

11. 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: $\Lambda^i = g\frac{q}{\pi^i_{1,0}}$.

parameter β^i for each sensor i ($1 \leq i \leq \nu$). On their turn, the β^is are given as input to the sensor models, thus closing the loop.

Within each sensor model, given the value of β^i and employing a numerical technique, we derive the unknown parameter α^i. α^i is estimated so that the sensor throughput given by (3) approximates the value previously predicted by the network model using (5). In Fig. 2, this procedure is highlighted by the inner loop around the block of the sensor model. We point out that obtaining a precise estimate of α^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 3.1 and Fig. 4). We compute the stationary probability of state W for sensor i as follows: $\pi^i_w = \prod_{k \in H^i}(\pi^k_S + \pi^k_N)$, using the most recent estimate of the stationary probabilities of the sensor next-hops. The transition probability f^i is estimated as

$$f^i = 1 - \prod_{k \in H^i}\left(1 - p\frac{\pi^k_R}{\pi^k_S + \pi^k_N}\right), \tag{13}$$

where $p\frac{\pi^k_R}{\pi^k_S+\pi^k_N}$ 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: $w^i = f^i\frac{\pi^i_w}{1-\pi^i_w}$. Once we have numerically solved the DTMC of each sensor [26], we can compute all metrics of interest described at the end of Section 3.1 and, in particular, a new estimate of the generation rates Λ^is (using (2)) 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.

The Markov chain describing each sensor node has $4B$ states, where B is the node buffer size. Using an approximate matrix geometric technique, we can reduce the number of states to $4L$, where L is a parameter to be chosen (essentially, we assume that, for queue length larger than L, the stationary state probability distribution decays geometrically with the queue length). Experimentally, we have found that $L = 5$ leads to an approximate solution very close to the exact one. Furthermore, we observe that the FPA procedure exhibits good convergence properties under low-medium traffic load conditions: Less than 10 iterations are usually required to have the worst relative error fall below a threshold of 10^{-4}. Under high traffic load, however, we need to impose in the model solution that $\alpha^i + \beta^i \leq 1$, $1 \leq i \leq \nu$, to obtain convergence.

3.5 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 3.1. 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 $\overline{D}$, 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

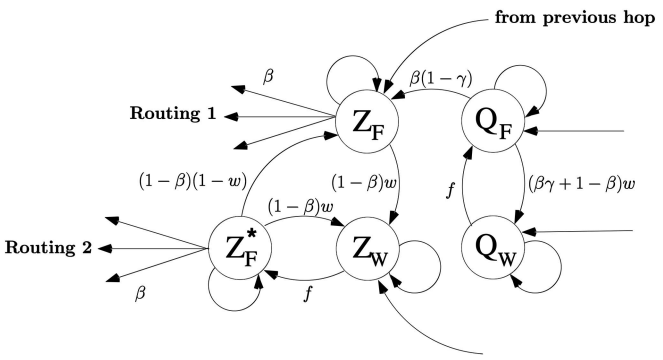

Fig. 11. Portion of the DTMC used to compute the transfer delay distribution representing the states related to the same node.

$$\overline{D} = \frac{\sum_{i=1}^{\nu}\overline{B}^i}{C}, \tag{14}$$

where $\overline{B}^i$ is the average buffer occupancy at sensor i (see (4)). The network energy consumption per time slot $\overline{E}$ can be divided into three contributions. The first one is the sum of the energy consumption at each sensor, which depends on the node operational state; it is given by

$$\sum_{i=1}^{\nu}\left[\pi^i_S E_s + (\pi^i_N + \pi^i_R)E^{(proc)}\right], \tag{15}$$

where E_s and $E^{(proc)}$ are the values of energy consumption in sleep mode and in idle mode, respectively (see Section 2.1). The other two contributions are 1) the energy required to transmit and receive data units and 2) the energy spent during transitions from sleep to active state. They are given by

$$\sum_{i=1}^{\nu}\left[T^i\sum_{j \in H^i}\left(E_{i,j}R(i,j)\right) + \pi^i_{S_0}q\ E^t\right]. \tag{16}$$

