Cost- and Collision-Minimizing Forwarding Schemes for Wireless Sensor Networks: Design, Analysis, and Experimental Validation

Michele Rossi, *Member*, *IEEE*, Nicola Bui, and Michele Zorzi, *Fellow*, *IEEE*

Abstract—The paper presents an original integrated MAC and routing scheme for wireless sensor networks. Our design objective is to elect the next hop for data forwarding by jointly minimizing the amount of signaling to complete a contention and maximizing the probability of electing the best candidate node. Toward this aim, we represent the suitability of a node to be the relay by means of locally calculated and generic cost metrics. Based on these costs, we analytically model the access selection problem through dynamic programming techniques, which we use to find the optimal access policy. Hence, we propose a contention-based MAC and forwarding technique, called Cost- and Collision-Minimizing Routing (CCMR). This scheme is then thoroughly validated and characterized through analysis, simulation, and experimental results.

Index Terms—Routing protocols, distributed applications, algorithm/protocol design and analysis, dynamic programming, wireless sensor networks.

✦

---

1 INTRODUCTION AND RELATED WORK

FORWARDING algorithms for Wireless Sensor Networks (WSNs) should be *simple*, as sensor nodes are inherently resource constrained. Moreover, they should also be *efficient* in terms of energy consumption and quality of the paths that are used to route packets toward the data gathering point (referred to here as sink). A trend in recent research is to select the next hop for data forwarding locally [1], [2], [3], [4], [5], [6] and without using routing tables. Such a localized neighbor election is aimed at minimizing the overhead incurred in creating and maintaining the routing paths. Often, nodes are assumed to know their geographical location. Such a knowledge can be exploited to implement online routing solutions where the next hop is chosen depending on the advancement toward the sink. However, in addition to the maximization of the advancement, other objectives such as the maximization of residual energies should be taken into account. The schemes in [1], [2], [3], [4], [5], [6], and [7] are *localized MAC/routing algorithms* (LRAs), where nodes only exchange information with their one-hop neighbors (i.e., the nodes within transmission range). This local information exchange is essential to achieve scalability, while avoiding the substantial communication costs incurred in propagating path discovery/update messages.

GeRaF [1] is an example of a geographical integrated MAC and routing scheme where the forwarding area (whose nodes offer geographical advancement toward the

sink) is subdivided into a number of priority regions. The next hop is elected by means of a channel contention mechanism, where the nodes with the highest priority (i.e., closest to the sink) contend first. This has the effect of reducing the number of nodes that simultaneously transmit within a single contention, while increasing the probability of electing a relay node with a good geographical advancement. The authors in [2] propose Contention-Based Forwarding (CBF). In their scheme, the next hop is elected by means of a distributed contention. CBF makes use of biased timers, i.e., nodes with higher advancements respond first to contention requests. The value of the timers is determined based on heuristics. A similar approach is exploited in [3], where the authors propose Implicit Geographic Forwarding (IGF). This technique accounts for biased timers as well. Response times are calculated by also considering the node's residual energy and a further random term. Advancements, energies, and random components are encoded into cost metrics. The random term improves the performance when multiple nodes have similar costs.

In [4], the authors improve the performance of LRAs by presenting the concept of *partial topology knowledge forwarding*. Sensors are assumed to know the state of the nodes within their communication range (called *knowledge range* in [4]) only. Their goal is to optimally tune, based on the local topology, the communication range (local view) at each sensor in order to approach globally optimal routing. Reference [5] proposes the MACRO integrated MAC/routing protocol. This is a localized approach relying on priority regions as [1] by, in addition, exploiting power control features for improved energy efficiency. A common denominator among these forwarding schemes is that they are all based on some sort of cost metrics, which are locally computed, and take into consideration the goodness of a node to be elected as the relay. Costs are often calculated by accounting for the progress toward the sink, but other

---

● *The authors are with the Consorzio Ferrara Ricerche (CFR), via Saragat 1, 44100 Ferrara, Italy, and also with the Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy. E-mail: {rossi, zorzi}@dei.unipd.it, buincl@unife.it.*

factors such as residual energy and transmission power are also considered [8]. We, however, note that the next-hop election is achieved by means of cost-aware heuristics that are not optimal. In addition to the above schemes, there is a whole line of research dealing with the design of good sleeping schedules [9], [10] to prolong the network lifetime. A recent paper [11] continues this work by proposing RMAC, which is an integrated MAC and routing technique where sleeping times are carefully set so as to minimize the latency, while retaining the energy savings of S-MAC [9].

The forwarding scheme we propose in this paper is designed to be reactive to the network dynamics and to elect the next hop with extremely low overhead through online optimal strategies. For these reasons, we integrate routing with a contention-based MAC not requiring time synchronization (unscheduled and stateless). Our algorithm can be seen as a generalization of [1], [5], and [6], as contentions are carried out by considering cost-dependent access probabilities instead of geographical [1] or transmission power-aware [5] priority regions. Moreover, the optimization performed in the present work is a nontrivial extension of the approaches in [6] and in [12]. In particular, the channel contention follows an optimization process over multiple access slots and, for each slot, over a two-dimensional cost-token space (justified and formally presented in Section 2.1); this considerably improves the performance of the forwarding scheme. In addition, the contention strategy we devise here is optimal rather than heuristic, and we add a new dimension to carry out the optimization (i.e., the node "cost," to be defined shortly). Also, as our solution provides a method to locally and optimally elect the next hop for a given knowledge range (transmission power), we note that it can be readily coupled with previous work [4]. Finally, our technique can be used in conjunction with advanced sleeping behavior algorithms [10], [11]. This is possible due to the stateless nature of our scheme, which makes it well adaptable to system dynamics.

More specifically, we present an original forwarding technique coupling the desirable features of previous work, such as the local nature of the next-hop election and the definition of suitable cost metrics, with *optimal access policies*. We consider a sensor network where traffic flows from the nodes to the sink, and we let the forwarding process be at a generic node, which has a number of awake neighbors. The objective of the algorithm we propose is to elect the next hop among them by maximizing the probability that MAC contention successfully chooses the best node in the set. The goodness of a potential relay node is represented through a generic cost metric, normalized in [0, 1], which is associated with each node in the network. Costs do not need to be exchanged among nodes, may be dynamically varying, and are calculated on demand by the candidate relay nodes when the contention starts. As a simple example, consider a number of candidate forwarders $1, 2, \ldots, n$, each with a geographical advancement toward the sink $a_i \geq 0$, where $i = 1, 2, \ldots, n$. For pure geographical routing, the cost of node i can be defined as $c_i = 1 - a_i/R$, where R is the transmission range of the nodes. In our scheme, we use these costs to drive the channel contention and elect a

relay. In doing so, we extend the approach in, e.g., [1] and [5], where the node selection is based on a priori assigned geographical regions, each with a given priority. In our case, in fact, we exploit cost regions that are dynamically (and optimally) adapted as the contention evolves, as will be clear from the analysis in Sections 2.3 and 3. In addition, costs can be related to further quantities, depending on the application/usage scenario. One might, for example, account for further metrics such as the ability of a node to aggregate the packet to be transmitted with those in its buffer, link qualities, residual energies, and so forth. Our analysis is general and can be adapted to any of these cases.

Based on these costs, we formulate the relay node selection as a dynamic programming optimization problem (Section 2). Subsequently, we find the optimal access policy, and we use it to derive a contention-based MAC and forwarding scheme, named Cost- and Collision-Minimizing Routing (CCMR) (Section 3).

Hence, we show the effectiveness of our solution by means of extensive and detailed simulation results, where we compare CCMR against state-of-the-art algorithms [2], [3] belonging to the same class of protocols, while considering realistic cost metrics. Finally, we complete our study by presenting experimental results and describing the software implementation of CCMR.

The main contributions of this paper are listed as follows:

- We analytically characterize the joint routing and relay election problem, finding optimal online policies.
- We use these results for the design of a practical solution for WSNs, which we call CCMR.
- CCMR is compared against state-of-the-art solutions belonging to the same class of protocols, showing its effectiveness.
- We describe the software implementation of our algorithm and present experimental results to demonstrate the feasibility of our approach.

The structure of the paper is given as follows: The analytical framework, including the cost model and the characterization of the optimal access policy, is presented in Section 2. In Section 3, we derive a cost-aware forwarding technique exploiting the optimal policy. Sections 4 and 5 present simulation and experimental results to validate and characterize the proposed solution, respectively. In Section 6, we discuss the complexity of our approach, showing that CCMR can be efficiently run on resource-constrained devices. Finally, Section 7 concludes the paper.

2 ANALYTICAL FRAMEWORK

2.1 Cost Model

In this section, we introduce a simple analytical cost model that we adopt to design our scheme. In doing so, we explicitly account for the correlation ρ among costs, as this parameter affects the optimal channel access behavior the most. In the next sections, such a cost model is used to derive the optimal access policy and to design an integrated channel access/routing protocol. In Section 4, simulation

results are given to demonstrate the validity of the approach in the presence of realistic costs, depending on geographical advancements and energy levels. Further, in Section 5, we show experimental results where the cost is associated with geographical advancements and is used to implement a greedy geographical routing scheme.

Let us consider a generic set $\mathcal{S}_N$ of N nodes, where we refer to c_k as the cost associated with node $k \in \mathcal{S}_N$. In order to model the cost correlation, we assume that c_k is given by $c_k = \bar{c} + \gamma_k$, where $\bar{c}$ is a cost component common to all nodes, whereas γ_k is an additive random displacement uniformly distributed in $[-\alpha\bar{c}, \alpha(1-\bar{c})]$, $\alpha \in [0,1]$, and independently picked for every node k. $\bar{c}$ is drawn from a random variable with domain in [0, 1] and whose exact statistics depends on the specific environment where the sensors operate. With the above model, $\alpha = 0$ corresponds to the fully correlated case as all node costs collapse to $\bar{c}$. Conversely, $\alpha = 1$ gives the i.i.d. case ($\rho = 0$) as the costs of every pair of nodes in $\mathcal{S}_N$ are independent. Intermediate values of α lead to a correlation $\rho \in (0,1)$. The (linear) correlation coefficient between the costs of any two nodes r, $s \in \mathcal{S}_N$ is defined as $\rho_{r,s} = (E[c_r c_s] - E[c_r]E[c_s])/(\sigma_r \sigma_s)$, where $\sigma_s^2 = E[(c_s - E[c_s])^2]$, $c_r = \bar{c} + \gamma_r$, and $c_s = \bar{c} + \gamma_s$. Hence, the correlation coefficient is given by

$$\rho_{r,s} = \frac{\left(E[\bar{c}^2] - E[\bar{c}]^2\right)(1-\alpha)^2}{\left(E[\bar{c}^2] - E[\bar{c}]^2\right)(1-\alpha)^2 + \alpha^2/12}. \quad (1)$$

As an example, if $\bar{c}$ is drawn from a uniform distribution in [0, 1] ($E[\bar{c}^2] = 1/3$ and $E[\bar{c}]^2 = 1/4$), (1) becomes

$$\rho_{r,s} = \frac{(1-\alpha)^2}{(1-\alpha)^2 + \alpha^2}. \quad (2)$$

Note that as long as $E[\bar{c}^2] > E[\bar{c}]^2$, the correlation can be tuned by varying the parameter α, and as anticipated above, $\alpha = 0$ and $\alpha = 1$ lead to the fully correlated and to the i.i.d. case, respectively. Also, there is a one-to-one mapping between α and ρ as (1) is invertible.

