Wireless Netw (2008) 14: 55–70
DOI 10.1007/s11276-006-7791-8

Statistically assisted routing algorithms (SARA) for hop count based forwarding in wireless sensor networks

Michele Rossi · Michele Zorzi · Ramesh R. Rao

Published online: 9 June 2006

Abstract The main goal of this paper is to provide routing–table-free online algorithms for wireless sensor networks (WSNs) to select cost (e.g., node residual energies) and delay efficient paths. As basic information to drive the routing process, both node costs and hop count distances are considered. Particular emphasis is given to greedy routing schemes, due to their suitability for resource constrained and highly dynamic networks. For what concerns greedy forwarding, we present the Statistically Assisted Routing Algorithm (SARA), where forwarding decisions are driven by statistical information on the costs of the nodes within coverage and in the second order neighborhood. By analysis, we prove that an optimal online policy exists, we derive its form and we exploit it as the core of SARA. Besides greedy techniques, sub–optimal algorithms where node costs can be partially propagated through the network are also presented. These techniques are based on real time learning LRTA algorithms which, through an initial exploratory phase, converge to quasi globally optimal paths. All the proposed schemes are then compared by simulation against globally optimal solutions, discussing the involved trade–offs and possible performance gains. The results show that the exploitation of second order cost information in SARA substantially increases the goodness of the selected paths with respect to fully localized greedy routing. Finally, the path quality can be further increased by LRTA schemes, whose convergence can be considerably enhanced by properly setting real time search parameters. However, these solutions fail in highly dynamic scenarios as they are unable to adapt the search process to time varying costs.

Keywords Wireless sensor networks · Optimal routing · Greedy forwarding policies · Online optimization techniques · Protocol design · System analysis

1. Introduction

In the past few years, a great deal of work has been pursued to design efficient routing algorithms for Wireless Sensor Networks (WSNs) [1–10]. These networks should operate unattended for long period of times. Their task is to sense the environment and report their measurements to a single or a few special nodes referred to as *sinks*. These nodes are in charge of further processing the gathered data as well as forwarding it to some data collection point in the network. A crucial aspect in WSNs is to save energy, so as to prolong the network lifetime. However, energy saving does not come alone; equally important goals are to avoid loops in packet routing, control the packet delivery delays, minimize collisions at the MAC layer and include these features in scalable routing algorithms. Moreover, WSNs are much more energy/computationally constrained than, e.g., Mobile Ad hoc NETworks (MANETs [11]) and this makes these challenges more difficult to be solved. In addition, due to the importance of saving energy, routing schemes should be designed to limit the information exchanged to update routing tables and, at the same time, they should still be capable of selecting good paths. A commonly accepted approach to save energy is to periodically switch off the radio of some nodes in the network [4, 5, 12], thereby leading to a possibly

M. Rossi (✉) · M. Zorzi
Department of Information Engineering of the University of Padova, via Gradenigo 6/B, 35131 Padova, Italy
e-mail: {michele.rossi, michele.zorzi}@dei.unipd.it

R.R. Rao
University of California San Diego (UCSD), 9500 Gilman Drive, La Jolla, CA, 92093-0436, USA
e-mail: rrao@ucsd.edu

intermittently connected and highly dynamic environment. In such a scenario, conventional ad hoc routing schemes such as DSR [13], AODV [14] and others [15–17], could fail. Specifically, reactive schemes [13,14] could fail to discover a complete path, while proactive ones [17] could fail to converge, resulting in a network flooded with topology update messages. A possible solution is to not account for routing tables at all [2–4,7]. In fact, a generic sensor node only needs to know which is the best node to be selected as relay for the current transmission and does not necessarily need to have a full description of the network.

In line with this, in this paper we present routing solutions that we name Statistically Assisted Routing Algorithms (SARA). SARA is a family of forwarding techniques where the path selection is accomplished through a hop-by-hop forwarding process, accounting for the state of the nodes within transmission range, and statistically accounting for the state of the nodes placed outside the coverage area. Each node is assumed to know its own cost and hop count value. The cost is used to represent the state of a candidate relay as well as of the link connecting it to the current node. Possible cost models are discussed in Section 3. The hop count (HC), instead, is the number of hops separating the node from the destination (*sink*) in the minimum hop path [18]. HCs are used here just as coordinates are exploited in geographical routing [1,4,6,19,20], i.e., to have an indication of the distance that separates the current node from the sink and therefore drive the forwarding process towards a well–defined direction. In addition, nodes are assumed to have statistical knowledge about the minimum cost in their second order neighborhood, i.e., regarding the nodes that are placed two hops away and in the direction towards the sink. In Section 5.4, it is shown that an optimal online stationary routing policy under the considered system model exists. This policy is used as the core of the SARA scheme. In addition to greedy forwarding, in Section 5.5 we also present online optimization routing algorithms based on real time search techniques [21], where cost estimates are propagated through the network with the aim of approaching optimal paths. These solutions are effective as the network dynamics is slower than the dynamics of the path searching phase. We introduce these schemes to understand whether online search with the subsequent propagation of node costs can be feasible and, in such a case, what the expected gains are in terms of path cost.

The paper is structured as follows. In Section 2 we review the related work. In Section 3, we introduce the network models along with the assumptions that we have considered to carry out the analysis. In Section 4, we introduce the concept of path optimality by discussing the exact algorithm that is used to compute optimal paths. These are used as a benchmark for the considered online schemes. In Sections 5.1 to 5.4, we present several greedy routing algorithms by investigating optimal localized routing rules. In Section 5.5, we focus on sub–optimal algorithms where the cost information can be propagated through the network. These algorithms find cost–efficient paths through controlled search techniques. In Section 6, we compare the routing schemes with each other and with optimal off–line solutions, discussing their efficiency in terms of path quality and their suitability as a function of the network dynamics. Section 7 concludes the paper.

2. Related work

Several algorithms have been proposed so far to obtain scalable and efficient routing over WSNs. A widely considered approach to the routing problem consists of geographical routing [1,4,6,22–24], where nodes know both their own coordinates and the coordinates of the sink. Within geographical routing, data forwarding is usually achieved by delivering the packet to the node with the largest geographical advancement towards the sink. This technique is named as Most Forward within Radius (MFR) [25]. If the forwarding set is temporarily empty, and this could happen if all nodes in the set are asleep for energy saving purposes, the relay election fails and has to be re-scheduled to a later time. This may also happen due to the presence of low connectivity areas (*connectivity holes*). In the latter case, additional mechanisms [1], [26] are required to get around low connectivity regions. A common solution to achieve energy saving is to allow nodes to periodically turn off their radio [4,5,10,12]. In [4], the authors consider MFR geographical routing by presenting a cross-layer approach for the efficient selection of relay nodes. They divide the forwarding areas in priority regions according to the related advancements. Nodes in higher priority regions are the first to contend to be the relay. In [5,12] the authors propose topology management and MAC protocols for WSNs which aggressively exploit sleeping modes to save energy. In [10] the authors report both analytical and test bed results by showing the benefits of putting nodes to sleep and awake them when needed, i.e., in case of loss of connectivity. In [9], the authors focus on geographical localized routing, by introducing the new concept of optimal topology knowledge range. The main idea is to locally and dynamically adjust the transmission power to get the optimal amount of topology information and make efficient routing decisions. Papers [22–24] couple geographical routing with residual energy and link quality aspects; these are included into link cost metrics. The relay is elected through probabilistic channel access schemes where nodes with low cost are prioritized. We observe, however, that all geographical routing approaches have several problems which may be due to low connectivity areas or to the impossibility, in some environments (e.g. underwater sensor networks [27]), to get accurate position estimates. Moreover, dealing with node coordinates and related trigonometric formulas may be computationally expensive and/or energy consuming for resource constrained WSNs. The geographical knowledge can be avoided by algorithms such as [3,28,29] which use probabilistic, gossip-based or controller flooding algorithms. These solutions are valuable in highly intermittently connected networks, where the position information is not available or too expensive to achieve due to the high network dynamics. The main drawback of these solutions is the usually large overhead induced by the transmission of multiple copies of a single packet. Other solutions consider hop count (HC) routing [2, 7, 30–33], where the geographical information is substituted by an integer number representing the number of transmissions needed to reach the sink on the shortest path. In [2] the authors present directed diffusion, where the sink queries sensor nodes by disseminating an interest, i.e., a list of attribute–value pairs for the desired data. Subsequently, reverse paths are built from all potential sensing sources to the sink following the gradients created during the interest dissemination. In addition, reinforcement mechanisms are exploited to select high quality paths among the multiple paths available. In GRAB [7], HC based cost fields are built (during the interest dissemination) and maintained to provide each sensor with a criterion to forward data. Costs are accumulated from the sink up to the interested sensing units. However, the cost field needs to be periodically updated as link costs vary for the scheme to be effective. In [31,32], packets are forwarded along strictly decreasing HC gradients. Also, they present several techniques to deal with HC reconfiguration and energy–aware next hop selection. In the present paper, we propose online algorithms based on HC gradient. However, we do not propagate cost information from the sink to the sensor nodes but we rather pick the next hop on-the-fly and based on local cost information. For this reason, when the link cost dynamics is high, our scheme is more robust with respect to [7]. Also, our work generalizes previous approaches [2,31,32] as we model the path quality through generic cost functions by devising optimal routing policies that, in order to increase the path quality, also exploit non–decreasing HC directions and statistical measures of two–hops away link costs.

3. Network model

We model the network as a weighted graph $\mathcal{G} = (N, A)$, consisting of a set N of nodes that is composed of $|N| - 1$ nodes and one special node (the sink) whose function is to gather and process network messages. A is a set of ordered pairs (i, j), $i, j \in N$. (i, j) is referred to as the arc (or link) connecting node i with node j. Each link (i, j) has a cost c_{ij} that depends on both endpoints i and j. For the connectivity, we consider bi-directional links, but we assume asymmetrical costs, i.e., in general, $c_{ij} \neq c_{ji}$. To keep the analysis as general as possible, we do not specify here a specific propagation model. In fact, our framework is based on neighboring sets, i.e., on sets of nodes within coverage that satisfy certain properties (see Section 5.1). In fact, as the network may be highly dynamic, these sets may vary between subsequent forwarding actions and therefore depend on many factors such as connectivity model and node sleeping cycles. For this reason, it makes sense to obtain these sets on–demand when the forwarding decision has to be made. This may be implemented as in [4,22], where the next hop is elected by means of a probabilistic channel contention every time a packet has to be transmitted. In such a case, we might associate higher access probabilities to the more desirable nodes within range, i.e., to the lowest cost nodes. As an example, the cost may be expressed as a function of several quantities: $c_{ij} = f(P^s_{ij}, E^{res}_j, a_j)$, where P^s_{ij} is the link quality (packet success probability), E^{res}_j is the normalized residual energy of node j, and a_j is the normalized advancement towards the destination associated with node j.[1] Several forms for the cost function can be found in [22,24,34]. In [24], for instance, the cost is calculated as $c_{ij} = 1 - a_j P^s_{ij}$, where $a_j P^s_{ij}$ is the expected advancement towards the sink. In [34] the authors calculate link costs based on both energy (E^{res}_j) and information progress ($a_j P^s_{ij}$) as $c_{ij} = 1 - a_j P^s_{ij} E^{res}_j$. In [22] the cost is modeled through a weighted sum $c_{ij} = \xi(1 - E^{res}_j) + (1 - \xi)(1 - a_j)$, $\xi \in [0, 1]$ and P^s_{ij} is accounted by probabilistically ruling out nodes with a low P^s_{ij} in the channel access. In more detail, on receiving a forwarding request, the potential relay nodes calculate P^s_{ij} and decide not to participate in the subsequent channel contention if P^s_{ij} is too small. The nodes deciding to take part in the channel contention probabilistically contend to be the relay and contention probabilities are calculated as a function of link costs. We observe that the cost can also be associated with other factors such as transmission power and path delay [24]. In the present paper we do not account for a specific cost model, by however observing that our framework can work with any of the approaches above. Furthermore, throughout the paper we assume link costs c_{ij} to be independent of the starting node i, $c_{ij} = c_j, \forall\, i, j \in N$. This assumption is made to keep the notation simple. In fact, the analysis that we present next still holds in the general case where costs depend on both i and j. Also, throughout the paper and without loss of generality we consider normalized costs in [0, 1]. For every node $i \in N$ we define its hop count (HC(i)) as the minimum number of transmissions needed to get to the sink [2, 7, 30–33]. The interested reader is referred to the above references for more details on how HC fields can be created and maintained. We

