

Optimal Local Topology Knowledge for Energy Efficient Geographical Routing in Sensor Networks

Tommaso Melodia, Dario Pompili, Ian F. Akyildiz

Broadband and Wireless Networking Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
e-mail: {tommaso,dario,ian}@ece.gatech.edu

Abstract—Since sensor networks can be composed of a very large number of nodes, the developed protocols for these networks must be scalable. Moreover, these protocols must be designed to prolong the battery lifetime of the nodes. Typical existing routing techniques for ad hoc networks are known not to scale well. On the other hand, the so-called geographical routing algorithms are known to be scalable but their energy efficiency has never been extensively and comparatively studied. For this reason, a novel analytical framework is introduced. In a geographical routing algorithm, the packets are forwarded by a node to its neighbor based on their respective positions. The proposed framework allows to analyze the relationship between the energy efficiency of the routing tasks and the extension of the range of the topology knowledge for each node. The leading forwarding rules for geographical routing are compared in this framework, and the energy efficiency of each of them is studied. Moreover *Partial Topology Knowledge Forwarding*, a new forwarding scheme, is introduced. A wider topology knowledge can improve the energy efficiency of the routing tasks but can increase the cost of topology information due to signaling packets that each node must transmit and receive to acquire this information, especially in networks with high mobility. The problem of determining the optimal Knowledge Range for each node to make energy efficient geographical routing decisions is tackled by Integer Linear Programming. It is demonstrated that the problem is intrinsically localized, i.e., a limited knowledge of the topology is sufficient to take energy efficient forwarding decisions, and that the proposed forwarding scheme outperforms the others in typical application scenarios. For online solution of the problem, a probe-based distributed protocol which allows each node to efficiently select its topology knowledge, is introduced and shown to converge to a near-optimal solution very fast.

Index Terms—Wireless Sensor Networks, Mathematical programming/Optimization, Position Based routing, Topology Control.

I. INTRODUCTION

Recent advances in wireless communications and electronics are paving the way for the deployment of low-cost, low-power networks of untethered and unattended sensors and actuators. Sensor networks [1] differ from “traditional” ad hoc networks in many aspects. The number of nodes in a sensor network can be several orders of magnitude higher than in ad hoc networks, and the deployment of nodes is usually denser. Moreover, sensor nodes are limited in power, computational capacities and memory, and they may not have *global identification* (ID) because of the very large number of

nodes and the according overhead.

Because of the above constraints, sensor networks protocols and algorithms must possess self-organizing capabilities, i.e. sensors must be able to cooperate in order to organize and perform networking tasks in an efficient way. The primary design constraints of these algorithms are: *energy efficiency, scalability and localization*.

It has been pointed out in [2] that the improved energy efficiency can be obtained by designing protocols and algorithms with a *cross-layer approach*, i.e., by taking into account interactions among different layers of the communication process so that the overall energy expenditure can be minimized. In this paper we consider dependencies between physical and network layers with the objective to perform energy efficient routing tasks.

All networking tasks, such as routing, should perform well for wireless networks with an arbitrary number of nodes. A *scalable* algorithm performs well in a large network. The notion of scalability for an algorithm is strictly related to that of *localization*: in a scalable algorithm each node exchanges information only with its neighbors (*localized* information exchange) in a very large wireless network [3]. In a localized routing algorithm, each node decides on the next hop based only on the position of itself, of its neighbors, and of the destination node. As a result, the local node behavior tries to achieve global network objectives such as minimum latency, minimum energy consumption, etc. On the other hand, in a non-localized routing algorithm a node maintains an accurate description of the overall network topology to compute the next hop, so that a global objective can be maximized. The routing problem becomes then equal to the shortest path problem if the *hop count* is used as the global performance metric or the shortest weighted path if *power* [4] or *cost* [5][6] link metrics are used.

It has been shown in [7][8] that the routing protocols which do not use geographical location information are not scalable, e.g. AODV (Ad hoc on-demand Distance Vector), DSDV (Destination Sequenced Distance Vector) or DSR (Dynamic Source Routing). On the other hand, the recent availability of small, inexpensive and low-power GPS (*Global Positioning System*) receivers, together with techniques which can deduce relative sensor coordinates from signal strengths [9] encourage people to deploy *Geographical Routing* [16] (also *Position*

Based Routing) algorithms which are becoming most promising scalable solutions for critically power-constrained sensor networks.

For these reasons this paper deals with the interactions between topology control [10] and energy efficient geographical routing. The question we try to answer is “How extensive should be the Local Knowledge of the global topology in each sensor node, so that an energy efficient geographical routing can be guaranteed?”. This question is clearly related to the degree of localization of the routing scheme. If each sensor node could have the complete knowledge of the topology, it could then compute the “global” optimal next hop which minimizes the energy expenditure. However, the process of acquiring complete topology information has a cost, i.e., energy spent to exchange the signaling traffic.

We develop an analytical framework to capture the trade-off between what we call the *topology information cost*, which increases with the Knowledge Range of each node, and the *communication cost*, which decreases when the knowledge becomes more complete. We apply this analytical framework to different position based forwarding schemes [11][12][13][14][15] and demonstrate by using Monte Carlo simulations that a limited knowledge is sufficient to make energy efficient routing decisions.

With respect to existing literature on geographical routing, we try to better define the terms “localized” and “neighbor”. A “neighbor” for a certain sensor node is another node which falls into its *topology Knowledge Range*, denoted as KR in what follows.

Our main contributions are:

- 1) We introduce a novel *analytical framework* to evaluate the energy expenditure of geographical routing algorithms [16] for sensor networks;
- 2) We give an Integer Linear Programming (ILP) formulation of the *topology Knowledge Range* optimization problem;
- 3) We provide a detailed comparison of the leading existing forwarding schemes [11][12][13][14][15] and introduce a new scheme called *Partial Topology Knowledge Forwarding* (PTKF);
- 4) For the on-line solution of the problem we introduce the *PRobe-bAsed Distributed protocol for knowledge rANge adjustment* (PRADA), which allows the network nodes to select near-optimal Knowledge Ranges in a distributed way.

The remainder of the paper is organized as follows. In Section II we review the forwarding schemes [11][12][13][14][15] for geographical routing and other related work. In Section III we state the problem and in Section IV we provide a mathematical formulation of the optimization problem. In Section V we introduce the distributed protocol for Knowledge Range adjustment and in Section VI we show numerical results obtained using the above analytical framework. Finally, in Section VII we conclude the paper.

II. RELATED WORK

First we describe the existing position based forwarding rules which will be utilized in the remainder of the paper.

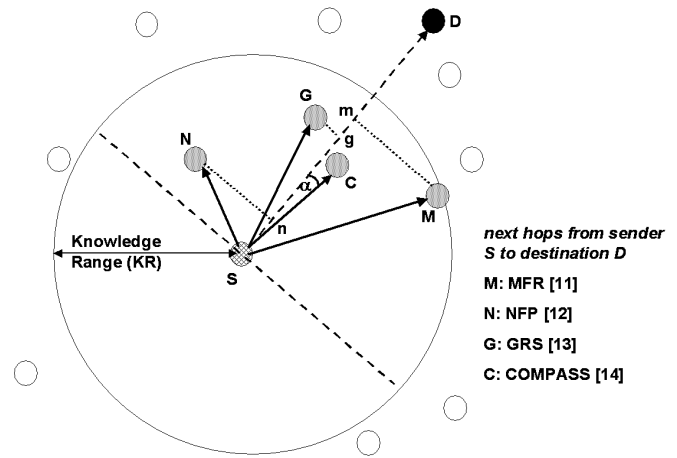


Fig. 1. Different Forwarding Schemes

A. Forwarding Rules

In a localized routing scheme, the node S (Fig. 1) which currently holds a message, knows only the position of its neighbors, i.e., the nodes within its Knowledge Range, and the destination node D .

Definition 1: Given a sender node S , and a destination node D , the *progress* of a generic node X , neighbor of S , is defined as the orthogonal projection of the line connecting S and X onto the line connecting S and D .

Takagi and Kleinrock [11] proposed the first geographical routing scheme based on the notion of progress. In their *Most Forward within Radius* (MFR) scheme [11], the packet is forwarded to the neighbor whose progress is maximum, e.g., the node M , whose progress is \overline{Sm} , in Fig. 1. Note that though node G is closer to the destination, its progress \overline{Sg} is smaller than \overline{Sm} . Hou and Li [12] discuss the *Nearest Forward Progress* (NFP) method which selects the neighbor with the minimum progress within the Knowledge Range of S , e.g., the node N in Fig. 1, whose progress is \overline{Sn} .