Besides this, with the above cost model, for a given $(\alpha, \bar{c})$ pair, all costs fall in the interval $[c_{min}, c_{max}]$, where $c_{min} = \bar{c} - \alpha\bar{c}$, and $c_{max} = \bar{c} + \alpha(1-\bar{c})$; thus, the cost set can be specified in terms of either $(\alpha, \bar{c})$ or (c_{min}, c_{max}). We shall see in Section 3 how these values can be optimally estimated as the contention evolves.

Note also that by specifying α and $\bar{c}$ or, equivalently, c_{min} and c_{max}, we only know that all costs are *uniformly* and independently distributed in the subset $[c_{min}, c_{max}] \subseteq [0,1]$, which means that our cost model has maximum entropy. In fact, for a given $(\alpha, \bar{c})$ pair, there is maximum uncertainty for the actual position of the costs in $[c_{min}, c_{max}]$. We observe that finding an optimal policy by considering this cost model makes sense from both a practical and a theoretical point of view. From a practical standpoint, this model requires only two parameters to statistically characterize the costs by accounting for their correlation. This is especially useful in sensor networks due to their inherently limited resources. From a theoretical point of view, we note that maximum entropy also corresponds to the worst case in terms of performance. In

fact, any other distribution able to track the cost correlation would lead to a more precise statistical description of the costs. Finally, the practical scheme we propose in Section 3, as the contention evolves, effectively searches the cost space using cost intervals of the type $[c_{min}, c_{max}]$. As the algorithm goes through successive phases, this interval becomes smaller and smaller until we locate the lowest cost node. Hence, although we approximate the actual cost distribution as uniform within each interval, such an approximation improves at every step, and this makes the approach general enough to deal with generic cost models. This is shown in Section 3.

2.2 State Space Representation and Problem Formulation

Let us consider the next-hop election problem for a given node in the network. Such an election is performed by means of MAC contentions, which usually consume resources in terms of both time and energy. Broadly speaking, our goal is to elect the relay node by maximizing the joint probability that a node wins the contention and it has the smallest cost (or a sufficiently small cost) among all active neighbors. The formal problem statement is given at the end of this section. Here, we refer to this election strategy as *optimal*. According to our scheme, the node sends a request (REQ) addressed to all nodes in its active (or forwarding) set, which is composed of all active neighbors providing a positive advancement toward the sink. Upon receiving the REQ, the active nodes in this set transmit a reply (REP) by considering a slotted time frame of W slots. Specifically, each node picks one of these slots according to its cost and uses it to transmit a REP to the inquiring node. The first node to send a REP captures the channel so that the nodes choosing a later slot refrain from transmitting their REPs.

To model the above scheme and find the *optimal* slot election strategy under any cost correlation value, we proceed as follows: For each node, we account for a cost and a token, the latter being a random number that is uniformly picked in [0, 1] at every contention round. Tokens are used to model cost-unaware access probabilities [13]. In more detail, when costs are fully correlated ($\rho = 1$), the nodes should pick the access slots by only considering their tokens, as their costs are all equivalent by definition. In this case, the aim of the algorithm is to select any node in the forwarding set by maximizing the probability of having a successful contention, and the solution reduces to the one in [12]. On the other hand, when costs are completely uncorrelated ($\rho = 0$), tokens must be disregarded, and the slot selection should be made on the basis of the node costs only. Finally, if the cost correlation is in (0, 1), both costs and tokens should be taken into account in the selection of the access slot. In addition, in order to simplify the problem formulation, access probabilities can be expressed in terms of access intervals as we explain next. For illustration, consider the case where $\rho = 1$, i.e., only tokens are accounted for in making access decisions. In this case, at any given node and for a given slot, accessing the channel with a given probability p is equivalent to accessing the channel if the token is within the interval $[0, p]$. When $\rho = 0$, the same rationale can be used for the costs, by defining
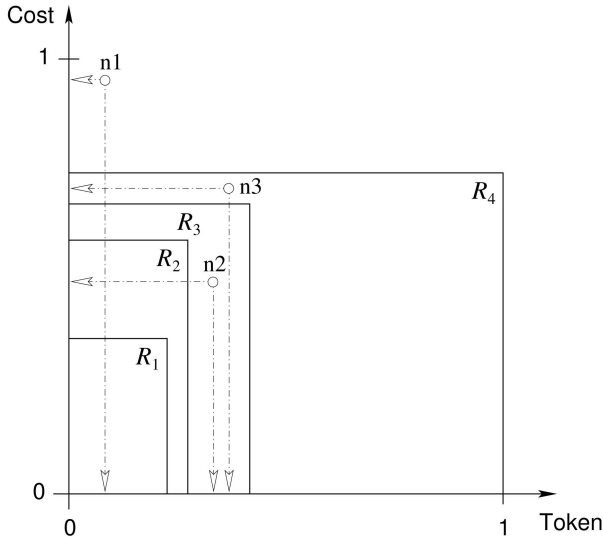
Fig. 1. Example of access region and node representation for $W = 4$.

intervals in the cost space. In the most general case ($\rho \in (0,1)$), we can define rectangular *access regions* spanning over both costs and tokens. For the sake of explanation, we illustrate the concept by means of Fig. 1, where we plot an access slot selection example for $W = 4$ slots. A formal treatment is given in Section 2.3. The active set is composed of the three nodes n1, n2, and n3, which are plotted in the token-cost space by means of white-filled circles. We associate the access regions R_1, R_2, R_3, and R_4 with the access slots 1, 2, 3, and 4, respectively. Note that $R_1 \subset R_2 \subset R_3 \subset R_4$; this property holds in the general case; see Section 2.3. For the slot selection, each node picks the access slot corresponding to the smallest region containing its (cost, token) pair. Specifically, node n1 cannot transmit in any of the slots as it is not within a valid access region. Moreover, in the first slot, none of the remaining nodes can access the channel. In fact, R_3 is the first region containing a node, n2 in our example, which therefore sends its REP in the third slot. Note that according to our slot selection strategy, n3 would be allowed to transmit its REP in slot 4. However, it refrains from transmitting the REP in this slot as it senses the ongoing communication of node n2. In this example, a single node (n2) accesses the channel, and this is the node with the minimum cost in the active set. We observe that collisions (multiple nodes select the same slot) are possible. Moreover, although it could also be possible that the winner of the contention is not the node with the minimum cost, our solution is aimed at minimizing the probability of occurrence of this event.

The problem to be solved can be formulated as follows: For a given active set $\mathcal{S}_N$, characterized by the number of nodes N therein and their cost correlation ρ, and for a given number of contention slots W, our objective is to find the sequence of access regions $R_1, R_2, \ldots, R_W$ maximizing the joint probability that a node wins the contention and has a sufficiently low cost. The term "sufficiently low" means that the absolute value of the difference between the cost of the winner and the minimum cost in the set is smaller than or equal to a certain parameter $\varepsilon \in [0,1]$ that we use to define the optimality criterion. In more detail, if

$\varepsilon = 0$, we aim at electing the node with the smallest cost in $\mathcal{S}_N$, whereas if $\varepsilon \in (0,1]$, we relax our optimality requirement. We mathematically formulate and solve this problem in Section 2.3.

2.3 Optimal Access Schedules: Analysis

We represent the generic access region R_i, $i \in \{1, 2, \ldots, W\}$, by means of a cost-token pair (c_i, t_i), where $c_i, t_i \in [0, 1]$. With (c_i, t_i), we understand that the region R_i is identified by the two intervals $[0, c_i]$ and $[0, t_i]$ for the cost and the token spaces, respectively. We observe that as tokens are uniformly and independently drawn in [0, 1], the only fact that counts for the optimization over the token space is the length of the token interval. For the costs, our assumption is also correct as we aim at electing a node with a small (possibly the smallest, see later) cost in the active set $\mathcal{S}_N$. Moreover, we observe that $R_1, R_2, \ldots, R_W$ is an *increasing sequence*. With this term, we mean that if $R_i = (c_i, t_i)$, $i \in \{1, 2, \ldots, W-1\}$, the region associated with slot $i + 1$ must be $R_{i+1} \supset R_i$, i.e., (c_{i+1}, t_{i+1}) should comply with one of the following three cases: 1) $c_{i+1} > c_i$ and $t_{i+1} = t_i$, 2) $c_{i+1} = c_i$ and $t_{i+1} > t_i$, and 3) $c_{i+1} > c_i$ and $t_{i+1} > t_i$. In fact, if $R_i = (c_i, t_i)$ and no node accesses the channel in slot i, it does not make sense to have $R_{i+1} = R_i$, as in this case, no node will access the channel in slot $i + 1$ as well, and this is trivially inefficient. To proceed with our analytical formulation, we introduce the following definitions:

Definition 1. *Consider an active set $\mathcal{S}_N$ of N nodes and a constant $\varepsilon \in [0, 1]$. We define node $k \in \mathcal{S}_N$ as ε-optimal if $c_j \geq \max(0, c_k - \varepsilon)$, $\forall j \in \mathcal{S}_N$, $j \neq k$, i.e., if none of the remaining nodes has a cost strictly smaller than $c_k - \varepsilon$.*

Definition 2. *We say that slot i is **silent** if no node chooses this slot and that there is a **collision** if two or more nodes pick the same slot i. In addition, we say that a node **wins** in slot i if it is the only sensor picking this slot and all previous slots $1, 2, \ldots, i - 1$ were silent. Finally, we say that a given contention round is **successful** according to a given $\varepsilon \in [0, 1]$ (optimality criterion) if there is a node that wins in any slot in $\{1, 2, \ldots, W\}$ and this node is ε-optimal.*

Note that the following analysis is conditioned on the values α (ρ), the number of nodes in the active set N, and the common cost component $\bar{c}$. For readability, we do not explicitly state these conditions. In order to get to the recursive expression of the probability of a successful event, we introduce the following quantities by considering the cost model in Section 2.1:

1. We introduce the probability $P_T(t_a, t_b)$ that the token of a given node falls within the generic interval $[t_a, t_b] \subseteq [0, 1]$. This probability is given by $P_T(t_a, t_b) = t_b - t_a$.

2. We introduce the probability $P_C(c_a, c_b)$ that the cost of a given node $k \in \mathcal{S}_N$ falls in the generic cost interval $[c_a, c_b] \subseteq [0, 1]$. We note that for node k, $P_C(c_a, c_b) = P\{\bar{c} + \gamma_k \in [c_a, c_b]\}$. Therefore, $P_C(c_a, c_b) = \int_{c_a - \bar{c}}^{c_b - \bar{c}} f(\gamma) \, d\gamma$, where $f(\cdot)$ is the probability density function (pdf) of the random cost displacement γ_k. $f(\gamma)$ equals $1/\alpha$ for $\gamma \in [-\alpha\bar{c}, \alpha(1 - \bar{c})]$ and zero

otherwise. By solving the above integral, we obtain the following closed-form expression:

$$P_C(c_a, c_b) = \begin{cases} 1, & \alpha = 0 \\ & \text{and } \bar{c} \in [c_a, c_b], \\ [\min(c_b, c_{max}) - \max(c_a, c_{min})]\alpha^{-1}, & \alpha \in (0, 1] \\ & \text{and } c_b \geq c_{min} \\ & \text{and } c_a \leq c_{max}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $c_{min} = \bar{c} - \alpha\bar{c}$, and $c_{max} = \bar{c} + \alpha(1 - \bar{c})$; see Section 2.1.