[1] The residual energy and the advancement are usually normalized with respect to the initial energy reserve and the maximum transmission range, respectively.

do not address these issues here, as the focus of the present paper is on routing schemes exploiting the HC information. We therefore assume node HCs as given and without estimation errors. We define $\mathcal{N}_i$ as the set containing all the neighboring nodes of node i. We also define $\mathcal{N}_i(n)$, $\mathcal{N}_i(n-1)$ and $\mathcal{N}_i(n+1)$, $n \in \mathbb{N}^+$ as the sets of neighboring nodes of node i with HC equal to n, $n-1$ and $n+1$, respectively. We define a path $\mathcal{P}$ from node s to node d as an ordered list $\mathcal{P} = \{s, r_1, r_2, \ldots, r_k, d\}$, where nodes s and d are referred to as the source and the destination, respectively, whereas r_j, $j \in \{1, 2, \ldots, k\}$ are referred to as relay nodes. The cost $C(\mathcal{P})$ associated with path $\mathcal{P}$ is calculated as[2]

$$C(\mathcal{P}) = c_{sr_1} + \sum_{j=1}^{k-1} c_{r_j r_{j+1}} + c_{r_k d} \tag{1}$$

In addition, for each path $\mathcal{P}$ we consider its length, expressed as $D(\mathcal{P})$, i.e., the number of links in path $\mathcal{P}$. In the sequel, we refer to $D(\mathcal{P})$ as path delay. The choice of selecting an additive cost function as the path cost criterion is reasonable since additive metrics arise in many settings. For instance, end-to-end delay, delay jitter, maximum total residual energy and reliability all correspond to the sum of link weights. In the reliability case, in fact, the sum of the logarithms of the link failure probabilities can be used to minimize their product [18].

4. Optimal routing as a bicriteria optimization problem

Our main concern in this paper is to devise efficient online algorithms to provide both *cost* and *delay* efficient paths. Of course, it is valuable to compare the goodness of these paths against optimal paths to understand how close we can get to optimal performance. Hence, we introduce here the technique that we have considered to find optimal solutions. Optimal paths are found by solving a bicriteria optimization problem, where the two optimization criteria are the total path cost C and the path delay D. In particular, for a given network topology and node cost distribution we are interested in finding the collection of all the optimal paths connecting every pair of nodes in the network. The word *optimal* deserves some explanation. Here, we are to solve a multi–objective optimization problem, where in general the two objective functions represent non–comparable criteria. The problem is that the meaning of *optimum* is not well defined in this context [35]. To overcome this fact and to give an interpretation to the notion of optimum, we utilize the concept of *Pareto optimality* [36]. The Pareto optimum almost always does not give a single solution but a set of solutions called *non-dominated* set [35]. This set, for a given source-destination pair (s, d), consists of the collection of all the non–dominated, loop free paths connecting node s to node d. Here, we obtain non–dominated set by means of standard bicriteria Shortest Path Tree (SPT) algorithms [37]. For illustration, consider Fig. 1, where we report seven points in the cost (C) delay (D) plane corresponding to all paths connecting a given (s, d) pair. The non–dominated Pareto set is {P1, P4, P5}. For instance, P4 dominates both P2 and P7 as it provides better performance for both metrics C and D. However, P1 and P4 do not dominate each other as solution P4 has a smaller cost and a longer delay. In practice, non–dominated solutions perform better for one criterion and worse for the second one. Hence, it does not make sense to pick dominated solutions as they are inefficient, i.e., they can be outperformed for both metrics by a non–dominated solution. In Section 6, we compare online routing algorithms against the minimum cost non–dominated solution OPT-CO (P5 in Fig. 1) and the non-dominated solution with the shortest delay OPT-DO (P1 in Fig. 1).

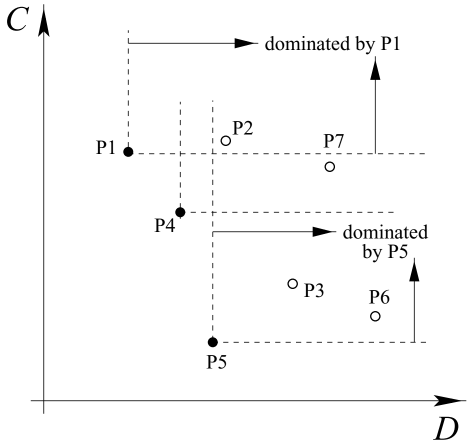

Fig. 1 A graphical example for the Pareto optimal set (black filled circles)

5. Sub–optimal forwarding strategies

In this section, we investigate two possible sub–optimal routing strategies. First, in Sections 5.1 to 5.4 we focus on localized greedy forwarding strategies. Later on, in Section 5.5 we consider online search algorithms, where a controlled back–propagation of the costs in the network is allowed with the aim of selecting quasi globally optimal paths.

5.1. Greedy forwarding techniques

Assume that the current node is node i with $\mathrm{HC}(i) = n$, and that a next hop in set $\mathcal{N}_i$ has to be chosen for forwarding

[2] Under the hypothesis of additive cost function, see [18].

purposes. The next hop can be either in set $\mathcal{N}_i(n-1)$ or $\mathcal{N}_i(n)$, set $\mathcal{N}_i(n+1)$ is not considered since, as shown in Section 6, the nodes therein rarely lead to an increase of the path quality. Let nodes $j_{n-1} \in \mathcal{N}_i(n-1)$ with cost c_{n-1} and $j_n \in \mathcal{N}_i(n)$ with cost c_n be the minimum cost nodes in sets $\mathcal{N}_i(n-1)$ and $\mathcal{N}_i(n)$, respectively. If in $\mathcal{N}_i(n-1)$ ($\mathcal{N}_i(n)$) there is more than one node with the same minimum cost, the minimum cost node j_{n-1} (j_n) is selected by breaking ties arbitrarily. Nodes with both the same minimum cost and HC distance in our framework are, in fact, equivalent. The relay node is selected by means of the generic function $P_{n-1}(c_{n-1}, c_n)$. In particular, node j_{n-1} is selected with probability $P_{n-1}(c_{n-1}, c_n)$, whereas node j_n is selected with probability $P_n(c_{n-1}, c_n) = 1 - P_{n-1}(c_{n-1}, c_n)$. In this paper, we investigate the following three specializations of the probabilistic scheme presented above

(1) In the first scheme, node $j_{n-1} \in \mathcal{N}_i(n-1)$ is always selected as the relay node ($P_{n-1}(c_{n-1}, c_n) = 1$, $\forall$ (c_{n-1}, c_n)). This strategy may be far from being optimal from the cost point of view, but always corresponds to the shortest path in terms of number of hops.

(2) In the second scheme, node $j \in \{\mathcal{N}_i(n-1) \cup \mathcal{N}_i(n)\}$ with the lowest cost is selected as the relay node. The function $P(\cdot)$ is specified as follows: $P_{n-1}(c_{n-1}, c_n) = 1$ if $c_{n-1} \le c_n$ and $P_{n-1}(c_{n-1}, c_n) = 0$ otherwise. In the case where $c_{n-1} = c_n$, node $j_{n-1} \in \mathcal{N}_i(n-1)$ is preferred to achieve a positive advancement towards the destination.

(3) In the last scheme, which can be seen as a generalization of the previous one, if $c_{n-1} \le c_n$ node $j_{n-1} \in \mathcal{N}_i(n-1)$ is always selected as before (which corresponds to $P_{n-1}(c_{n-1}, c_n) = 1$ for $c_{n-1} \le c_n$). If on the other hand $c_{n-1} > c_n$, a probabilistic selection is made between nodes j_{n-1} and j_n. In particular, node j_{n-1} is selected with probability $P_{n-1}(c_{n-1}, c_n) = p_{n-1}(c_{n-1}, c_n)$, whereas node j_n is selected with probability $1 - p_{n-1}(c_{n-1}, c_n)$. (Note that the previous scheme corresponds to $p_{n-1}(c_{n-1}, c_n) = 0$ for $c_{n-1} > c_n$.) Suitable forms for the function $p_{n-1}(c_{n-1}, c_n)$ are presented in Section 5.3.

5.2. Analysis of greedy forwarding strategies

In this section, we characterize both energy expenditure and delay of the previously proposed greedy forwarding schemes. We assume that node i is the node currently occupied by the forwarding process, where $\text{HC}(i) = n$. For analytical tractability we assume to have, at every node i, $K_{n-1} \ge 1$ and $K_n \ge 1$ nodes in $\mathcal{N}_i(n-1)$ and $\mathcal{N}_i(n)$, respectively. K_{n-1} and K_n are assumed to be constant. Observe that this assumption is not restrictive and can be easily removed through a more cumbersome analysis, but leaving the approach unchanged.[3] We indicate the cost of node i as $c_i \in [0, 1]$, where C_i is the related r.v. Moreover, we express the cumulative distribution function (cdf) of the cost at node i as $F_i(c) = \text{Prob}\{C_i \le c\}$. Costs of different nodes are assumed to be independent. Now, consider N nodes, randomly picked from $\mathcal{N}_i$. We define a further r.v. X_N as the minimum cost among these N nodes, $X_N = \min\{C_1, C_2, \ldots, C_N\}$. The cdf of X_N is given by

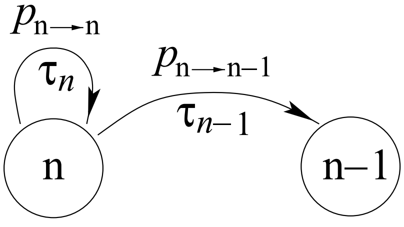
$p_{\text{n}\rightarrow\text{n}}$
τ_n
$p_{\text{n}\rightarrow\text{n}-1}$
τ_{n-1}
n
n–1