Finn [13] proposes the *Greedy Routing Scheme* (GRS), based on the geographical distance where the node S selects from its neighbors the closest one to its destination, e.g., G in Fig. 1. In the so-called *Compass Routing* method [14], the message is forwarded to a neighbor, e.g., C in Fig. 1, such that the direction SC is the closest to the direction SD , i.e., the angle $\angle CSD$ is minimum. In the so-called *Random Progress Forwarding* (RPF) method [15] a random next hop is selected among the nodes within the Knowledge Range. Let us now introduce the following

Definition 2: Given a sender node S and a destination node D , the *advance* of a generic neighboring node X is defined as the distance between S and D minus the distance between X and D .

A sufficient condition for a geographical routing scheme to be loop free is that only next hop nodes with positive *advance* can be selected. According to Definition 2, a generic neighbor has a positive advance with respect to a sender node if it is closer than the sender to the destination. When a routing scheme selects next hop nodes only if they have positive advance, then

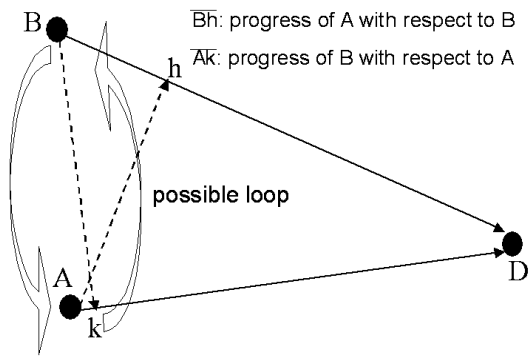


Fig. 2. Counterexample on the notion of progress

the overall path is guaranteed to be loop free. On the other hand, a positive progress for each next hop is not a sufficient condition for a routing scheme to be loop free, as can be inferred from the counterexample in Fig. 2, where three nodes, A , B and a destination node D are shown. A is a possible next hop for B and vice versa, since both nodes A and B have positive progress with respect to each other ($\overline{Ak} > 0$, $\overline{Bh} > 0$). However, this does not avoid loops. Both nodes could choose the other as next hop, thus generating a loop.

Conversely, loops are avoided when the positive advance criterion is used as a necessary condition for a node to be the next hop. Referring again to the example in Fig. 2, when a positive advance is a necessary condition for a node to be next hop, A is feasible next hop for B , but not vice versa, since A is closer than B to the destination ($\overline{AD} < \overline{BD}$). Since positive advance is a stronger condition, and guarantees loop free paths, we assume a positive advance as a necessary condition for a node to be the next hop in what follows. In other words, a node must choose the next hop among the nodes within its Knowledge Range *and* with positive advance with respect to the destination node, for all the considered forwarding schemes.

B. Other Related Work

Here we review related work on geographical routing, which constitutes the background of our work.

An excellent survey on position based routing techniques for ad hoc networks is given in [16], [17]. The methods to determine absolute and relative coordinates for network nodes, i.e., on location update techniques are reviewed in [18].

Most of the prior research assumes that nodes can work either in *greedy mode* or in *recovery mode*. In the greedy mode, the node that currently holds the message can forward it towards the destination. The recovery mode is entered when a node fails to forward a message in the greedy mode, since none of its neighbors is a feasible next hop. Usually this occurs because the node observes a void region between itself and the destination. For example the *Greedy Perimeter Stateless Routing* (GPSR), introduced in [19], makes *greedy* forwarding decisions (as *GRS* in Section II.A). When a packet reaches a concave node, the GPSR tries to recover by routing around the perimeter of the void region. Recovery mechanisms, which

allow a packet to be forwarded to the destination when a concave node is reached, are out of the scope of our paper. Here we assume that the packet is directly forwarded to the destination whenever such a node is reached.

The so-called Trajectory Based Forwarding (TBF) is proposed in [20] where the packet is forwarded along a predefined parametric curve encoded in the packet at the source. Several localized algorithms for power, cost and power-cost efficient routing are proposed and their efficiency is analyzed in [21]. Scalability properties of different ad hoc routing techniques, such as flat, hierarchical and geographical routing are discussed in [22]. A topology control algorithm called GAF given in [23] identifies, based on position information, nodes that are equivalent from a routing perspective and adaptively turns unnecessary nodes off in order to maintain a constant level of performance.

A taxonomy of location systems is given in [9] for ubiquitous computing applications including location sensing techniques and properties as well as a survey of commercially available location systems. In [24] it is shown how to derive position information for all nodes using *Angle of Arrival* (AOA) capabilities, when only a fraction of the nodes have positioning capabilities. Finally a distributed location service (GLS) is described in [7], where a node sends its position updates to its location servers without knowing their actual identities. This information is then used by the other nodes in the network to perform geographical routing operations.

III. PROBLEM SETUP

First we describe the Neighborhood Discovery Protocol which allows each node to gather information about its neighborhood. We then introduce the network model and define some notions. The network model is followed by the energy efficiency model. Finally we develop a new forwarding scheme called *Partial Topology Knowledge Forwarding* (PTKF).

Let us consider the following *Neighborhood Discovery Protocol*. Node S in Fig. 3 periodically sends a *Neighborhood Discovery* packet, called *ND-packet*, to gather information about its neighbor nodes, at a power level that allows the packet to be received by all nodes within its chosen *Knowledge Range* (KR in Fig. 3).

As a result, nodes N_1 , N_2 and N_3 receive the ND-packet while other nodes do not. Then, the nodes which received the ND-packet reply with a *Location Update* packet, called *LU-packet*. This contains the geographical position of the node. Now the question we are trying to answer is what should the Knowledge Range (KR) of each node be in the network so that the energy required by the network to perform the routing tasks is minimized. It is intuitive that increasing the KR may result in more efficient routing decisions. However, this comes with the penalty that more energy is needed to exchange signaling traffic.

A. Network Model

The network of sensor nodes is represented as $(\mathcal{V}, \mathcal{D})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ is a finite set of nodes in a finite-dimension terrain, with $N = |\mathcal{V}|$, and \mathcal{D} is the matrix whose

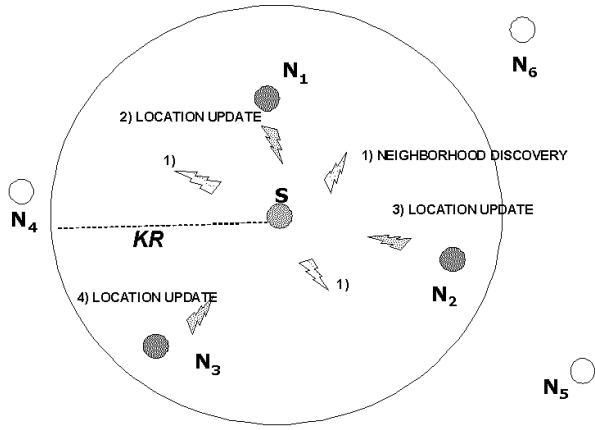


Fig. 3. Neighborhood Discovery Protocol

element (i, j) contains the value of the distance between nodes v_i and v_j . We associate each node k with its Knowledge Range, r_k , based on the Neighborhood Discovery protocol as explained above. Thus, the array $\underline{\mathbf{R}} = [r_1, r_2, \dots, r_N]$ describes the KRs of all nodes in the network. Let \mathcal{S} be the set of traffic sources and \mathcal{D} the set of destination nodes. We define $\mathcal{P} = \{(s, d) : s \in \mathcal{S}, d \in \mathcal{D}\}$ as the set of source-destination connections. The information rate of each connection is described by the traffic matrix $P = [p_{ij}]$, where p_{sd} represents the average information rate (bits/s) between a source node $s \in \mathcal{S}$ and a destination node $d \in \mathcal{D}$.

Definition 3: A loop-free Forwarding Rule \mathcal{F} , given a node v_i , its KR r_k and a destination node v_d , associates the node v_i with another node v_n in $\mathcal{V} \setminus \{v_i\}$, in such a way that the path $\{v_i, v_n, \dots, v_d\}$ obtained by recursively applying the rule is composed of distinct nodes.

Definition 4: v_n is called *next hop* of node v_i towards v_d with KR r_i , according to \mathcal{F} , which we indicate with $v_n = l_{v_i}^{\mathcal{F}}(v_d, r_i)$.

Note that for the sake of simplicity we will refer to a generic node v_n as n in what follows. We will also omit the index \mathcal{F} . Thus, $l_{v_i}^{\mathcal{F}}(v_d, r_i)$ is referred to as $l_i(d, r_i)$.

Given the set of KRs of all nodes $\underline{\mathbf{R}}$, the rule \mathcal{F} induces *paths* among any possible source-destination pair in the network. Thus,

$$\mathcal{F} : \underline{\mathbf{R}} \rightarrow \mathbf{x}_{ij}^{sd}(\underline{\mathbf{R}}) \quad (1)$$

where $x_{ij}^{sd}(\underline{\mathbf{R}}) = 1$ iff the link between node i and node j is part of the path between node s and node d with the given choice $\underline{\mathbf{R}}$ of ranges, when we apply the forwarding rule \mathcal{F} .