3. We now calculate the probability $P_{\text{noTX}}\{R_i\}$ that none of the nodes transmit in slots $1, 2, \ldots, i$ (i.e., region R_i is empty). The cost and token intervals associated with R_i are $[0, c_i]$ and $[0, t_i]$, respectively. $P_{\text{noTX}}\{R_i\}$ is found as follows:

$$P_{\text{noTX}}\{R_i\} = [1 - P_T(0, t_i)P_C(0, c_i)]^N, \quad (4)$$

where N is the number of nodes in $\mathcal{S}_N$, $P_T(\cdot)$ and $P_C(\cdot)$ are defined in points 1 and 2, and we used the fact that tokens and costs are independent by construction.

4. In the following, we focus on the probability $P_{\text{noTX}}\{R_i | R_{i-1}\}$ that none of the N nodes access the channel in a given slot i given that no node transmitted in slots $1, 2, \ldots, i - 1$. By using Bayes' formula, we can write

$$P_{\text{noTX}}\{R_i | R_{i-1}\} = \frac{P_{\text{noTX}}\{R_i\}}{P_{\text{noTX}}\{R_{i-1}\}}, \quad (5)$$

where we used the fact that region R_i contains region R_{i-1}, and therefore, if R_i is empty, R_{i-1} must also be empty.

5. We now consider the event that the generic slot i is successful. In particular, we refer to S_i as the event that a single node transmits in slot i, that its cost is ε-optimal, and that no nodes transmitted in slots $1, 2, \ldots, i - 1$. The probability of this event is referred to as $P\{S_i\}$ and is computed as follows:

$$P\{S_i\} = N \left[\int_0^{c_{i-1}} P_T(t_{i-1}, t_i)\xi(c, N - 1)f_{\text{cost}}(c) \, dc \right.$$
$$\left. + \int_{c_{i-1}}^{c_i} P_T(0, t_i)\xi(c, N - 1)f_{\text{cost}}(c) \, dc \right]. \quad (6)$$

In the above equation, we integrate over the cost region $[0, c_i]$ by splitting this integration interval into $[0, c_{i-1}]$ and $(c_{i-1}, c_i]$. In both terms, we use the cost pdf $f_{\text{cost}}(c)$ to account for the fact that the node transmitting in slot i (the winner of the contention) has cost equal to c. Considering the first integral, $P_T(t_{i-1}, t_i)$ gives the probability that the token of this node falls in $(t_{i-1}, t_i]$ so that the node is entitled to transmitting in slot i but not in slots $1, 2, \ldots, i - 1$. For the second integral, we instead consider the probability that the token of the winner is in $[0, t_i)$ (term $P_T(0, t_i)$). In fact, the integration interval $(c_{i-1}, c_i]$

already accounts for the fact that the winner cannot pick a slot in $1, 2, \ldots, i - 1$. The factor N is due to the N ways in which it is possible to elect a winner. In fact, in set $\mathcal{S}_N$, there are N nodes that, due to the symmetry of the cost statistics, all have the same probability of winning the contention (this probability is given by $P\{S_i\}/N$). Finally, the term $\xi(c, N - 1)$, which appears in both integrals, returns the joint probability that the remaining $N - 1$ users do not transmit in any of the slots up to and including slot i and their cost is larger than or equal to $c - \varepsilon$ (this ensures ε-optimality for the node transmitting in slot i). $\xi(c, N - 1)$ is obtained as follows:

$$[P_C(\max(0, c - \varepsilon), c_i)P_T(t_i, 1) + P_C(c_i, 1)]^{N-1}, \quad (7)$$

where for each of the $N - 1$ remaining users, we account for the mutually exclusive cases where the cost of the node is larger than c_i (term $P_C(c_i, 1)$) and the case where the node cost is within $[0, c_i]$ is larger than or equal to $c - \varepsilon$ (this accounts for ε-optimality; see term $P_C(\max(0, c - \varepsilon), c_i)$) and the token is outside the interval $[0, t_i]$ (term $P_T(t_i, 1)$). By factoring out the common terms and writing $P_T(\cdot)$ as in point 1, (6) can be simplified as follows:

$$P\{S_i\} = N \left[t_i \int_0^{c_i} \Psi(c) \, dc - t_{i-1} \int_0^{c_{i-1}} \Psi(c) \, dc \right], \quad (8)$$

where $\Psi(c) = \xi(c, N - 1)f_{\text{cost}}(c)$. To express the above equation in closed form, we need to solve the following integral: $I(x) = \int_0^x \Psi(c) \, dc$. By considering that $f_{\text{cost}}(c) = 1/\alpha$ if $c \in [c_{min}, c_{max}]$ and zero otherwise and that $c_{min} \geq 0$, $I(x)$ can be rewritten as follows:

$$I(x) = \int_{c_{min}}^{\min(x, c_{max})} \xi(c, N - 1) \, \alpha^{-1} \, dc \quad x > c_{min}, \quad (9)$$

and $I(x) = 0$ for $x \leq c_{min}$. The calculation of $I(x)$ involves the following two cases: 1) $c_{min} < x \leq c_{max}$ and 2) $x > c_{max}$. For readability, we skip the tedious calculations for these two cases, and we give the closed-form solution for $I(x)$ as follows:

$$I(x) = \left\{ [\min(x, c_{min} + \varepsilon, c_{max}) - c_{min}][k_1(k_2 - c_{min}) + P_C(c_i, 1)]^{N-1} \right.$$
$$+ \frac{[k_1(k_2 - \min(x, c_{min} + \varepsilon, c_{max}) + \varepsilon) + P_C(c_i, 1)]^N}{k_1 N}$$
$$\left. - \frac{[k_1(k_2 - \min(x, c_{max}) + \varepsilon) + P_C(c_i, 1)]^N}{k_1 N} \right\} \alpha^{-1},$$
$$k_1 = (1 - t_i)\alpha^{-1},$$
$$k_2 = \min(c_i, c_{max}), \quad (10)$$

$P\{S_i\}$ is obtained in closed-form as follows:

$$P\{S_i\} = \begin{cases} N(t_i - t_{i-1})(1 - t_i)^{N-1}, & \alpha = 0, \bar{c} \in [0, c_{i-1}], \\ Nt_i(1 - t_i)^{N-1}, & \alpha = 0, \bar{c} \in (c_{i-1}, c_i], \\ 0, & \alpha = 0, \bar{c} \notin [0, c_i], \\ N[t_i I(c_i) - t_{i-1} I(c_{i-1})], & \alpha \in (0, 1]. \end{cases} \quad (11)$$

Consider the first line of the previous equation. As $\alpha = 0$, all costs are equal, and the optimization is carried out on the token space. Thus, the probability of having a winner is equal to the probability that one of the nodes has the token in $(t_{i-1}, t_i]$, as if it had the token in $[0, t_{i-1}]$, it could have accessed the channel in a previous slot $(< i)$, and that the remaining $N-1$ users have their tokens in $(t_i, 1]$, i.e., they are entitled to sending their REPs in a later slot $(> i)$. As above, N accounts for the number of ways in which it is possible to elect a winner. Similar considerations apply for the second and third lines of (11). Also, note that for $\alpha = 0$, ε-optimality is always verified as all costs are equal.

6. We finally consider the last building block for our analysis, which consists of $P_{\text{succ}}\{R_i | R_{i-1}\}$, the probability of having a successful reply in the generic slot i (the region to be considered in this case is R_i), given that all previous slots (region R_{i-1}) are silent. This probability is readily found via Bayes' formula by considering the results in points 3 and 5:

$$P_{\text{succ}}\{R_i | R_{i-1}\} = \frac{P\{S_i\}}{P_{\text{noTX}}\{R_{i-1}\}} \ . \qquad (12)$$

Note that $P\{S_i\}$ is the joint probability that there is a success in slot i and no other nodes transmit in slots $1, 2, \ldots, i$, whereas $P_{\text{succ}}\{R_i | R_{i-1}\}$ is the probability of success in slot i conditioned on the event that no nodes transmitted in slots $1, 2, \ldots, i-1$. Next, we derive the maximum probability that a single contention round is **successful** for a given optimality criterion ε. This probability is written as a function of the access regions, the number of nodes N in $\mathcal{S}_N$, the common cost component $\bar{c}$, and the cost correlation ρ. For a generic slot i, we define $\varphi(i, R_{i-1})$ as the maximum probability to have a successful reply in some slot in $i, i+1, \ldots, W$ given that all previous slots $1, 2, \ldots, i-1$ were silent and that the region associated with the last slot $(i-1)$ is R_{i-1}. This probability is found according to a dynamic programming formulation [14], as detailed next. For the last slot W and for a given region R_{W-1}, $\varphi(W, R_{W-1})$ is found by maximizing over $R_W \supset R_{W-1}$ the probability of having a successful reply in the last slot given that all previous slots were silent $(P_{\text{succ}}\{R_W | R_{W-1}\})$. To find $\varphi(W-1, R_{W-2})$, we proceed by applying a backward recursion as follows: For a given (R_{W-2}, R_{W-1}) pair, the maximum probability of having a success in any of the last two slots $(W-1$ or $W)$ is given by the probability of having a success in slot $W-1$ given that all previous slots were silent $P_{\text{succ}}\{R_{W-1} | R_{W-2}\}$ summed to the probability that slot $W-1$ is also silent $P_{\text{noTX}}\{R_{W-1} | R_{W-2}\}$ multiplied by the maximum success probability in the last slot $\varphi(W, R_{W-1})$. $\varphi(W-1, R_{W-2})$ is found by maximizing the latter calculation over the feasible values of R_{W-1}, i.e., $R_{W-1} \supset R_{W-2}$. The same reasoning can be recursively written for each access slot by means of the optimality equation (13).
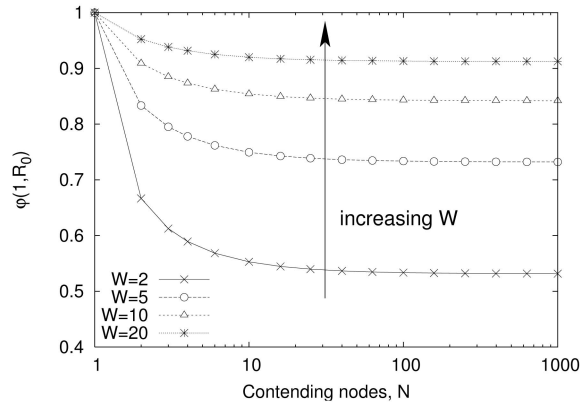

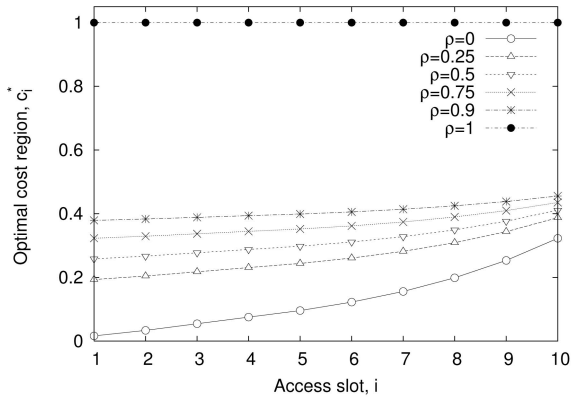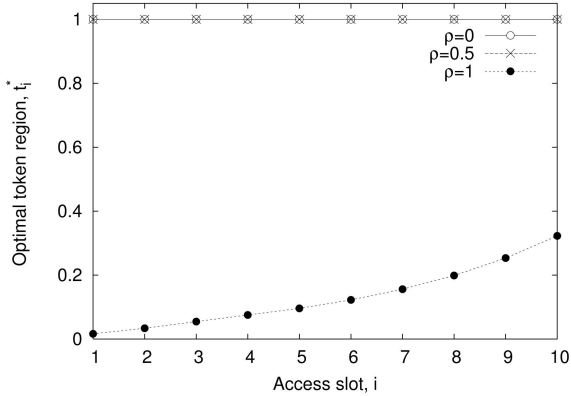Fig. 2. $\varphi(1, R_0)$ as a function of N.