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 transfer of one individual data unit, generated at a given source, toward the sink. The Markov chain includes 5ν states, plus one state representing the arrival of the data unit at the sink. For the sake of clarity, let us focus on a data unit stored into a generic sensor node; we can identify five different states which are shown in the diagram of Fig. 11. States labeled as Q_W and Q_F represent the data unit enqueued into the sensor buffer after other data units, waiting to be transmitted. The subscripts W and F have the same meaning described in Section 3.1, representing the 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 forwarded to another node, and it transits to

state Z_F. Indeed, 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. If the data unit is not forwarded because of channel contention (which occurs with probability $1-\beta$) and all next-hops become unavailable, the data unit transits to state Z_W. 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. When this happens, a transition occurs to state $Z_F^\star$, which specializes Z_F during the initial time slot in which one of the next-hops becomes available again. This specialization is done because the routing of the data unit to one of the next-hops is different between states Z_F and $Z_F{}^\star$: From state Z_F, we use the routing probabilities given by (6); in state $Z_F^\star$, 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. In the latter case, the routing probabilities are expressed by

$$R^\star(i,j) = K\left[\prod_{k\in N_{i,j}}\left(1-\frac{p\pi_R^k}{\pi_S^k+\pi_N^k}\right)\right]\frac{p\pi_R^j}{\pi_S^j+\pi_N^j}. \qquad (17)$$

Transition probabilities among the states of Fig. 11 are reported in the diagram, except for self-transitions that can be derived from the others. In the figure, γ is the parameter of the geometric decay that characterizes the queue length distribution at a sensor and can be computed from the analysis of the DTMC representing the detailed behavior of a node. (Here, the calculation is omitted due to lack of space; the interested reader can refer to [27] for further details on the used methodology.) Finally, 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.

4 Results

In this section, we present a collection of results obtained exploring the parameter space of the network scenario described in Section 2. Analytical predictions derived from the global system solution presented in Section 3.4 are compared against detailed simulations of the network system.

We have used an ad hoc, discrete-time simulator, where the duration of the sensor's active and sleep phases are deterministic and equal to $1/p$ and $1/q$, respectively. With regard to channel access and data routing, the same assumptions are made as described in Section 2. However, it is not just a simulation of the analytical model presented in Section 3 which would be trivial. Our sensor network can, in theory, be described by a discrete time Markov process in which the state at time k is represented by a vector $\boldsymbol{S}[k]=\{s_1[k],s_2[k],\ldots,s_\nu[k]\}$, where $s_i[k]$, the state of node i at time k, is one of the states depicted in Fig. 3. Unfortunately, the number of states of the resulting Markov chain that we refer to as the "exact model" grows exponentially with the number of nodes in the network. Even considering finite-size buffers, the complexity of the exact model would be prohibitive. For example, in the simplest case in which the buffers can store a single data unit, we would have 4^ν different states (each sensor could be in one of states $\{R_0,R_1,S_0,N_1\}$) so that only networks comprising very few nodes could be studied exactly.[12] Our analytical work is actually meant to demonstrate that accurate predictions about the system behavior can be obtained with limited computational complexity even for very large systems. To prove that, we simulate the theoretical exact model of the system and compare the results with those obtained by the approximate model described in Section 3. In the following, we report some details about the simulator implementation.

At the beginning of each simulation, we generate a random topology that, when all sensors are active, is connected at the physical layer. We then apply the Dijkstra algorithm to compute once and for all the best M routes available to each node according to the energy cost defined in Section 2.2. To simulate the ideal MAC protocol described in Section 2.3, we adopt the following strategy: At the beginning of each time slot, all sensors can potentially transmit or receive data units during the slot. To solve the contention on the channel, we extract a random permutation of the indexes $1,2,\ldots,\nu$ associated to the sensor nodes. Then, we inspect each sensor according to the order resulting from the permutation. If the sensor is able to transmit a data unit to one of its next-hops, it is allowed to do so. Moreover, as a consequence of the CSMA/CA mechanism with handshaking, all nodes within radio range of the transmitting sensor are no more allowed to receive until the beginning of the next slot. Similarly, all nodes within radio range of the receiving next-hop are not allowed to transmit during the current slot. In this way, we randomly and fairly choose the transmissions that take place in the network during each time slot.