B. Energy Model

An accurate model for energy consumption per bit at the physical layer is

$$E = E_{elec}^{trans} + \beta d^\alpha + E_{elec}^{rec} \quad (2)$$

where

E_{elec}^{trans} is the energy utilized by transmitter electronics (PLLs, VCOs, bias currents, etc) and digital processing. This energy is independent of distance;

E_{elec}^{rec} is the energy utilized by receiver electronics, and

βd^α accounts for the radiated power necessary to transmit over a distance d between source and destination.

As in [28], we assume that

$$E_{elec}^{trans} = E_{elec}^{rec} = E_{elec} \quad (3)$$

Thus the overall expression for E in eq. 2, which we refer to as *link metric* hereafter, simplifies to

$$E = 2 \cdot E_{elec} + \beta d^\alpha \quad (4)$$

According to this link metric, the *topology information cost* for node v_i is expressed as:

$$C_i^{INF}(r_i) = [L_D \cdot \beta r_i^\alpha + (N_i(r_i) + 1) \cdot L_D \cdot E_{elec} + \sum_{m \in \zeta_i(r_i)} L_U \cdot \beta d_{mi}^\alpha + 2N_i(r_i) \cdot L_U \cdot E_{elec}] \cdot \frac{1}{T_M} \quad (5)$$

with

α is the path loss ($2 \leq \alpha \leq 5$);

β is a constant [*Joule/(bits · m^α)*];

L_D is the length of neighborhood discovery packets [*bits*];

L_U is the length of location update packets [*bits*];

E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit [*Joule/bits*];

$N_i(r_i)$ is the number of neighbors of node v_i when its Knowledge Range is r_i ;

$\zeta_i(r_i)$ is the set containing the indices of the nodes in range r_i of node i ;

T_M is the period between two consecutive neighborhood discovery messages [*sec*];

The expression βr_i^α represents the energy needed to transmit one bit at distance r_i ; thus $L_D \cdot E_{elec} + L_D \cdot \beta r_i^\alpha$ is the energy needed for node i to transmit the ND-packet in its Knowledge Range, where as each of the $N_i(r_i)$ nodes in its KR “spends” only $L_D \cdot E_{elec}$ to receive the ND-packet. By adding these two components we obtain the first line of eq. 5. Then, each of the $N_i(r_i)$ nodes transmits an LU-packet. The energy expenditure has again a constant factor, $L_U \cdot E_{elec}$, plus a factor, $L_U \cdot \beta d_{mi}^\alpha$, which depends on the distance between the transmitting node v_m and node v_i . Moreover, v_i spends $L_U \cdot E_{elec}$ to receive each of the $N_i(r_i)$ LU-packets. By adding all these components, and dividing by T_M , which depends on the mobility rate of the nodes in the network, we obtain the final expression for C_i^{INF} . In other words, C_i^{INF} is the *average energy* (measured in *watts*) which is needed to allow node v_i to obtain topology information within the range r_i .

The *communication cost* for node v_i can be computed from:

$$C_i^{COM}(\underline{\mathbf{R}}) = \sum_{(s,d) \in \Pi_i(\underline{\mathbf{R}})} [\beta d_{ii}^\alpha(d, r_i) + 2E_{elec}] \cdot p_{sd} \quad (6)$$

with

$$\Pi_i(\underline{\mathbf{R}}) = \{(s, d) \text{ s.t. } x_{ij}^{sd} = 1 \text{ for at least one } j\} \quad (7)$$

The set $\Pi_i(\underline{\mathbf{R}})$ contains all source-destination pairs whose path includes v_i as a transit node, as well as those for which v_i is the source. Thus, in eq. 6 we sum over all the connections in which v_i is a transmitting node. Note that each term has a distance-independent component $2 \cdot E_{elec}$ (the energy needed to transmit

and receive one bit), and a distance dependent component, $d_{i l_i(d, r_i)}^\alpha$, which represents the α -th power of the distance between node v_i and $v_{l_i(d, r_i)}$, its next hop towards v_d when its KR is r_i . Every term is then multiplied by the average bit rate of the communication p_{sd} . Thus, $C_i^{COM}(\mathbf{R})$ is measured in *watts* and represents the *average energy* expenditure for all the communications node v_i is involved in. We can now state the *total cost for node v_i* as:

$$C_i^{TOT}(\mathbf{R}) = C_i^{COM}(\mathbf{R}) + C_i^{INF}(r_i), \forall i. \quad (8)$$

Note that while the information cost of each node only depends on its own KR, the communication cost depends on the KR's of all nodes involved in the communication process.

C. Partial Topology Knowledge Forwarding (PTKF)

Here we describe a novel forwarding scheme called Partial Topology Knowledge Forwarding (PTKF). This is essentially a *shortest weighted path* routing scheme with a *power* link metric. Consider a node S which must forward a message to a given destination D . Given its KR, S knows the position of all nodes inside this range and the position of the destination node. The topological view of S is constituted by node D and by all the nodes in the KR with positive advance with respect to D , so that the loop freedom condition holds. To evaluate the next hop towards the destination node, a link metric of $2 \cdot E_{elec} + \beta d_{ij}^\alpha$, according to eq. 4, is assumed to be the cost of the link between each node pair v_i and v_j . A shortest weighted path algorithm (such as Bellman-Ford's) is executed to calculate the path towards the destination. The message is forwarded to the first node N in this shortest path. The node N calculates, in its turn, the optimal path towards the destination D , but this time according to its own KR. This can actually result in a very different path being chosen by N compared to the path calculated by S . It is easy to see the existing trade-off between the communication cost and the information cost for this scheme. Note that, unlike the forwarding schemes described in Section II.A, this is not a greedy scheme. This scheme is more localized the smaller the KR of each node becomes. However, we will demonstrate by using realistic models that "small" KR's are chosen when energy efficiency is the major concern.

IV. INTEGER LINEAR PROGRAMMING FORMULATION

Our objective is to select the vector of Knowledge Ranges (KR) \mathbf{R} which minimizes the energy expenditure of the overall network, given the set of connections \mathcal{P} and a Forwarding Rule \mathcal{F} :

$$\mathbf{P} : \min_{\mathbf{R}} C^{TOT} = \sum_{i \in \mathcal{V}} (C_i^{COM} + C_i^{INF}) \quad (9)$$

Here we give an *Integer Linear Programming* (ILP) formulation of the problem.

To linearize an inherently non-linear problem we consider discrete values of the Knowledge Ranges. The granularity of this quantization can be whatever, but obviously finer-grained transmission ranges increase the complexity of the problem. Each variable r_i , $0 \leq r_i \leq r^{max}$ assumes one out of the k_{max} discrete, equidistant values in the set $\{r^0, r^1, \dots, r^{k_{max}-1}\}$,

with $r^k - r^{k-1} = \Delta_r$, $\forall k$ s.t. $1 \leq k \leq k_{max} - 1$, with $r^0 = 0$ and $r^{max} = r^{k_{max}-1}$. We refer to the set of indices $\{0, 1, \dots, k_{max} - 1\}$ as \mathcal{R} .

We introduce the following notations and variables:

$r(k)$ is the k -th Knowledge Range;

$r^\alpha(k)$ is the α -th power of the k -th KR;

$N_i(k)$ is the number of neighbors for node v_i when it selects the k -th KR

$f_{dk}^{ij} = 1$ iff, according to \mathcal{F} , node v_j is next hop for node v_i , when v_d is destination, and the k -th Range is chosen;

$a_{ij}^k = 1$ iff node v_j is in the k -th KR of node v_i ;

d_{ij}^α is the α -th power of the distance between nodes v_i and v_j .

We introduce the following routing variables:

$x_{ij}^{sd} = 1$ iff link $i - j$ is part of the path between v_s and v_d .

The assignment variables are:

$y_i^k = 1$ iff node v_i uses k -th Knowledge Range. We refer to the variables y_i^k as *Knowledge Range indices*.