The maximum probability of having a successful round is finally given by $\varphi(1, R_0)$, where $R_0 = (0, 0)$ by construction. The optimal access policy is given by the sequence of access regions $R_1^*, R_2^*, \ldots, R_W^*$ leading to $\varphi(1, R_0)$. In particular, the optimal policy specifies for each pair (i, R_{i-1}) an access region $R_i^* = (c_i^*, t_i^*)$ maximizing the right size of the optimality equation. Both the maximum probability that a single contention round is successful $\varphi(1, R_0)$ and the optimal access regions can be found by numerical approximation and recursive fixing techniques [14]. The results of these computations are discussed in Section 2.4.

$$\varphi(i, R_{i-1}) = \begin{cases} \max_{R_i \supset R_{i-1}} P_{\text{succ}}\{R_i | R_{i-1}\}, & i = W, \\ \max_{R_i \supset R_{i-1}} \big[P_{\text{succ}}\{R_i | R_{i-1}\} \\ \quad + P_{\text{noTX}}\{R_i | R_{i-1}\} \, \varphi(i+1, R_i) \big], & i < W. \end{cases} \qquad (13)$$

2.4 Optimal Access Schedules: Discussion of Results

As a first result, in Fig. 2, we show the probability $\varphi(1, R_0)$ of having a **successful** contention round using the optimal policy, by averaging over $\bar{c}$ (uniformly distributed in [0, 1]). The parameters for this figure are $\varepsilon = 0$ and $\rho = 0.5$. Perfect knowledge is assumed at the transmitter for the number of contenders N, the cost correlation ρ, and $\bar{c}$. As expected, $\varphi(1, R_0)$ increases with an increasing number of access slots W: increasing W from 2 to 10 almost doubles the performance, whereas further increasing it $(10 \rightarrow 20)$ only provides marginal improvements. Also, for a given W, the success probability quickly stabilizes $(N \geq 10)$ to its asymptotic value. In our implementation of the scheme, we chose $W = 10$ to obtain a reasonable trade-off between complexity and effectiveness.

In Figs. 3 and 4, we plot the optimal access regions for costs and tokens, respectively. Notably, the value of ρ does have an impact on the shape of the regions. In practice, the case $\rho = 0$ is the most selective in the sense that high costs, for any given slot, are penalized the most. Also, we observe that for $\rho = 1$, all costs are equal by construction, and hence, they should not affect the slot selection process. This is in fact verified in Fig. 3, where cost regions are all equal to one

Fig. 3. c_i^* for $N = 10$, $W = 10$, and $\overline{c} = 0.5$.

Fig. 5. c_i^* for $N = W = 10$, $\overline{c} = 0.5$, and $\varepsilon = 0.1$.

Fig. 4. t_i^* for $N = 10$, $W = 10$, and $\overline{c} = 0.5$.

Fig. 6. t_i^* for $N = W = 10$, $\overline{c} = 0.5$, and $\varepsilon = 0.1$.

for $\rho = 1$. This concept can be remarked by looking at Fig. 4, where we plot the token regions t_i^* for the same system parameters. In this case, t_i^* are equal to one for $\rho \in [0, 1)$. This means that for these values of ρ, the tokens do not influence the slot selection, which is only driven by the costs. On the other hand, for $\rho = 1$, costs are no longer relevant to the access policy. We finally observe that *token* regions t_i^* when $\rho = 0$ are equal to the *cost* regions c_i^* for $\rho = 1$. This suggests a sort of *duality* between costs and tokens in these two extreme cases.

By analyzing the obtained results for $\varepsilon = 0$, we found an interesting connection with the findings in [12], whose theory allows finding the optimal access probability for every slot when all costs are equal ($\rho = 1$). If N is the number of contenders, i is the generic access slot, and $N \geq 2$, we define $f_s(N)$ as [12]

$$f_s(N) = \begin{cases} 0, & s = 1, \\ \frac{N-1}{N - f_{s-1}(N)}, & s \geq 2. \end{cases}$$

According to [12], the optimal value of the probability that a user selects i as its access slot is

$$p_i^* = \frac{1 - f_{W-i}(N)}{N - f_{W-i}(N)} \left(1 - p_1^* - p_2^* - \cdots - p_{i-1}^*\right). \quad (14)$$

Choosing the access slots according to the above p_i^*s leads to the maximization of the probability that a single node accesses the channel during a given contention round. Note that this distribution does not depend on the node costs ($\rho = 1$ in our framework). In fact, for $\rho = 1$, our problem
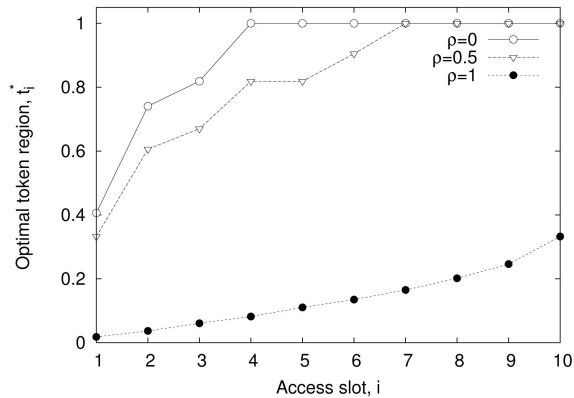
reduces to the one considered in [12]. Hence, in this case, our token regions t_i^* must lead to the same slot selection policy dictated by the p_i^*s in (14). According to our scheme, a node picks the generic slot i if and only if its token is in $(t_{i-1}^*, t_i^*]$. The probability of such an event is $\pi_i^* = t_i^* - t_{i-1}^*$, and since $p_i^* = \pi_i^*$ for $\rho = 1$, we have $t_i^* = t_{i-1}^* + p_i^*$. This was validated in all our results. In addition, by recalling that t_i^* for $\rho = 1$ equals c_i^* for $\rho = 0$, we can claim that the p_i^*s also give the optimal cost regions for $\rho = 0$, i.e., $c_i^* = c_{i-1}^* + p_i^*$. Further, we note that for $\rho \in (0, 1)$ costs are simply rescaled in $[c_{min}, c_{max}]$ (see cost model in Section 2.1) and are distributed uniformly in this interval. As the difference between the two cases $\rho = 0$ and $\rho \in (0, 1)$ is only given by the size of the cost interval and not by the cost distribution, which is still uniform, we infer that the optimal cost regions c_i^* for $\rho \in (0, 1)$ must be a rescaled version of those for $\rho = 0$. Accordingly, for $\rho \in (0, 1)$, we must have that

$$c_i^* = c_{i-1}^* + p_i^*(c_{max} - c_{min}), \quad (15)$$

where $c_0^* = c_{min}$. Hence, for $\rho \in [0, 1)$, the c_i^*s can be calculated by means of (15), whereas $t_i^* = 1 \; \forall \; i$ (note that for $\rho = 0$, we have $c_{min} = 0$ and $c_{max} = 1$; thus, (15) still holds). Finally, for $\rho = 1$, we have that $c_i^* = 1 \; \forall \; i$, and the t_i^*s are obtained as $t_i^* = t_{i-1}^* + p_i^*$. p_i^* can be tabulated for a given W and for several values of N according to (14). This is a practical and exact method to derive R_i^* when $\varepsilon = 0$.

Now, we discuss the case where $\varepsilon > 0$ whose example results in terms of cost and token regions are reported in Figs. 5 and 6, respectively. As can be observed from these

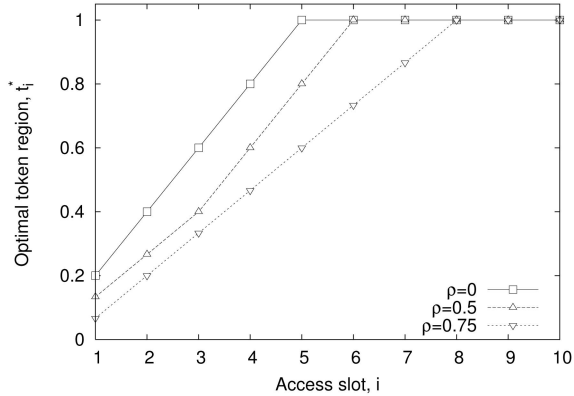Fig. 7. c_i^*: quantized state space.
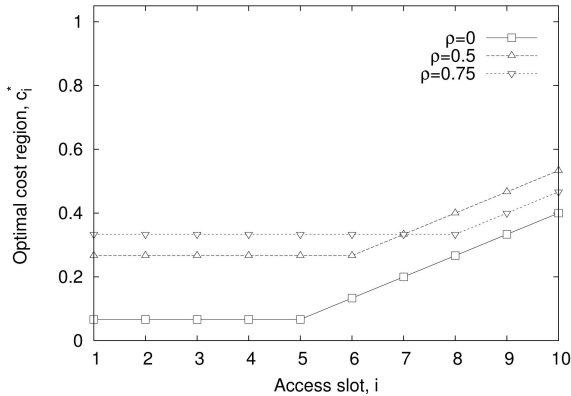
Fig. 8. t_i^*: quantized state space.

plots, for $\rho \in [0, 1)$, the optimal policy varies both c_i^* and t_i^* concurrently, i.e., the duality between costs and tokens and the result in (15) do not hold in this case. Finally, in Figs. 7 and 8, we report c_i^* and t_i^* when the cost/token space $[0, 1]$ is quantized in 20 equally spaced levels, by considering $\varepsilon = 0$, $N = 10$, $W = 10$, and $\bar{c} = 0.5$. This may be the case for resource-constrained devices. In these settings, our analysis still holds by just redefining the access regions R_i as a cost-token pair (c_i, t_i), where costs and tokens take values in discrete and finite sets composed of 20 points. As reported in the figures, now, token regions are varied first, while keeping the cost region fixed. Subsequently, as the token regions t_i^* saturate to the maximum value (one), the optimal policy starts varying c_i^*. This behavior compensates for the lack of precision due to the state space quantization. We finally observe that $\varphi(1, R_0)$ decreases when regions are quantized. As an example, $\varphi(1, R_0)$ for $N = 10$ and $W = 10$ is about 0.85 for a continuous state space, whereas it drops to 0.748 for $\rho = 0$ (best case) and about 0.592 on the average.

