Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 5 (2007) 974–997
www.elsevier.com/locate/adhoc

Power-aware single- and multipath geographic routing in sensor networks ☆

Shibo Wu, K. Selçuk Candan *

Department of Computer Science, Fulton School of Engineering, Arizona State University, Tempe, AZ 85287, United States

Received 9 June 2005; received in revised form 14 November 2005; accepted 29 April 2006
Available online 9 June 2006

Abstract

Nodes in a sensor network, operating on power limited batteries, must save power to minimize the need for battery replacement. We note that the range of transmission has a significant effect on the power consumption of both the transmitting node and listeners. This paper first presents a *Geographical Power Efficient Routing* (GPER) protocol for sensor networks. Each sensor node makes *local* decisions as to how far to transmit: therefore, the protocol is power efficient, localized, highly distributed, and scalable. In GPER, given a final destination, each node first establishes a subdestination within its maximum radio range. The node, however, may decide to relay the packet to this subdestination through an intermediary node or alter the subdestination if this will preserve power. Traditional deterministic geographic routing algorithms aim at achieving close to the shortest weighted paths. However, they normally stick to the same paths for the same source/destination pairs. This may conversely drain the nodes on these paths and result in short network life when the communication in the network is unevenly distributed. Thus, we further investigate a set of probabilistic multipath routing algorithms, which generate braided multipaths based only on *local* information. The algorithms have less communication and storage overhead than conventional on-demand multipath routing algorithms, while providing greater resilience to node failures. Simulations on NS2 show that GPER almost halves the power consumption in the network relative to alternative geographic routing algorithms. Furthermore, in situations where the communication tasks are non-uniformly distributed, probabilistic multipath routing contributes up to an additional 30% to network lifetime.

Keywords: Wireless sensor networks; Geographic routing; Localized routing; Power-aware routing

---

1. Introduction

Applications of wireless sensor networks vary from personal area networks, where all wireless devices are located physically close to each other (in potentially architected configurations), to wide-area networks, where sensors are placed (potentially randomly) on a very large open terrain for in situ observations. The advance of low-cost, miniaturized sensor technologies makes it possible to deploy a large number of detection equipment on an unknown and uneven terrain for various measurement and surveillance applications. The *sensor nodes* are delivered and scattered on a specific region and *prime nodes* act as conduits between sensors and

---

☆ This work is partially supported by NSF grant # 0308268.

* Corresponding author. Tel.: +1 480 965 2770.
E-mail addresses: shibo.wu@asu.edu (S. Wu), candan@asu.edu (K.S. Candan).

1570-8705/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2006.04.007

the external data processing units. Various types of data collected and filtered by the sensors are delivered to the prime nodes via self-organizing wireless sensor networks. These sensor networks serve as the information conduit between the sensing devices and the deliberative and reactive processes that lie within or outside the network. In wide-area wireless sensor networks, such as the next generation smart dust-style sensing environments, the number of sensors deployed in the system can be large. The network therefore has to function in a fully distributed and scalable manner. Furthermore, in most cases, it is impractical to replace or recharge the batteries of the already deployed sensors. Since wireless communication can incur comparable or significantly higher energy cost than sensing and computation tasks [36], routing algorithms have to be scalable, power efficient, and robust.

In wide-area wireless sensor networks with large numbers of nodes, traditional table-driven and on-demand routing protocols are not directly applicable [35,16,32,31,34,37]. First, for wide-area sensor networks, the global routing table can grow unmanageably. Secondly, exchanging routing tables until a stable state is reached becomes unacceptable for sensor networks with large number of nodes. In [39] and its variant [24], the route table is generated by running Bellman–Ford algorithm; however each node only exchanges route tables with a subset of its neighbors, which is called the enclosure of this node. Sending messages directly to a node outside the enclosure will cost more energy than forwarding the message through nodes in the enclosure. In [25], the best path is chosen among the minimal power consumption paths and paths that maximize minimal residual power with the trade-off determined by a parameter. This algorithm is centralized because each node must know the remaining power of all nodes and power consumption to transmit a packet along any two nodes in the network. In [4], the authors developed a flow redirection algorithm to take message flow from the path of shortest lifetime and give it to the path of longest lifetime. To calculate the lifetime of the nodes, the message rate must be known. To reduce the information exchange overhead in wireless networks, several on-demand routing protocols have been proposed [16,32,34]. In its most basic form, an on-demand routing protocol will flood a route discovery message into the network and obtain the best path to the destination within the response. This of course will cause significant overhead if the number of nodes is large. A number of methods have been proposed to constrain the number of nodes that will rebroadcast the route discovery message [19].

Since the topology of a sensor network is determined by the geographic topology of its nodes, we can leverage the geographic information in developing routing algorithms that use only *local* information [2,6,18,14,21,22,41,47,26,54] thus minimizing the route discovery and messaging costs. In this paper, we first describe the *Geographical Power Efficient Routing* (GPER) protocol initially proposed in [51], in which each node makes local decisions as to how far to transmit the data. Given a final destination, by GPER each node establishes a subdestination within its immediate neighborhood, defined as the maximum distance it can transmit to. The packet may, however, be transferred to an intermediary relay node if this is likely to preserve power. This intermediary node, then, may alter the subdestination based on its own radio range and its neighborhood status. The results presented in Section 5 show that, whether networks have uniform or non-uniform node density distributions, using GPER can greatly reduce the power consumption.

We then note that choosing and sticking to minimal power consuming paths may conversely drain the nodes on these paths and result in short network life when the communication in the network is unevenly distributed. Multipath algorithms aim at eliminating power drainage due to repeated use by routing consecutive messages between the same end-points through different paths. The use of multiple paths improves load balancing: the traffic between a node-destination pair is split among multiple paths so that the energy utilization is spread across nodes in the network[1] However, since not all of the paths used during multipath routing can be optimal, multipath routing algorithms have to consider the trade-off between drainage due to repeated use and power over-consumption due to suboptimal path selection. Therefore, in Section 4, we extend the single-route GPER algorithms to handle effective multipath routing. The resulting protocols establish multiple paths between source and destination pairs and improve sensor network

---

[1] Multipath routing can also increase resilience to node failures when multiple copies of data are *simultaneously* sent along different paths. In this paper, however, we focus on the load balancing aspect of multipath routing.

lifetime. Most, previous multipath algorithms require flooding route requests into the network to identify and establish desirable alternative paths. However, we note that, for reasons of robustness and energy efficiency, the multipath routing algorithms should use localized decisions for choosing paths. In this paper, we propose and investigate a family of probabilistic routing algorithms (MGPSR, $\mathrm{MGPER}_{\mathrm{sub}}$, and $\mathrm{MGPER}_{\mathrm{hop}}$), which can generate braided multipaths, based only on local information. Unlike most existing multipath algorithms, the algorithms we proposed in this paper eliminate the overhead of flooding route request into the network and can construct the multipaths on the fly. Thus, our algorithms are scalable and efficient. They have less communication and storage overhead than conventional on-demand multipath routing algorithms, while providing greater resilience to node failures.

2. Related work and contributions of this paper

In this paper, we focus on the network lifetime and the minimization of the power usage. Geographic routing is preferred (in some cases required) in sensor networks where coordinates of the nodes determine the topology of the network. In [46], the topology of the network is assumed to be known for each node. k minimum energy node-disjoint and link-disjoint paths are calculated by executing a minimum weight k node-disjoint paths algorithm. In [2,21,18,6,14], greedy routing algorithms minimizing the number of hops in a mobile wireless network are proposed. By these algorithms, there is no exchange of routing tables (as in proactive routing protocols [39,24,25]) or flooding of route discovery messages into the network (as in reactive routing protocols [16,32,34]). The routing path is not determined prior to the actual packet forwarding and the routing decision at each node is made locally. Traffic between the same source–destination pair tends to follow the same path. When greedy routing fails, all these algorithms resort to a recovery mode which is denoted as planar graph routing, while planar graph routing is a recovery strategy which preserves the stateless property of greedy routing algorithms. Planar graph routing is called FACE routing in [2,21] and perimeter routing in [18]. The routing algorithms in [6,41,22] are similar to the above greedy algorithms, although [22,41] are mainly for lossy wireless sensor networks and [6] focuses on improving the hop count during the recovery phase.

2.1. Geographic power-aware routing protocols

In MECN [39] and its variant SMECN [24], each node calculates an enclosure such that sending messages directly to a node outside the enclosure will cost more energy than forwarding the message through some nodes in the enclosure. A distributed Bellman–Ford shortest weighted path based algorithm is run to generate the optimal routing table for each node. Instead of exchanging routing tables with all nodes in the neighborhood, each node only exchanges its routing table with those neighbors in its enclosure; therefore, the number of messages and the power consumption for generating the optimal routing tables for all nodes are reduced.

In localized power-aware routing protocols [48,47,54,29], the transmitting node s locally estimates the total power consumption for forwarding the packet to the destination d through each of its neighbors, n (i.e. the sum of $\rho(s,n)$ and $\rho_{\mathrm{est}}(n,d)$, where ρ denotes power consumption and ρ_{est} denotes an estimate of the power needed to route a message from the neighbor to the final destination) and chooses the neighbor that minimizes the total power consumption as the next hop. Ideally, the more accurate the estimation is, the more power saving that can be achieved. In LAPAR, [54], $\rho_{\mathrm{est}}(n,d)$ is over-estimated because it assumes that no relays will be used from the selected neighbor n to destination d. This over-estimation becomes larger when the distance between n and d is large. Due to the over-estimation, total power consumption is mostly dominated by $\rho_{\mathrm{est}}(n,d)$, therefore, in most cases, the neighbor closest to the destination is chosen as the next hop. SP in [48,47], on the other hand, assumes that a set of optimally placed relay nodes exist between n and d and, thus, $\rho_{\mathrm{est}}(n,d)$ is optimal. Unfortunately, this assumption does not usually hold and, depending on the power model, $\rho_{\mathrm{est}}(n,d)$ can be greatly over- or under-estimated. The estimation error becomes larger when d is far away from n. As discussed above, over-estimating $\rho_{\mathrm{est}}(n,d)$ usually causes the neighbor closest to the destination chosen as the next hop, while under-estimating $\rho_{\mathrm{est}}(n,d)$ usually causes the neighbor with smallest $\rho(s,n)$ be selected as the next hop, resulting in large hop count as well as large power consumption. In [48,47], it is also proposed that the estimate $\rho_{\mathrm{est}}(n,d)$ should be multiplied by a constant whose value is determined by the density of the network. However, such a constant can also not easily be calculated, especially when the network is not uniformly distributed. In PTKF [29], $\rho_{\mathrm{est}}(n,d)$ may be either over- or under-estimated, depending on the size of the distance-independent power consumption. Furthermore, none of [48,47,54,29] takes the receiving power consumption of the neighbors who are not involved in routing into consideration. As discussed above, such inaccurate estimations of the overall power consumption from the neighbor to the destination will result in routing paths with large network-wide power consumptions.

Note that pure power consumption metric may be misleading in the long term; some nodes participate in routing packets for more source–destination pairs and this may result in their early failures. Refs. [44,3,11,48,47] take the residual power of neighbors into consideration, such that energy critical nodes are avoided during routing. Singh et al. [44] proposed to use a function $f(n) = 1/g(n)$, where $g(n)$ is the remaining power of the neighbor n, to denote node n's reluctance to be involved in relaying packets. The protocols in [3,11,48,47] are variations which combine two metrics: the power consumption of the path and the residual power on the path. However obtaining accurate residual power estimations of neighbors can be costly. In this paper, we discuss multipath routing protocols which aim at preventing early power failures even when such residual power estimations are not available. We also show how these protocols can be augmented if residual power estimations of neighbors are available for relay selection.