Fig. 2 Transition diagram for the greedy forwarding technique

$$F_{min}(c|N) = 1 - \prod_{j=1}^{N} [1 - F_j(c)] \tag{2}$$

We observe that, when all nodes are characterized by the same cost distribution, i.e., $F_i(c) = F(c)\ \forall\ i$, $f_{min}(c|N)$, the probability density function (pdf) of X_N, is $f_{min}(c|N) = N(1 - F(c))^{N-1} f(c)$, where $f(\cdot)$ is the node cost pdf. Now, referring to the greedy forwarding algorithms in Section 5.1, at the generic node i with $\text{HC}(i) = n$, the probabilities of selecting a node with HC equal to $n-1$ or n are derived as follows

$$\begin{aligned} p_{n\to n} &= \int_0^1 \int_0^1 g(c_{n-1}, c_n) P_n(c_{n-1}, c_n)\, \mathrm{d}c_n \mathrm{d}c_{n-1} \\ p_{n\to n-1} &= \int_0^1 \int_0^1 g(c_{n-1}, c_n) P_{n-1}(c_{n-1}, c_n)\, \mathrm{d}c_n \mathrm{d}c_{n-1} \end{aligned} \tag{3}$$

where $p_{n\to n}$ is the probability of selecting a relay node j with $\text{HC}(j) = n$, given that the current node i has $\text{HC}(i) = n$, whereas $p_{n\to n-1}$ is the probability of selecting a node j with $\text{HC}(j) = n-1$. $g(c_{n-1}, c_n) = f_{min}(c_{n-1}|K_{n-1}) f_{min}(c_n|K_n)$ is the joint pdf of the minimum costs for the K_{n-1} and K_n nodes in sets $\mathcal{N}_i(n)$ and $\mathcal{N}_i(n-1)$, respectively. $P_n(c_{n-1}, c_n)$ and $P_{n-1}(c_{n-1}, c_n) = 1 - P_n(c_{n-1}, c_n)$ are defined in Section 5.1. Now, refer to the transition diagram in Fig. 2. The forwarding process is a discrete time process that, at every stage, evolves by selecting a next hop that is either in set $\mathcal{N}_i(n-1)$ or in set $\mathcal{N}_i(n)$. Let stage 0 be the discrete instant at which this process enters node i with $\text{HC}(i) = n$ for

[3] The full approach also requires the joint distribution of K_{n-1} and K_n. However, the delay and cost metrics obtained in this section can be easily plugged in the full analysis through a further expectation.

the first time. We define a *cycle* as the forwarding history between time 0 and the stage in which a node with HC $n-1$ is selected for the first time, i.e., the first stage where a positive hop count advancement towards the destination occurs. The number of stages it takes to complete a cycle is referred to as the *cycle duration*, D, and is the number of subsequent forwarding events needed to complete the cycle. The aim of the following analysis is to characterize the cycle in terms of cost and delay. The mean $E[D]$ and the variance σ_D^2 of the delay D, are derived as

$$\begin{aligned} E[D] &= p_{n\to n-1}\sum_{n=1}^{+\infty} n\ p_{n\to n}^{n-1} = \frac{1}{p_{n\to n-1}} \\ \sigma_D^2 &= E[D^2]-E[D]^2 = \frac{1-p_{n\to n-1}}{p_{n\to n-1}^2} \end{aligned} \tag{4}$$

where $f_D(n) = p_{n\to n}^{n-1}p_{n\to n-1}$, $n \geq 1$ is the cycle duration probability mass function (pmf). The computation of the cost metrics is slightly more complicated. We refer to transitions $n \to n-1$, $n \to n$ and to their associated costs as τ_{n-1}, τ_n and c_{n-1}, c_n, respectively (see Fig. 2). The k-th order moments of the cost associated with each transition, $E[C^k|\tau_{n-1}]$ and $E[C^k|\tau_n]$, are given by

$$\begin{aligned} E[C^k|\tau_{n-1}] &= \int_0^1\int_0^1 g_{n-1}(c_{n-1},c_n)\ c_{n-1}^k\ \mathrm{d}c_n\mathrm{d}c_{n-1} \\ E[C^k|\tau_n] &= \int_0^1\int_0^1 g_n(c_{n-1},c_n)\ c_n^k\ \mathrm{d}c_n\mathrm{d}c_{n-1} \end{aligned} \tag{5}$$

where $g_x(c_{n-1},c_n) = g(c_{n-1},c_n)P_x(c_{n-1},c_n)/p_{n\to x}$ and $x \in \{n-1, n\}$. The total expected cost for a single transition is given by $E[C] = p_{n\to n-1}E[C|\tau_{n-1}] + p_{n\to n}E[C|\tau_n]$. To find the cost associated with a cycle we assume that, at every decision stage, the set of possible forwarding nodes is independent of the previous one. Owing to this assumption, both mean and variance of the cost associated with a cycle can be computed through the sum of means and variances associated with each transition, respectively. If C_{cycle} is the r.v. of the total cost of a cycle of length k, then $C_{cycle} = C_{\tau_n}^1 + C_{\tau_n}^2 + \cdots + C_{\tau_n}^{k-1} + C_{\tau_{n-1}}^k$, where $C_{\tau_i}^j$ is the r.v. associated with transition τ_i at in the j-th stage of the cycle. Mean and variance of a cycle are derived as

$$\begin{aligned} E[C_{cycle}] &= \sum_{n=1}^{+\infty} f_D(n)\ [(n-1)E[C|\tau_n] + E[C|\tau_{n-1}]] \\ &= E[C|\tau_{n-1}] - E[C|\tau_n] + \frac{E[C|\tau_n]}{p_{n\to n-1}} \\ \sigma_{C_{cycle}}^2 &= \sigma_C^2(\tau_{n-1}) - \sigma_C^2(\tau_n) + \frac{\sigma_C^2(\tau_n)}{p_{n\to n-1}} \end{aligned} \tag{6}$$

where $E[C_{cycle}]$ and $\sigma_{C_{cycle}}^2$ are the mean and the variance of the cost associated with a cycle, respectively. $\sigma_C^2(\tau_i) = E[(C - E[C|\tau_i])^2|\tau_i]$ is the variance of the cost associated with transition τ_i. In Eq. (6) we introduced the further assumption that the r.v. governing the cost of a transition remains the same for all stages, i.e., it is invariant with respect to time and space.[4] In what follows, we specialize the analysis presented above to the first two schemes proposed in Section 5.1. To this end, we introduce a further assumption on the r.v. C governing node costs, by assuming C to be uniformly distributed in [0, 1], i.e., $f_{min}(c|N) = N(1-c)^{N-1}$. This is also reasonable if we relate the cost to the residual energy as proposed in Section 3. The third scheme proposed in Section 5.1 is treated in detail in Section 5.3.

[Scheme 1] The duration of each cycle is deterministically equal to one stage and $p_{n\to n-1} = 1$. According to Eq. (4), $E[D] = 1$, $\sigma_D^2 = 0$, and the expected value of the cost in a cycle $E[C_{cycle}]$ and its variance $\sigma_{C_{cycle}}^2$ are

$$\begin{aligned} E[C_{cycle}] &= \frac{1}{1+K_{n-1}} \\ \sigma_{C_{cycle}}^2 &= \frac{E[C_{cycle}]^2\ K_{n-1}}{2+K_{n-1}} \end{aligned} \tag{7}$$

[Scheme 2] The transition probability $p_{n\to n-1}$ is[5]

$$p_{n\to n-1} = \frac{K_{n-1}}{K_{n-1}+K_n} \tag{8}$$

therefore

$$\begin{aligned} E[D] &= \frac{K_{n-1}+K_n}{K_{n-1}} \\ \sigma_D^2 &= \frac{K_n\ E[D]}{K_{n-1}} \\ E[C|\tau_{n-1}] &= \frac{1}{1+K_{n-1}+K_n} \\ E[C^2|\tau_{n-1}] &= \frac{2E[C|\tau_{n-1}]}{2+K_{n-1}+K_n} \end{aligned} \tag{9}$$

where $E[C^k|\tau_n] = E[C^k|\tau_{n-1}]$, $k \geq 1$. This result is reasonable since the selection scheme makes no difference between

[4] This assumption is reasonable since, in all practical situations, the length of a cycle is usually bounded to a few transitions.

[5] In this case $p_{n\to n-1}$ equals the probability that the node with the lowest cost is in set $\mathcal{N}_i(n-1)$. As expected, the probability of this event is increasing in K_{n-1}. Correctly, as $K_{n-1} = K_n$ this probability equals 0.5.

nodes in $\mathcal{N}_i(n-1)$ and $\mathcal{N}_i(n)$, but it simply selects the node with the lowest cost. Finally,

$$E[C_{cycle}] = \frac{K_{n-1} + K_n}{K_{n-1}(1 + K_{n-1} + K_n)}$$
$$\sigma^2_{C_{cycle}} = \frac{E[C_{cycle}]^2 \; K_{n-1}}{2 + K_{n-1} + K_n} \tag{10}$$

Observe that σ^2_D is quadratic in K_n/K_{n-1} and that $E[C_{cycle}]$ and $\sigma^2_{C_{cycle}}$ are linear and quadratic in $(K_{n-1}+K_n)/K_{n-1}$, respectively. Therefore, this scheme is expected to lead to long paths with high variance when $K_n \gg K_{n-1}$.[6] Also, mean and variance of this scheme are the same as in Eq. (7) for $K_n = 0$.

5.3. One–stage statistically–assisted optimal greedy policy

In the third scheme presented in Section 5.1, we consider a probabilistic selection of the relay node. Let us focus first on a shortened version of the general decision process introduced above and labeled as a cycle. The results achieved for the shortened cycle, referred to here as *one–stage statistically–assisted greedy policy*, is generalized to the full cycle in Section 5.4. Assume that the forwarding process at stage 0 is in node i with HC$(i) = n$. Let c^0_{n-1} and c^0_n be the costs associated with $j_{n-1} \in \mathcal{N}_i(n-1)$ and $j_n \in \mathcal{N}_i(n)$, respectively. If node i chooses j_{n-1} as the next hop, the HC distance at stage 1 is $n-1$ and the cost associated with this transition is c^0_{n-1}. Hence, the cycle is completed with a total cost of c^0_{n-1}. Otherwise, if node j_n is selected, we force the next node (stage 1) to be picked from $\mathcal{N}_{j_n}(n-1)$ and the cycle is completed with a total cost of $c^0_n + c^1_{n-1}$. Therefore, at the generic node i, the decision maker has two possible actions $a \in \mathcal{A} = \{a_{n-1}, a_n\}$ to be taken: action a_{n-1} corresponds to picking j_{n-1}, whereas action a_n corresponds to picking j_n in the first stage and a node in the set $\mathcal{N}_{j_n}(n-1)$ in the second stage, i.e., to transition $n \to n \to n-1$. $\mathcal{A}$ is the action set. Observe that the two quantities c^0_{n-1} and c^0_n are known by node i. c^1_{n-1} is instead an unknown quantity described by the r.v. C. The best that can be done is therefore to make the forwarding decision based on the two known quantities and to statistically account for the unknown value c^1_{n-1}. For each action, we shall define the rewards as $R_{a_{n-1}} = c^0_n + c^1_{n-1} - c^0_{n-1}$ and $R_{a_n} = -R_{a_{n-1}}$, where $R_{a_{n-1}}$ is intended to weigh the convenience of selecting action a_{n-1} over a_n. In practice, $R_{a_{n-1}}$ gives us the cost spared in selecting a_{n-1} instead of a_n. Similarly, R_{a_n} weighs the convenience of selecting a_n over a_{n-1}. Now, we say that node j_n is preferable over j_{n-1} if the expected reward $E_a[R|K_{n-1}]$ is positive, where $\overline{R}_a = E_a[R|K_{n-1}]$ is the expectation taken over the action set $\mathcal{A}$ given that the number of nodes in $\mathcal{N}_{j_n}(n-1)$ is K_{n-1}