We set the system parameters as follows: $r=0.25$, $\eta=2$, $E^{(amp)}=0.057$ mJ/slot, $E^{(ele)}=E^{(proc)}=0.24$ mJ/slot, $E^{(sleep)}=300$ nJ/slot, and $E^t=0.48$ mJ. 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: $G=g\nu q/(p+q)$, 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.

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,

12. When $\nu=10$, we would already have more than one million states.

TABLE 3
Network Capacity and Mean Data Delivery Delay
(Averaging the Results of Several Topologies)

	ν = 200		ν = 400	
	50% S – 50% A	80% S – 20% A	50% S – 50% A	80% S – 20% A
Network capacity *(C)*	*0.877 (0.873)*	*0.736 (0.811)*	*0.956 (0.969)*	*0.895 (0.930)*
Average delay *($\overline{D}$)*	*101.4 (168.0)*	*262.78 (236.5)*	*46.13 (43.9)*	*84.25 (128.8)*

we investigate the actual network capacity C and the average data delivery delay $\overline{D}$ that we can obtain for different values of the system parameters. Table 3 shows the results of this study comparing analytical predictions (in brackets) and simulation results averaged over several different topologies. In all of these experiments, $p = 0.1$ and $M = 6$; 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 (ν) 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.

Next, we present some results obtained by considering $\nu = 400$.

Fig. 12 shows the average data unit delivery delay expressed in time slots as a function of the sensor distance from the sink for $G = 0.4$ and 0.9. We set $p = q = 0.1$ and $M = 6$. 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 (recall that a one-hop transmission is completed in one slot).

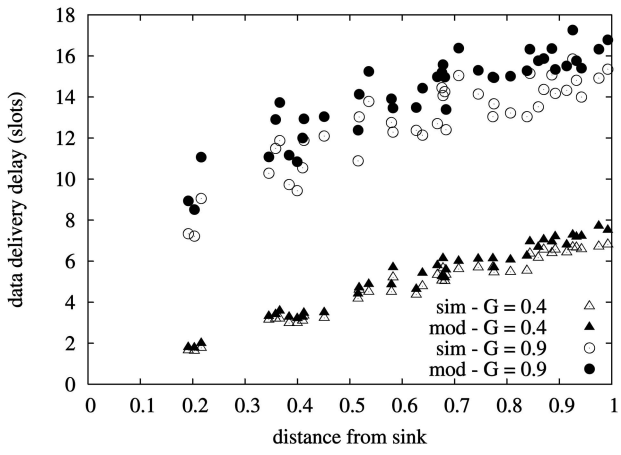

Fig. 12. Average data unit delivery delay versus the sensor distance from the sink for varying traffic load conditions. Analytical and simulation results are compared.

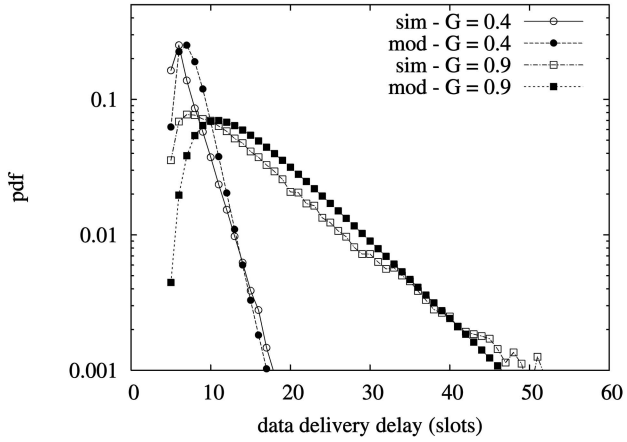

Fig. 13. Probability density function (pdf) of the data delivery delay for the farthest source from the sink under different load conditions. Analytical and simulation results are compared.