We can now express the problem as:

P: Optimal Topology Knowledge Ranges Problem:

Minimize:

$$C^{TOT} = \sum_{i \in \mathcal{V}} (C_i^{COM} + C_i^{INF}) \quad (10)$$

Subject to:

$$\sum_{k \in \mathcal{R}} y_i^k = 1, \forall i \quad (11)$$

$$\sum_{j \in \mathcal{V}} (x_{sj}^{sd} - x_{js}^{sd}) = 1, \forall s \in \mathcal{S}, \forall d \in \mathcal{D} \text{ s.t. } s \neq d; \quad (12)$$

$$\sum_{j \in \mathcal{V}} (x_{dj}^{sd} - x_{jd}^{sd}) = -1, \forall s \in \mathcal{S}, \forall d \in \mathcal{D} \text{ s.t. } s \neq d; \quad (13)$$

$$\sum_{j \in \mathcal{V}} (x_{ij}^{sd} - x_{ji}^{sd}) = 0, \forall s \in \mathcal{S}, \forall d \in \mathcal{D}, \forall i \in \mathcal{V} \quad (14)$$

s.t. $s \neq d, i \neq s, i \neq d;$

$$x_{ij}^{sd} \leq \sum_{k \in \mathcal{R}} (y_i^k \cdot f_{dk}^{ij}), \forall s \in \mathcal{S}, \forall d \in \mathcal{D}, \forall i, j \in \mathcal{V}; \quad (15)$$

$$x_{sj}^{sd} = \sum_{k \in \mathcal{R}} (y_s^k \cdot f_{dk}^{sj}), \forall s \in \mathcal{S}, \forall d \in \mathcal{D}, \forall j \in \mathcal{V} \text{ s.t. } s \neq d \quad (16)$$

$$C_i^{INF} = (L_N \cdot \beta \cdot \sum_{k \in \mathcal{R}} (y_i^k \cdot r^\alpha(k)) + (\sum_{k \in \mathcal{R}} (y_i^k \cdot N_i(k)) + 1) \cdot L_N \cdot E_{elec} + \sum_{m \in \mathcal{V}} (L_U \cdot \beta \cdot d_{mi}^\alpha + 2 \cdot L_U \cdot E_{elec}) \cdot \sum_{k \in \mathcal{R}} (y_i^k \cdot a_{im}(k))) \cdot \frac{1}{T_M}, \forall i \in \mathcal{V}. \quad (17)$$

$$C_i^{COM} = \sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} \sum_{j \in \mathcal{V}} (x_{ij}^{sd} \cdot p^{sd} \cdot (2 \cdot E_{elec} + \beta \cdot d_{ij}^\alpha)), \forall i \in \mathcal{V}. \quad (18)$$

The constraint (11) imposes the existence of a single Knowledge Range index different from zero for each node. The constraints (12)(13)(14) express conservation of flows [25], while

the constraints (15)(16) impose that paths are built according to the forwarding rule defined by the input parameters f_{dk}^{ij} . Finally the constraints (17) and (18) express the information and communication cost with the Knowledge Range index notation, respectively. Note that given a forwarding rule \mathcal{F} , expressed by the f_{dk}^{ij} parameters, the assignment of the routing (x_{ij}^{sd}) variables is completely dependent on the choice of Knowledge Ranges (y_i^k variables). Once the values of the y_i^k variables have been selected, the set $\mathcal{X} = \{x_{ij}^{sd}\}$ defines the path from source to destination for any connection in \mathcal{P} .

V. PRADA: A DISTRIBUTED PROTOCOL FOR TOPOLOGY KNOWLEDGE ADJUSTMENT

The solution of the ILP problem is not feasible in a practical setting due to its complexity and centralized nature. Here we introduce the *PRobe-bAseD Distributed protocol for knowledge rAnge adjustment* PRADA, which determines the KRs on-line in a distributed way. The objective of PRADA is to allow network nodes to select stable and efficient topology Knowledge Ranges (KRs). This global target is achieved through local decisions and by means of probe packets exchanged among the nodes. The main idea behind PRADA is to allow each node to adjust its KR according to the feedback information it receives from neighboring nodes involved in the same multihop connections. A quick convergence to a near-optimal solution and robustness are the key features of PRADA.

To trade off between the topology information cost and the communication cost, each node which is part of the path of a particular connection (as a source or a transit node), periodically probes its possible KRs. For each of them the node evaluates the increase/decrease in energy expenditure when selected that KR could affect the network operation. To clearly understand the rationale behind PRADA we point out that while the information cost of each node only depends on its KR, the communication cost depends on the KRs of all nodes involved in the communication process. Thus, the communication cost must be monitored with probe packets.

PRADA is executed at each node v_i that has an active role in the network, as a source or a transit node, in a certain set of connections \mathcal{P}_i . For each connection p_k in this set, v_i selects the next hop $l_{v_i}^{\mathcal{F}}(v_d^k, r_{probe})$, where v_d^k is the destination node of the k -th connection, according to the selected forwarding rule \mathcal{F} and to its current KR. Periodically, each active node selects a certain KR to be probed, different from the current one, in the discrete set of possible KRs. We refer to the selected KR as r_{probe} and to the current KR as $r_{current}$. Then the node calculates:

$$C_i^{TOT}(r_{probe}) = C_i^{INF}(r_{probe}) + \sum_{p \in \mathcal{P}_i} c_i^p(r_{probe}) \quad (19)$$

where $c_i^p(r_{probe})$ is the cost of the transmissions along the path from v_i to the destination of the connection p , with KR r_{probe} . This way, the node can calculate the communication cost, from the node itself to all the destinations, plus the information cost that this new KR r_{probe} would cause.

If $C_i^{TOT}(r_{probe}) < C_i^{TOT}(r_{current})$, the value of the KR is updated ($r_{current} = r_{probe}$).

Probe packet:				
Source geographical coordinates	Destination geographical coordinates	Cumulative communication cost	KR _{probe}	F/R flag

Incremental cost record for each KR:		
Row 1:	Destination 1 geographical coordinates	Incremental communication cost 1
Row 2:	Destination 2 geographical coordinates	Incremental communication cost 2
Row n:	Destination n geographical coordinates	Incremental communication cost n

Fig. 4. Structures of Probe Packet and of Incremental Cost Record Table

Let us describe the fields of the probe packets to explain how this information is obtained. As shown in Fig. 4, a probe packet has five fields. The first two contain the geographical coordinates of the source and the destination. The third contains a parameter called *Cumulative Communication Cost* and the fourth contains the value r_{probe} of KR. The last field is a one-bit flag, which is equal to 1 if the packet is on the *forward* path towards the destination, or equal to 0 if it is on the *reverse* path. The cumulative communication cost field, initialized to 0 when the packet is created, is updated hop-by-hop by adding the incremental communication cost, i.e., the communication cost necessary to reach the next hop, to the communication cost stored in the packet. This way, the partial cumulative communication costs are computed hop-by-hop along the path from the sender to the destination.

Algorithm 1 PRADA

```

begin
  randomly select  $r_{probe} \neq r_{current}$ 
  for each  $p_k \in \mathcal{P}_i$  do
     $v_i \rightarrow l_{v_i}^{\mathcal{F}}(v_d^k, r_{probe})$ : probe packet
  end for
  wait for return packets
   $C_i^{TOT}(r_{probe}) = C_i^{INF}(r_{probe}) + \sum_{p \in \mathcal{P}_i} c_i^p(r_{probe})$ 
  if ( $C_i^{TOT}(r_{probe}) < C_i^{TOT}(r_{current})$ ) then
     $r_{current} = r_{probe}$ 
  end if
end

```

After choosing a KR r_{probe} , for each of the connections in \mathcal{P}_i the node sends a probe packet to the relevant next hop and waits for its return. When a node receives a probe packet on the forward path, it looks into the *Incremental Cost Record* table to check if it already knows the incremental communication cost needed to reach this destination. If it does, there is no need to forward the probe packet to the destination. The probe packet is sent back with the updated information and the path bit is set to *reverse*. If it does not, the packet is forwarded to the next hop towards the destination

	Scenario 1	Scenario 2	Scenario 3
Terrain	(10mx10m)	(10mx10m)	(50mx50m)
KRs	(0,2,4,6,8)m	(0,2,4,6,8)m	(0,5,...,20)m
α	4	3	varies
L_D	128 bits	128 bits	128 bits
L_U	128 bits	128 bits	128 bits
T_M	1s	varies	1s
E_{elec}	varies	50pJ/bit	50nJ/bit
β	100pJ/bit/m $^\alpha$	100pJ/bit/m $^\alpha$	100pJ/bit/m $^\alpha$
traffic	10Kbit/s	100Kbit/s	100Kbit/s

TABLE I
PARAMETERS OF THE MODEL USED FOR SIMULATIONS

in order to evaluate the communication cost. The packet is forwarded until a node with information for that destination or the destination itself is reached. The pseudocode in this page (Algorithm 1) describes the operations performed by a node v_i which executes PRADA.

In order to reach stability, we choose to update the KR only if the moving average of the communication cost for the last N_{probe} values gathered is lower than the cost of the current range. In the experiments we assume that all the KRs are probed with the same probability. More sophisticated strategies can also be implemented in order to selectively scan the KRs, aimed at saving transmission power, e.g. by avoiding values of KR that are not likely to bring any benefit and providing a better estimate of the cost.

VI. PERFORMANCE EVALUATION

We implemented the forwarding schemes described in Section II-A, PTKF given in Section III-C and PRADA, given in Section V. We further implemented the ILP problem in AMPL [26] and solved it with the CPLEX [27] solver.

We are particularly interested in scenarios, such as those encountered in sensor networks applications, where the density of nodes is very high. However, due to the computational complexity of the problem we investigate, and to the large amount of the input data, a state-of-the-art workstation can find the optimal solution with CPLEX for networks with at most 100 nodes. Thus, we consider small geographical areas in order to take into account the effects of high node densities on the problem. The model depends on several input parameters, and on the appropriate choice of these parameters which are highly dependent on the technology and on the target applications. Our choice for these parameters was motivated by the model presented in [28]. However we also vary these parameters in order to study their relevant effects on the network performance. Moreover, we believe that a realistic tuning of these parameters must be aided by real hardware implementation of the considered protocols.