$$\begin{aligned}\bar{R}_a &= P_{n-1}\big(c^0_{n-1}, c^0_n\big)R_{a_{n-1}} + P_n\big(c^0_{n-1}, c^0_n\big)R_{a_n} \\ &= (1-2P_n)\big(c^1_{n-1} - \Delta\big)\end{aligned} \tag{11}$$

where $P_n = P_n(c^0_{n-1}, c^0_n)$ and $\Delta = c^0_{n-1} - c^0_n$. Now, we proceed by finding the value of P_n, P^{opt}_n, maximizing the expected reward. For the sake of generality, we relax the assumption regarding the number of nodes in set $\mathcal{N}_{j_n}(n-1)$, by allowing this number to be derived from a r.v. L whose pmf is $\phi(l)$. This pmf gives the probability that l nodes with HC $n-1$ are within radio range given that the current node has hop count n. To determine a P_n that is in some sense optimal, we need to define an optimality criterion. In this analysis, we use the total expected reward $\bar{R} = E[R]$ as our criterion, where the reward function is averaged over the number of nodes l in set $N_{j_n}(n-1)$, c^1_{n-1} and the action set $\mathcal{A} = \{a_{n-1}, a_n\}$

$$\begin{aligned}\bar{R} &= \sum_{l=1}^{+\infty}\left[\phi(l)\int_0^1 \bar{R}_a \;\mathrm{d}F_{min}\big(c^1_{n-1}|l\big)\right] \\ &= \sum_{l=1}^{+\infty}\left[\phi(l)\int_0^1 (1-2P_n)\big(c^1_{n-1} - \Delta\big)\;\mathrm{d}F_{min}\big(c^1_{n-1}|l\big)\right] \\ &= (1-2P_n)\left[\sum_{l=1}^{+\infty}\phi(l)E[C|l] - \Delta\right] \\ &= (1-2P_n)(E[C] - \Delta)\end{aligned} \tag{12}$$

where $E[C] = E[c^1_{n-1}]$ is the total expected minimum cost of the nodes in $\mathcal{N}_{j_n}(n-1)$. Eq. (12) gives us the optimal policy, i.e., the value of P_n that has to be selected to maximize $\bar{R}$, i.e., $P^{opt}_n = \text{argmax}_{P_n}\bar{R}$. Noting that $\bar{R}$ is a linear function of P_n and that its behavior depends on the sign of $(E[C] - \Delta)$, the optimal policy P^{opt}_n is found as

$$P^{opt}_n(\Delta) = \begin{cases} 0 & \Delta \le E[C] \\ 1 & \Delta > E[C] \end{cases} \tag{13}$$

The optimal policy is therefore a stationary policy [38] whose behavior depends on the difference $\Delta - E[C]$. Hence, at the generic node i with HC$(i) = n$, the optimal one–stage forwarding action is to proceed towards node j_n with probability $P_n = 1$ when $\Delta > E[C]$, and to node j_{n-1} otherwise.

[6] This, as shown in Section 6, is also the typical case for densely populated wireless sensor networks.

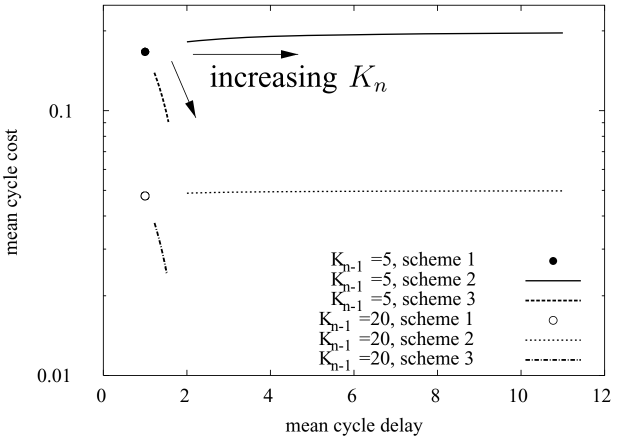

Fig. 3 $E[D_{cycle}]$ vs $E[C_{cycle}]$ for the three greedy schemes by varying K_n from K_{n-1} to $10K_{n-1}$

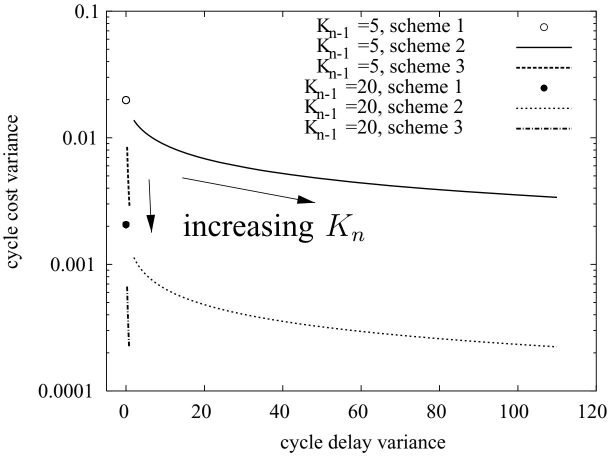

Fig. 4 $\sigma^2_{C_{cycle}}$ vs $\sigma^2_{D_{cycle}}$ for the three greedy schemes by varying K_n from K_{n-1} to $10K_{n-1}$

Observe that $E[C]$ depends both on node cost statistics and on the pmf $\phi(\cdot)$.[7]

Now, we derive cost and delay metrics for the policy where, at each stage, the action is chosen independently of the previous forwarding history and according to Eq. (13), i.e., we relax the constraint that action a_n must be always followed by action a_{n-1}. In order to be in line with what is derived in the previous section, we consider again K_{n-1} and K_n to be the constant number of nodes with HC $n-1$ and n at every stage, respectively. Moreover, we assume the costs to be independent and uniformly distributed in [0, 1]. The k-th order moment of the cost conditioned on transition τ_{n-1} is found as

$$E[C^k|\tau_{n-1}] = \int_0^1 \int_0^{E[C|K_{n-1}]} g_{n-1}(c_{n-1}, c_n)\, c_{n-1}^k \, \mathrm{d}c_{n-1}\mathrm{d}c_n \quad (14)$$

where $g_{n-1}(c_{n-1}, c_n) = g(c_{n-1}, c_n)/p_{n\to n-1}$ and $p_{n\to n-1}$ is given by

$$\begin{aligned} p_{n\to n-1} &= \int_0^1 \int_0^{E[C|K_{n-1}]} g(c_{n-1}, c_n)\, \mathrm{d}c_{n-1}\mathrm{d}c_n \\ &= 1 - K_n \frac{(1 - E[C|K_{n-1}])^{1+K_{n-1}}}{1+K_{n-1}} F_1 \end{aligned} \quad (15)$$

where $F_1 = {}_2F_1[1, 1-K_n; 2+K_{n-1}; 1-E[C|K_{n-1}]]$ and ${}_2F_1[\cdot]$ is the Gauss's hypergeometric function [39]. The other quantities are derived as in Section 5.2.

In Figs. 3 and 4, we report the cycle performance in terms of cost/delay mean and variance for the three greedy schemes discussed above. The curves are obtained by keeping K_{n-1} constant and varying K_n. Node costs are considered to be uniformly distributed in [0, 1]. As expected, by increasing K_n and/or K_{n-1} both cost mean and variance decrease. Scheme 1 gives optimal delay performance. Scheme 2, as K_n increases, outperforms scheme 1 in terms of cost variance, whereas it performs worse than scheme 1 for every other metric. In particular, it is inefficient from the delay point of view since $E[C_{cycle}]$ and $\sigma^2_{C_{cycle}}$ are linear and quadratic in K_n, respectively. Scheme 3 is always better than the other two forwarding techniques in terms of cost. Moreover, it also presents a satisfying behavior for what concerns the delay metric. In this sense, both mean delay and delay variance remain close to the optimal delay technique (scheme 1). The following observations are in order here. In geographical routing, relay nodes are usually picked among the neighbors leading to a positive advancement towards the destination [1,4,6,9]. Observe that nodes in set $\mathcal{N}_i(n-1)$ give a positive advancement towards the destination, exactly as the nodes considered in geographical schemes. However, nodes in $\mathcal{N}_i(n)$ may or may not provide such a positive advancement. Therefore, the forwarding set considered by HC based algorithms is broader. Therefore, more solutions can be reached. This means that HC routing has the potential of considering paths that in geographical schemes are discarded *a priori* and that, if correctly configured, has the potential of getting closer to optimal solutions.

5.4. Multi–stage optimal greedy routing policy

In this section, we generalize the one–stage optimal policy presented above to the full cycle structure by obtaining the optimal policy in the multi–stage case. Let $t \geq 0$ and node i, with HC$(i) = n$, be the current forwarding stage and the currently occupied node, respectively. Let $j_{n-1}^t \in \mathcal{N}_i(n-1)$ and $j_n^t \in \mathcal{N}_i(n)$ and c_{n-1}^t and c_n^t, be the minimum cost nodes and their costs at stage t, respectively. The action set and the

[7] Observe that, when the cost statistics are uniform and the number of nodes in set $\mathcal{N}_{j_n}(n-1)$ is equal to the constant K_{n-1}, $E[C] = E[C|K_{n-1}]$ reduces to $1/(1+K_{n-1})$.

decision maker's state are defined as $\mathcal{A}_t = \{a_{n-1}^t = j_{n-1}^t, a_n^t = j_n^t\}$ and $X_t = (c_{n-1}^t, c_n^t)$, respectively. The cost of taking action $a(t) \in \mathcal{A}_t$ when in state X_t is $C(X_t, a(t)) \geq 0$. Furthermore, let stage 0 be the beginning of a new cycle. For any policy π, the long term expected cost when $X_0 = X$ is the initial state is [38]

$$V_\pi(X) = E_\pi\left[\sum_{n=0}^{+\infty} C(X_n, a(n))\bigg| X_0 = X\right] \tag{16}$$

We refer to $V(X) = \inf_\pi V_\pi(X)$ as the minimum expected cost under any policy. A policy π^* is optimal if $V_{\pi^*}(X) = V(X)$, $\forall$ X. According to the dynamic programming framework [38] the optimal policy is characterized by the optimality equation as follows

$$V(X_t) = \min_{a(t)\in\mathcal{A}_t}\left[C(X_t, a(t)) + \int_{\mathcal{D}_X} V(X_{t+1})\, \mathrm{d}F(X_{t+1})\right] \tag{17}$$

where X_{t+1} is the state at stage $t+1$ (next forwarding stage), the term $\int_{\mathcal{D}_X} V(X_{t+1})\, \mathrm{d}F(X_{t+1})$ accounts for the average cost incurred in future decisions, $\mathcal{D}_X$ and $F(X_{t+1})$ are the domain set and the cdf of X_{t+1}, respectively. In what follows, we consider $C(X_t, a_{n-1}^t) = c_{n-1}^t$ and $C(X_t, a_n^t) = c_n^t$. The forwarding process is modeled here as an optimal stopping problem, where at stage t the decision maker can decide to either continue (select j_n^t) or stop (select j_{n-1}^t). In the former case, a cost $C(X_t, a_n^t) = c_n^t$ is paid and the cycle is continued towards j_n^t, whereas in the latter the cycle is ended with a final cost $C(X_t, a_{n-1}^t) = c_{n-1}^t$ and the future cost term $\int_{\mathcal{D}_X} V(X_{t+1})\, \mathrm{d}F(X_{t+1})$ is zero. Given that, $V(X_t)$ (Eq. (17)) is rewritten as

$$\min\left[C\big(X_t, a_{n-1}^t\big), C\big(X_t, a_n^t\big) + \int_{\mathcal{D}_X} V(X_{t+1})\, \mathrm{d}F(X_{t+1})\right] \tag{18}$$