The results that we discussed above highlight some interesting characteristics of the optimal policy and demonstrate the validity and the flexibility of our analytical formulation. In Section 3, we consider the case $\varepsilon = 0$ by exploiting the result in (15) to devise an efficient cost- and collision-minimizing forwarding technique.

3 COST- AND COLLISION-MINIMIZING ROUTING

In this section, based on the previously discussed results, we present an integrated channel access and routing scheme that we name as CCMR. Our cross-layer design relies on the definition of the costs, which are used in the channel access to discriminate among nodes. This is achieved by accounting for routing metrics such as the geographical advancement, right in the cost calculation. Realistic cost models are presented in Section 4, where we report extensive simulation results to validate our approach. Next, we outline our integrated scheme by considering the costs as given:

1. Consider a generic node n. When the node has a packet to send, it first senses the channel according to a CSMA policy, as done in, e.g., IEEE 802.15.4. If the channel is sensed idle for a predetermined

interval, the contention starts. The contention to elect the next hop works in rounds and ends as soon as a round is **successful**. At the generic round $r \geq 1$, node n sends a request (REQ) including its identifier and an estimate for the number of contenders N and specifies a cost interval $[c_{min,r}, c_{max,r}]$, where $c_{min,1} = 0$, and $c_{max,1} = 1$. We detail how this interval is modified for $r > 1$ in point 3 below.

2. All active devices providing a positive advancement toward the sink contend for the channel. Upon receiving the REQ, at round $r \geq 1$, every node considers W access slots and calculates cost and token regions as follows: The node first computes a decay function $d(r) = r\beta/(r\beta + 1)$ depending on the round number r and on a constant $\beta > 0$. If $(c_{max,r} - c_{min,r}) > d(r)$, the c_i^*s are calculated by means of (15) and $t_i^* = 1 \,\forall\, i$; otherwise, $c_i^* = 1 \,\forall\, i$ and $t_i^* = t_{i-1}^* + p_i^* \,(t_0^* = 0)$, where p_i^* are as in (14) and $i \in \{1, 2, \ldots, W\}$. In other words, $d(r)$ is used to estimate when costs can be assumed to be equal, and therefore, the corresponding theory for $\rho = 1$ should be used. We refer to the cost region associated with the last slot W as $c_{W,r}^*$. Subsequently, using these access regions and its own cost, the node picks a slot in $\{1, 2, \ldots, W\}$ according to the scheme in Section 2 and schedules a reply (REP) in this slot.

3. Three cases can occur: a) All slots are **silent**. In this case, node n infers that none of the active nodes has cost in $[c_{min,r}, c_{W,r}^*]$ and starts round $r + 1$ by sending a new REQ, including the interval $[c_{min,r+1}, c_{max,r+1}]$, where $c_{min,r+1} = c_{W,r}^*$, and $c_{max,r+1} = c_{max,r}$. b) Multiple nodes send their REP in the same slot so that a **collision** occurs. Node n assumes that at least two nodes have cost in $[c_{min,r}, c_{W,r}^*]$. Hence, the node sends a REQ including the new interval $[c_{min,r+1}, c_{max,r+1}]$, where $c_{min,r+1} = c_{min,r}$, and $c_{max,r+1} = c_{W,r}^*$. c) A single node responds to the REQ (**success**): node n sends the packet to this node, which subsequently replies with an acknowledgment, and the contention is concluded.

The decay function $d(r)$ is used to tune the maximum number of contentions before considering the costs as fully correlated. That is, as the interval $[c_{min,r}, c_{max,r}]$ becomes sufficiently tight, access regions are calculated as in [12].
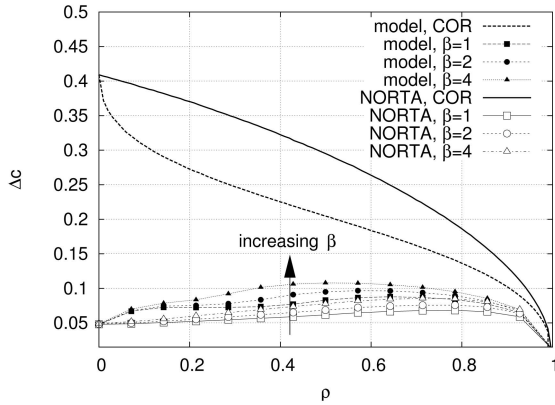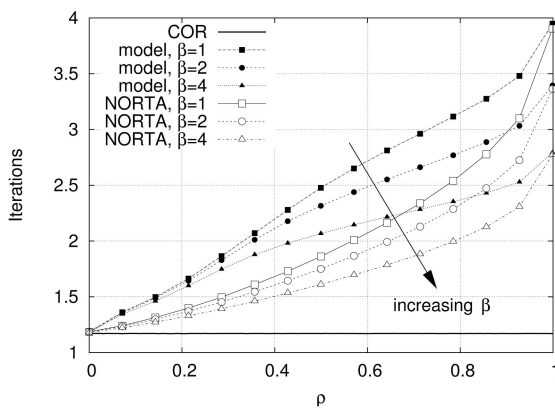
Fig. 9. Δc as a function of ρ.

Fig. 10. Rounds needed to complete a contention.

The parameter β acts as a knob to determine the decay threshold as a function of r.

In Figs. 9 and 10, we plot the average distance between the minimum cost in the active set and that of the winner of the contention (Δc) and the average number of contentions to elect the next hop, respectively. The considered system parameters are $N = 10$, $W = 10$, $\varepsilon = 0$, and $\bar{c}$ is uniformly distributed in $[0, 1]$. The performance of the scheme of Tay et al. [12] (COR) is reported for comparison. All results are obtained with the cost model in Section 2.1 (referred to as "model" in the graphs) and with the NORTA method [15] ("NORTA"), which we selected to test the effectiveness of our scheme in the presence of uniform costs with general correlation structure. NORTA is a known framework for representing random vectors whose elements have arbitrary marginal distributions and correlation matrix. First, we note that the performance of CCMR improves in terms of both Δc and the number of contention rounds when nodes' costs are derived using NORTA. In fact, we designed CCMR for the case where there is maximum uncertainty (maximum entropy) about the actual values of the nodes' costs. For any other distribution, having a higher degree of structural relationship among costs, our scheme is thus expected to have better performance. In addition, these results demonstrate that the algorithm is robust against ρ, is very effective in promoting low-cost nodes, and has delay very close to the optimum (which is given by COR). Also, suitable trade-offs can be achieved by

TABLE 1
Impact of Erroneous Estimates of N on Performance for $\beta = 2$

	No error		Error 25%		Error 50%	
N	Δc	Iterations	Δc	Iterations	Δc	Iterations
7	0.057	2.57	0.059	2.7	0.062	2.93
14	0.086	3.27	0.088	3.4	0.088	3.52
21	0.096	4.24	0.097	4.37	0.097	4.47

varying β. In the remainder, we use $\beta = 2$, which gives a good trade-off between cost and delay.

In the results above, we considered a perfect estimate of N. In Table 1, we report the worst case performance in terms of the cost error Δc for three selected values of $N \in \{7, 14, 21\}$, where we introduce some uncertainty on our knowledge of the number of contenders. As indicated by the results, the proposed scheme is robust against estimation errors. In particular, for an increasing error, the delay (the number of iterations to complete the contention) is only slightly increased, while the cost performance is almost unaffected. A further verification of this fact is given in the next section, where we report the multihop performance of the algorithm.

We finally observe that we assumed that collisions are detected with probability one and are always due to the transmission of REQ messages and that simultaneous transmissions always collide. We note, however, that in practice these assumptions may not be entirely accurate, due to, e.g., capture effect, parallel transmissions, channel errors, etc. All these assumptions are removed in the results shown in Sections 4 and 5. In addition, the presence of hidden terminals can lead to either colliding REPs or to the reception of multiple (disjoint) REPs for the same REQ. In the former case, CCMR reacts by reshaping the channel access regions so that in the following contention round, the colliding nodes are likely to respond in different slots. In the very worst case, where nodes' costs are equal, CCMR still resolves the contention as its access regions, after a number of rounds (see parameter β), become independent of the cost metric. In case multiple REPs are received, CCMR is again not affected as data packets are unicast.

4 SIMULATION RESULTS

The following results are obtained by means of the event-driven simulation tool presented in [16], which we complemented with PHY and MAC modules for sensor networking. Interuser interference is accurately modeled through the calculation of the received Signal-to-Interference-plus-Noise Ratio (SINR) for each pair of nodes. Bit errors at the PHY layer are derived from SINR measurements, according to [17] (see chapter 5 and references therein). For the energy consumption, we adopt the model in [18], i.e., idle, reception, and transmission modes consume 26.1, 47.1, and 90.6 mW, respectively. For the simulation results, we assume a maximum transmission range and a bit rate of $R = 30\,\mathrm{m}$ and $B = 38{,}400$ bps, respectively. Both sensors and sink are uniformly placed within a square-shaped simulation area of side 100 m. In the results that we show next, we use our solution to
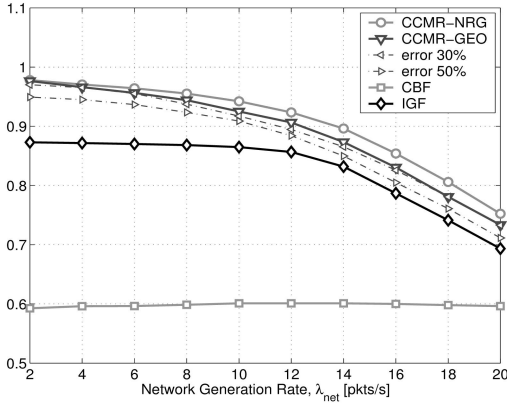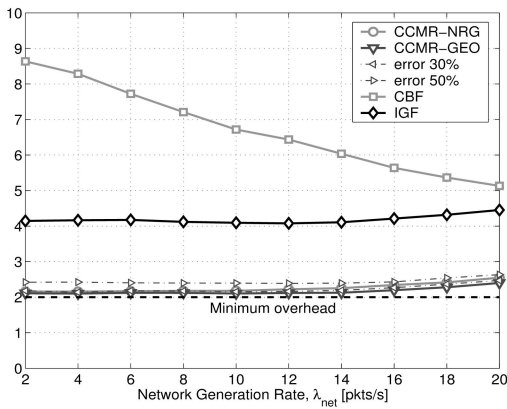
Fig. 11. Delivery rate versus λ_{net}.

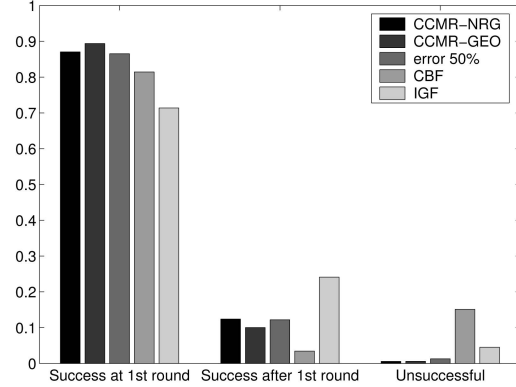Fig. 12. Average protocol overhead.