2.2. Geographic multipath routing protocols

Unfortunately, traditional geographic routing algorithms, power-aware or not, choose and reuse paths based on geographical information and thus potentially depleting resources along common paths. To prevent resource depletion by repetitive use, in Section 4, we propose probabilistic geographical multipath routing protocols, which calculates different paths for different messages/packets between a given source–destination pair.

Multipaths are generally classified as node-disjoint multipaths and braided multipaths (where paths are not completely node-disjoint). For instance, [50,23] aim to find node-disjoint multipaths whereas [33,42] consider braided multipaths. Refs. [9,28] consider both kinds of multipaths. In [33] the reply message is redirected to the shortest of the least selected cached path that does not create a reply loop. Besides, [33] tries to decompose route replies to reconstruct shorter, more diverse paths. In [9,28] upon receiving the route request messages, the destination sends out k reinforcement messages one by one, among which the kth message is sent to the neighbor with the kth highest quality (e.g. lowest delay). When an intermediate node receives the kth reinforcement message, it queries its neighbors in the sequence of high to low quality for one that has not been on any of the previous paths yet, and then forwards the reinforcement message to this neighbor. With few adaptations, the above approach can be applied to construct braided multipaths. Lee and Gerla [23] proposes to rebroadcast duplicate route request that arrives from a different incoming link and a smaller hop count than the first received route request. This approach can discover maximally disjoint paths at the cost of transmitting more route requests. Wu and Harms [50] requires all received route request messages to be cached instead of dropped in the middle nodes. This will ensure that none of the paths from source to destination is lost. If the path in the route request message is not node-disjoint with previously received paths, the destination sets a *redirection* flag on in the reply message. When a middle node receives a reply message with *redirection* flag on, it searches in the local cache for a path that is node-disjoint with the remaining path back to source in the route reply message. If such a path can be found, then *redirection* flag is cleared, and the reply message is redirected along this path. In [42], the destination initiates the route discovery process, and each intermediate node forwards the request only to the neighbors that are closer to the source and farther from destination. The source will calculate the multipaths based on the received route requests.

All of the above approaches (and the others that rely on broadcasting) have the problem of scalability in wireless network setups: broadcasting of route request messages is wasteful. In [26] a set of routing schemes, which do not require broadcasting, were proposed to forward a message m along different paths. In the *alternate* method, the ith received copy of message m is forwarded to ith best neighbor, according to the selected criterion. In the *disjoint* method, each intermediate node, upon receiving m, will forward it to its best neighbor among those who never received the message. Each neighbor therefore has to be associated with a list of messages that it has forwarded, however, maintaining such information can be impractical when the number of messages is large. In the multipath method, the source node forwards a copy of m to each of k best neighbors according to distance from the destination. Each of the k copies follows the original, alternate, or disjoint method.

Unlike the above schemes, including [26], our goal is not to identify different paths that will be simultaneously used to improve resilience of the system, but to make sure that different messages between a given source–destination pair are routed through different paths to prevent resource depletion. *Therefore, we propose probabilistic geographical multipath routing protocols that (a) do not need request broadcasting and (b) different paths are calculated for different messages between a given source–destination pair, without diverging from the optimal known path significantly.*

Probabilistic selection of paths[2] has been used along with traditional and on-demand routing algorithms for improving the resilience of the network to failures [52]. A common aspect of the existing multipath protocols is that they use an *on-demand routing approach* to search for multiple paths: the source node broadcasts a route request message into the network. The IDs of the visited nodes are recorded into the messages. The destination node sends reply messages to the source node along the reverse-paths in the routing request messages. As we discussed above, on the other hand, geographic routing is more suitable than on-demand routing for sensor networks.

In this paper, we investigate the use of probabilistic routing techniques specifically for *improving the network lifetime*. In [14], a similar approach is used to trade-off load balancing and optimal path length, but it does not consider power efficiency and the effect of probabilistic routing on network lifetime is not explored. Probabilistic geographic multipath routing we discuss in this paper has several advantages compared to conventional multipath routing algorithms discussed above.

- First, the source node does not need to broadcast route discovery messages. Such broadcasting overhead would become unacceptable for large sensor networks. Probabilistic routing also eliminates the overhead of saving multiple paths at source, since the multipaths do not need to be known prior to routing.
- These algorithms adapt to topology changes much better than conventional algorithms. In conventional algorithms, if any node on a path is dead for whatever reason, this path is dead as a whole. In the case that the multipaths are node-disjoint, those nodes that are alive on the dead path are just wasted and cannot be used to forward messages any more. Even if the multipaths are not node-disjoint (braided multipaths), the number of multipaths that the source node can save is limited, so that a small number of node failures may cause a large portion of saved multipaths useless. When none of the multipaths are available, rebroadcasting would be required to establish new set of multipaths. In the proposed algorithms, each node on path acts just like the source node; it calculates alternative next hops, selects one of them as next hop by probability and forwards the message. The paths are generated on the fly, and a newly generated path may be different from all the previous paths, so the number of multipaths that can be generated are unlimited.

2.3. Contributions of this paper

In this paper, we first propose the *Geographical Power Efficient Routing* (GPER) protocol. Unlike, for instance, GPSR where nodes send messages as far as they can, GPER uses relays (intermediary nodes among its neighbors) when sending packets toward the neighbor closest to the destination (subdestination). In GPER, since each node knows whereabouts of the nodes within its power range, each intermediate node locally identifies the most power efficient path to the subdestination within its reach and forwards the packet to the next node (relay) on this locally identified path. The packet, however, may not follow this path exactly since the subdestination changes from node to node. Two mechanisms, force routing and perimeter routing, are applied in GPER to prevent infinite loops. Therefore, the contributions include:

- A protocol *RouteWithinNeighbors* that enables each sensor to choose the best next node in its radio range (Section 3.3).

---

[2] In a completely different context, the term “probabilistic routing” denotes routing algorithms which use delivery predictability vectors to increase the likelihood of message delivery in mobile networks [27]. Our use of the term differs from this.

- The GPER protocol, which builds on *RouteWithinNeighbors* and dynamic subdestination adjustment and forced routing techniques, to establish routes for destinations that are not within the radio range of the source (Section 3.4).

In Section 5, we show that GPER can greatly reduce the power consumption for various types of node density distributions.

We also propose three probabilistic geographic multipath routing algorithms that complement geographic routing solutions, GPSR [18] and GPER (Section 4):

- MGPSR builds on GPSR. It probabilistically selects the next hop from a set of alternatives.
- $\text{MGPER}_{\text{sub}}$ builds on GPER. Like MGPSR, $\text{MGPER}_{\text{sub}}$ probabilistically selects the subdestination from a set of alternatives; but then calculates the next hop relative to this selected subdestination.
- The $\text{MGPER}_{\text{hop}}$ also builds on GPER. Unlike MGPSR and $\text{MGPER}_{\text{sub}}$, $\text{MGPER}_{\text{hop}}$ selects an optimal subdestination as in GPER, but calculates a set of alternative next hops for this subdestination and probabilistically selects the next hop within this set.

In the proposed multipath algorithms, we use three routing criteria (distance to the destination, deviation_angle, residual power) for calculating and selecting among alternatives (Section 4.4). In Section 5.2, we experimentally compare and contrast the properties of these algorithms and highlight under what conditions each is most efficient. The experiment results show that when the routing tasks are uniformly distributed in a network, since the nodes are more or less evenly used, multipath routing may not contribute to network lifetime. However, in situations where the communication tasks are non-uniformly distributed, multipath routing contributes to the network lifetime.

3. GPER: geographical power efficient routing

In this section, we first introduce the *Geographical Power Efficient Routing* (GPER) protocol, which enables nodes make local decisions as to how far to transmit the data to reduce the end-to-end power consumption. GPER consists of two complementary protocols. The first one aims to identify the best next hop within the nodes that are in the radio range of the source. Unlike the others in the literature [2,6,18,14,21], this algorithm not only considers which neighbor node has the maximum progress towards the destination, but also how much power can be saved if intermediary relays are used to reach this neighbor. The second algorithm aims to establish routes between nodes that are not in the radio ranges of each other. This algorithm builds on the first one by introducing novel dynamic subdestination adjustment and forced routing techniques. Before we start discussing these two protocols, we first present the wireless network and power models that underline the proposed approach.

3.1. Wireless network model

In this paper, we adopt a commonly used sensor and ad hoc network model [5,7,24,39,48,47,54,29]: A set, S, of nodes is located in a two dimensional geographic area, G. Each node $v_i \in S$ has coordinates, $coord(v_i) = \langle x_i, y_i \rangle$. Each node knows its own coordinates. This can be achieved either through an internal GPS device or through a separate calibration process. The location of a node acts as its ID and network address. Therefore, there is no need for a separate ID establishment protocol. Each packet is marked with the location of the next hop and that node picks up the packet. The transmission ranges are assumed to be circular and each node has the same the maximum transmission radius.[3]

In this paper, without a loss of generality, we assume that the wireless channel is reliable so that packets are not lost during transmission. Note that the wireless channels in real sensor networks can be unreliable. Routing strategies for such lossy sensor networks have been proposed in [22,41]. In [22,41], the current node is assumed to know the packet reception rate of its neighbors and only those neighbors with reception rate larger than a threshold are considered for forwarding. Therefore the quality of the wireless link between current node and the next hop node is ensured. The routing protocols proposed in this paper can be extended for lossy wireless networks by setting appropriate threshold value as in [41].

---

[3] This assumption is required for the recovery mechanism in the proposed algorithms to work.

3.2. Power model and its effects on routing

Several metrics [45] have been proposed for measuring power consumption. The most commonly used metrics are energy consumption for each packet [38] and the system lifetime [4,25]. A wireless sensor node consumes power while (a) transmitting data, (b) while receiving, as well as (c) while being idle.

- *Power consumption during transmission of data:* During data transmission, the transmission power of the node is generally modeled as being distance-sensitive. In this paper, we assume the commonly accepted channel power model, $\rho = a \times \delta^{\gamma} + b$, where ρ denotes the transmission power, δ denotes the distance between the sender and the receiver, and γ is the power loss constant, typically between 2 and 4 [38,5,24]. In the power model, a and b are the distance-relative and constant terms of the power consumption. In several works, the constant term, b, is assumed to be negligible [5,54,25]. In [7,30], however, it is shown that in short distances b can be quite significant. In either case, each sensor node, v, has a maximum communication range, *range*(v). We call the set of nodes within this range the *neighborhood* of v and denote as $N_v(\subseteq S)$. The nodes in the network are able to adjust their transmission range depending on how far they need/choose to transmit.

  The power consumption for transmitting a packet is therefore $E = \rho \times t$, where t is the transmission time of the packet and t equals to the length of the packet divided by the channel data rate. In this paper, we assume constant packet size, therefore total energy consumption for transmission is proportional to ρ.