We present simulation results for the scenarios illustrated in Table I.

In Scenario 1, all nodes are sources with 10 Kbit/s flows directed towards a unique sink node. In Fig. 5 we show

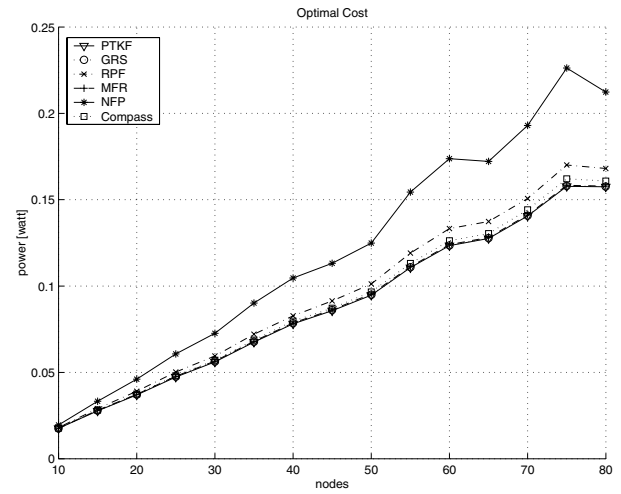


Fig. 5. Scenario 1 - Optimal Cost for the implemented forwarding schemes, $E_{elec} = 50 \cdot 10^{-9} J/bit$

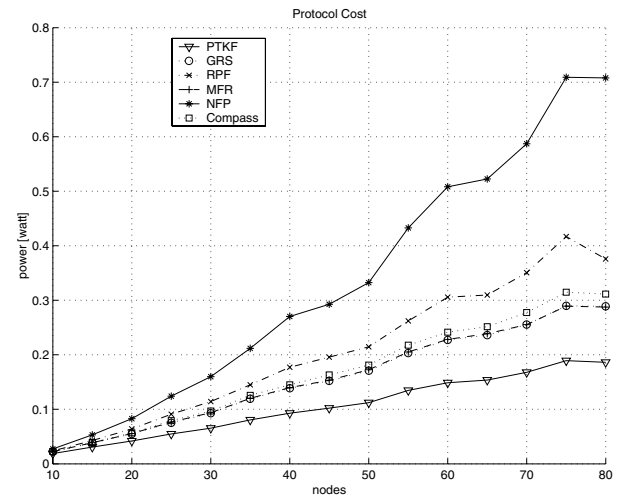


Fig. 6. Scenario 1 - Cost with PRADA for the implemented forwarding schemes, $E_{elec} = 50 \cdot 10^{-9} J/bit$

the optimal cost (the minimum of the objective function of problem **P**, eq. 10), with increasing number of nodes for all the implemented forwarding schemes (described in Sections II-A and III-C). The value chosen for the parameter E_{elec} is $50 \cdot 10^{-9} J/bit$ [28]. Note that the confidence intervals are not shown for the sake of clarity. Since the area of the terrain is very limited, multi-hop is often not energy efficient, which leads source nodes to directly transmit to the destination. For this reason, many forwarding schemes show similar performance. In Fig. 6 we show the total cost for all the implemented forwarding schemes in Scenario 1 obtained by applying PRADA with $N_{probe} = 3$.

In Fig. 7 we compare the optimal cost obtained for PTKF with three different approaches for the solution of the optimization problem, with 95% confidence intervals. The problem is solved with CPLEX (optimal solution), with a greedy local search heuristic, and by applying the distributed protocol PRADA introduced in Section V. CPLEX finds the optimal

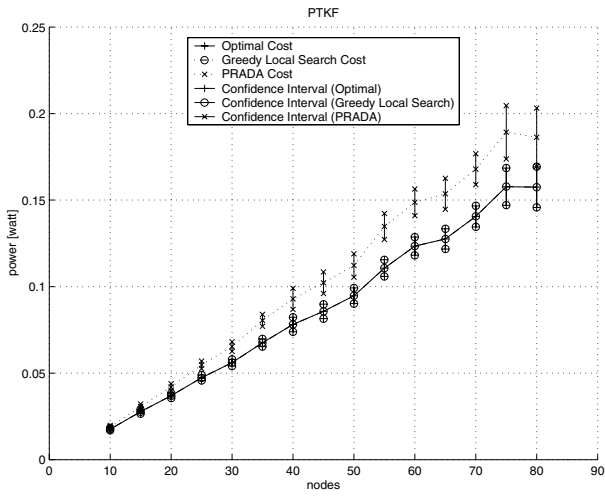


Fig. 7. Scenario 1 - Comparison of Optimal Cost for PTKF with different approaches, $E_{elec} = 50 \cdot 10^{-9} J/bit$

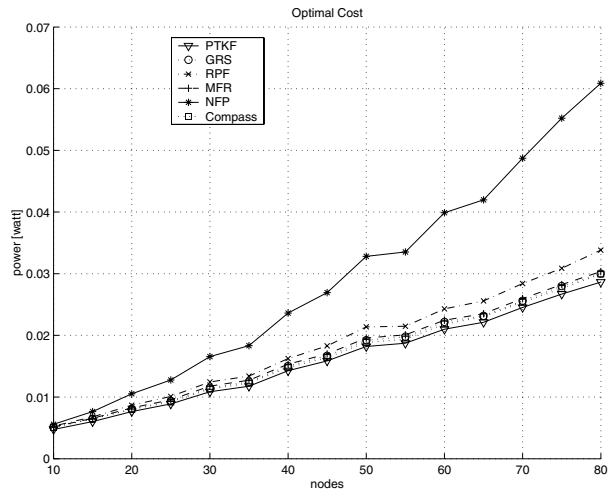


Fig. 9. Scenario 1 - Optimal Cost for the implemented forwarding schemes, $E_{elec} = 50 \cdot 10^{-10} J/bit$

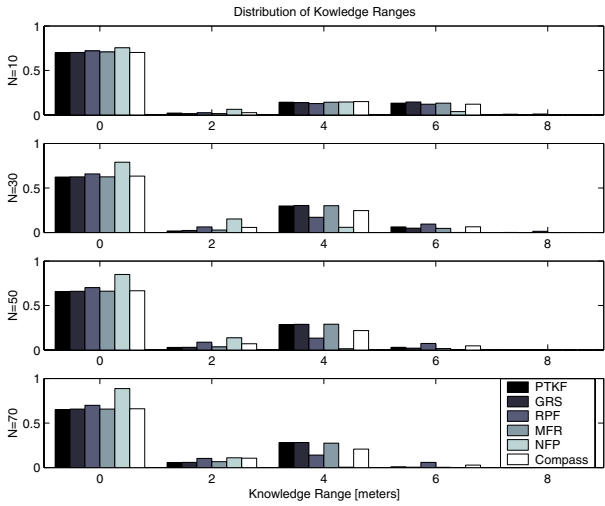


Fig. 8. Scenario 1 - Distribution of values of Knowledge Range, $E_{elec} = 50 \cdot 10^{-9} J/bit$

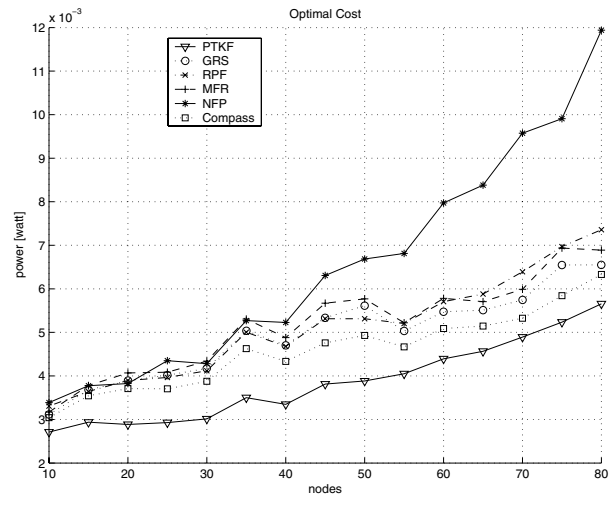


Fig. 10. Scenario 1 - Optimal Cost for the implemented forwarding schemes, $E_{elec} = 50 \cdot 10^{-11} J/bit$

solution for mixed integer problems by using a branch and bound algorithm. The greedy local search heuristic basically scans the nodes one after another and selects for each of them the KR which minimizes the cost; the process is repeated periodically until the stability is reached. Results obtained with PRADA are also given where the PRADA curve is very close to those obtained with CPLEX and with the greedy local search heuristic. This behavior, as will be shown, becomes more evident when the problem becomes more localized.