We now define the following set

$$\mathcal{B}_1 = \left\{X_t : C\big(X_t, a_{n-1}^t\big) \leq C\big(X_t, a_n^t\big) + \int_{\mathcal{D}_X} C\big(X_{t+1}, a_{n-1}^{t+1}\big)\, \mathrm{d}F(X_{t+1})\right\} \tag{19}$$

Set $\mathcal{B}_1$ contains all states for which stopping is at least as good (on average) as continuing for one more stage and then stopping. The policy that stops as the process enters $\mathcal{B}_1$ is called *one–stage look–ahead policy*. Next, we investigate this policy first. $\mathcal{B}_1$ simplifies to

$$\mathcal{B}_1 = \left\{X_t : c_{n-1}^t - c_n^t \leq \mathcal{E}\right\} \tag{20}$$

$$\mathcal{E} = E\big[c_{n-1}^{t+1}\big] = \int_0^1 c_{n-1}^{t+1}\, \mathrm{d}F_{min}\big(c_{n-1}^{t+1}\big) \tag{21}$$

$\mathcal{E}$ and $F_{min}(\cdot)$ are the expected minimum cost among nodes with HC $n-1$ and its cdf at stage $t+1$, respectively. Hence, the one–stage optimal policy tells us to stop at stage t by selecting node j_{n-1}^t and ending the cycle, if $c_{n-1}^t - c_n^t \leq \mathcal{E}$. We must continue towards j_n^t otherwise. Not surprisingly, this is the same result obtained in Eq. (13). Before proceeding with the discussion on the optimality of the above one–stage policy we need to introduce some quantities. At every stage $t \geq 0$, the decision maker has to make a decision in the set $\mathcal{A}_t = \{a_{n-1}^t, a_n^t\}$. In case decision a_{n-1}^t is made, node j_{n-1}^t is selected and the cycle ends with a total cost $C_{tot}(t)$, where

$$C_{tot}(t) = C_{par}(t) + c_{n-1}^t \tag{22}$$

$$C_{par}(t) = \begin{cases} 0 & t < 1 \\ \sum_{k=0}^{t-1} c_n^k & t \geq 1 \end{cases} \tag{23}$$

On the other hand, if decision a_n^t is made, the cycle is continued towards j_n^t with a partial cost $C_{par}(t+1)$. Note that when $C_{par}(t+1) \geq C_{tot}(t)$ there is no point in further searching for a better path and the cycle should end. The minimum cost of the paths encountered by the decision maker since the beginning of the cycle is

$$C_{tot}^{min}(t) = \min_{0\leq k\leq t}\{C_{tot}(k)\} \tag{24}$$

In Fig. 5, we plot a diagram for a decision tree when the decision maker is at stage $t = 2$. $C_{tot}^{min}(2)$ is the minimum cost between all paths that the decision maker could have selected to end the cycle up to and including stage 2. $C_{tot}(3) = C_{par}(t+1) + \mathcal{E}$ is the cost associated with continuing for one more period and then stopping. Next, we prove that the one–stage policy dictated by set $\mathcal{B}_1$ is not globally optimal.

Theorem 1 *The forwarding policy dictated by set $\mathcal{B}_1$ is not globally optimal.*

Proof: Let P1 be the one–stage policy defined by set $\mathcal{B}_1$ and let t_1 be the generic stage at which set $\mathcal{B}_1$ is entered for the first time. Moreover, let P2 be a second policy that stops according to set $\mathcal{B}_1$ as long as $C_{tot}(t) = C_{tot}^{min}(t)$ and that stops with probability one as $C_{tot}(t) > C_{tot}^{min}(t)$. As P1 stops at time t_1, the following inequalities must hold: $c_{n-1}^0 - c_n^0 > \mathcal{E}$, $c_{n-1}^1 - c_n^1 > \mathcal{E}, \ldots, c_{n-1}^{t_1-1} - c_n^{t_1-1} > \mathcal{E}, c_{n-1}^{t_1} - c_n^{t_1} \leq \mathcal{E}$. Note

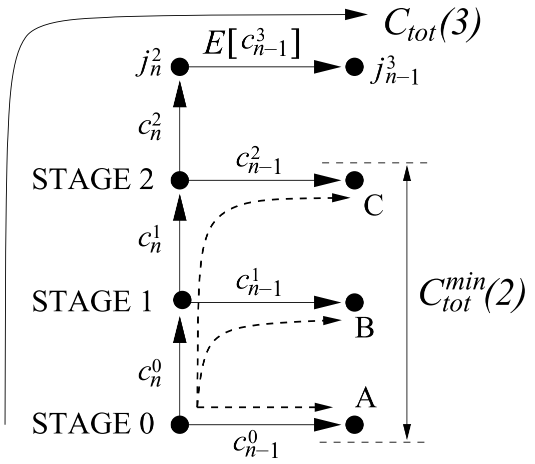

Fig. 5 Decision tree at stage $t = 2$. c^3_{n-1} is the unknown cost of the minimum cost node j^3_{n-1} with HC $n - 1$. A, B, C are all the possible paths encountered so far: $C^{min}_{tot}(2) = \min\{\text{Cost}(A), \text{Cost}(B), \text{Cost}(C)\}$

that for $t_1 \geq 1$ the previous constraints do not suffice to guarantee that $C_{tot}(t_2)$ is equal to the minimum cost $C^{min}_{tot}(t_2)$ for every stage $t_2 < t_1$. Hence, on average there is a non zero probability that P2 stops at $t_2 < t_1$ given that t_1 is the stopping stage for policy P1. Moreover, since this is true for every $t_1 \geq 1$ and the average cost of stopping at $t_2 < t_1$ is strictly lower than the average cost of stopping at t_1, the expected total cost of policy P2 is strictly lower than the expected total cost of policy P1. The theorem follows as a policy (P_2) exists with a strictly lower expected cost than policy P_1. □

To find the optimal policy, let us consider the following extended version of the forwarding problem. At the generic stage t, the decision maker must choose an action in set $\mathcal{A}_t = (a^t_{n-1}, a^t_n)$, as above. At every decision stage t, the decision maker knows the previously encountered costs $\{c^0_{n-1}, c^0_n, c^1_{n-1}, c^1_n, \ldots, c^t_{n-1}, c^t_n\}$ and can evaluate the minimum cost of all paths encountered so far $C^{min}_{tot}(t)$ (Eq. (24)). If we assign a cost $C^{min}_{tot}(t)$ to action a^t_{n-1}, the decision maker should stop as soon as $C^{min}_{tot}(t) = C^{min}_{tot}(t+1)$ (*stopping rule*), where $C^{min}_{tot}(t+1) = \min(C^{min}_{tot}(t), C_{tot}(t+1))$. This is equivalent to stopping when prolonging the cycle of one more stage ($t \to t+1$) does not lead to any advantage in terms of total cost. It is worth noting that this *stopping rule* aims at driving the forwarding process towards those paths for which $C^{min}_{tot}(\cdot)$ is strictly decreasing, i.e., to seek the lowest cost solution. From the discussion above, it follows that the *stopping rule* is verified iff $C^{min}_{tot}(t) \leq C_{tot}(t+1)$, ($C_{tot}(t+1) = C_{par}(t+1) + c^{t+1}_{n-1}$). The corresponding one–stage policy is dictated by the following stopping set

$$\mathcal{B}_2 = \left\{X_t : C^{min}_{tot}(t) - C_{par}(t+1) \leq \mathcal{E}\right\} \qquad (25)$$

```
$T \leftarrow \emptyset$; $C_{acc} \leftarrow 0$; $C^{min}_{tot} \leftarrow +\infty$;
repeat
    $i \leftarrow$ current node;
    $j_{n-1} = \text{argmin}_{z \in N_i(n-1)}\{c_z\}$;
    $j_n = \text{argmin}_{z \in N_i(n)}\{c_z\}$;
    $C^{min}_{tot} = \min\{C^{min}_{tot}, C_{acc} + c_{n-1}\}$;

    if $(C^{min}_{tot} - (C_{acc} + c_n) \leq \mathcal{E})$ or $(j_n \in T)$ then
        - select next hop $j \leftarrow j_{n-1}$;
        - $C_{acc} \leftarrow 0$; $C^{min}_{tot} \leftarrow +\infty$;
    else
        - select next hop $j \leftarrow j_n$;
        - $C_{acc} \leftarrow C_{acc} + c_n$;
    end

    Update tabu list $T$:
    begin
        - $T \leftarrow T \cup j$;
        - Delete nodes with age $\geq$ $tabulen$ from $T$;
    end
until $i$ = sink node;
```

Algorithm 1: Statistically–Assisted greedy Routing Algorithm (SARA). A tabu list T is used to prevent loops or ping-ponging between nodes at the same HC distance.

where, in this case $X_t = (c^0_{n-1}, c^0_n, c^1_{n-1}, c^1_n, \ldots, c^t_{n-1}, c^t_n)$, while $\mathcal{E}$, $C_{par}(t+1)$ and $C^{min}_{tot}(t)$ are defined in Eqs. (21), (23) and (24), respectively. The one–stage look–ahead policy dictated by set $\mathcal{B}_2$ states that it is convenient to stop at stage t whenever the expected cost of stopping at stage $t+1$ is greater than or equal to the minimum cost of all paths encountered through stages 0 to t. Next, we prove that the forwarding strategy dictated by set $\mathcal{B}_2$ is globally optimal.