  Note that besides the power cost for data packets, additional power consumption in the network comes from transmitting and receiving the control packets generated by MAC layer, for example ACK packets. Although this packets are generally smaller than the data packets, they are also subject to similar power models.
- *Power consumption during the reception of data:* Sensor nodes also consume power during data *receiving* process; though the receiving power, l, while listening to the incoming data is constant (or distance-insensitive). Unfortunately, during a transmission, all nodes within the transmission range will *receive* the data (and thus consume power), even though the data might be addressed to only one of them. This makes the use of large radii for communication unpractical [8,10]. If a sender could adjust the transmission range based on how far it needs to transmit, instead of always using maximum transmission range, the number of nodes within its transmission range would not be larger than needed and therefore the total receiving power consumption would be reduced.
- *Power consumption during the idle and sleep times:* Sensor nodes have two different types of non-active states. In the *idle listening* state the nodes are not actively transmitting or receiving, but they nevertheless consume power. In fact, nodes can consume up to 50–100% of their receiving power during idle listening [10,7,53]. In the *sleep* state, the node essentially turns itself off and power consumption is negligible relative to the transmission, retrieval, and idle listening states.To reduce the idle listening state consumption, various, synchronized and asynchronous, sleep and wake-up based solutions have been proposed. In GeRaF [55], for instance, nodes may turn off to save power. When a node wants to transmit a packet towards a destination, it broadcasts a message in its entire radio range. Depending on the status of the nodes close to the target, zero or more active nodes will receive the message; hence the actual node that will receive is not known a priori by the sender, but rather is decided (probabilistically) after the transmission has taken place, according to nodes' own locations towards the destination. In Piconet [1], each node goes to sleep mode randomly and periodically wakes up. STEM [40], on the other hand uses two radios one for data and the other, low-duty-cycle mode radio, for waking-up other nodes. The transmitting node wakes up the receiver by sending a wake-up tone or a beacon. PAMAS [43] requires that nodes that are not able to receive and send packets to turn off the wireless interface. A separate control channel enables nodes to determine when and for how long to sleep. Other works on dynamically turning nodes to sleep mode to save power can be found in [12,15].

  In this paper, we assume that the sensor network implements a protocol that turns inactive nodes off to minimize their power consumption. Although we do not focus on the implementation of this orthogonal protocol, we note that since nodes are identified simply by their locations, the protocols we develop in this paper can easily be extended to a GeRaF-like model, using contention resolution and retransmission protocols as in [55]. We also note that multipath routing provides greater resilience when node availability is unpredictable.

Based on the above discussion, we see that while a node is transmitting, the total power consumption in the network will depend on the transmission power parameters of the sender (a, b, δ, and γ), the receiving power of the listener (l), and the any involuntary listeners in the network neighborhood of the transmitter (determined by the density, D, of nodes, as well as the range, δ, of transmission). In fact, as shown in Fig. 1, during multihop routing, not only those nodes that are actively participating to the route, but also those nodes that are close to these nodes will consume power: they will not be able to stay in the sleep state as they may involve in routes themselves and they may spend power as they involuntarily listen to the transmissions that are in their neighborhood. Let us consider a situation, where a source, s is sending a packet to a destination, d, in its range through an omnidirectional radio message. Let us denote the distance between s and d as δ. While s is communicating with d, the transmission power consumption by s is: $(a \times \delta^\gamma + b)$ and the receiving power consumption by the nodes in the neighborhood is $(\pi \times \delta^2 \times D \times l)$. Therefore, the total power consumption, $\rho(s,d)$, in its neighborhood will be

$$\varrho(s,d) = b + (\pi \times D \times l) \times \delta^2 + a \times \delta^\gamma.$$

Since, our aim in this proposal is to minimize the overall power consumption in the network (i.e., to improve the network lifetime), we will use $\varrho(s,d)$ as the power model for communication between s and d, instead of $\rho(s,d)$ which ignores the retrieval power consumption of the nodes within the neighborhood of transmission. Now, let us consider a situation, where a third node, c, is located exactly halfway between s and d, as shown in Fig. 2, this would give us

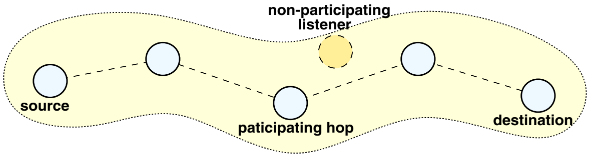


Fig. 1. The range of effected nodes during multihop routing.

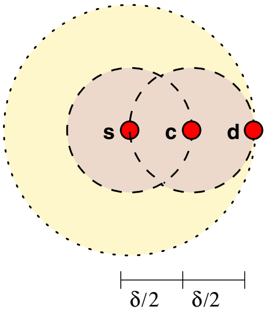


Fig. 2. The effects of using an intermediary node for routing: the number hops involved in communication increases; but the total coverage area (i.e., the number of neighbors affected) drops.

$$\begin{aligned}\varrho(s,c,d) &= \varrho(s,c) + \varrho(c,d) = 2 \\ &\quad \times \left[b + (\pi \times D \times l) \times \left(\frac{\delta}{2}\right)^2 + a \times \left(\frac{\delta}{2}\right)^\gamma \right] \\ &= 2 \times b + \frac{(\pi \times D \times l)}{2} \times \delta^2 + \frac{a}{2^{\gamma-1}} \times \delta^\gamma.\end{aligned}$$

Thus, we can compute the power consumption difference between the first alternative (forwarding the message directly to d using the maximum power range) and the second one (using a relay node c which requires a smaller range), as

$$\begin{aligned}\Delta\varrho_{\text{saved}} &= \varrho(s,d) - \varrho(s,c,d) \\ &= \frac{\pi \times D \times l}{2} \times \delta^2 + \frac{(2^{\gamma-1} - 1)}{2^{\gamma-1}} \times a \times \delta^\gamma - b.\end{aligned}$$

This suggest that even when b is comparable to $a \times \delta^\gamma$ (as reported in [30,7]), when a closer relay is used, the overall power consumption drops due to the savings $\left(\frac{\pi \times D \times l}{2} \times \delta^2\right)$ at the neighboring sensors who do not get to spend power to receive packets unnecessarily.

3.2.1. Distance adaption and MAC layer protocols

A network protocol which can adapt the transmission distance to obtain power savings requires extra support from the underlying MAC layer. Naturally, this has impact on the design of MAC level protocols and has effects on the performance.

In particular, the underlying MAC protocol has to be designed to support variable transmission radio ranges. Let us consider, for instance, the traditional 802.11 MAC protocol. To perform collision avoidance, traditional 802.11 MAC protocol requires RTS and CTS packets being transmitted before the actual data transfer. In a fixed range scenario, transmitting RTS and CTS using the maximum radio range solves the hidden terminal problem.

On the other hand, transmitting control packets RTS and CTS, introduces the *exposed terminal problem* decreasing the network utilization [17,20,39]. To reduce network contention, a distance-adaptive protocol would send the data, ACK, as well as the other control packets with just-enough power to match the desired transmission radio range. In fact, by reducing the transmission range (from the maximum possible range to a adaptively chosen smaller range), the scheme we present in this paper contributes to the reduction in network contention both for data and control packets used by the MAC layer. For example, in Fig. 2, when node s transmits directly to d, all other nodes in the dotted region would be refrained from transmitting data or control packets; however, when c is used as relay, only those nodes in one of the smaller dashed regions would need to be refrained. Therefore, more concurrent transmissions are possible within the same neighborhood.

For this reason, an adaptive MAC layer sends the RTS and CTS packets with the same radio range as the data and ACK packets would use. Note that conventional MAC protocols are designed such that if the RTS and CTS transmission ranges of *all* nodes in the wireless network are the same, then collisions can be avoided. However, in a sensor network, where the power of the nodes significantly vary from time to time, it is not realistic to assume that all the RTS and CTS packets can be sent with enough power to cover the same maximum range. Thus, collisions due to power variations are not avoidable. On the other hand, distance adaptive transmission of RTS, CTS, and data packets, as presented in this paper, significantly reduces the area of coverage needed for transmissions (i.e., network contentions) and provides protection against performance degradations.

3.2.2. Relayed routing and hop counts

Scaling down the transmission ranges (to save power) increase the number of hops that is required for end-to-end transmission. Of course, using a higher number of hops may increase the overall end-to-end transmission delay in the network. In this paper, we focus on the total power consumption rather than the end-to-end delay.[4] In particular, we see that under the right conditions, using relays will improve the network lifetime. We use this observation to develop our protocols.

3.3. Power efficient routing within the radio range

As discussed above, given a source, s, and a destination node, $d \in N_s$, within the neighborhood of s, the minimum energy consuming transmission strategy may either be to directly transmit the message to d or to use an intermediary node c to relay the message to d. The following example demonstrates this second case. Consider the scenario in Fig. 3. In this figure, node A wants to send a message to node D, which is in A's radio range. Let us assume that, based on its power status and its knowledge about the neighbors in its radio range, A discovers that the minimum energy path to D is $A \rightarrow B \rightarrow C \rightarrow D$ and forwards the packet to B (Fig. 3(a)). Next B finds out that the minimum energy path from B to D is $B \rightarrow C \rightarrow D$ and forwards the packet to C (Fig. 3(b)). C in turn forwards the packet to the destination D (Fig. 3(c)).

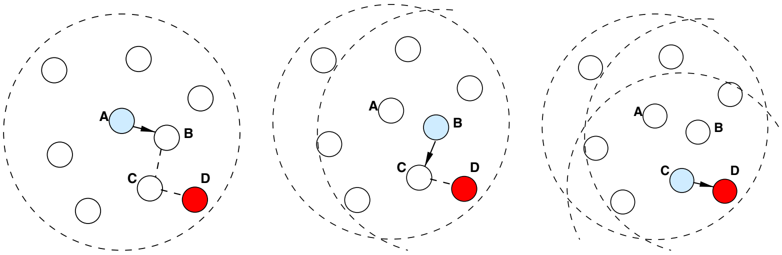
The task of the source node s, then, is to select the location of the next relay node in its radio range. When this neighborhood is sparse, it may be possible for each node to maintain a view of its neighborhood.[5] In this case, s finds the next relay node using a shortest path algorithm (where the edge weights correspond to the expected power consumption based on the power model $a \times \delta^{\gamma} + b$). Fig. 4 outlines the *RouteWithinNeigbors* algorithm for power efficient routing within the radio range.

Unfortunately, the neighborhood graph, $NGraph_s$, constructed by s in *RouteWithinNeighbors* may not reflect the actual topology of the nodes in V_s, since it is possible that the distance of two nodes, $v_i, v_j \in V_s$ exceeds their respective radio ranges. In this case, communication through the edge $\langle v_i, v_j \rangle \in E_s$ between these two nodes may be physically impossible. This problem can be addressed by letting s determine the power ranges of all neighbors at network initialization time and eliminating the invalid edges in E_s. If, on the other hand, the power range of the nodes in the network are constant and identical, such a corrective action is not

---

[4] On the other hand, note that using smaller-ranged transmissions will reduce network contention and, thus, may actually help improve the end-to-end delay.

[5] Since the number of nodes in the radio range of a given node is much smaller than the total number of nodes distributed in a wide-area wireless network, it may be possible for each node to learn and save the coordinates of its neighbors (N_v) at the system initiation phase.