In Fig. 8 we show the distribution of the values of the KRs in Scenario 1, with $N = 10, 30, 50$ and 70 nodes. The average KR is, in this Scenario, below 1.5 meters, and it is easy to see that most nodes either have a KR equal to 0 (that is, they “prefer” to know nothing about their neighborhood and directly transmit to destination) or they try to know “far” nodes ($4, 6$ meters) to use them as intermediate relays. As a result, it is either efficient to directly transmit to destination or use at most one intermediate node as relay.

By decreasing the E_{elec} parameter, we decrease the weight of the component in energy expenditure (link metric in eq. 4) which is independent of the distance. It becomes more energy efficient to select multi hop paths, since the overall distance independent part of the energy expenditure increases with the number of hops. We would obtain the same effect by increasing the area of the terrain, but we would have a less dense terrain.

It can be inferred by comparing Figures 7, 9, 10 and 11 that the more multi hop paths are energy efficient (low values for E_{elec}), the more PTKF (Section III-C) outperforms the other schemes. In the above figures, the values for E_{elec} are $50 \cdot 10^{-9}, 50 \cdot 10^{-10}, 50 \cdot 10^{-11}$ and $50 \cdot 10^{-12} J/bit$ respectively.

For $E_{elec} = 50 \cdot 10^{-12} J/bit$, the cost obtained with PRADA is optimal, as can be seen from Fig. 12. When the distance independent term E_{elec} in eq. 4 becomes small compared to the area of the terrain, multi hop paths become more energy efficient. When this occurs, by selecting KRs which are opti-

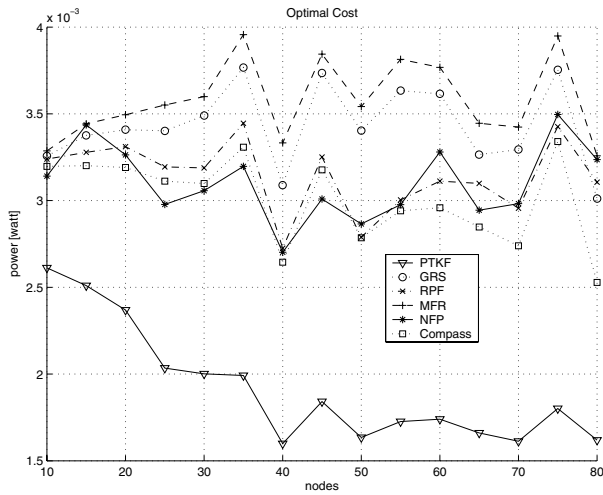


Fig. 11. Scenario 1 - Optimal Cost for the implemented forwarding schemes, $E_{elec} = 50 \cdot 10^{-12} J/bit$

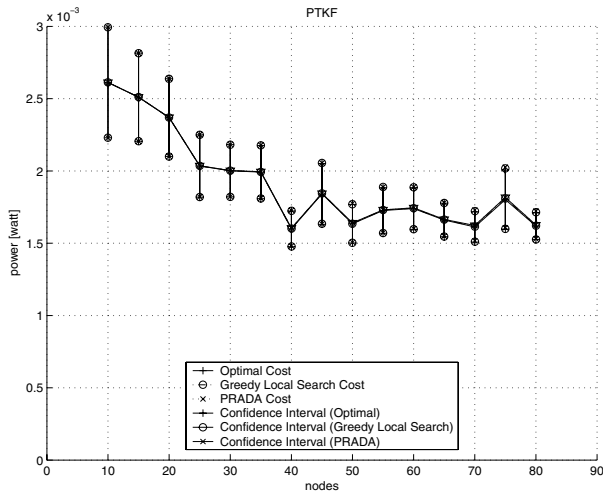


Fig. 12. Scenario 1 - Comparison of Optimal Cost for PTKF with different approaches, $E_{elec} = 50 \cdot 10^{-12} J/bit$

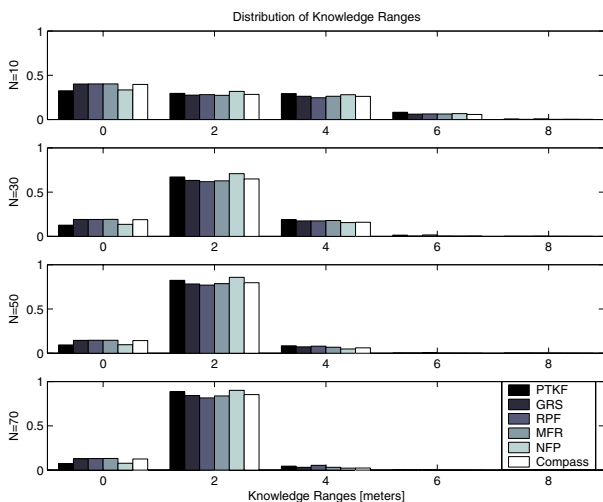


Fig. 13. Scenario 1 - Distribution of values of Knowledge Range, $E_{elec} = 50 \cdot 10^{-12} J/bit$

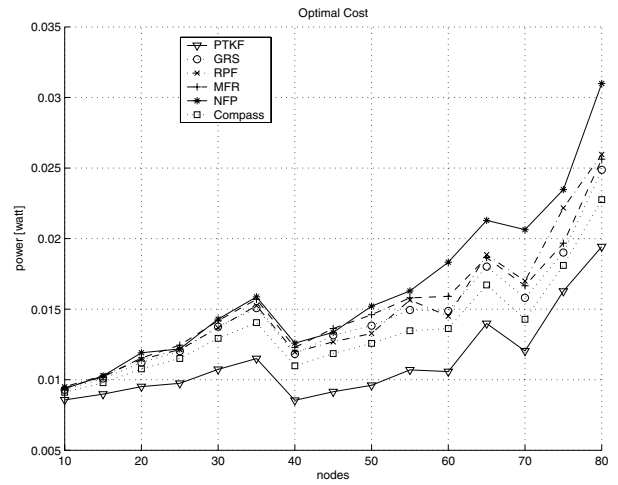


Fig. 14. Scenario 2 - Optimal Cost for the implemented forwarding schemes, $T_M = 0.01s$

mal only locally, as PRADA does, we obtain globally optimal solutions, because the problem becomes more localized when E_{elec} decreases. In Fig. 13 we demonstrate that it is more energy efficient to select near nodes (KRs are 2 meters), as E_{elec} decreases. This is particularly true when the density of the nodes increases.

In Scenario 2, all nodes are sources with 100 Kbit/s flows directed towards a unique sink node. In Fig. 14 we report optimal costs with increasing number of nodes for all the implemented forwarding schemes (Section II-A). Again, PTKF (Section III-C) performs better than the other forwarding schemes. More greedy schemes such as Nearest Forward Progress (NFP) and Most Forward within Radius (MFR), both described in Section II-A, consume more energy.

In Fig. 15 we give optimal paths for all the considered forwarding schemes in a Simulation with 50 nodes. Fig. 16 shows the total cost for all the implemented forwarding schemes in Scenario 2 obtained by applying PRADA with $N_{probe} = 3$. Fig 16 and 14 are almost identical, which is explicitly shown by Fig. 17 where we compare the results obtained for PTKF with the three different optimization approaches (CPLEX, greedy local search, PRADA). In Fig. 18 we depict the information cost (eq. 17) and the communication cost (eq. 18) for PTKF, again with the three different approaches. The communication cost is shown to highly exceed the information cost when relatively high data rate flows must be supported. In Fig. 19 we show the average value of the Knowledge Range with increasing number of nodes for all the proposed schemes. It is obvious that a very limited knowledge of the topology is needed in average, less than 2 meters.

In Figures 20 and 21 we show the average convergence dynamics of PRADA to the optimal solution with 70 and 40 nodes. At every *step*, any sensor node selects and probes randomly one of its KRs. For 70 nodes, after 3000 steps we obtain a near-optimal solution. In Fig. 22 we assume a lower mobility rate, thus, we set $T_M = 1$. As can be seen in Fig. 22, for lower rates of mobility PTKF even more evidently outperforms the other schemes. A more extended