Theorem 2 *The forwarding policy dictated by set $\mathcal{B}_2$ is globally optimal.*

Proof: The result follows by showing that $\mathcal{B}_2$ is a closed set of states (see [38], Theorem 2.2, p. 54, or [40], p. 164). In particular, $\mathcal{B}_2$ is closed if $X_t \in \mathcal{B}_2$ implies that $X_\tau \in \mathcal{B}_2$, $\forall\ \tau > t$. The set rule can be re-written as $C_{par}(t+1) + \mathcal{E} \geq C^{min}_{tot}(t)$. Now, assume that this rule is verified at the generic time t with X_t, then it is verified for $X_\tau, \forall\ \tau > t$ since $C^{min}_{tot}(t)$ and $C_{par}(t+1)$ are non increasing and non decreasing in t, respectively. □

The policy above is optimal in the sense that it leads to the minimum expected long term cost among all online policies exploiting first (c_n and c_{n-1}) and second order ($\mathcal{E}$) cost information. The Multi–Stage algorithm derived from set $\mathcal{B}_2$ is named SARA (Statistically–Assisted greedy Routing Algorithm); in Algorithm 1, we report its full version, where a tabu list T is used to avoid cycling and ping-ponging between nodes at the same HC distance. This algorithm is a straightforward implementation of Eq. (25) (set $\mathcal{B}_2$).

```
foreach $i \in N$ do $\tilde{V}_0(i) \leftarrow \gamma \mathrm{HC}(i)$;
$T \leftarrow \emptyset$;
(1) $i \leftarrow$ source node;
repeat
    (2) $\mathcal{N}_i(\mathcal{R}) = \{i \in \mathcal{N}_i$ s.t. $\mathcal{R}$ holds and $i \notin T\}$;
    (3) $j^\star = \mathrm{argmin}_{j \in \mathcal{N}_i(\mathcal{R})}\{\tilde{V}(j) + \beta\, c_{ij}\}$;
    (4) $\tilde{V}(i) = \min_{j \in \mathcal{N}_i(\mathcal{R})}\{\tilde{V}(j) + \beta\, c_{ij}\}$;
    (5) Break ties arbitrarily. $i \leftarrow j^\star$;
    (6) Update tabu list $T$:
    begin
        - $T \leftarrow T \cup j^\star$;
        - Delete nodes with age $\geq$ $tabulen$ from $T$;
    end
until $i =$ sink node;
```

Algorithm 2: LRTA–based routing algorithm.

5.5. LRTA–Based routing algorithm

In this section, we allow node costs to be partially propagated through the network. We assume that the forwarding process is at node i. Let $\mathcal{A}$ be the set of possible actions, where action $a_j = j$ corresponds to electing node $j \in \mathcal{N}_i$ as the relay node. Let $V(j)$ be the cost associated with the minimum cost path among all paths connecting node j to the sink. Assuming $V(j)$ to be known for every node $j \in \mathcal{N}_i$, the optimal action a_{opt} at node i is [38]

$$a_{opt} = \mathrm{argmin}_{j \in \mathcal{N}_i}\{c_{ij} + V(j)\} \tag{26}$$

Hence, once the $V(j)$s are specified for every node, the optimal cost path is derived applying Eq. (26). The problem to be solved is to obtain the $V(j)$s with low communication overhead. To this end, we consider the LRTA* [21,41] algorithm. Similar techniques were also used in GEAR [20]. Here, we extend the previous work by introducing new approaches to control and limit the cost propagation. LRTA* is a real–time search technique capable of learning and adapting its runtime path selections over unknown environments. This task is carried out only based on information gathered in the neighborhood of the problem solver's current stage, i.e., the look ahead search is limited to depth one as for greedy schemes. However, differently from greedy techniques, LRTA* allows for the propagation of cost estimates through the network. The goodness of such cost estimates is refined as the algorithm is repeated and is exploited at each iteration to drive the searching process. Next, we present an adapted version of LRTA* to our routing problem. At the beginning of the algorithm, each node $i \in N$ independently sets an initial heuristic estimate $\tilde{V}_0(i)$ of $V(i)$. As will be shown next, the convergence speed is highly dependent on the goodness of such an initial estimate. Let us refer to $t \geq 0$ as the t-th iteration of the algorithm, where 0 is the step just before the first iteration starts. At the beginning, a packet is generated by a source node s, that is the starting point of the algorithm (step (1) in Algorithm 2). Then, a subset ((2) in Algorithm 2) of nodes is derived from $\mathcal{N}_s$. This set is referred to as $\mathcal{N}_s(\mathcal{R})$, and is composed by all nodes verifying condition $\mathcal{R}$. Two conditions are introduced here: 1) DO (*Delay Optimized*): given the current node i and a node $j \in \mathcal{N}_i$, the condition $\mathcal{R}_{DO}$ is verified if $\mathrm{HC}(j) < \mathrm{HC}(i)$. The optimal solution, in this case, is the Pareto optimal path corresponding to the minimum cost path in the shortest path set. 2) CO (*Cost Optimized*): the condition $\mathcal{R}_{CO}$ is verified if $\mathrm{HC}(j) \leq \mathrm{HC}(i)$. The optimal solution, in this case, is the Pareto optimal path corresponding to the minimum length path in the minimum–cost path set. As above, only the nodes in $\mathcal{N}_i(n-1)$ and $\mathcal{N}_i(n)$ are considered for routing purposes. Neglecting the nodes in $\mathcal{N}_i(n+1)$ is very effective, since the state space considered in the searching phase is considerably reduced with a subsequent reduction of the algorithm convergence time. In addition to rule $\mathcal{R}$, a *tabu list* T, which contains the last $tabulen$ visited nodes is stored in the packet to be forwarded. This list is used in the CO algorithm to prevent loops or ping-ponging between nodes at the same HC distance. After building the set $\mathcal{N}_i(\mathcal{R})$, the algorithm selects the next hop $j^\star \in \mathcal{N}_i(\mathcal{R})$ as in line (3), i.e., by summing the estimated cost at node $j \in \mathcal{N}_i(\mathcal{R})$ ($\tilde{V}(j)$) to the cost associated with transition $i \to j$ (c_{ij}) and then choosing the node that minimizes this sum. The presence of the further factor β will soon be clarified. The statement in line (4), instead, corresponds to an iterative propagation of the minimum costs $\tilde{V}(i)$. Observe that, at the first step (time $t = 1$) of the algorithm $\tilde{V}(i)$ is only a rough estimate of the true cost. LRTA* enjoys the *convergence* property [21,41]. In more detail, the initial cost estimate $\tilde{V}_0(i)$ is said to be admissible if $\tilde{V}_0(i) \leq V(i)$, where $V(i)$ is the true cost estimate at node i. When the initial heuristic is admissible for all nodes in the network, the algorithm converges to the optimal solution for a sufficiently large but bounded step t [41]. For example, the trivial setting $V_0(i) = 0$, $\forall\, i \in N$, is admissible and the convergence is also guaranteed. However, in such a case the number of iterations to get to the optimal solution could be quite high. In [41] it has been proven that when the admissibility hypothesis does not hold, then it is sufficient to have $\tilde{V}_0(i) \leq V(i)(1+\varepsilon)$, $\varepsilon \geq 0$, $\forall\, i \in N$, to provide a nontrivial bound on the solution cost. In this case the initial heuristic estimates are said to be ε–admissible and the final path cost is guaranteed to be bounded by $(1+\varepsilon)V(i)$. In our algorithm, we derive the initial heuristic estimates as $\tilde{V}_0(i) = \gamma \mathrm{HC}(i)$. That is, every node independently computes its cost estimate by multiplying its hop distance by a predefined constant $\gamma \in [0, 1]$. As an example, in Fig. 6, the cost difference between CO and the optimal cost solution is plotted as a function of the algorithm step t for two different γ values. These results are for a network where node costs are uniformly distributed in $[0, 1]$ and nodes are uniformly distributed

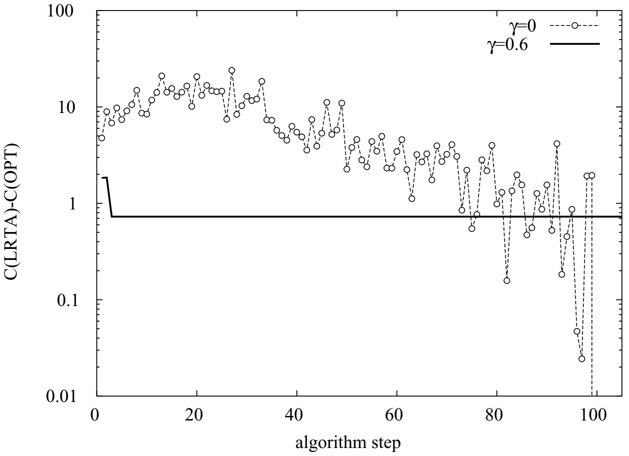

Fig. 6 Cost difference between the optimal solution (OPT) and the path found by LRTA*–CO. The plot was obtained for a source destination pair (s, d), where $\text{HC}(s) = 15$

with an average number of nodes within range equal to $\lambda_n = \lambda\pi R^2 = 15$, where λ is the node density (nodes/m^2). For what concerns signal attenuation and connectivity, we consider the unit disk [42] model, i.e., nodes i and j can communicate with each other iff $d(i, j) \leq R$, where $d(\cdot)$ is the Euclidean distance between the two nodes and R is the constant transmission range.[8] Clearly, a γ-weighted initial heuristic can considerably speed up the convergence time. In our example, as $\gamma = 0$ the convergence to the optimum is achieved around step 98, whereas a bounded but non-optimal solution is soon achieved for $\gamma = 0.6$. However, the good result obtained for $\gamma = 0.6$ is not always guaranteed as the final solution depends on the correctness of the initial guess in the selection of γ. Too high a value of γ could, in fact, prevent the search from properly exploring the state space therefore leading to suboptimal solutions. To cope with this problem, we consider a new strategy to propagate the costs. In lines (3) and (4) of Algorithm 2, we introduce a further factor $\beta \in [0, 1]$. This is a constant that is used here with the aim of limiting the searching phase. In particular, the contribution of the generic cost estimate $\tilde{V}$ in lines (3), (4), after $n \geq 1$ iterations of the algorithm, is given by $\beta^n \tilde{V}$, i.e., the costs propagated from n hops away are scaled by the factor β^n. This is somehow equivalent to the discount factor in dynamic programming [38]. In general, the rationale behind discounting is that, at the generic node i, it could be reasonable to choose the next hop by having a limited knowledge of the network status rather than having its complete view. This is the reason for introducing β, i.e., to fully consider the cost (c_{ij}) of the nodes placed one hop away and to progressively neglect the costs of those nodes that are placed far away. The solution is, in general, sub–optimal. More details are given in the next section.

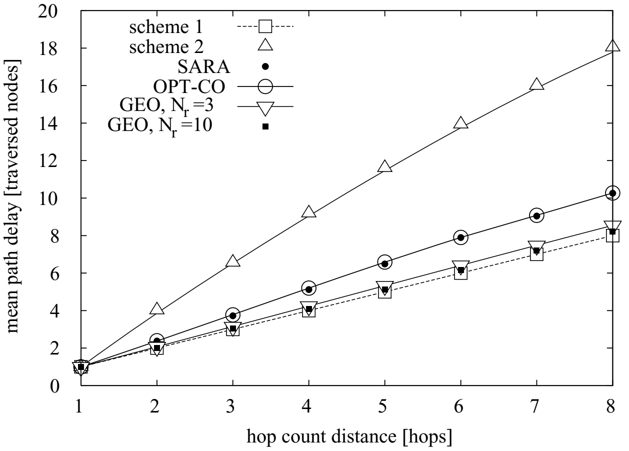

Fig. 7 Mean path delay: comparison between greedy routing schemes

6. Results

We consider a unit disk connectivity model, where the node transmission range is $R = 1$ m. Nodes are uniformly placed within a square area of 18×18 m^2 with the sink positioned in its center. The node density is λ and $\lambda_n = \lambda\pi R^2$ is the average number of nodes within coverage. In the sequel, we assume $\lambda_n = 15$. Moreover, we consider node costs to be uniformly distributed in [0, 1]. Also, every node has a perfect estimate of $\mathcal{E}$ (Eq. (21)). The performance reported next is therefore valid from a theoretical point of view and gives us the maximum achievable gains (ideal MAC and perfect cost estimates). Observe that, to implement the SARA scheme, we need to propagate cost information along with request and reply packets used within the channel contention. Estimates for $\mathcal{E}$ can therefore be obtained by overhearing signaling traffic and by appropriate low–pass filtering. This issue is however left for future research. In addition to the HC based schemes discussed above, we consider an idealized geographical routing algorithm where we subdivide the relaying area into a number N_r of priority regions, according to the related advancement towards the destination [4]. Moreover, the relay is always the lowest cost node within the non-empty region with the highest priority, i.e., the lowest cost node leading to the maximum advancement towards the destination. Results are obtained with a C written simulator through a Monte Carlo approach. First, the topology is created by assigning node positions and costs. Subsequently, a source node with a given HC distance is randomly picked and a path from the source to the sink is created for each routing scheme. The process is iterated until 95% confidence intervals are within $+/-$ 1% of the reported mean values. As an example, in Figs. 11 and 12, 95% confidence intervals are

[8] More sophisticated propagation models are also possible. However, since they do not represent the main issue of this paper, we leave their study for future research.