(a) (b) (c)

Fig. 3. Routing within the radio range: *A* uses an intermediary node instead of directly transmitting to, *D*, which is in its neighborhood.

```
RouteWithinNeighbors($d$)
  • $s$ is the current node (the node in which this routine is executed at);
  • $d$, within the radio range of $s$, is the destination node
  • this routine identifies the next node on the minimum power consumption path from $s$ to $d$
{
  1. $s$ constructs a neighborhood graph $NGraph_s = (V_s, E_s, l)$ such that
       • all the nodes in $V_s$ are neighbors of $s$;
       • there is an edge in $E_s$ for every pair of nodes in $V_s$; and
       • for each edge $e \in E_s$, the label $l(e)$ gives the expected power consumption between the end nodes of $e$
  2. $s$ runs a shortest path algorithm on its neighborhood graph $NGraph_s$ with $s$ as source and $d$ as destination;
  3. the routine returns the next node on the path between $s$ and $d$;
}
```

Fig. 4. Algorithm used by node *s* to identify the next relay node on the path to node *d*, when the node *d* itself is within the neighborhood of *s*.

necessary as the minimum energy consumption path between s and d will never contain a physically impossible edge.[6] To see this, consider the following: suppose nodes, v_i and v_j, and the edge $\langle v_i, v_j\rangle$ appears in the minimum energy consumption path from s to d. If $\varrho(v_i, v_j) > \varrho(s, d)$, then s could transmit directly to d. If the number of nodes in the radio range is small, each node can use *small* routing tables for its immediate neighborhood. During initialization, each node runs the algorithm once for the small number of nodes in its neighborhood and saves the coordinates of the first relays for each one.

3.4. Power efficient routing outside the radio range

RouteWithinNeighbors is applicable when the destination is within the radio range of the current node. In order to leverage *RouteWithinNeighbors* when dealing with the more general case where destination node is beyond the radio range, we introduce the concept of *dynamically adjusted subdestination* nodes. Given a destination node, *d*, the source node, *s*, selects the node *u*, within its neighborhood, closest to *d* and closer to *d* than itself, as the subdestination. In a sense, this is similar to those greedy algorithms in [18,2,21] which try to minimize the number of hops that the packet takes by *greedily* choosing the neighbor maximizing progress towards the destination. On the other hand, in GPER, once the subdestination node, *u*, is identified, *RouteWithinNeighbors* can use local relay nodes to efficiently

---

[6] This assumption is required for planar graph routing in the recovery phase of GPER (Section 3.4.3).

deliver the packet to u instead of directly transmitting it to u. Furthermore, as described next, each local relay node may make dynamic adjustments on the subdestination; hence the packet may skip u altogether on its way to d.

3.4.1. Dynamic subdestination adjustment

In *RouteWithinNeighbors*, the source node sets up a *locally optimal* minimum power consumption path to its subdestination *assuming* that the packet will be forwarded through the nodes on the chosen path. On the other hand, *irrespective of the assumptions made by this node*, the next node acts independently and calculates a new subdestination based on its own power range and neighborhood. Therefore, rather than committing to a fixed subdestination until it is reached, each relay node makes adjustments and prevents costly deviations from the destination due to earlier misjudgments.

Fig. 5 shows an example. In this figure, the source, A, is trying to route a packet to destination G. For this purpose, it chooses a subdestination, D in its radio range, and a path $A \rightarrow B \rightarrow C \rightarrow D$ to this subdestination. It then forwards the packet to the next hop on this path, which is B (Fig. 5(a)). Once it receives the packet, B considers its own neighborhood and establishes F as its own subdestination for reaching G (Fig. 5(b)). B then chooses the path $B \rightarrow E \rightarrow F$ and routes the packet to E instead of to C as was assumed priorly by A. This dynamic subdestination selection process will continue as E considers its own neighborhood for the next step and the packet may never need to reach the node D chosen as the subdestination by the source node A.

3.4.2. Forced routing to prevent infinite loops

As described above, after s identifies its subdestination u to reach d, it calculates a low power route to u using *RouteWithinNeighbors*. In most cases, the next relay node r on the resulting path will be closer to the destination d than s. However, if r is further from d than s, forwarding the packet to r may lead into a loop. Fig. 6(a) and (b) illustrates how this can happen: in Fig. 6(a), B is closer to D than A. First, A chooses C as the subdestination to D and forwards the packet to B; then B chooses F as the subdestination to D and forwards the packet back to A, since A is on the minimum energy consumption path from B to F. This causes an infinite loop between node A and B.

In order to guarantee that the routing algorithm is free of such infinite loops, in GPER we introduce a *forced routing* mechanism as a preventive measure. Forced routing is applied when a potential loop is identified: let us assume that s

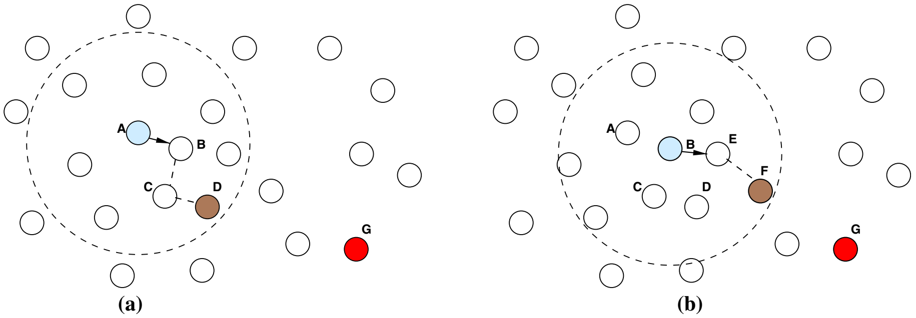


Fig. 5. Dynamic subdestination selection: (a) A chooses D as its subdestination, (b) at the next step, B chooses F as its subdestination.

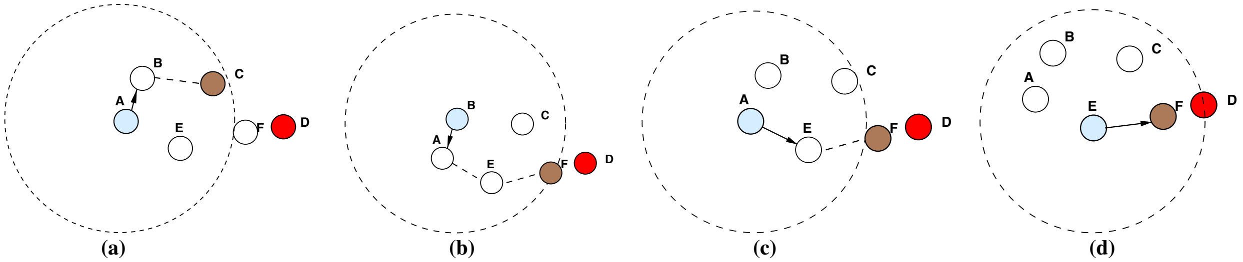


Fig. 6. (a, b) A loop due to dynamic subdestination selection, (c, d) the force routing solution.

is trying to route a packet to the final destination d, and it identifies the subdestination u in its neighborhood. Furthermore, let us assume that s decides to route the packet to r on a low power path to u. If it turns out that the next hop r is further from the final destination d than s itself, s declares a *forced routing* status. Under forced routing, the subdestination u is kept fixed until it is reached. Furthermore, the *RouteWithinNeighbors* is augmented to constrain the path such that $\delta(r,u) < \delta(s,u)$, i.e. r is closer to u than s. The augmented version of the in-neighborhood routing algorithm is called $\mathit{RouteWithinNeighbors}_{\mathrm{forced}}$.

It is possible that a given packet reaches the same node more than once even when the forced routing scheme described above is used. Each such loop is *temporary*. Consider Fig. 6 once again. In Fig. 6(b), when B realizes that A is further from the destination D than itself, it marks the packet as in force routing mode and fixes the packet's subdestination to F. After this, B may still forward the packet back to A, since A is closer to the newly fixed subdestination F than B. This time, on the other hand, A will recognize that the packet is in force routing mode with fixed subdestination F. Therefore, this time A will decide to use E instead of B as the next hop (Fig. 6(c)). Fig. 6(d), then, shows how the packet proceeds towards the fixed subdestination F. Therefore, loops of the form $A \rightarrow B \rightarrow A$ are temporary and the packet will approach its destination.

To show that the result is indeed free of infinite loops, we need to show that the number of times each packet is forwarded is finite. Let us define *ForceRoutingRun* as the period which starts when the packet enters force routing mode and ends when the packet reaches the corresponding fixed subdestination. During a *ForceRoutingRun*, each forward will bring the packet closer to the fixed subdestination, so the number of forwards in each *Force- RoutingRun* is finite. Assuming that there is a subdestination closer to the final destination, the total number of forwards will be finite. Therefore, forced routing guarantees that the path generated by GPER is free of infinite loops when the identified subdestination is closer to the destination than the current node. In an arbitrarily structured wireless network, however, there may be cases in which there may not be a suitable subdestination (Fig. 7). In next subsection, we examine this.

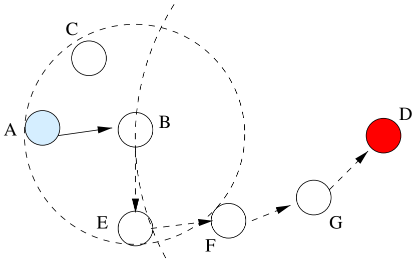


Fig. 7. The lack of a suitable subdestination at B.

3.4.3. Planar perimeter with forced routing

So far we assumed that each node can identify a proper subdestination within its neighborhood; however, there are scenarios in which no neighbor is closer to the destination than the current node; i.e., there are no suitable subdestinations. For example in Fig. 7, node B realizes that all of its neighbors $\{A, C, E\}$ are further than itself to the destination D. Such scenarios have been extensively studied in [18,2,21] and planar perimeter routing is introduced to gradually forward the packet to a node closer to the destination than the current node. When there is no suitable subdestination, under perimeter routing, each packet traverses the graph of the network using a right-hand rule, which requires that if the node visited before B was A, then the next edge to traverse is the first counter-clockwise edge about B from edge (A, B). In Fig. 7, B would choose C as the next node to follow. To ensure that routes will be found when they actually exist, the graph of the network is planarized before the next edge to traverse is calculated [18,2,21]. Note that when no node in the network is located closer to the destination, the edge where face traversal was started is traversed twice in the same direction and the message will be dropped.

In GPER, we adopt the planar perimeter routing approach presented in [18,2] to tackle the scenarios when no suitable subdestination is available. As with [18,2], we requires that all sensor nodes in the network have circular communication radio ranges and the same transmission radius. On the other hand, to save power, we implement perimeter routing approach through *RouteWithinNeighbors* which may use intermediary relay nodes when this helps reduce the power consumption. However, to ensure that *RouteWithinNeighbors* implements perimeter routing without making dynamic adjustments which may destroy the overall counter-clockwise progression of the perimeter routing approach, we also use forced routing along with *RouteWithinNeighbors*.

Consequently, when the next hop for planar perimeter routing is determined (for instance *C* in the above example), the packet is also marked to be in force routing with this fixed subdestination (i.e., *C* will be the fixed subdestination until it is reached). The proof that there are no infinite loops under planar perimeter mode extended by force routing follows the fact that original planar perimeter routing has no infinite loops [18,2].