Fig. 13. Distribution of the contention outcome.

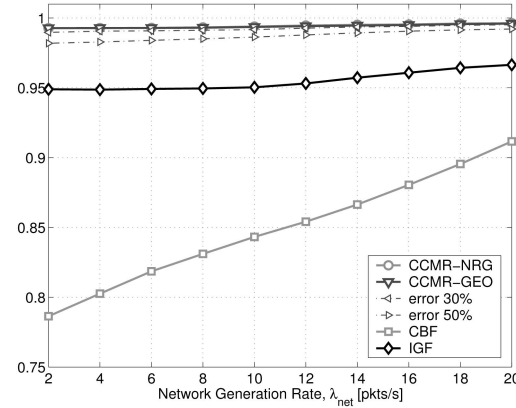Fig. 14. Probability of a successful contention.
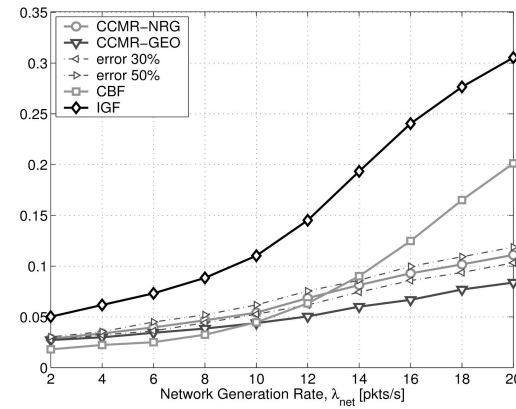
Fig. 15. Duration of the channel contention [in seconds].

deliver the data to the sink by exploiting geographical coordinates. Geographical forwarding is considered here to show the validity of the approach. However, we stress that our scheme, through minor modifications, works over virtual coordinates as well, e.g., hop counts [6]. In fact, different topologies just translate into a different definition of the costs. We run extensive tests varying the number of nodes (excluding the sink) N_u from 25 up to 150, all of which generate traffic according to a Poisson process with an intensity of λ packets per second per node. We plot the performance as a function of the total packet generation rate $\lambda_{net} = \lambda N_u$.

In the following, we compare our CCMR scheme against IGF [3], [19] and CBF [2]. In both schemes, the nodes in the active set respond to the inquiring node by exploiting a timer-based approach. In particular, upon receiving a REQ, each node replies after a time period, which is calculated as a function of its cost. Costs are derived from the geographical advancement only [2] or by also considering the node residual energy [3]. The main difference between [2], [3], and our scheme is that in our approach, nodes contend by jointly optimizing over a multislot frame, whereas in [2] and [3], the response time is a continuous quantity calculated by means of heuristics. We consider here two versions of CCMR. The first, called CCMR-GEO, assumes a cost model as in [2], i.e., the cost associated with the generic node n is calculated as $c_n = 1 - a_n/R$, where a_n is the geographical advancement provided by the node. In

the second scheme, referred to as CCMR-NRG, c_n is calculated as $c_n = 1 - (a_n/R)(e_r/E)$, where e_r is the node's residual energy, and E is the initial energy reserve. This is in line with the cost model in [3]. For both CCMR-GEO and CCMR-NRG, we assume a decay factor $\beta = 2$ (see Section 3). All forwarding techniques are implemented on top of a standard Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC, where the channel is sensed before starting new contentions, and nodes back off in case of colliding REQs.

Figs. 11, 12, 13, 14, 15, 16, and 17 consider a network scenario with $N_u = 50$ randomly distributed nodes, all of which generate data traffic. This translates to an average of
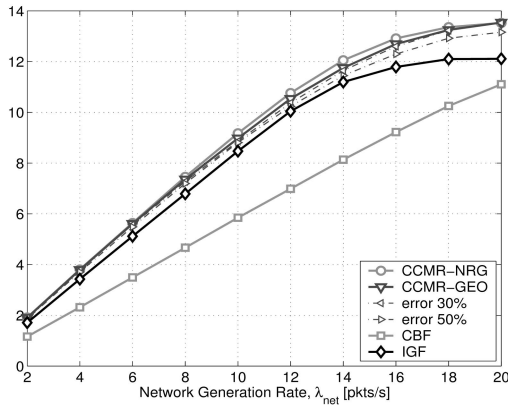
Fig. 16. Packets delivered to the sink per second.

Fig. 17. Delivery rate versus hop distance from the sink.

seven active nodes in the forwarding area. The performance achieved for different values of N_u is discussed at the end of this section. All plots, in order to show the robustness of our approach, also include CCMR-GEO by considering erroneous estimates of the number of active nodes N. We do not report such curves for CCMR-NRGs as they revealed a nearly identical behavior. These estimation errors are accounted for by randomly drawing N from a uniform distribution defined in the interval $[N - \Delta N, N + \Delta N]$, where ΔN is computed as a percentage of the actual value of N (ΔN is 30 percent and 50 percent in our plots).

As a first result, in Fig. 11, we report the packet delivery rate, calculated as the total number of received packets divided by the total number of packets generated. In all cases, CCMR obtains better performance than the other schemes, and at low λ_{net}, it delivers almost all packets. CCMR-NRG performs slightly better than CCMR-GEO as energy-aware costs allow the redistribution of the data flows, avoiding excessive congestion at specific nodes. Note that CBF suffers from a low delivery rate, even at low λ_{net}. This is due to the timer-based mechanisms adopted to respond to the REQs. In fact, in case the difference among the response times picked by multiple nodes is shorter than the time needed to complete the carrier sense operation, the REPs collide with high probability. In IGF, this problem is reduced thanks to the addition of a random quantity in the response times. In our solution, instead, slots are designed so as to allow for the completion of a full carrier sense operation. Note that this does not completely prevent collisions in CCMR, as sensors can still select the same access slot. However, after a collision event, the scheme adapts its policy to avoid such an event in the subsequent round with high probability.

Fig. 12 reports the overhead, defined as the average number of REQs and REPs required to transmit a data packet to the relay. Note that CCMR is very close to the optimal performance (1 REQ and 1 REP): keeping the control traffic low is beneficial as this means improvements in terms of channel capacity (interference) and energy consumption. A further reduction of the overhead, not treated in detail here, would be obtained by transmitting back-to-back multiple packets to the winner of the contention. In Fig. 13, we show the percentage of
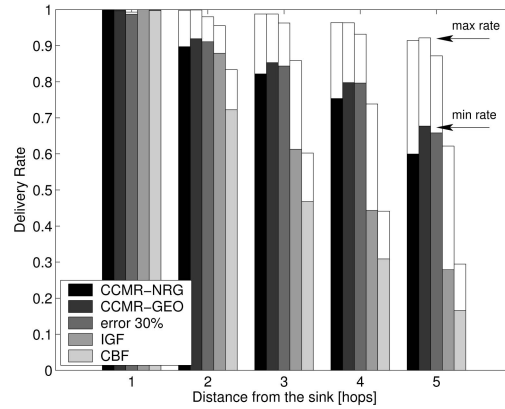
packets successfully sent in one and multiple contention rounds and the percentage of unsuccessful contentions (more than seven failed attempts in a row). Further results on the contention are given in Figs. 14 and 15, plotting the probability of successfully electing a next hop and the associated average delay, respectively. Note that both IGF and CBF improve as λ_{net} increases as a higher traffic means a lower number of active nodes taking part in the contentions and, in turn, a lower number of collisions. The results in Fig. 16 show the number of packets delivered to the sink per second in steady state. This metric saturates for increasing λ_{net}. Although from this figure, one might conclude that IGF and CCMR roughly lead to the same performance, by looking at the microscopic behavior of the schemes, it can be shown that this is not the case. To this end, in Fig. 17, we focus on the delivery rate as a function of the minimum number of hops separating the traffic sources from the sink. For each scheme, colored histograms are used to report the worst case performance, whereas white-filled histograms indicate the best achievable delivery rate. Best and worst cases are found by varying λ_{net} from 1 to 20 pkts/s. Both versions of CCMR outperform the remaining schemes by leading to a weaker dependence of the delivery rate on the distance from the sink. Finally, we obtained the same plots for different values of the density (N_u), which show better performance for higher values of N_u and a roughly similar behavior for lower densities: note that CCMR properly adapts its access policy according to the size of the active set, whereas IGF and CBF do not.

These results clearly show that CCMR is able to provide substantial performance enhancements compared to other schemes. As an example, in Fig. 12, we see that the signaling overhead of our scheme is close to optimal, which is half that of IGF and substantially smaller than that of CBF. Further advantages are shown by all other plots. Perhaps the most remarkable result is that CCMR provides even more substantial improvements in multihop networks (Fig. 17). In this case, all schemes tend to perform well in proximity of the sink, while the performance is considerably degraded for the nodes placed farther away (two or more hops from the sink). Our scheme can be seen to scale considerably better in these situations.
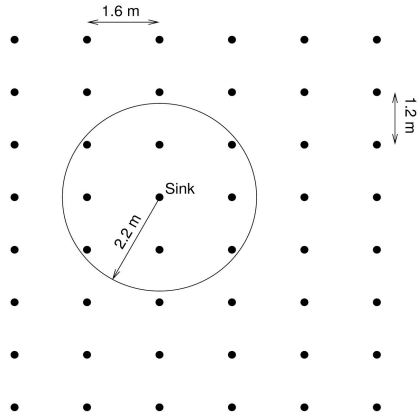
Fig. 18. Testbed layout.

5 EXPERIMENTAL RESULTS

In this section, we present our software implementation of CCMR, and we validate our scheme through experimental results. Section 5.1 presents the testbed we used for our experiments in terms of node placement, hardware platform, and achievable rates. Section 5.2 describes the obtained results.

5.1 Testbed

Our experiments were run on the SignetLab testbed, deployed in the Department of Information Engineering, University of Padova [20]. The hardware platform consists of EyesIFXv2 [21] sensor nodes, developed by Infineon Technologies. These sensors are equipped with a TDA5250 radio transceiver (working in the 868-MHz frequency band), an MSP430 16-bit microcontroller with 10-Kbyte on-chip RAM, 48-Kbyte flash/ROM, and 512 Kbytes of serial EPROM. The radio chip is a low-power FSK/ASK half-duplex transceiver with typical data rates of 19.2 Kbps. The testbed consists of 48 of these nodes placed on an 8×6 grid suspended 2.4 m above the floor, where the maximum distance between any pair of nodes in the network is 15 m; see Fig. 18. We define $N_u = 47$ as the sink does not generate traffic. For further information about the testbed, sensor nodes, and software tools we used to carry out the experiments, the reader is referred to [20].

We wrote a software module for TinyOS [22] implementing CCMR. The source code of this software can be downloaded from [23]. The CCMR library takes 6,086 bytes of ROM and 2,696 bytes of RAM. The RAM is used as follows: In the default configuration of the algorithm, 2,304 bytes are allocated for two queues, which are used by CCMR to store generated and received packets, respectively. However, the size of these queues can be tuned according to design constraints. In fact, an alternative solution consists of implementing the queues at the application layer, which leads to a RAM occupancy of 392 bytes. As a final remark, the optimal regions in Section 3 (see also (14) and (15)), which are used to drive channel contention, were tabulated using only 400 bytes of ROM. Finally, the CCMR code together with all the TinyOS modules that need to be linked for a working application (radio, timers, etc.) take about 24 Kbytes of ROM, which is just half the memory available on our platform. This leaves room for developing more complex applications on top of CCMR.