Fig. 13 shows the probability density function (pdf) of the data unit delivery delay expressed in time slots for $p = q = 0.1$ and $M = 6$. 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 $G = 0.4$ and 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. 14 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 (14), while the average energy consumption is computed using (15) and (16). In the plot, we use a logarithmic scale for the values of delivery delay and of the abscissa. Notice that the average number of active sensors in the network at a given time slot is strictly related to the value of q/p. 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. For instance, $q/p = 1$ means that on average an equal number of nodes are in sleep and active mode, and the fraction of active sensors grows with increasing values of q/p. 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 eight times smaller, while the energy consumption grows by a factor of 4. As for the impact of G, we observe that the

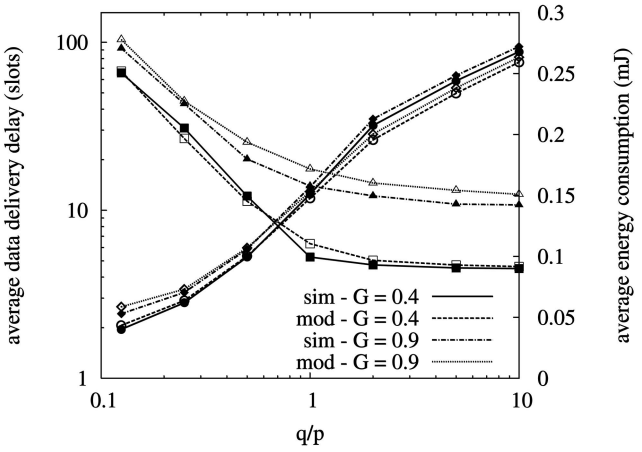

Fig. 14. Trade-off between average data unit delivery delay and average network energy consumption versus $q = p$. Analytical and simulation results are compared for varying load conditions. Triangular and square markers indicate the delay curves, while circles and rhombi denote the energy curves.

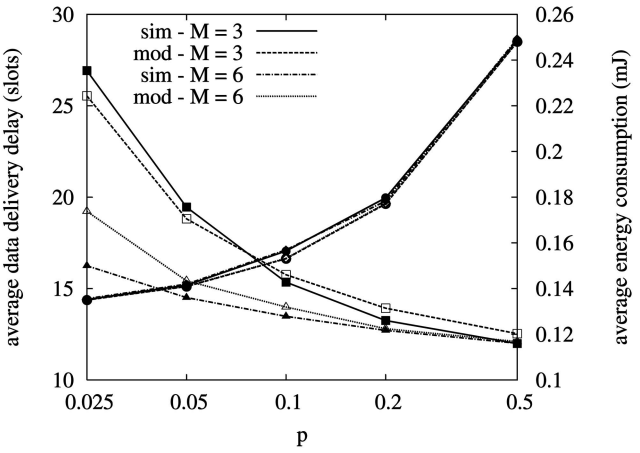

Fig. 15. Trade-off between average data unit delivery delay and average network energy consumption as a function of the sleep/active transition rates (p and q with $q = p$). Analytical and simulation results are compared for different values of the maximum number of available routes (M). Triangular and square markers indicate the delay curves, while circles and rhombi denote the energy curves.

load conditions are relevant to the delay performance, while they 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, Fig. 15 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 probability to forward a data unit through the network, even when the system dynamics are slow.

5 Conclusions

We considered a sensor network where nodes send their data to a sink node by using multihop transmissions. Sensors alternate between sleep and active mode; while in sleep mode, sensors consume low power but they cannot receive or transmit. We developed an analytical model which enables us to evaluate the network performance in terms of several performance metrics, among which the distribution of the data delivery delay. We showed the good accuracy of our model by comparing analytical and simulation results. Furthermore, we were able to investigate the trade-offs existing between energy saving and system performance, as the sensors dynamics in sleep/active mode vary. The model can be easily modified to account for some aspects that have not been addressed in this work, such as the error process over the wireless channel and the limited buffer capability available at the sensor nodes.

Acknowledgments

This work was supported by the Italian Ministry of University and Research through the PRIMO project.

References

[1] A. Sinha and A.P. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks," *IEEE Design and Test of Computers,* vol. 18, no. 2, pp. 62-74, Mar.-Apr. 2001.

[2] S. Singh and C.S. Raghavendra, "PAMAS: Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks," *ACM Computer Comm. Rev.,* pp. 5-26, July 1998.

[3] W. Ye, J. Heidemann, and D. Estrin, "An Energy Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM,* June 2002.

[4] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient Mac Protocol for Wireless Sensor Networks," *Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03),* 2003.