3.4.4. GPER protocol

Fig. 8 presents the Geographic Power Efficient Routing (GPER) protocol which takes all the above issues into account. An important aspect of GPER

```
$RoutePacket_s(pack, d)$
    • $s$ is the current node (the node in which this routine is executed at);
    • $pack$ is the packet being routed
    • $d$, outside of the radio range of $s$, is the destination node
    • this routine routes $pack$ to the next node on the minimum power consumption path
      from $s$ to $d$
{
    1. $s$ finds the neighbor that is closest to $d$ and save it to $n$;
    2. if (ForcedRouting flag is on for $pack$)
        (a) $s$ extracts the Subdest of $pack$ and save it to $n$;
        (b) if ( $s == n$)
            i. $s$ clears the ForcedRouting flag for $pack$;
    3. if (ForcedRouting flag is off for $pack$)
        (a) if (PeriRouting flag is on for $pack$)
            i. extract PeriStartAt from $pack$
            ii. if($s$ is closer to $d$ than PeriStartAt)
                A. clear both PeriRouting and ForceRouting flags for $pack$
            iii. else
                A. set ForcedRouting flag on for $pack$
                B. calculate next node for planar perimeter routing and save it to $n$
                C. $s$ sets $n$ to be the subdestination, Subdest, of $pack$;
        (b) else if ($n$ is further than $s$ to $d$)
            i. set both PeriRouting and ForcedRouting flags on for $pack$
            ii. set PeriStartAt field in $pack$ to $s$
            iii. calculate next node for planar perimeter routing and save it to $n$;
            iv. $s$ sets $n$ to be the subdestination, Subdest, of $pack$;
    4. if(ForceRoutingflag is on for $pack$)
        (a) $NextHop = RouteWithinNeighbors_{forced}(n)$;
    5. else
        (a) $NextHop = RouteWithinNeighbors(n)$;
    6. if (ForcedRouting flag is off for $pack$ and $NextHop$ is further than $s$ to $d$)
        (a) $s$ sets ForcedRouting flag on for $pack$;
        (b) $s$ sets $n$ to be the subdestination, Subdest, of $pack$;
        (c) $NextHop = RouteWithinNeighbors_{forced}(n)$;
    7. $s$ forwards the packet to $NextHop$;
}
```

Fig. 8. GPER packet routing algorithm used by node *s* to identify the next relay node on the path to node *d*. In this case, the node *d* is not within the neighborhood of *s*.

is that this performance is achieved with only local information; that is, the coordinates of the neighbors are the only information needed. The overhead of GPER is the same as GPSR since GPER is using similar local information and forced routing in GPER does not incur any additional power consumption overhead.

4. Multipath routing with alternatives

As discussed in Section 2, routing algorithms, power-aware or not, choose and reuse paths between end-points and thus may potentially deplete resources, especially if two end-points communicate for an extended period of messages. To cope with this challenge within the context of GPER, in this section, we introduce and investigate three geographic multipath routing algorithms (MGPSR, $\mathrm{MGPER}_{\mathrm{sub}}$, and $\mathrm{MGPER}_{\mathrm{hop}}$), which generate alternative paths probabilistically based only on local information. These algorithms identify and use different routes for different messages between a source and a destination. As a result, as shown in Fig. 9, they take away from the load on the nodes that are on the preferred path between the end-points and distribute this load on other nodes that are in the neighborhood. Of course, as these paths are less perfect than the preferred path, they are expected to introduce a degree of *overhead*, but experiments show that multipath routing improves the lifetime of the network, especially in cases where the communication is non-uniformly distributed in the network (i.e. some source–destination pairs communicate more often than others). Probabilistic selection of paths has been used along with traditional and on-demand routing algorithms for improving the resilience of the network to failures [52]. As we discussed in the previous section, on the other hand, geographic routing with localized decision making is more suitable than on-demand routing for sensor networks. Existing related works, such as [14], do not consider power efficiency and the effect of probabilistic routing on network lifetime.

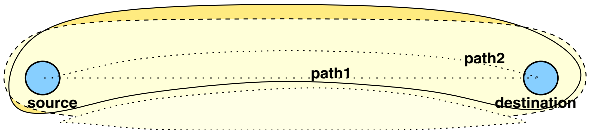


Fig. 9. Multipaths between a single pair of source and destination.

4.1. *k-MGPSR: multipath GPSR with k alternative subdestinations*

Following the ideas of GPSR [18], we first introduce k-MGPSR. Unlike previous multipath algorithms, k-MGPSR does not establish multiple alternative paths *before* actual packet routing; instead, when a node needs to forward a packet, it probabilistically chooses the next node. In the original GPSR algorithm, each node chooses the neighbor that is closest to the destination as the next hop. In k-MGPSR, however, each node identifies k alternative next hops as the k closest neighbors to the destination, and probabilistically selects one from these to be the next hop. Consequently, in k-MGPSR, the nodes used as relays are not the same for different messages routed between the same pair of source and destination. In this way, k-MGPSR generates braided multipaths.

Fig. 10 shows an example, where node S wants to send a packet to node D. Fig. 10(a) shows that when GPSR is used, the message always follows the path $S \to H \to L \to D$. Fig. 10(b) shows that when using k-MGPSR (k is assumed to be 3 in MGPSR), the source node S has three choices, $\{F, G, H\}$, for the next hop and it will select one of these as the next hop randomly. Fig. 10(c) shows the case after S forwards the message to H; at this point, H has three choices for its next hop: $\{K, L, J\}$. Fig. 10(d), on the other hand, shows the alternative case where node S has forwards the message to node G instead of H; in this case, due to its location G has two choices for its next hop. Overall, the resulting alternative routing paths include among others: $S \to H \to L \to D$, $S \to H \to K \to D$, $S \to G \to J \to M \to D$.

In k-MGPSR, each node uses a discrete probability distribution function for choosing a node from the k alternatives. If a uniform probability distribution is used, a node would select each one of the alternatives using the same probability value: $1/k$. If a normal distribution is used, higher probabilities are given to those nodes that are closer to the destination. For example, when k is equal to 3, the probabilities for choosing the three alternatives, based on their distances to the destination, are 0.55, 0.31, and 0.14, respectively. Obviously, when k becomes larger, more neighbors are contained in the alternative set, increasing the number of alternative paths. However, when k is large, the message may depart further from the direction of source to destination, wasting power resources along the way.

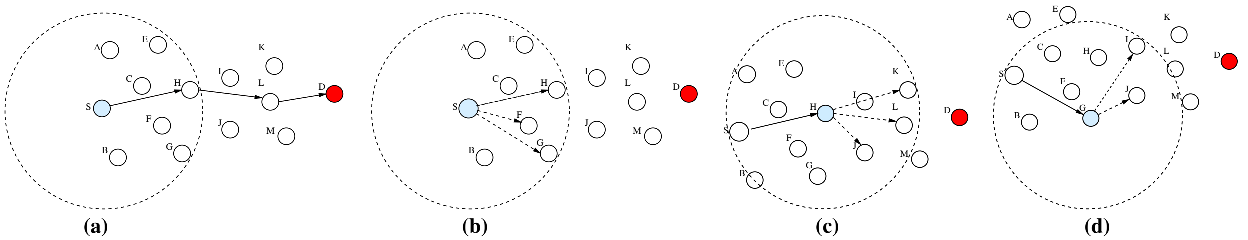


Fig. 10. Example for GPSR and MGPSR.

4.2. *k-MGPER$_{sub}$: multipath GPER with k alternative subdestinations*

As described above, in GPER each node along the path first chooses a subdestination as in GPSR, but then it chooses an intermediate node between itself and the subdestination as the relay and forwards the message to this relay. Thus, in GPER, the subdestination or the next hop can be chosen among alternatives. In this section, we introduce *k*-MGPER$_{\text{sub}}$, which selects its subdestination among alternatives and then computes the next hop relative to this subdestination.

Similar to *k*-MGPSR, in *k*-MGPER$_{\text{sub}}$ the current node calculates *k* closest neighbors relative to the destination as multiple alternative subdestinations. The current node then probabilistically select one from these alternative subdestinations. As in GPER, based on the selected subdestination, *k*-MGPER$_{\text{sub}}$ chooses a node within its neighborhood as the next hop. Since the subdestination and the corresponding alternative next hops may be different each time messages arrive on the same node, different messages may take different paths toward the same destination, resulting in braided multipaths.

Fig. 11(a) shows that GPER always uses the same path, $S \to C \to H \to I \to L \to D$, to forward messages from node S to node D. In Fig. 11(b) where 3-MGPER$_{\text{sub}}$ is used, node S has three 3 alternative subdestinations: $\{H, F, G\}$ whose next hops are $\{C, F, B\}$ respectively. Node S will probabilistically select one from $\{C, F, B\}$ as the actual next hop. Fig. 11(b) shows the case that node S selected C as its next hop. Node C calculates its alternative subdestinations to be $\{H, I, J\}$ whose corresponding next hops are all node H. Fig. 11(d) shows the case that node S has selected F as next hop and at node F the alternative subdestinations are $\{H, I, J\}$ and the corresponding next hops are $\{H, J\}$. The alternative routing paths from S to D generated using 3-MGPER$_{\text{sub}}$ include among others: $S \to C \to H \to I \to L \to D$, $S \to F \to J \to M \to D$, $S \to F \to H \to I \to K \to D$.

4.3. *k-MGPER$_{hop}$: multipath GPER with k alternative next hops*

k-MGPER$_{\text{hop}}$ differs from *k*-MGPER$_{\text{sub}}$ in that, in *k*-MGPER$_{\text{hop}}$ the subdestination is simply fixed to be the neighbor that is closest to the destination. The alternatives are introduced while identifying the next hop towards this subdestination. As described above, the next hop in GPER is the second node on the path of minimum power usage from the current node to the subdestination. Such a path is calculated by a node by running a shortest path algorithm *locally* (within the immediate neighborhood constructed by the current node and its neighbors). In MGPER$_{\text{hop}}$, the first alternative next hop is the next hop identified by GPER. To calculate an alternative next hop, this node is removed from the immediate neighborhood graph and the shortest

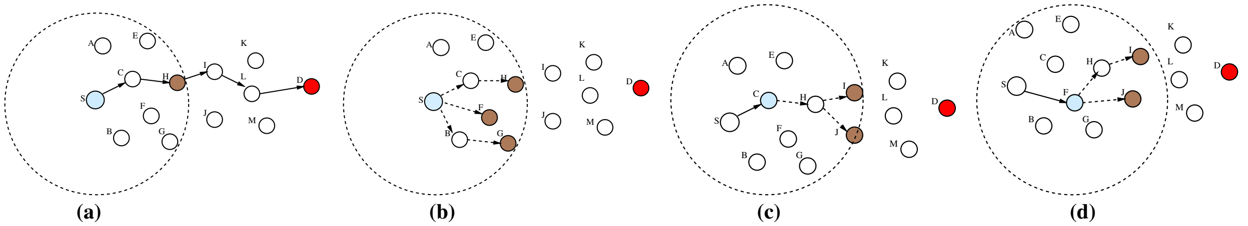


Fig. 11. Example for GPER and MGPER$_{\text{sub}}$.

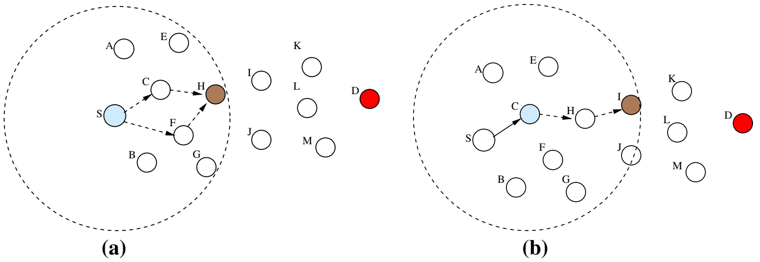


Fig. 12. Example of $\text{MGPER}_{\text{hop}}$ routing.

path algorithm is reapplied (within the immediate neighborhood). This continues until k alternative next hops are obtained or the subdestination becomes an alternative. Within the set of alternative next hops, the current node probabilistically picks one to be the next hop.