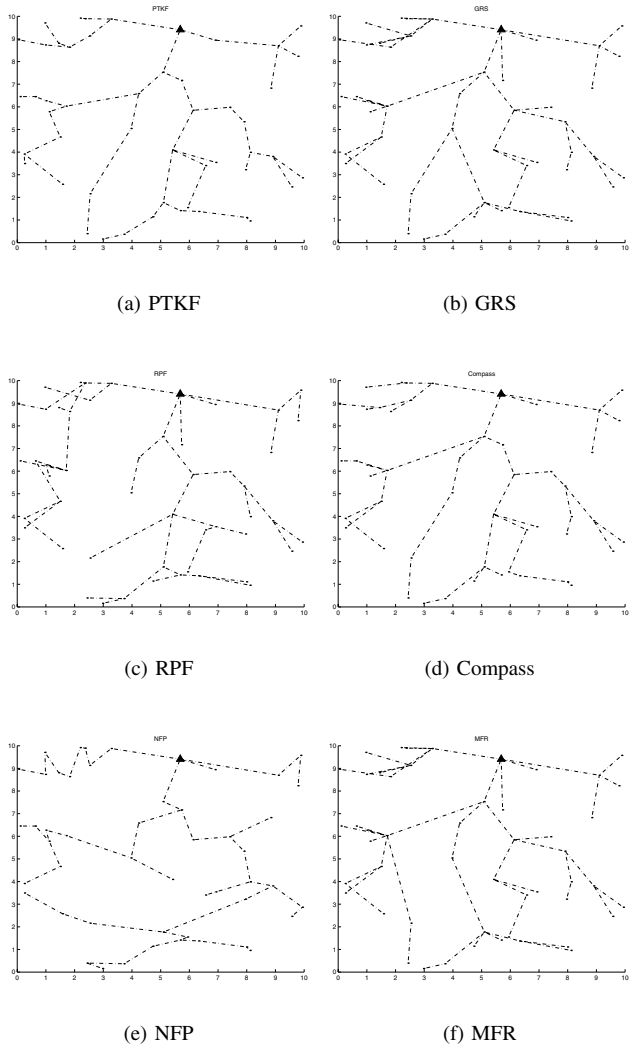


Fig. 15. Optimal Routing Trees with different Routing schemes - Scenario 2, 50 nodes.

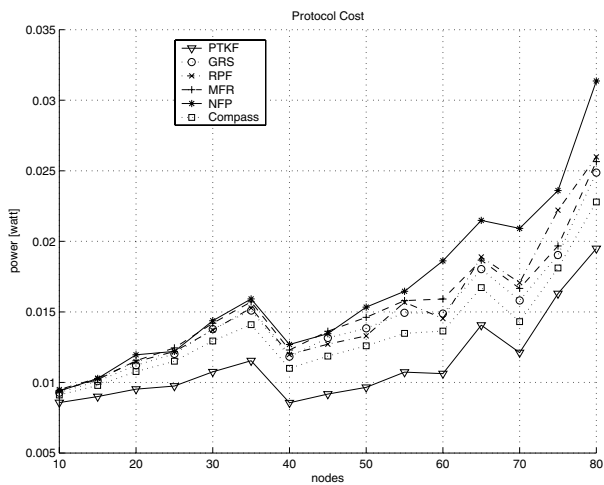


Fig. 16. Scenario 2 - Cost with PRADA for the implemented forwarding schemes, $T_M = 0.01s$

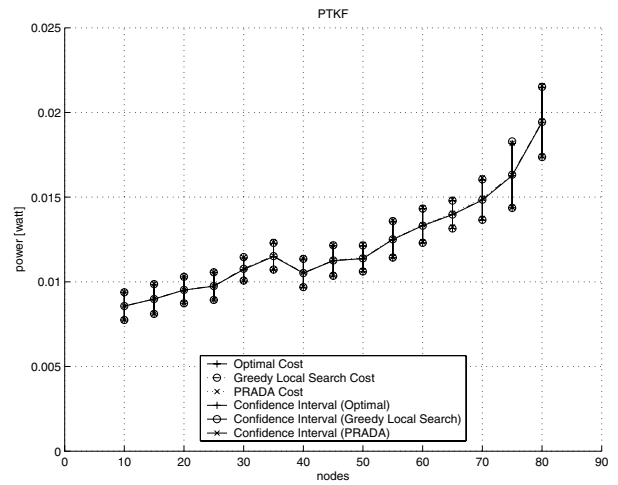


Fig. 17. Scenario 2 - Comparison of Optimal Cost for PTKF with different approaches, $T_M = 0.01s$

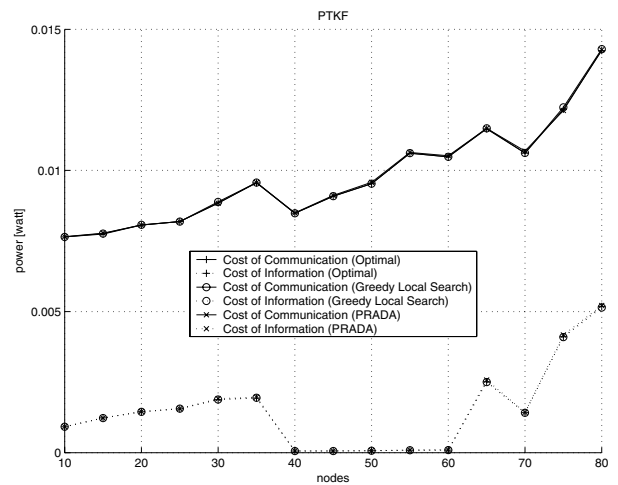


Fig. 18. Scenario 2 - information cost and communication cost for PTKF, $T_M = 0.01s$

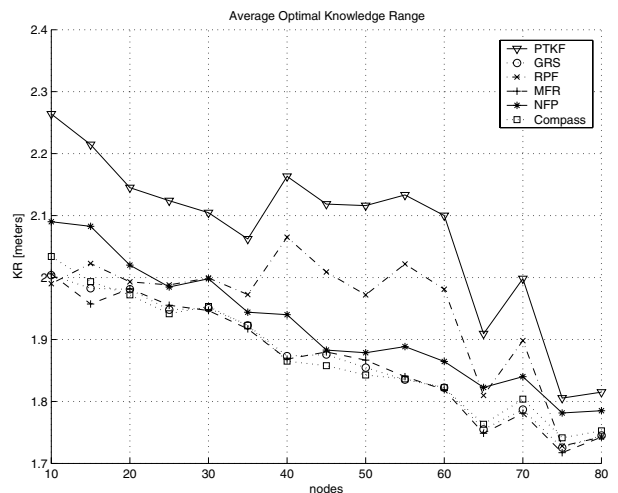


Fig. 19. Scenario 2 - Average KR with different forwarding schemes, $T_M = 0.01s$