[5] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Topology Management for Sensor Networks: Exploiting Latency and Density," *Proc. MobiHoc '02,* June 2002.

[6] R. Zheng, J. Hou, and L. Sha, "Asynchronous Wakeup for Power Management in Ad Hoc Networks," *Proc. MobiHoc '03,* June 2003.

[7] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing for Wireless Ad Hoc Networks Using Partial Information," *IEEE Personal Comm.,* Feb. 2001.

[8] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. MobiCom,* July 2001.

[9] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory,* vol. 46, Mar. 2000.

[10] C. Florens and R. McEliece, "Packet Distribution Algorithms for Sensor Networks," *Proc. IEEE INFOCOM,* Mar. 2003.

[11] R. Kannan, S. Sarangi, S.S. Iyengar, and L. Ray, "Sensor-Centric Quality of Routing in Sensor Networks," *Proc. IEEE INFOCOM,* Apr. 2003.
[12] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On Network Correlated Data Gathering," *Proc. IEEE INFOCOM,* Mar. 2004.
[13] S. Shakkottai, R. Srikant, and N.B. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," *Proc. IEEE INFOCOM,* Apr. 2003.
[14] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* Apr. 2005.
[15] W. Choi and S.K. Das, "A Novel Framework for Energy-conserving Data Gathering in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1985-1996, Mar. 2005.
[16] M. Cardei, M.T. Thai, Y. Li, and W. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1976-1984, Mar. 2005.
[17] M.X. Cheng, L. Ruan, and W. Wu, "Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 2638-2645, Mar. 2005.
[18] H. Liu, P. Wan, C.-W. Yi, X. Jia, S. Makki, and N. Pissinou, "Maximal Lifetime Scheduling in Sensor Surveillance Networks," *Proc. IEEE INFOCOM,* pp. 2482-2491, Mar. 2005.
[19] R. Cristescu and M. Vetterli, "On the Optimal Density for Real-Time Data Gathering of Spatio-Temporal Processes in Sensor Networks," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* Apr. 2005
[20] Y. Sung, L. Tong, and H.V. Poor, "Sensor Activation and Scheduling for Field Detection in Large Sensor Arrays," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* Apr. 2005.
[21] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms," *Proc. MobiHoc,* June 2004.
[22] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine,* pp. 102-114, Aug. 2002.
[23] F. Bennett, D. Clarke, J.B. Evans, A. Hopper, A. Jones, and D. Leask, "Piconet: Embedded Mobile Networking," *IEEE Personal Comm. Magazine,* vol. 4, no. 5, pp. 8-15, Oct. 1997.
[24] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Int'l Conf. System Sciences (HICSS '00),* Jan. 2000.
[25] P. Karn, "MACA: A New Channel Access Method for Packet Radio," *Proc. Ninth Computer Networking Conf.,* pp. 134-140, Sept. 1990.
[26] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains.* Princeton Univ. Press, 1994.
[27] W.J. Neuts, *Matrix Geometric Solutions in Stochastic Models.* Dover, 1981.

Carla-Fabiana Chiasserini received a summa cum laude degree in electrical engineering from the University of Florence in 1996. She did her graduate work at the Politecnico di Torino, Italy, receiving the PhD degree in 1999. Since then, she has been with the Department of Electrical Engineering at Politecnico di Torino, where she is currently an assistant professor. Since 1999, she has worked as a visiting researcher at the University of California, San Diego. Her research interests include architectures, protocols, and performance analysis of wireless networks for integrated multimedia services. She is a member of the editorial board of the *Ad Hoc Networks Journal* (Elsevier) and has served as an associate editor of the *IEEE Communications Letters* since 2004.

Michele Garetto (S'01-M'04) received the master's degree in telecommunication engineering and the PhD degree in electronic and telecommunication engineering, both from Politecnico di Torino, Italy, in 2000 and 2004, respectively. In 2002, he was a visiting scholar with the Networks Group of the University of Massachusetts, Amherst, and, in 2004, he held a postdoctoral position in the Electrical and Computer Engineering Department at Rice University, Houston. His research interests are in the field of performance evaluation of wired and wireless communication networks.