Fig. 12 shows how k-$\text{MGPER}_{\text{hop}}$ works. In Fig. 12(a), the subdestination at node S is node H, and using $\text{MGPER}_{\text{hop}}$ S calculates the alternative next hops to be $\{C,F\}$. Fig. 12(b) shows the case where node S has selected node C as the next hop. The alternative routing paths generated using $\text{MGPER}_{\text{hop}}$ include $S \to C \to H \to I \to L \to D$, $S \to F \to H \to I \to L \to D$, $S \to C \to H \to J \to L \to D$, and $S \to F \to H \to J \to L \to D$.

Note that in the three multipath routing algorithms (k-MGPSR, k-$\text{MGPER}_{\text{sub}}$ and k-$\text{MGPER}_{\text{hop}}$) discussed above, the criterion for calculating the k alternatives is the distance from the neighbor to the destination. Other criteria are also possible. In the next subsection, we will investigate another related approach for multipath routing.

4.4. Angle constrained routing

One risk of using a probabilistic next hop selection is that paths may deviate too much from the optimal path, thus consuming excessive power resources. In angle constrained routing, to prevent the alternative paths from deviating too much from the destination, we place a constraint on the maximum allowed deviation angle when choosing the next hop: given a current node S and its neighbor set N_S, we define a deviation_angle of $N \in N_S$ relative to a point P as the cross angle of the line segments $Line(S,N)$ and $Line(S,P)$. Within a given deviation_angle constraint, the alternatives are further constrained to be closer to the destination than the current node to prevent loops on the routing paths. This could help if some of alternatives may deviate significantly from an optimal path, wasting resources.

In hybrid $\text{MGPER}_{\text{hop}}$, the subdestination is fixed before the alternative next hops are chosen and next hops are constrained based on their distance; thus, a separate angle constraint may not be necessary. However, in $\text{MGPER}_{\text{sub}}$ routing, the subdestinations themselves are chosen among alternatives, some of which may deviate significantly from an optimal path. Thus an angle constraint (rather than or in addition to a constraint on the number of alternatives) may be more desirable. Yet, when additional angle constraining is used, the number of alternative next hops may reduce, so angle constraining may hurt the performance when the alternatives paths were not significantly deviating from optimal. Thus, it is important to make sure that deviation_angle is properly chosen and is not over-constraining.

The deviation_angle may either be constant or may vary based on the position of the node relative to the destination. When the current node is near to the destination, the range can be relatively large since the message will not deviate much from the destination in short distance. When the current node is far from the destination, the range should be small to constrain the deviation of the message from the destination. Thus a natural formula for computing the deviation angle range, $\alpha \leqslant \pi/2$, is

$$\alpha = \arctan\left(\frac{Max\_Dev}{Distance(S,D) - Radio\_Range}\right),$$

where S is current node, D is the destination node, *Radio_Range* is the radio range of the current node, and *Max_Dev* is the maximum projected deviation from the destination. We note one possible disadvantage of the above formula would be that, the deviation angles farther away from the destination would be extremely tightly constrained. Therefore it would be impossible to benefit from potentially useful alternatives, when further away from the destination. Since the nodes closer to source and the destination are likely to drain faster, this may not be desirable, especially in cases where one node always acts as a source and the other node always acts as a destination. In such cases, a minimum on the deviation angle may be need. Here, we do not further investigate how the deviation angle should vary with the distance from the destination.

4.5. Multipath routing with additional information

It is intuitive that, if additional information (such as network topology, residual power, and message flow patterns in the network) are available, good alternative paths are easier to select. In general, the overhead of collecting and maintaining such global information is likely to be unacceptable. However, collecting such information *within* the neighborhood may be possible. In this section, we investigate the cases when the residual power of the neighbors can be considered when choosing a relay among available alternatives.

4.5.1. Residual power of neighbors is recorded by each node

One way to obtain the residual power of the neighbors is to require each transmitted packet to include a value, indicating the residual power on the transmitting node. Each receiving node would record the residual power value for the source node and use that when choosing a relay. Given estimates of the residual powers of the neighbors, instead of being selected probabilistically, the next hop will the one with maximum residual power among the alternatives. Of course, these are only rough estimates, as nodes also consume power while receiving packets and how much power a node has consumed since the last time it transmitted a packet cannot be known.

4.5.2. Residual power of neighbors is not maintained

It is also possible to select the neighbor with highest residual power, without recording the residual power information of all neighbors [13]. The idea is that the sender does not determine the receiver, but instead neighbors compete with each other to be the actual packet receiver based on their residual power. One possible way to achieve this is that after receiving the RTS control packet, each neighbor sets its own CTS timer, whose timeout value is inversely proportional to its own residual power [13]. This ensure that the neighbor with highest residual power will send out the CTS control packet first.

By choosing the alternative with maximum residual power as the next hop, it is expected that no node whose power reduces much faster than the others and the power consumption is distributed more evenly across the network. As shown in the experiment section (Section 5), this leads to improved network lifetime.

Note that in the multipath algorithms proposed in this paper, only the selected k alternatives need to compete with each other based on their residual powers. We achieve this by including the IDs of the alternative nodes in the RTS packet.

4.6. Multipath routing combined with planar routing

As in GPER and those algorithms in [18,2,21], if there is no neighbor closer to the destination than the current node, the message enters the perimeter mode. In perimeter mode, the message will traverse around the network in a strictly defined way guaranteed to arrive at a node that is closer to the destination if such a node exists in the network. Then the message proceeds normally towards the destination. When there is no node in the network closer to the destination, the edge where perimeter routing was started is traversed twice in the same direction and the message will be dropped.

5. Evaluation

In this section, we present results obtained through a (modified) NS2 simulation to validate the efficiency and effectiveness of GPER and the probabilistic multipath routing algorithms, presented in this paper. The simulation setup used in this section is described in Table 1. The MAC802.11 protocol and the wireless interface has been modified to enable variable transmission ranges. In the experiments, the MAC address of each node is known by

Table 1
Simulation setup

Sim. parameter	Value
Simulator	ns-2.27
Network size	1000 m × 1000 m
Number of nodes	1000
Channel data rate	11 Mbps
Mac protocol	Mac802.11
Transmission power	1.3 W
Receiving power	0.9 W
Initial energy of nodes	1 J
Message size	128B
Antenna	OmniAntenna

all nodes in its neighborhood, so that no ARP packets are generated at the link layer during message transmission. To achieve this, initially each node broadcasts its MAC address within its neighborhood using maximum transmission range. The power cost for broadcasting location information is omitted since we are assuming static sensor networks and the power cost for collecting location information is only during the time of network initialization. Moreover, the power cost for tracking neighboring nodes in GPER is the same as the other geographic routing protocols, such as GPSR.

In this paper, we focus on transmission and receiving power consumption and we assume that there exists a separate protocol to minimize the idle listening power consumption [1,40]. The power consumption model used in the experiments is described in detail in Section 3.2. However, to account for the worst case scenario, where the transmitting node's power consumption ($a \times \delta^{\gamma} + b$) is dominated by the constant b, we assume that the transmission power is $\sim b$. Thus, the savings reported in this section, are mainly due to the reduction in the number of receivers and overall receiving power consumption. *Naturally, the savings would be even larger in more common cases where the transmission power consumption is distance-sensitive.*

5.1. Experiments results for GPER

We first conducted a set of simulations to verify the GPER in networks with three different kinds of node density distributions: uniform, lightly non-uniform, heavily non-uniform.

5.1.1. Routing in networks with uniform sensor densities

First, 1000 sensor nodes are distributed in the network in an uniformly random way. To observe different network densities, we varied the maximum transmission range of sensor nodes between 67 m and 150 m. A larger radio range means a network that is denser, in the sense there will be more nodes in each node's neighborhood. We routed 100 messages from random sources to random destinations and calculated the average power consumption by using two geographic routing protocols GPSR and GPER. The optimal power consumption is also calculated using an idealized shortest path algorithm (which assumes full knowledge of the network) and is included here for comparison.

The average power consumption for messages using GPSR and GPER is shown in Fig. 13(a). *OPT* refers to the average power consumption obtained by the idealized optimal routing algorithm, which assumes that each node has the complete knowledge of the network topology. Therefore by OPT, each node can calculate the optimal path to any destination node. From the figure, we can see that when using GPER, the average power consumption across different transmission ranges is significantly reduced. We also note that as the maximum transmission radio range of the nodes becomes larger (i.e., as the density of the network increases), the power consumption of GPSR suffers significantly whereas GPER does not suffer at all. In fact, GPER even sees some improvement as there are more alternatives to choose from. The fact that GPER reduces the overhead of the neighbors can be observed in Fig. 13(b), which shows the number of nodes participating to the paths as well as the number of all (participating and non-participating) listeners. In Fig. 13(b), the transmission range of each node is 100 m. We can see that GPSR has the least number of hops on the chosen paths, but it also has most listeners. By GPER, although the number of hops is increased, the transmission range is reduced at each hop. The

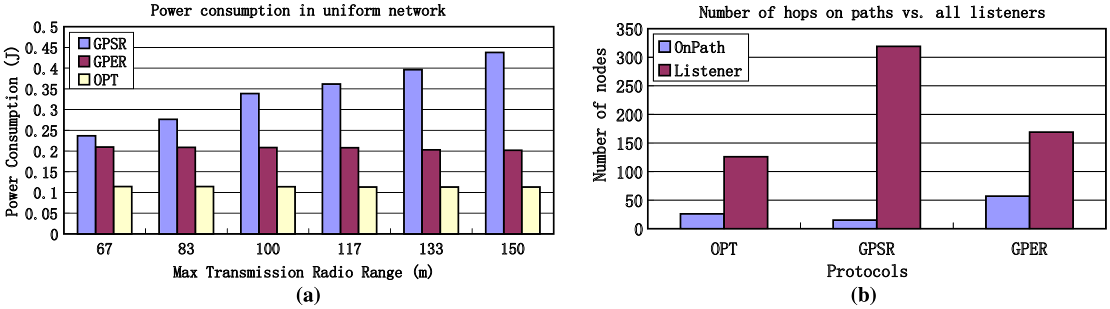


Fig. 13. Comparison of (a) routing power consumption for uniform sensor distribution and (b) the corresponding numbers of actual relay nodes and non-participating listeners when the transmission range is 100 m.

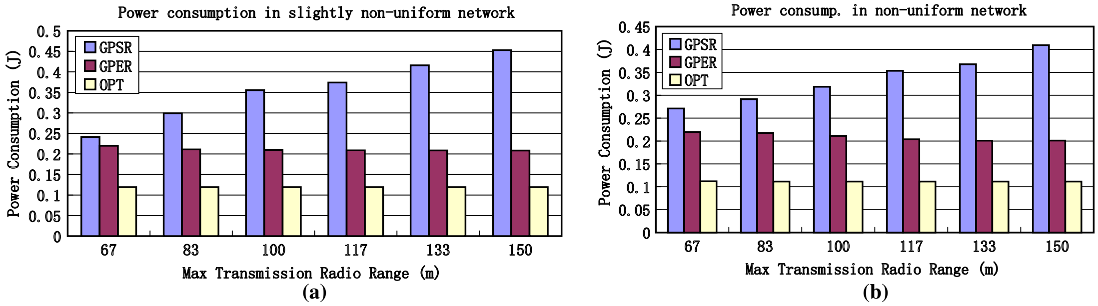


Fig. 14. Power consumptions for (a) slightly and (b) heavily non-uniform sensor distribution.

effect of reducing the number of listening nodes due to smaller transmission radio range at each hop exceeds the effect of increased number of hops. Therefore, GPER provides a good balance between the transmission range and the number of hops on paths, without requiring global information.