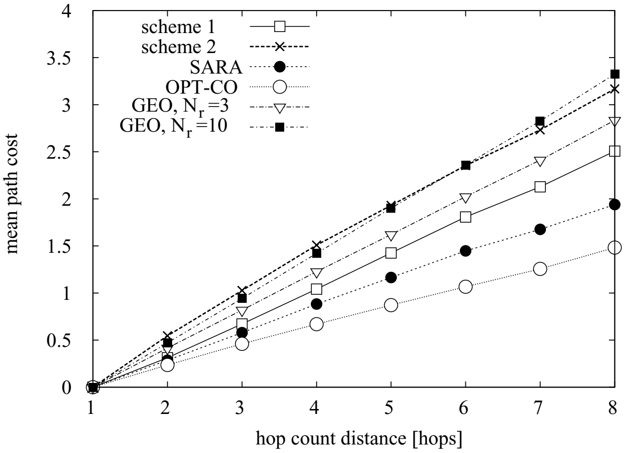

Fig. 8 Mean path cost: comparison between greedy routing schemes

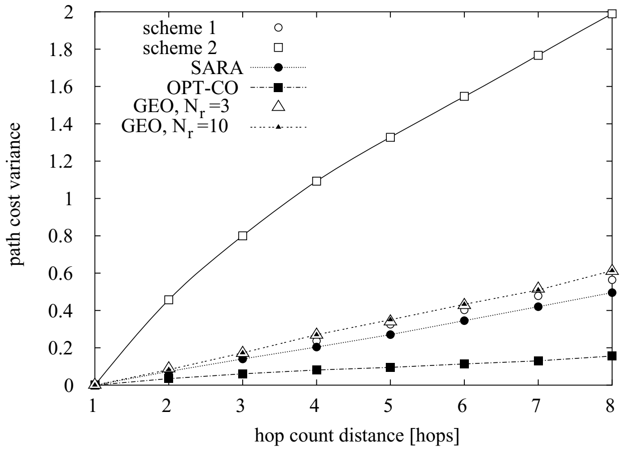

Fig. 9 Cost variance: comparison between greedy routing schemes

shown for the case $\beta = 1$, $\gamma = 0$. They are not plotted for every graph/curve for the sake of readability of the plots.

In Fig. 7, we report the mean path delay as a function of the hop distance between source node and sink. The delay is measured here by the number of traversed nodes to reach the sink, i.e., the number of packet exchanges (which is lower bounded by the hop count value of the source node). As expected, scheme 1 is the one leading to the shortest delay, while in scheme 2, the delay substantially increases with increasing HC distance. SARA achieves good results, staying quite close to the optimal performance of scheme 1. Note also that SARA and the non-dominated cost optimal scheme (OPT-CO, see Section 4) present a very similar behavior in terms of delay performance. Geographical forwarding (GEO) exhibits almost optimal delay performance for both $N_r = 3$ and $N_r = 10$ [4]. Mean path costs are plotted in Fig. 8, where the advantage offered by statistically–assisted techniques can be clearly observed. SARA is the on-line routing scheme performing best, by roughly halving the cost of scheme 1. In Fig. 9, we focus on cost variances. Scheme 2 is the one with the worst performance, whereas GEO, SARA and scheme 1 stay all reasonably close to the optimal solution (OPT-CO). In Fig. 10, we plot the complementary cumulative distribution of the exceeding cost with respect to the non-dominated minimum cost path (OPT-CO). This distribution is plotted for an initial HC distance between source and sink of 8 hops. The idealized geographical forwarding is the scheme with the poorest performance: in 10% of the cases its cost exceeds the optimal one (OPT-CO) by 180−250%, whereas SARA and the one–stage optimal policy exceed the optimal cost solution by 82%. It is also worth observing that SARA leads to substantial improvements with respect to both scheme 1 and 2. In particular, the exploitation of two–hop neighbor statistical measures ($\mathcal{E}$) leads to improvements up to one order of magnitude (see performance gap between scheme 1 and SARA). These results are due to the fact that in pure geographical routing the relay selection is driven by *the maximum geographical advancement towards the destination only*. That is, for a given set of awake nodes within range, the relay selection is carried out mainly considering geographical advancements towards the sink. Moreover, as discussed in Section 5.3, the set of nodes considered for forwarding in geographical routing is, in general, a subset of the nodes considered in HC forwarding. GEO performance could be improved considering a different technique to jointly weigh costs and distances. As a general observation for all schemes, we stress that actual routing performance strongly depends on the considered MAC. To this respect, MAC approaches such as [4], [22] can be used to get information on local costs on-the-fly. Also, SARA could be implemented in such a way that nodes $j \in \mathcal{N}_i(n)$ account for a cost $c_j + \mathcal{E}$, whereas node $j \in \mathcal{N}_i(n-1)$ consider c_j only. These costs may subsequently be used to select the access probabilities in the contention to be the relay [4,22]. This topic is the object

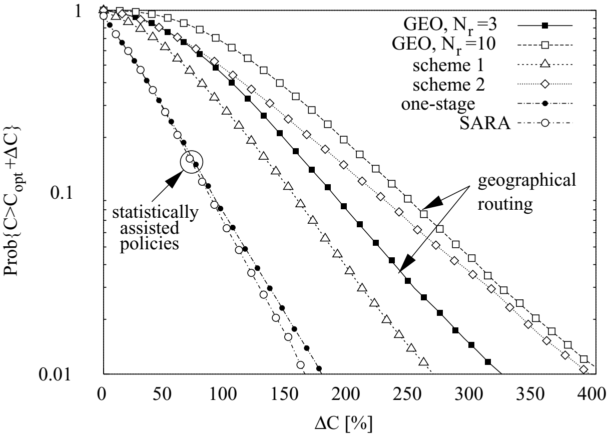

Fig. 10 Cumulative complementary distribution function (ccdf) of the cost difference between the proposed algorithms and the optimal cost non–dominated solution (OPT-CO), C_{opt})

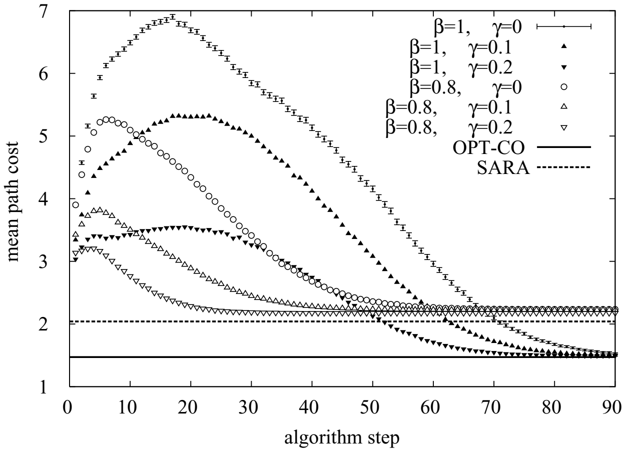

Fig. 11 LRTA–CO routing scheme: Average cost as a function of the algorithm step. The cost of the Pareto optimal lowest cost path (OPT-CO) and of the solution found by SARA are also plotted for comparison

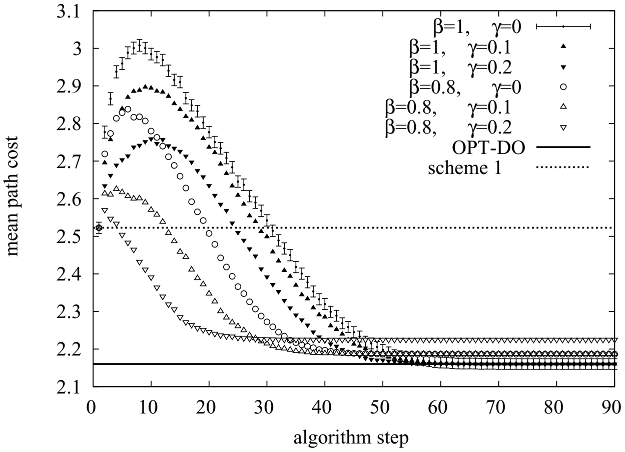

Fig. 12 LRTA–DO routing scheme: Average cost as a function of the algorithm step. The cost of the Pareto optimal shortest path (OPT-DO) and of the solution found by the shortest–path greedy scheme (scheme 1) are also plotted for comparison

of our current research. We also stress that, differently from the cost model considered in the paper, where node costs are independently drawn, real networks may show spatial correlation. In such a case, the cost statistics at future stages may depend on the current forwarding choice. However, as the aim of SARA is to go around high cost nodes, we expect it to still work satisfactorily, especially in the presence of highly correlated regions. Assume, for instance, that nodes in $\mathcal{N}_i(n-1)$ have high costs, then due to the correlation structure we may expect their descendants to have high costs as well and, in this case, the selection of a node in $\mathcal{N}_i(n)$ with smaller cost increases the probability of circumnavigating the high cost region. We leave these issues open for future research. In Figs. 11 and 12, we compare greedy routing with LRTA–based schemes. Both figures report the average path cost as a function of the algorithm complexity, i.e., of the number of iterations of the algorithm. Since cost estimates can be propagated back through piggy–backing, i.e., by inserting them into ACKs, the complexity corresponds to the number of packet transmissions that are needed for the scheme to converge. For comparison, in Fig. 11 we also plot the non-dominated optimal cost performance (OPT-CO) and the mean path cost found by SARA. As expected, an appropriate setting of γ and β can substantially decrease the algorithm convergence time. The setting $\gamma = 0$, $\beta = 1$ is of course unusable since in this case the algorithm starts with no knowledge about the network state. Hence, the exploratory phase is more risky and characterized by high initial costs. However, after a large number of iterations (≈ 90), $\beta = 1$ leads very close to optimal solutions. This was verified in all of our experiments and is an important finding as it tells us that the $\mathcal{R}_{CO}$ strategy is good enough to select good paths. This motivates our claim in Section 5.1 that the selection of nodes in the set $\mathcal{N}(n+1)$ can be avoided. Of course, there could be cases where the selection of nodes in $\mathcal{N}(n+1)$ is instead necessary, but these are rather unlikely. In Fig. 12 we focus on shortest paths only, where the label OPT-DO means non-dominated shortest path solution (see Section 4). In this case, the best that a greedy scheme can do is what scheme 1 does. In such a setting, LRTA-based routing seems to be more effective as its cost is well below the cost of greedy forwarding. Moreover, thanks to an appropriate selection of β and γ ($\beta = 0.8$, $\gamma = 0.2$)[9] LRTA routing is convenient over the greedy scheme only after 3 packet transmissions. The optimization process also converges to a low cost. This behavior is due to the further reduction of the searching state space enabled by condition $\mathcal{R}_{DO}$. We observe that the good results of LRTA algorithms were obtained for static environments, where node costs remain unchanged during the algorithm execution. As node costs vary during the search phase the solution of LRTA schemes highly degrades leading to worse performance than with online schemes.