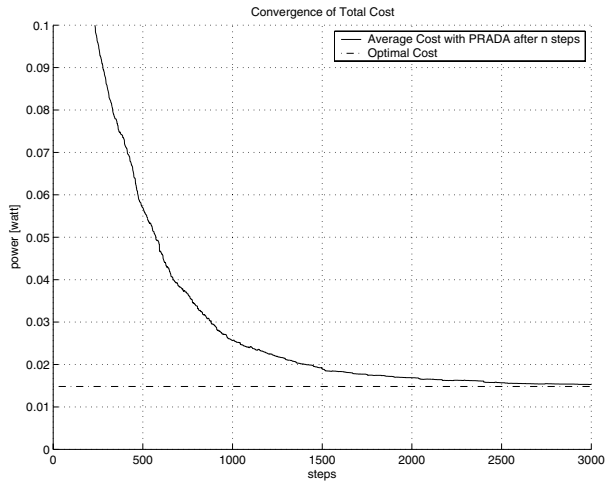


Fig. 20. Scenario 2 - Convergence of PRADA with PTKF, 70 nodes, $T_M = 0.01s$

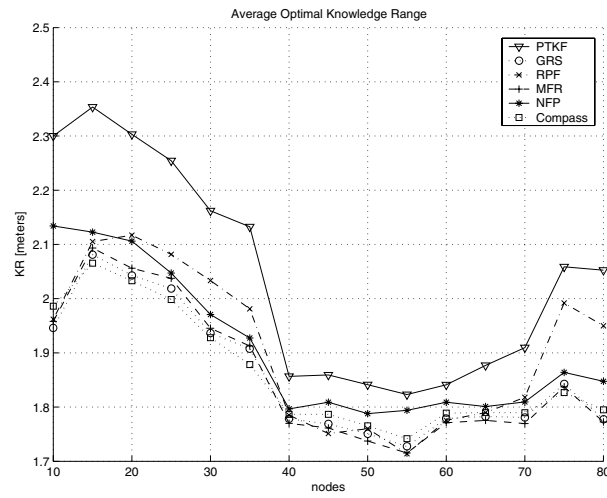


Fig. 23. Scenario 2 - Average KR with different forwarding schemes, $T_M = 1$

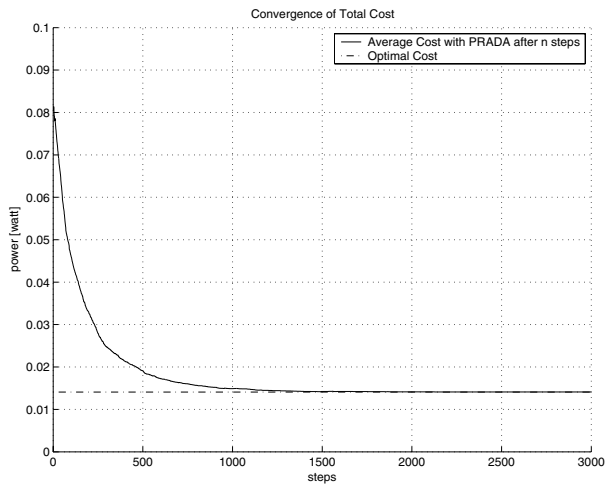


Fig. 21. Scenario 2 - Convergence of PRADA with GRS, 40 nodes, $T_M = 0.01s$

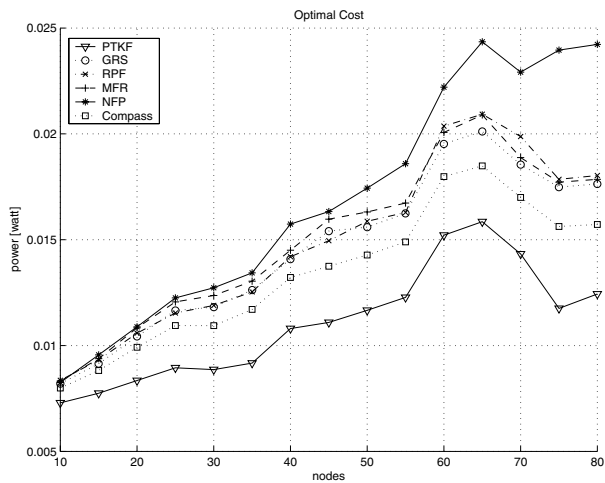


Fig. 22. Scenario 2 - Optimal Cost for the implemented forwarding schemes, $T_M = 1$

local topology knowledge brings benefits in terms of energy to the scheme which best exploits this information. This is confirmed by Fig. 23 that shows how the average KR's increase in general, and particularly for PTKF which is able by its nature to better take advantage of a more extended knowledge. Still, the extension of local knowledge of the topology is very limited compared to the terrain dimensions.

In Scenario 3, 100 Kbit/s traffic flows are simultaneously generated by sensor nodes in the network towards a sink node, but the terrain is bigger ($50m \times 50m$). Figures 24 and 25 report optimal cost with increasing number of nodes for all the implemented forwarding schemes with $\alpha = 3$ and $\alpha = 5$, respectively. For high values of the parameter α the optimal cost decreases as the node density increases, while for low values of α the increase in the amount of traffic overcomes the positive effect of a higher node density. Again, in all the experiments of Scenario 3 PTKF is shown to perform better than any other scheme. This is more evident again when multi-hop paths are energy efficient, that is, when α is higher (the distance dependent part of the cost has a higher weight). Again more greedy schemes, such as Nearest Forward Progress (NFP) [12] and Most Forward within Radius (MFR) [11], both described in Section II-A, are shown to lead to higher energy consumptions.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we solve the problem how to determine optimal local topology knowledge for energy efficient geographical routing for sensor networks. We give an Integer Linear Programming Formulation of the problem which constitutes a framework for the analysis of the energy efficiency of different forwarding schemes. We show that only a limited local topology knowledge is needed to take energy efficient routing decisions. We introduce a distributed protocol called PRADA which quickly achieves a near-optimal solution.

Future research will include the extension of the model, primarily to include features such as battery and bandwidth

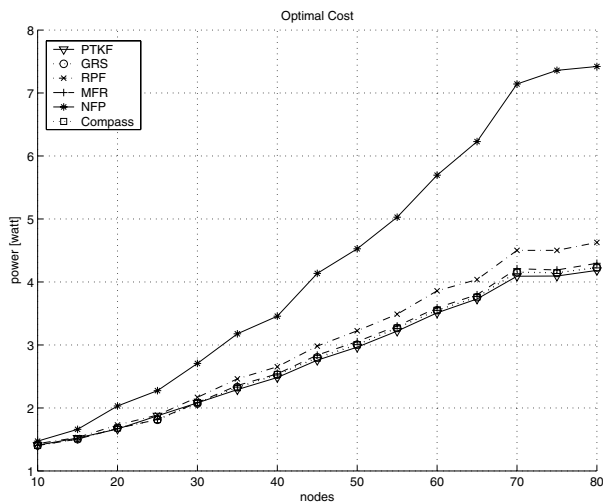


Fig. 24. Scenario 3 - Optimal Cost for the implemented forwarding schemes, $\alpha = 3$

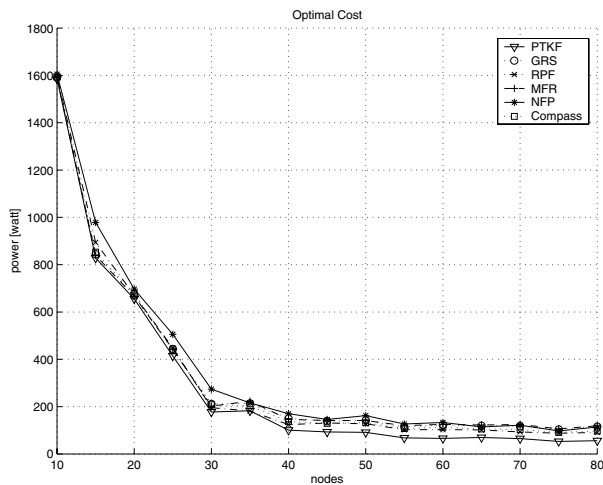


Fig. 25. Scenario 3 - Optimal Cost for the implemented forwarding schemes, $\alpha = 5$

constraints for the nodes. Moreover, the considered schemes will be implemented in a tool simulating all layers of the communication task to evaluate the effect of the signaling traffic.

ACKNOWLEDGMENTS

The authors would like to thank Zhaosong Lu of the School of Industrial and Systems Engineering at the Georgia Institute of Technology, for his suggestions to formulate the ILP Problem.

REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks (Elsevier) Journal*, pp. 393-422, March 2002.
- [2] Z.J. Haas, "Design Methodologies for Adaptive and Multimedia Networks," *IEEE Communications Magazine*, Vol. 39, no. 11, November 2001, pp 106-07.

- [3] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. IEEE/ACM Mobicom 1999*, Seattle, pp. 263-270.
- [4] V. Rodoplu, T. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE Journal of Selected Areas in Communications*, Vol. 17, no. 8, August 1999, pp. 1333-1344.
- [5] S.M. Woo, S. Singh, C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. of IEEE/ACM Mobicom 1998*, pp. 513-529.
- [6] J. Chang, L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," *Proc. of IEEE Infocom 2000*, pp. 22-31.
- [7] J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. IEEE/ACM Mobicom 2000*, pp. 120-30.
- [8] R. Jain, A. Puri, R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *IEEE Personal Communications*, Feb. 2001, pp. 48-57.
- [9] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computers*, Aug. 2001, pp. 14-25.
- [10] R. Ramanathan, R. Rosales-Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," *Proc. IEEE Infocom 2000*.
- [11] H. Takagi, L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Transactions on Communications*, Vol. 32, no. 3, 1984, pp. 246-57.
- [12] T.C. Hou, V.O.K. Li, "Transmission Range Control in multihop packet radio networks," *IEEE Transactions on Communications*, Vol. 34, no. 1, pp. 38-44.
- [13] G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," *ISI res. rep ISU/RR- 87-180*, Mar. 1987.
- [14] E. Kranakis, H. Singh, J. Urrutia, "Compass routing on geometric networks," *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999.
- [15] R. Nelson, L. Kleinrock, "The spatial capacity of a slotted ALOHA multihop packet radio network with capture," *IEEE Transactions on Communications*, Vol. 32, no. 6, 1984, pp. 684-694.
- [16] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," *IEEE Communications Magazine*, Vol. 40, No. 7, July 2002, 128-134.
- [17] S. Giordano, I. Stojmenovic, L. Blazevic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," *Ad Hoc Wireless Networking*, Kluwer, 2003
- [18] I. Stojmenovic, "Location Updates for Efficient Routing in ad hoc network," *Handbook of Wireless Networks and Mobile Computing*, Wiley, 2002, 451-471.
- [19] B. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. of ACM/IEEE MobiCom 2000*.
- [20] D. Niculescu, B. Nath, "Trajectory based forwarding and its applications," *Tech. Rep. DCS-TR-48*, Rutgers University, June 2002.
- [21] I. Stojmenovic, X. Lin, "Power-Aware Localized Routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 11, November 2001, 1122-1133.
- [22] X. Hong, K. Xu, M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, July/August 2002.
- [23] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of ACM/IEEE Mobicom 2001*, Rome, Italy, July 2001.
- [24] D. Niculescu, B. Nath, "Localized positioning in ad hoc networks," *Elsevier's Journal of Ad Hoc Networks*, Special Issue on Sensor Network Protocols and Applications. Vol. 1, Issues 2-3, Pages 211-349, September 2003.
- [25] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications," Prentice Hall, February 1993.
- [26] R. Fourer, D. M. Gay, B. W. Kernighan "AMPL: A Modeling Language for Mathematical Programming," Duxbury Press / Brooks/Cole Publishing Company, 2002.
- [27] www.cplex.com
- [28] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, October 2002.