5.1.2. Routing in non-uniform networks

In order to observe the effect of slight non-uniformities in the network, we divided the network into 25 cells of size 200 m × 200 m and distributed the 1000 nodes, such that 12 randomly chosen cells received 54 nodes and the other 13 cells received 27 nodes. Fig. 14(a) shows the comparison of average power consumption to route 100 from random sources to random destinations in a slightly non-uniform network. Fig. 14(b) on the other hand, shows the power consumption for a sample situation, where dense cells receive 4 times the number of nodes than sparse cells, while the total number of nodes in the network remains 1000. These results (along with Fig. 13 for uniform distribution of nodes) shows that GPER scales well to various node distributions. GPSR also adapts to density differences, but not as well.

5.2. Experimental results for multipath geographic routing algorithms

In this section, we present experimental results to test and contrast the performance of the probabilistic multipath routing algorithms presented in this paper under different situations. The specific setups used for individual experiments are detailed below along with evaluation of the impacts on the performance.

5.2.1. Characteristics of routing paths

Before studying the impacts of the various settings on the network lifetime directly, we conducted a set of experiments to investigate the characteristics of routing paths obtained using different probabilistic multipath routing algorithms. For this purpose, we uniformly distributed the 1000 nodes in the network. The maximum transmission radio range for each node is set to 100 m. The alternatives are calculated based on the distance from neighbor to the destination and the number of alternatives considered, k, is set to 3. We routed messages from the bottom-left corner to the top-right corner of the space 100 times. Figs. 15 and 16 show various properties of multipaths compared with the properties of the paths obtained using single path algorithms. *OPT* refers optimal paths. Once again, only for comparison purposes, optimal paths are calculated using global information.

Average power consumption: Fig. 15 shows the average power consumption of different routing algorithms. In this figure, we first see that GPER family of algorithms consume significantly less power than GPSR family of algorithms. Since the multipath versions of the algorithms are using alternatives that are not necessarily best, the average power consumption of multipath algorithms could be higher than their single path versions.

Interestingly, the power consumption of multipath algorithms is very close to their corresponding

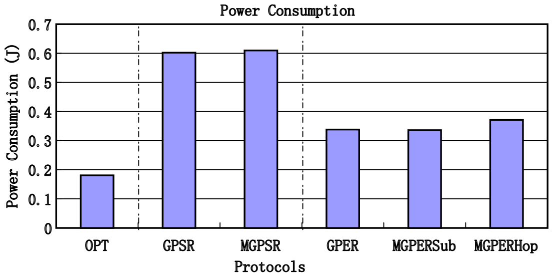
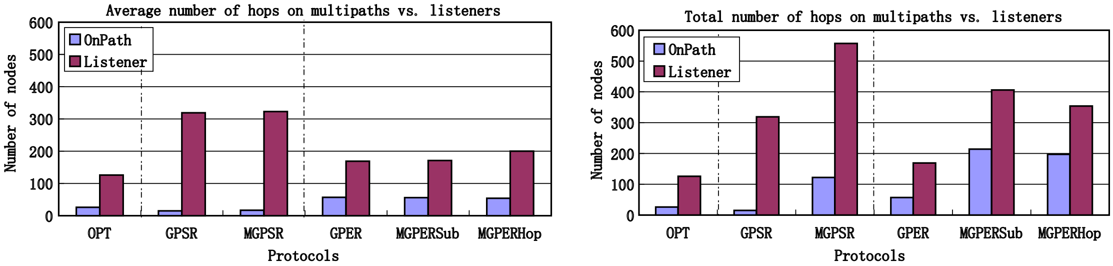


Fig. 15. Average power usage of different algorithms.



(a) **(b)**

Fig. 16. (a) Average number of hops vs. average number of listeners and (b) total number of nodes involved as hops and listeners.

single path algorithms. In fact, $\text{MGPER}_{\text{sub}}$ even consumes slightly less power than GPER. These can be understood more clearly by studying the number of listeners involved in the paths (Fig. 16(a)). Here, we can see that the average numbers of hops on multipaths and listeners are very close to their corresponding single path versions, which indicates that average power for transmitting messages and average power for receiving messages in multipath algorithms both are very close to the values by their respective single path algorithms. The fact that using multipath algorithms may reduce average power consumption highlights the fact that the GPSR and GPER are already suboptimal and that using close alternatives does not cause much penalty in average power consumption.

Delay: Again, Fig. 16(a) shows that the average numbers of hops on multipaths are very close to their corresponding single path versions. Therefore, the overall delay of multipath routing is as good as the single path version of the algorithms.

Load distribution: Fig. 16(b) shows the total number of nodes that are involved in the 100 routing tasks as either hops or listeners. As expected, the total numbers of nodes on multipaths acting as hops or listeners are larger than their single path counterparts. Thus, in multipath algorithms per message power consumption (which is mostly the same as the power consumption of the singlepath algorithms, Fig. 15) is distributed to more nodes in the network. Thus, the average power consumption on individual nodes is reduced and multipath algorithms are likely to achieve better network lifetime.

5.2.2. Network lifetime

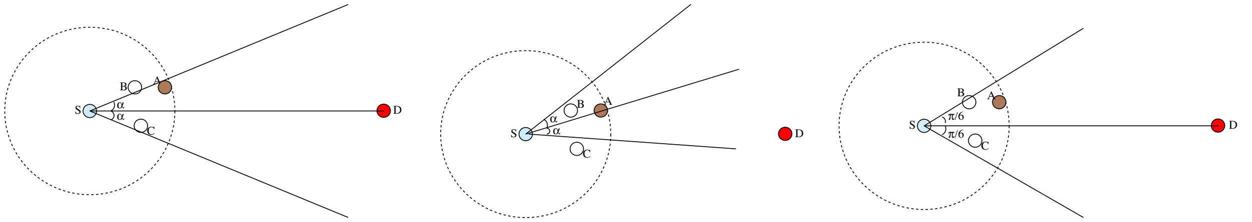
To observe the impact of multipath routing on the network lifetime in detail, we conducted a separate set of experiments. In this section, we use three different definitions for network lifetime [42,49]:

- *Time to number of dead nodes exceeds a given value:* Under this definition, a given network's lifetime is the duration of time until the number of dead nodes in the network exceeds a specified value; a node is declared inactive/dead when its power level drops below that is required to route messages to its neighbors.
- *Time to first routing task failure:* When the routing tasks are non-uniformly distributed, nodes closer to the few end-points are more critical than the rest of the nodes. Thus, the active node ratio (which measures the ratio of the active nodes in the entire network) may not the be the right network lifetime indicator. Thus, we also measured the network lifetime in terms of the duration of time until the routing task failure.

For each family of multipath routing algorithms introduced in this paper, we also experimented with different next hop selection criteria:

- *Dist/Prob(D/P):* Alternatives are calculated by distance criteria and the selection among alternatives is probabilistic (Sections 4.1–4.3).
- *Angle:* Alternatives are angle constrained (deviation_angle with distance) and the selection among alternatives is probabilistic (Section 4.4). We experimented with three angle constraining schemes:
  - *A/P:* Angle range is computed relative to the final destination and constrained around it.
  - *AS/P:* Angle range is computed relative to the final destination, but constrained around the subdestination chosen at each step.
  - *AC/P:* Angle range is constant ($\pi/6$) around final destination.

Fig. 17 shows an example for each of the angle constraint discussed above. In Fig. 17(a), nodes $\{A, C\}$ are the alternatives under the A/P constraint. In Fig. 17(b), nodes $\{A, B\}$ are the alter-



(a) A/P **(b)** AS/P **(c)** AC/P

Fig. 17. Angle constrain based routing examples: S is the source node; D is the destination node; A is the subdestination selected by GPER. Alternatives (a) and (b) are using the angle, α, chosen by the angle-constrained routing scheme discussed in Section 4.4; whereas (c) is using a constant angle, $\pi/6$.

natives under the AS/P constraint. In Fig. 17(c), nodes $\{A, B, C\}$ are the alternatives under the AC/P constraint.

- *Dist/Res(D/R):* Alternatives are calculated by distance criteria and the one with max residual power is always selected. In our experiments, we applied the MAC competition technique as discussed in Section 4.5.2 to achieve this.

We uniformly distributed the 1000 nodes in the network. The maximum transmission radio range for each node is set to 100 m. We repeated each experiment, for two k values (small – 3 and large – 10). The initial energy for each node is set to 1 J and a new message is generated every 10 s in the network (i.e., the lifetimes reported here are in terms of *#_messaging_tasks* × 10 s). Each experiment is repeated 50 times. The results are presented next.

Multipath routing is not good for uniformly distributed routing tasks: Table 2 shows the average network lifetime, where routing tasks are uniformly distributed in the network. The first part of the table is based on time to first task failure criterion, while the second part is based on the number of messages it takes to have 10 nodes that are power-dead:

- We see from these two tables that, when routing tasks are uniformly distributed in the network, multipath routing algorithms do not improve the network lifetime significantly. In fact, they may slightly hurt the lifetime as they use suboptimal path for load distribution.
- In general, network lifetime for uniform routing tasks is worse when k is large. Since the tasks are already uniformly distributed in the network, each node is used about equally, even without the help of multipath routing. Thus, to save power for each node, each path should simply use least power. When k is larger, due to inherent suboptimality of multipath routing, on the average, paths will cost more power, so the network lifetime decreases.

Multipath routing significantly improves the network lifetime for non-uniformly distributed routing tasks: In many real situations, however, the commu-

Table 2

As expected, multipath routing **is not good** for uniformly distributed routing tasks (the best solutions in their classes are shown in bold, the best overall solution is underlined)

	Time before first task fail				Time before 10 dead nodes			
	GPSR 808	GPER 866			GPSR 850	GPER 1066		
	MGPSR	MGPER$_{\text{sub}}$		MGPER$_{\text{hop}}$	MGPSR	MGPER$_{\text{sub}}$		MGPER$_{\text{hop}}$
D/P/3	791	803	>	748	828	**1006**	>	976
D/P/10	781	**805**	>	756	818	986	>	888
A/P/3	776	**866**	>	832	834	**1002**	>	978
A/P/10	740	828	>	795	841	965	>	846
AS/P/3	795	**826**	>	782	842	**1025**	>	886
AS/P/10	751	803	>	724	840	973	>	850
AC/P/3	756	**864**	>	845	835	**1014**	>	965
AC/P/10	725	793	<	802	811	983	>	884
D/R/3	814	**882**	>	862	862	**1080**	>	1002
D/R/10	800	864	>	846	842	988	>	946

Table 3
Multipath routing significantly **improves** the lifetime for non-uniformly distributed routing tasks (the best solutions in their classes are shown in bold, the best overall solution is underlined)