In our testbed, sensors have a radio transmission range of several tens of meters. This makes the network a single-hop environment. Hence, in order to test our protocol in a multihop scenario, we forced the transmission range to 2.2 m by discarding all the received packets whose transmitter was farther from the receiver. Next, we refer to this scenario as pseudo multihop (PMH). The impact of this somewhat artificial setting is discussed in the following section (see comment to Fig. 20).

Next, we characterize the implication of our reduction of the transmission range by calculating the modified theoretical capacity of our network. The following analysis has been used to tune the generation rate in the experiments so as to test CCMR under a noncongested regime. We tested our protocol for a data gathering application where all nodes had to send their packets to a sink, placed in the center of the testbed area. The network capacity can be upper bounded in the following way [24]. We assume a greedy routing scheme so that each sensor sends its data to the neighbor maximizing the geographical advancement toward the sink. We assume a perfect TDMA for the transmissions, where nodes take turns in transmitting according to an ideal and optimal schedule. In addition, we assume that all nodes are backlogged. Next, we compute the number of transmissions (or TDMA slots) so that all nodes can transmit exactly one of their own generated packets to the sink. The generic sensor i has to forward its own data, as well as the data generated by sensors placed farther away. Referring to p_i^g, p_i^r, and p_i^t as the number of packets generated, received, and transmitted by node i, respectively, we have that $p_i^t = p_i^g + p_i^r$. The number of packets received is calculated as $p_i^r = \sum_n \zeta_{ni} p_n^t$, where n spans over all nodes in the network, whereas ζ_{ni} accounts for the routing policy. In particular, for our greedy routing, ζ_{ni} equals zero for all nodes n that are not within the transmission range of node i, as well as for all sensors that do not choose i as their relay. ζ_{ni} is instead equal to one if and only if i is the best relay for node n (i.e., i provides maximum advancement). Note that the ζs are determined once network layout and routing rules are specified. Thus, the total number of packets received at node i can be computed as

$$p_i^r = \sum_n \zeta_{ni} \left(p_n^g + p_n^r \right), \qquad (16)$$

where ζ_{ni} is equal to one for all nodes n picking node i as their (unique) relay node and is zero otherwise. This formula can be calculated iteratively by first obtaining $p_n^t = p_n^g$ for the nodes having the maximum hop count h_{max}; these nodes do not relay any traffic; thus, $p_n^r = 0$. We subsequently compute p_i^r for the nodes with hop count $h_{max} - 1$ through (16). We continue in this way processing one hop count at a time. The total number of transmissions in the network is thus calculated as $p_{tot}^t = \sum_n p_n^t = \sum_n (p_n^g + p_n^r)$, where n spans over all nodes. We solved these equations numerically and found that 47 nodes, in our grid scenario, need at least $p_{tot}^t = 168$ transmissions to deliver all the
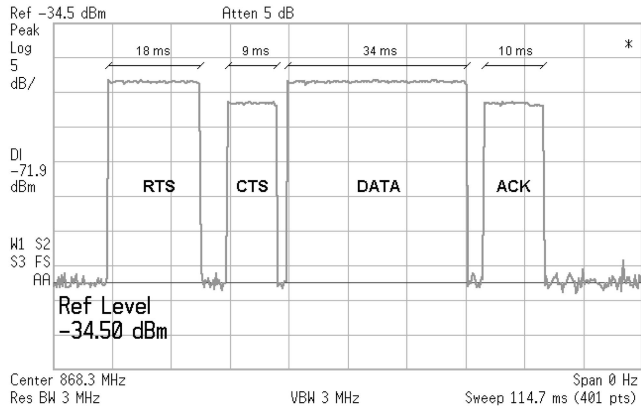
Fig. 19. Snapshot of a single channel contention. Here, the receiver responds in the first slot. Hence, this is the shortest possible time to complete a contention, which is about 85 ms. The longest time is 265 ms, which happens when a node replies in the last slot ($W = 10$ and the slot duration is 20 ms).

generated data to the sink. The upper bound for the capacity can be derived as

$$C_{net} = \frac{R_{max} N_u}{p_{tot}^t}, \qquad (17)$$

where N_u is the number of original packets to send (one per node but the sink), and p_{tot}^t is the total number of transmissions for their delivery to the sink through a greedy routing over multiple hops (without aggregating data). R_{max} is the maximum sustainable packet rate for a single link, i.e., $R_{max} = 1/T_{min}$, where T_{min} is the minimum amount of time for the transmission of a single packet. In our settings, $T_{min} \approx 85$ ms, as shown by the screenshot in Fig. 19, which is obtained using a spectrum analyzer, a log-periodic antenna, and a pair of nodes (transmitter and receiver) running our algorithm (channel assessment, REQ/REP/DATA/ACK). Hence, $T_{min} p_{tot}^t$ is the minimum time needed to deliver the N_u original packets through a perfect TDMA. Equation (17) returns $N_u/(T_{min} p_{tot}^t)$, which is the packet rate under perfect TDMA and no collision. Hence, $C_{net} \approx 3.3$ packets second.

Practically, we also have to consider the effect of the CSMA mechanism, the impact of packet collisions (we do not have a perfect schedule in our experiments), and the nonideality of the radio equipment. These factors will all reduce the achievable capacity. Further considerations on this will be given in Section 5.2.

5.2   Results

All sensors generate traffic according to a Poisson arrival process. The network generation rate λ_{net} is varied from 0.2 to 1.5 packets per second to respect the capacity limits we found in the previous section. The generation rate for a single node is $\lambda = \lambda_{net}/N_u$. We repeated at least 10 experiments for each value of λ_{net}, where each node generates 100 packets for each experiment. For the transmission of RTS messages, we adopted an IEEE 802.11-like CSMA technique. In particular, in case of a busy medium, we set the backoff timer so that new RTSs cannot interfere with ongoing contentions. Costs are
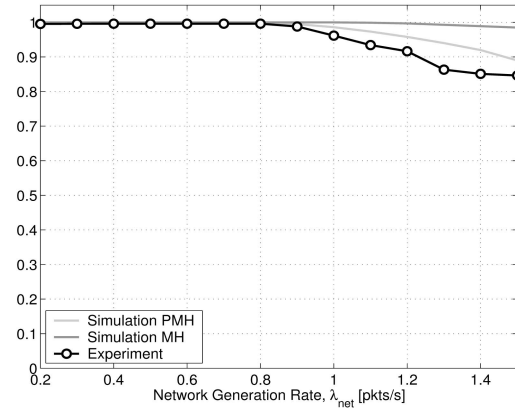

Fig. 20. Delivery rate versus λ_{net}.

calculated considering the geographical advancements provided by the nodes, i.e., for node n, $c_n = 1 - a_n/R$, where a_n is the advancement toward the sink of node n, and R is the maximum transmission range.

During each experiment, the packets received at the sink are collected, storing the source and the number of hops traversed. Also, each node saves statistics about all its (one-hop) contentions. The performance metrics we show here are the *delivery rate*, defined as the average number of packets delivered to the sink over the total number of packets generated, the *fraction of duplicate packets* reaching the sink, and the *distribution of the contention outcome*. This last metric represents the statistics of the number of contention rounds for the election of the next hop.
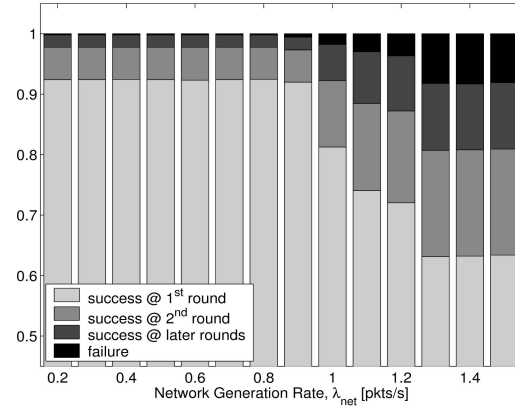
Fig. 20 shows the delivery rate, averaged over all nodes, as a function of λ_{net}. Both simulation and experimental points are shown in the plot. To this end, we used the simulator presented in Section 4, which was configured to reproduce, as closely as possible, our experimental setting. Two types of simulations were run. In the former, nodes are in a real multihop scenario (referred to as MH in the figure), whereas the latter reproduces the PMH environment that we used in our experiments. The results show that there is some difference between simulation and experiments, because of the many complex effects that are not accounted for in the simulation (and which lead to some performance degradation), as well as between the simulation results for the MH and PMH environments (with MH performing a little better that PMH). Despite these differences, which are in agreement with what we intuitively expected, the behavior of the curves is consistent, and the qualitative trend is very similar (e.g., the value of traffic for which the network starts losing packets is the same in all cases). This confirms the correctness of our simulation approach and validates the use of the PMH approach in our experiments.

In Fig. 20, we can assert that our scheme works properly, providing a delivery rate close to one until the traffic saturates the network. This happens for a $\lambda_{net} \approx 1$ packets/second, which is smaller than the capacity bound calculated in the previous section (about 3.3 packets/second). This is basically due to the collisions and to the nonideality of MAC and sensor hardware. We finally observe that the nodes placed farther away from the sink (having a longer
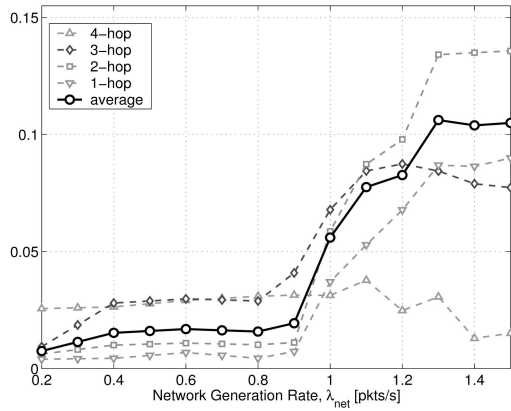
Fig. 21. Fraction of duplicate packets versus λ_{net}.
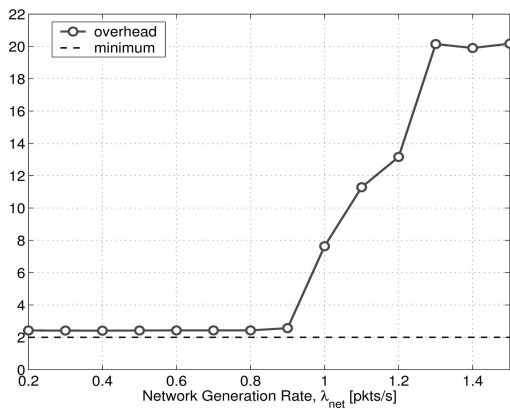
Fig. 22. Protocol overhead versus λ_{net}.

Fig. 23. Distribution of the contention outcome versus λ_{net}.

hop count distance) are those most affected as the traffic load increases, as expected.