7. Conclusions

In this paper we discussed online routing strategies over hop count coordinates for wireless sensor networks. In the first part of the paper we focused on greedy schemes which make routing decisions based on exact cost information for in range devices and on cost estimates for out–of–sight nodes. In this respect, we proposed SARA, a family of routing algorithms which route packets exploiting an online optimal policy. Further, we investigated LRTA based routing schemes, where node costs are partially propagated through the network to

[9] This are the values that gave us a good trade–off between cost and delay in all our measurements.

find quasi optimal paths. These routing techniques are then compared with each other and with (off-line) globally optimal solutions. The results of SARA are very good, as we show that the exploitation of the expected information related to nodes two hops away can effectively reduce path costs with respect to pure greedy forwarding (information on nodes within range only) and lead very close to optimal solutions. LRTA-based schemes also provide good solutions in terms of cost. They are however affected by possibly long convergence times and their applicability is therefore limited to static networks.

Acknowledgments This material is based upon work partially supported by the National Science Foundation under Award Number 0331690 and by the European Commission under contracts IST-004400 (Embedded WiSeNts) and IST-4-027227-IP (e-SENSE).

References

1. B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *ACM MobiCom*, Boston, Massachusetts, US (Aug. 2000) pp. 243–254.
2. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. on Networking*, Vol. 1 (Feb. 2003) pp. 2–16.
3. A. Lindgren, A. Doria and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, No. 3 (July 2003) pp. 19–20.
4. M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Trans. on Mobile Computing*, Vol. 2 (Oct.-Dec. 2003) pp. 337–348.
5. C. Schurgers, V. Tsiatsis, S. Ganeriwal and M.B. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Trans. on Mobile Computing*, Vol. 1, No. 1 (Jan.-Mar. 2002) pp. 70–80.
6. G. Xing, C. Lu, R. Pless and Q. Huang, "On greedy geographic routing algorithms in sensing-covered networks," in *ACM MobiHoc*, Roppongi Hills, Tokyo, Japan (May 2004) pp. 31–42.
7. F. Ye, G. Zhong, S. Lu and L. Zhang, "GRAdient broadcast: a robust data delivery protocol for large scale sensor networks," *ACM Wireless Networks*, Vol. 11, No. 3 (May 2005) pp. 285–298.
8. X. Chen and A.L. Murphy, "Enabling disconnected transitive communication in mobile ad hoc networks," in *ACM Workshop on Principles of Mobile Computing (POMC)*, Newport, Rhode Island, US (Aug. 2001).
9. T. Melodia, D. Pompili and I. Akyildiz, "On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks," *IEEE J. Select. Areas Commun.*, Vol. 23, No. 3 (Mar. 2005) pp. 520–532.
10. A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," *IEEE Trans. on Mobile Computing*, Vol. 3, No. 3 (Jul.-Aug. 2004) pp. 272–285.
11. M. Ilyas, *The Handbook of Ad Hoc Wireless Networks*. CRC Press (2002).
12. Wei Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. on Networking*, Vol. 12, No. 3 (June 2004) pp. 493–506.
13. D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinsky and H. Korth, Eds. Norwell, MA: Kluwer (1996) Ch. 5, pp. 153–181.
14. C.E. Perkins, E.M. Belding-Royer and S.R. Das, "Ad-hoc on demand distance vector routing," IETF Mobile Ad-hoc Networks (MANET) Working Group RFC. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt
15. J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM MobiCom*, Dallas, Texas, US (Oct. 1998) pp. 85–97.
16. A.S. Tanenbaum, *Computer Networks*. Prentice Hall (1989).
17. G. Pei, M. Gerla and T.-W.Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *IEEE ICC*, New Orleans, Louisiana, US (June) pp. 70–74.
18. R. Guérin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Trans. Networking*, Vol. 10, No. 5 (Oct. 2002) pp. 613–620.
19. P. Bose, P. Morin, I. Stojmenović and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *ACM Workshop for Distributed Algorithms and Methods for Mobile Computing and Communications (DialM)*, Seattle, Washington, US (Aug. 1999) pp. 48–55.
20. Y. Yu, R. Govindan and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," in *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023* (May 2001).
21. R.E. Korf, "Real–time heuristic search," *Artificial Intelligence*, Vol. 42 (1990) pp. 189–211.
22. M. Rossi and M. Zorzi, "Cost efficient localized geographical forwarding strategies for wireless sensor networks," in *TIWDC 2005*, Sorrento, Italy (July 2005) to be published in LNCS, Springer.
23. K. Seada, M. Zuniga, A. Helmy and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *ACM SenSys 2004*, Baltimore, Maryland, US (Nov. 2004) pp. 108–121.
24. S. Lee, B. Bhattacharjee and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *ACM MobiHoc 2005*, Urbana-Champaign, Illinois, US (May 2005) pp. 230–241.
25. L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *IEEE National Communication Conference*, Birminghan, Alabama (1978) pp. 4.3.1–4.3.5.
26. F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, "Geometric ad–hoc routing: of theory and practice," in *ACM Symposium on Principles of Distributed Computing (PODC)*, Zurich, Switzerland (2003) pp. 63–72.
27. I.I. Akyildiz, D. Pompili and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Elsevier Ad Hoc Networks*, Vol. 3 (Mar. 2005) pp. 257–279.
28. A. Beaufour, M. Leopold and P. Bonnet, "Smart-tag based data dissemination," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, Atlanta, Georgia, US (Sept. 2002) pp. 68–77.
29. H. Sabbineni and K. Chakrabarty, "Location-aided flooding: an energy-efficient data dissemination protocol for wireless sensor networks," *IEEE Trans. on Computers*, Vol. 54, No. 1 (Jan. 2005) pp. 36–46.
30. F. Ye, A. Chen, S. Lu and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *IEEE ICCCN*, Scottsdale, Arizona, US (Oct. 2001) pp. 304–309.
31. C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks," in *IEEE MILCOM*, McLean, Virginia, US (Oct. 2001) pp. 357–361.

32. K.-H. Han, Y.-B. Ko and J.-H. Kim, "A novel gradient approach for efficient data dissemination in wireless sensor networks," in *IEEE VTC*, Los Angeles, California, US (Sept. 2004) pp. 2979–2983.
33. S. Basagni, M. Elia and R. Ghosh, "ViBES: virtual backbone for energy saving in wireless sensor networks," in *IEEE MILCOM*, Monterey, California, US (Oct.-Nov. 2004) pp. 1240–1246.
34. X. Jinghao, B. Peric and B. Vojcic, "Energy-aware and link-adaptive routing metrics for ultra wideband sensor networks," in *IEEE Networking with Ultra Wide Band and Workshop on Ultra Wide Band for Sensor Networks*, Rome, Italy (July 2005) pp. 1–8.
35. V. Chankong and Y.Y. Haimes, *Multiobjective Decision Making: Theory and Metodology*. New York: Elsevier Science Publishing Co. (1983).
36. V. Pareto, *Cours d'Economie Politique*. Lausanne, Switzerland: F. Rouge & Cie., Vol. 1 (1896).
37. D.-Z.Z. Du and P.M. Pardalos, *Network Optimization Problems: Algorithms, Applications and Complexity*. World Scientific Pub Co Inc. (1993).
38. S.M. Ross, *Introduction to Stochastic Dynamic Programming*. Academic Press (1983).
39. J. Seaborn, *Hypergeometric Functions and Their Applications*. New York: Springer–Verlag (1991).
40. D.P. Bertsekas, *Dynamic Programming and Optimal Control: Volume II*. Athena Scientific (2001).
41. M. Shimbo and T. Ishida, "Controlling the learning process of real–time heuristic search," *Artificial Intelligence*, Vol. 146 (2003) pp. 1–41.
42. B.N. Clark, C.J. Colbourn, and D.S. Johnson, "Unit disk graphs," *Discrete Mathematics*, Vol. 86, No. 1-3 (Aug. 1991) pp. 165–177.

Michele Rossi was born in Ferrara, Italy on October 30th, 1974. He received the Laurea degree in Electrical Engineering (with honors) and the Ph.D. degree in Information Engineering from the University of Ferrara in 2000 and 2004, respectively. Since 2000 he has been a Research Fellow at the Department of Engineering of the University of Ferrara. During 2003 he was on leave at the Center for Wireless Communications (CWC) at the University of California San Diego (UCSD), where he did research on wireless sensor networks. In November 2005 he joined the Department of Information Engineering of the University of Padova, Italy, where he is currently an Assistant Professor. Michele Rossi is currently part of the EU funded Ambient Networks and eSENSE projects. His research interests include: TCP/IP protocols over wireless networks, performance analysis of link layer retransmission techniques, routing and access selection in heterogeneous wireless networks and MAC/routing algorithms for wireless sensor networks.

Michele Zorzi was born in Venice, Italy, in 1966. He received the Laurea degree and the Ph.D. degree in Electrical Engineering from the University of Padova, Italy, in 1990 and 1994, respectively. During the Academic Year 1992/93, he was on leave at the University of California, San Diego (UCSD), attending graduate courses and doing research on multiple access in mobile radio networks. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering of the University of Ferrara, Italy, and in 2003 joined the Department of Information Engineering of the University of Padova, Italy, where he is currently a Professor. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, and energy constrained communications protocols.

Dr. Zorzi from 2003 to 2005 was the Editor-In-Chief of the *IEEE Wireless Communications Magazine*, and currently serves on the Editorial Boards of the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Mobile Computing*, 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) and the *IEEE Journal on Selected Areas in Communications* (Multi-media Network Radios).

Ramesh R. Rao was born in Sindri, India, where he completed his undergraduate work at the Regional Engineering College of the University of Madras in Tiruchirapalli, obtaining a BE (Honors) degree in Electronics and Communications in 1980. He completed his graduate work at the University of Maryland, College Park, Maryland where he received his M.S. and Ph.D. Professor Rao is currently a Professor at the University of California, San Diego (UCSD) at the department of Electrical and Computer Engineering in the Irwin and Joan Jacobs School of Engineering, where he has been a member of the faculty since 1984. Professor Rao is the former director of UCSD's Center for Wireless Communications (CWC), and currently serves as the Qualcomm Endowed Chair in Telecommunications and Information Technologies, and as the Director of the San Diego Division of the California Institute of Telecommunications and Information Technology [Cal-(IT)^2]. As Director of the San Diego Division of Cal-(IT)^2, he leads several interdisciplinary and collaborative projects. His research interests include architectures, protocols and performance analysis of computer and communication networks, and he has published extensively on these topics. Since 1984, Professor Rao has authored over 100 technical papers, contributed book chapters, conducted a number of short courses and delivered invited talks and plenary lectures. He is currently supervising both masters and doctoral students.