	Time before first task fail				Time before 10 dead nodes			
	GPSR 350	GPER 326			GPSR 408	GPER 400		
	MGPSR	$\text{MGPER}_{\text{sub}}$		$\text{MGPER}_{\text{hop}}$	MGPSR	$\text{MGPER}_{\text{sub}}$		$\text{MGPER}_{\text{hop}}$
D/P/3	372	395	>	375	440	456	>	428
D/P/10	388	**402**	>	370	444	**467**	>	425
A/P/3	368	378	>	368	434	**447**	>	426
A/P/10	**390**	390	>	363	428	442	>	425
AS/P/3	358	**377**	>	355	452	**454**	>	408
AS/P/10	348	365	>	340	438	418	>	406
AC/P/3	378	**392**	>	376	455	**458**	>	438
AC/P/10	385	384	>	366	437	448	>	431
D/R/3	398	**414**	>	388	472	**478**	>	448
D/R/10	387	404	>	376	453	464	>	433

nication end-points are not uniformly distributed in a network. The sensors closer to activity are likely to act as sources and the nodes close to the *prime nodes* which communicate with the outside networks are more likely to be end-points. A worry in this case is that the number of alternatives are significantly constrained closer to the end-points and the networks will start to fail as the nodes closer to the end-points are depleted. Therefore, we also investigated the impact of multipath routing in such situations. Experiment results showed that overall multipath routing significantly helps the network lifetime when tasks are distributed non-uniformly. Table 3 shows the average network lifetime when only 10 (randomly selected) end-point pairs are communicating. This table lists lifetimes until *the first task failure* and until *10 nodes die*:

- The main observation is that multipath routing approaches provide significantly longer (up to 30%) lifetimes compared to their singlepath counterparts. This is especially important as the lifetime of networks in the presence of non-uniform tasks is significantly less than the lifetime in the presence of uniform tasks (Table 3 vs. Table 2). Thus, optimization of the network lifetime for non-uniform tasks is especially critical and multipath routing achieves that.
- The most successful strategy is $\text{MGPER}_{\text{sub}}$ along with the distance and residual power criteria and the number of alternative is 3. In fact, this strategy is the winner under both network lifetime definitions. When residual power information is not available, the distance based probabilistic selection with $\text{MGPER}_{\text{sub}}$ provides the best results.
- These tables show that $\text{MGPER}_{\text{sub}}$ consistently provides a larger improvement than $\text{MGPER}_{\text{hop}}$. In fact, as the number of failures in the network increase, $\text{MGPER}_{\text{hop}}$'s performance starts falling below that of GPER alone.
- For $\text{MGPER}_{\text{sub}}$ and MGPSR, when angle routing used, increasing the alternatives too much is not necessarily advantageous. However, when the alternatives are calculated based on the distance criteria, large k (10 vs. 3 in the experiments) help improve the lifetime. In this case, the larger flexibility provided by $\text{MGPER}_{\text{sub}}$ provides significant savings by distributing the load better near the end-points.
- In all experiments with angle constrains, a constant angle range around the final destination performed better than or comparable to the other angle based alternatives. This points out to the fact that angle constraints computed based on the final destinations may be over-constraining, especially further away from the destination.

6. Conclusions

In this paper, we presented a new *Geographical Power Efficient Routing* (GPER) protocol, which enables the nodes to adaptively chooses the transmission range to minimize the power consumption of the sensor as well as of the non-participating listeners. We then presented three families of probabilistic geographical multipath routing protocols aimed at improving the network lifetime of wireless sensor networks. The protocols presented in this paper are highly power efficient, distributed, and scalable. Simulation results show that, for repeated routing tasks between a given source–destination pair, the average power cost of using multipath routing protocols is close to the power consumed when using the corresponding single path routing protocol. Most important results show that the network lifetime is improved especially when the endpoints in the network are repeatedly used, instead of being uniformly distributed. Among the nodes involved in multipath routing, those that are close to the source or the destination are used more frequently than others. Therefore, they drain out of power earlier than others. Simulations on NS2 show that GPER reduces the power consumption in the network close to 50% relative to GPSR and works well in networks with node density variations. In addition, multipath routing improves the lifetime up to 30%.

References

[1] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones, D. Leask, Piconet: embedded mobile networking, IEEE Personal Communications 4 (5) (1997) 8–15.

[2] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in: DIAL-M, 1999.

[3] J.H. Chang, L. Tassiulas, Routing for maximum system lifetime in wireless ad hoc networks, in: Annual Allerton Conference on Communication, Control and Computing, 1999.

[4] J.H. Chang, L. Tassiulas, Energy conserving routing in wireless ad hoc networks, in: INFOCOM, 2000, pp. 22–31.

[5] X. Cheng, B. Narahari, R. Simha, M. Cheng, D. Liu, Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics, Mobile Computing 2 (3) (2003) 248–256.

[6] S. Datta, I. Stojmenovic, J. Wu, Internal node and shortcut based routing with guaranteed delivery in wireless networks, Cluster Computing 5 (2) (2002) 169–178.

[7] L.M. Feeney, Energy efficient communication in ad hoc wireless networks, in: S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (Eds.), Mobile Ad Hoc Networking, Wiley, 2004.

[8] L.M. Feeney, An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks, Mobile Networks and Applications 6 (3) (2001) 239–249.

[9] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, Highly-resilient, energy-efficient multipath routing in wireless sensor networks, Mobile Computing and Communications Review 4 (5) (2001).

[10] Q. Gao, D. Holding, Y. Peng, K. Blow, Energy efficiency design challenge in sensor networks, LCS 2002, London, September 2002.

[11] J. Gomez, A.T. Campbell, M. Naghshineh, C. Bisdikian, Power-aware routing in wireless packet networks, in: IEEE International Workshop on Mobile Multimedia Communication, 1999.

[12] C. Guo, L.C. Zhong, J.M. Rabaey, Low power distributed mac for ad hoc sensor radio networks, in: GLOBECOM, 2001.

[13] B. Blum, T. He, S. Son, J. Stankovic, IGF: A state-free robust communication protocol for wireless sensor networks, Department of Computer Science, University of Virginia, USA, Tech. Rep. CS-2003-11, 2003.

[14] T. He, J.A. Stankovic, C. Lu, T.F. Abdelzaher, Speed: a stateless protocol for real-time communication in sensor networks, in: ICDCS, 2003.

[15] B. Hohlt, L. Doherty, E. Brewer, Flexible power scheduling for sensor networks, in: IPSN, 2004, pp. 205–214.

[16] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Mobile Computing, 1996, pp. 153–181.

[17] H. Karl, Relaying in wireless access networks – a capacity and energy-efficiency perspective, in: Wireless World Research Forum Kickoff Meeting, 2001.

[18] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: MobiCom, 2000, pp. 243–254.

[19] Y.B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: MobiCom, 1998, pp. 66–75.

[20] M. Krunz, A. Muquttash, Transmission power control in wireless ad hoc networks: challenges, solutions and open issues, IEEE Network 18 (2004) 8–14.

[21] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric ad hoc routing: of theory and practice, in: PODC, 2003, pp. 63–72.

[22] J. Kuruvila, A. Nayak, I. Stojmenovic, Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer, in: MASS, 2004.

[23] S.-J. Lee, M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, in: ICC, 2001.

[24] L. Li, J. Halpern, Minimum energy mobile wireless networks revisited, in: ICC, 2001.

[25] Q. Li, J. Aslam, D. Rus, Distributed energy-conserving routing protocols for sensor networks, in: IEEE HICSS, 2002.

[26] X. Lin, I. Stojmenovic, Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks, Journal of Parallel and Distributed Computing 63 (1) (2003) 22–32.

[27] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, in: Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2003.

[28] W. Lou, W. Liu, Y. Zhang, Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks, in: M. Cheng, Y. Li, D.-Z. Zhu (Eds.), Combinatorial Optimization in Communication Networks, Springer, 2006.

[29] T. Melodia, D. Pompili, I. Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, in: INFOCOM, 2004.

[30] R. Min, A. Chandrakasan, Mobicom poster: top five myths about the energy consumption of wireless communication, SIGMOBILE Mobile Computing and Communications Review 7 (1) (2003) 65–67.

[31] S. Murthy, J.J. Garcia-Luna-Aceves, An efficient routing protocol for wireless networks, Mobile Networks and Applications 1 (2) (1996) 183–197.

[32] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: INFOCOM, 1997, pp. 1405–1413.

[33] M. Pearlman, Z. Haas, P. Scholander, S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: MobiHOC, 2000, pp. 119–128.

[34] C. Perkins, Ad hoc on-demand distance vector routing, in: MILCOM '97 Panel on Ad Hoc Networks, 1997.

[35] C. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: SIGCOMM, 1994.

[36] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, Communications of the ACM 43 (5) (2000) 51–58.

[37] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, Geographic routing without location information, in: MobiCom, 2003, pp. 96–108.

[38] T.S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 1996.

[39] V. Rodoplu, T. Meng, Minimum energy mobile wireless networks, in: ICC, 1998, pp. 1633–1639.

[40] C. Schurgers, V. Tsiatsis, M. Srivastava, Stem: topology management for energy efficient sensor networks, in: IEEE Aerospace Conference, 2002.

[41] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks, in: SenSys, 2004.

[42] R.C. Shah, J.M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in: WCNC, 2002, pp. 350–355.

[43] S. Singh, C. Raghavendra, Pamas: power aware multi-access protocol with signalling for ad hoc networks, in: ACM Computer Communications, 1999.

[44] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: MOBICOM, 1998, pp. 181–190.

[45] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: MobiCom, 1998, pp. 181–190.

[46] A. Srinivas, E. Modiano, Minimum energy disjoint path routing in wireless ad hoc networks, in: MobiCom, 2003, pp. 122–133.

[47] I. Stojmenovic, S. Datta, Power and cost aware localized routing with guaranteed delivery in wireless networks, Wireless Communications and Mobile Computing 4 (2) (2004) 175–188.

[48] I. Stojmenovic, X. Lin, Power aware localized routing in wireless networks, IEEE Transactions on Parallel and Distributed Systems 12 (11) (2001) 1122–1133.

[49] Y. Wang, T. Arslan, An adaptive routing algorithm for long-lifetime wireless picoradio networks, in: SOC, 2003, pp. 96–99.

[50] K. Wu, J. Harms, Performance study of a multipath routing method for wireless mobile ad hoc networks, in: MASCOTS, 2001, p. 99.

[51] S. Wu, K.S. Candan, Gper: geographic power efficient routing in sensor networks, in: ICNP, 2004.

[52] H. Xie, L. Qiu, Y. Yang, Y. Zhang, On self adaptive routing in dynamic environments, in: ICNP, 2004.

[53] Y. Xu, J. Heidemann, D. Estrin, Usc/isi. Research report, USC/ISI, 2000.

[54] Y. Xue, B. Li, A location-aided power-aware routing protocol in mobile ad hoc networks, in: IEEE Globecom, 2001, pp. 25–29.

[55] M. Zorzi, P.R. Rao, Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance, IEEE Transactions on Mobile Computing 2 (2003) 337–348.

Shibo Wu received his B.S. degree in the Department of Precision Instruments and Mechanology, China in 2000. He is currently working on his Ph.D. degree in the Department of Computer Science and Engineering at Arizona State University. His current research focus is power-aware routing in wireless sensor networks. In particular, he is developing unicast, multicast, and multisource single destination data aggregation algorithms and protocols for power constrained ad hoc sensor networks. He has published his research at various premier networking and distributed computing venues.

K. Selçuk Candan is an associate professor at the Department of Computer Science and Engineering at the Arizona State University. He joined the department in 1997, after receiving his Ph.D. from the Computer Science Department at the University of Maryland at College Park. He has worked extensively on the management of heterogeneous information and information sources. His current research interests include quality adaptative data processing, distributed sensory data management, and efficient dynamic content delivery over different types of network architectures. He has published various articles in respected journals and conferences in related areas. He received his B.S. degree, first ranked in the department, in computer science from Bilkent University in Turkey in 1993.