In Fig. 21, we plot the fraction of duplicate packets delivered to the sink. We plot different curves according to the hop count distance traversed by the packets. In addition, with the bold curve, we report the same performance averaged over all hop counts. Up to the saturation point for the network ($\lambda_{net} \leq 1$ packets/second), the percentage of duplicate packets is reasonably low (below 5 percent) and, as expected, is an increasing function of the hop count distance. For higher λ_{net}, the duplication rate increases sharply, and this is due to the higher collision probability we get for a partially congested network. In particular, this behavior is representative of the increased probability of losing ACKs while delivering correctly the corresponding DATA. As the level of congestion grows further, the nodes become unable to get their messages through the next hop. In particular, the event of losing both the DATA and the corresponding ACK becomes more likely than that of losing only the latter. This justifies the decreased duplication rate when the network is heavily congested. In the figure, this phenomenon is evident for a hop distance of three and four hops. However, the same behavior occurs for the remaining curves as well for higher values of λ_{net}. The protocol overhead is reported in Fig. 22 and is comparable with the results in Fig. 12 until the traffic load saturates the network

(as shown in the figure, the average overhead is about 2.4, i.e., close to the optimal performance).

Fig. 23 reports the probability of completing the transmission of a packet to the selected next hop in a given number of contention rounds. This figure is obtained by averaging over all nodes. Under normal operating conditions, CCMR succeeds in a single round with high probability (higher than 0.9). Also, the probability of failure (corresponding to seven consecutive unsuccessful rounds) is negligible. As expected, the performance degrades for an increasing λ_{net}. In this case, the nodes with a small hop count distance still experience acceptable performance (see Fig. 20), while the sensors placed farther from the sink are almost unable to deliver their packets, leading to a trend similar to that in Fig. 17.

The experimental results we discussed above demonstrate the feasibility of our scheme, as well as its effectiveness in practical environments.

6 COMPLEXITY CONSIDERATIONS

The dynamic programming formulation in Section 2 is found numerically. Also, as anticipated in the analysis, cost/token regions R_i are quantized for numerical tractability. Assume that cost and token spaces (both in [0, 1]) are quantized into M levels. Finding the optimal access policy amounts to recursively solving (13), and an estimate of the complexity associated with this calculation can be derived as follows: For $i = W$, R_{i-1} can take M^2 values. Let (i, j) be the generic region R_{i-1} where i and j are the quantized values of cost and token, respectively. For a given pair (i, j), there are $M^2 - ij$ possible (quantized) regions R_i such that $R_i \supset R_{i-1}$, as required by (13). Thus, the number of computations required when $i = W$ is $\sum_{i=1}^{M} \sum_{j=1}^{M} (M^2 - ij) = (3M^4 - 2M^3 - M^2)/4$. The same applies to all remaining slots $1, 2, \ldots, W - 1$, which gives a total complexity of the order of $\mathcal{O}(3WM^4/4)$. This procedure is computationally demanding and cannot be exploited in a practical algorithm. Conversely, the online scheme in Section 3, for a given N and W pair, needs a unique p_i^* sequence (calculated by means of (14)). Cost and token regions are, in this case, a simple reshaping of the p_i^*s according to the limits of the cost intervals (see Section 2.4).

p_i^* sequences can be calculated offline and conveniently tabulated for their use in sensor nodes. In our current implementation of the scheme, a very satisfactory set of lookup tables for values of N up to 50 and $W = 10$ takes less than 400 bytes of ROM memory (see Section 5.1). The size of this table can be further reduced at the cost of a slight decrease in performance. The calculation of the optimal cost and token thresholds c_i^* and t_i^* in this case involves at most $3W$ additions and W multiplications (see (15) along with its accompanying text). This number of operations can be further reduced if a node stops calculating cost/token regions as soon as they include its own cost/token pair. This makes CCMR attractive for actual sensor devices, as is demonstrated by our own lightweight implementation of the scheme.

7 CONCLUSIONS

In this paper, we presented an original integrated channel access and routing technique for WSNs. Our design objectives were to minimize the energy consumption (the number of packets sent for each channel contention), while maximizing the probability of picking the best (*lowest cost*) node in the forwarding set. By analysis, we found an online algorithm that we called CCMR. We tested this scheme via simulation, by comparing its performance against that of state-of-the-art solutions. Besides proving the effectiveness of the scheme, our results show its robustness against critical network parameters. Finally, we demonstrated the feasibility of implementing CCMR as a lightweight software module for TinyOS and validated the protocol experimentally.

Further optimizations of the software implementing CCMR are possible, including the possibility of using the protocol as a basis for data aggregation and distributed network coding techniques.

REFERENCES

[1] M. Zorzi and R.R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Trans. Mobile Computing,* vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.

[2] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-Based Forwarding for Mobile Ad-Hoc Networks," *Elsevier's Ad Hoc Networks,* vol. 1, no. 4, pp. 351-569, Nov. 2003.

[3] T. He, B.M. Blum, Q. Cao, J.A. Stankovic, S.H. Son, and T.F. Abdelzaher, "Robust and Timely Communication over Highly Dynamic Sensor Networks," *Real-Time Systems,* vol. 37, no. 3, pp. 261-289, Dec. 2007.

[4] T. Melodia, D. Pompili, and I.F. Akyildiz, "Optimal Local Topology Knowledge for Energy Efficient Geographical Routing in Sensor Networks," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[5] D. Ferrara, L. Galluccio, A. Leonardi, G. Morabito, and S. Palazzo, "MACRO: An Integrated MAC/ROuting Protocol for Geographical Forwarding in Wireless Sensor Networks," *Proc. IEEE INFOCOM '05,* Mar. 2005.

[6] M. Rossi and M. Zorzi, "Integrated Cost-Based MAC and Routing Techniques for Hop Count Forwarding in Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 6, no. 4, pp. 434-448, Apr. 2007.

[7] M.C. Vuran and I.F. Akyildiz, "Spatial Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 14, no. 2, pp. 316-329, Apr. 2006.

[8] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient Geographic Routing in Multihop Wireless Networks," *Proc. ACM MobiHoc '05,* May 2005.

[9] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM '02,* June 2002.

[10] A. Keshavarzian, H. Lee, L. Venkatraman, D. Lal, K. Chintalapudi, and B. Srinivasan, "Wakeup Scheduling in Wireless Sensor Networks," *Proc. ACM MobiHoc '06,* May 2006.

[11] S. Du, A.K. Saha, and D.B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM '07,* May 2007.

[12] Y.C. Tay, K. Jamieson, and H. Balakrishnan, "Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 6, pp. 1048-1057, Aug. 2004.

[13] P. Popovski, F.H. Fitzek, and R. Prasad, "Batch Conflict Resolution Algorithm with Progressively Accurate Multiplicity Estimation," *Proc. Joint Workshop Foundations of Mobile Computing (DIAL M-POMC '04),* Oct. 2004.

[14] E.V. Denardo, *Dynamic Programming: Models and Applications,* second ed. Dover Publications, 2003.

[15] S. Ghosh and S.G. Henderson, "Behavior of the NORTA Method for Correlated Random Vector Generation as the Dimension Increases," *ACM Trans. Modeling and Computer Simulation,* vol. 13, no. 3, pp. 276-294, July 2003.

[16] L. Badia, N. Bui, M. Miozzo, M. Rossi, and M. Zorzi, "On the Exploitation of User Aggregation Strategies in Heterogeneous Wireless Networks," *Proc. 11th Int'l Workshop Computer-Aided Modeling, Analysis and Design of Comm. Links and Networks (CAMAD '06),* June 2006.

[17] B. Krishanamachari, *Networking Wireless Sensors,* first ed. Cambridge Univ. Press, 2005.

[18] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," *Proc. Second ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* Nov. 2004.

[19] D. Chen, J. Deng, and P.K. Varshney, "A State-Free Data Delivery Protocol for Multihop Wireless Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05),* Mar. 2005.

[20] R. Crepaldi, S. Friso, A. Harris III, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi, "The Design, Deployment, and Analysis of SignetLab: A Sensor Network Testbed and Interactive Management Tool," *Proc. Third Int'l Conf. Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom '07),* May 2007.

[21] Infineon Technologies, *EyesIFXv2 Version 2.0,* http://www.infineon.com, Aug. 2007.

[22] *TinyOS: An Open-Source OS for the Networked Sensor Regime,* http://www.tinyos.net/, Aug. 2007.

[23] N. Bui, R. Crepaldi, and M. Rossi, *CCMR Software Module for TinyOS,* http://dgt.dei.unipd.it/download, Aug. 2007.

[24] S. Toumpis and A. Goldsmith, "Capacity Regions for Wireless Ad Hoc Networks," *IEEE Trans. Wireless Comm.,* vol. 2, no. 4, pp. 736-748, July 2003.

Michele Rossi received the Laurea degree (with honors) in electrical engineering and the PhD degree in information engineering from the University of Ferrara in 2000 and 2004, respectively. From March 2000 to October 2005, he was a research fellow in the Department of Engineering, University of Ferrara. During 2003, he was on leave from the Center for Wireless Communications (CWC), University of California, San Diego (UCSD), where he did research on wireless sensor networks. In November 2005, he joined the Department of Information Engineering, University of Padova, Padova, Italy, where he is an assistant professor. He is also with Consorzio Ferrara Ricerche (CFR), Ferrara, Italy. He is currently part of the EU-funded SENSEI project. His research interests include data distribution through network coding for wireless ad hoc and sensor networks, routing and access selection in heterogeneous wireless networks, integrated MAC/routing algorithms, in-network data aggregation, and compressive sensing techniques for wireless sensor networks. He is a member of the IEEE.

Nicola Bui received the Laurea degree in telecommunication engineering from the University of Ferrara in 2005. He is currently with the Consorzio Ferrara Ricerche (CFR), Ferrara, Italy, and with the Department of Information Engineering, University of Padova, Padova, Italy. He is part of the EU-funded SENSEI project. His main research interests are centered around the design, simulation, and experimentation of protocols and applications for wireless sensor networks and embedded systems.

Michele Zorzi received the Laurea degree and the PhD degree in electrical engineering from the University of Padova, Padova, Italy, in 1990 and 1994, respectively. He has been a professor of telecommunications in the Department of Information Engineering, School of Engineering, University of Padova, since 2003. He is also with Consorzio Ferrara Ricerche (CFR), Ferrara, Italy. Prior to his current appointment, he was a faculty member at the Politecnico di Milano (1993-1996), a Research Scientist in the Center for Wireless Communications, University of California, San Diego (1995-1998), and an associate professor (1998-2000) and then professor (2000-2003) at the University of Ferrara. He is very active in the field of wireless communications and has published more than 200 papers in refereed journals and international conference proceedings. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, underwater acoustic sensor networks, and energy-constrained communications protocols. He was the editor in chief of the *IEEE Wireless Communications Magazine* from 2003 to 2005, is currently the editor in chief of the *IEEE Transactions on Communications*, and serves on the editorial boards of the *IEEE Transactions on Wireless Communications*, the *Wiley Journal of Wireless Communications and Mobile Computing*, and the *ACM/URSI/Kluwer Journal of Wireless Networks*. He was also guest editor for special issues in the *IEEE Personal Communications Magazine* ("Energy Management in Personal Communications Systems," June 1998) and the *IEEE Journal on Selected Areas in Communications* ("Multi-Media Network Radios," May 1999, and "Underwater Wireless Communications and Networks," Dec. 2008). He is a fellow of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.