Ad Hoc Networks 61 (2017) 16–32

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Bypassing holes in sensor networks: Load-balance vs. latency

Fan Zhou[a,*], Goce Trajcevski[b], Roberto Tamassia[c], Besim Avci[b], Ashfaq Khokhar[d], Peter Scheuermann[b]

[a] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[b] Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA
[c] Department of Computer Science, Brown University, Providence, RI 02912, USA
[d] Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA

ARTICLE INFO

Article history:
Received 30 August 2015
Revised 16 February 2017
Accepted 2 March 2017
Available online 4 March 2017

Keywords:
Geographical routing
Load-balancing
Communication latency

ABSTRACT

This work addresses problems that arise when geographic routing is used in the presence of holes in wireless sensor networks. We postulate that relying on the existing algorithms for bypassing a coverage hole may cause more severe depletion of the energy reserves among the nodes at (or near) that hole's boundary. This, in turn, will render some of those nodes useless for any routing (and/or sensing) purposes, thereby effectively enlarging the size of existing hole and inducing longer communication delays for certain *(source, sink)* pairs. We propose heuristics that address these complementary problems: (1) relieving some of the routing-load for the nodes around the boundary of a given hole, for the purpose of extending their lifetime; and (2) reducing the latency of the packets-delivery by using routes that are within certain bounds from the route based on the shortest-path. Our approaches are based on the idea that some of the packets that would (otherwise) need to be routed along the boundary of a given hole, should instead start to deviate from their original path further away from that hole. To investigate the potential benefits, we introduce approximations of the hole's boundary with a rectangle, a circle and an ellipse, respectively. We derive the bounds on reducing the routing latency for these three approximations. Our experiments demonstrate that the proposed approaches not only increase the lifetime of the nodes along the boundary of a given hole and yield a more uniform depletion of the energy reserves in its vicinity, but also reduce the communication latency, compared to the traditional face routing.

1. Introduction

One of the sources of potentially disrupting the expected quality of service and causing a communication delay in Wireless Sensor Networks (WSN), is the occurrence of holes – which is, regions inside the area of interest that are void of operational nodes for sensing and/or routing purposes [1]. While they may originate as early as the deployment stage of the network, however, holes may also be generated during its continuous operation, either because a significant amount of nodes within a close proximity of each other have depleted their energy-reserves, or because some environmental phenomenon (e.g., a forest fire) caused a physical destruction of a group of nodes.

The problem of detecting and bypassing holes in WSNs has generated a significant amount of research results [2–7]. The contexts of the existing works range from considering purely topological/connectivity relationships (e.g., [8]), to assuming some type of a geographic knowledge regarding the locations of the nodes [9,10]. When it comes to forwarding a packet from a given source to a given sink node, the typical approaches from the latter settings, a common assumption (cf. [4–6,8] is the combination of:

- *greedy forwarding* along the shortest path; and
- *around-perimeter forwarding (a la' face routing)*, in order to bypass the hole while routing packets towards the sink.

Although these works do guarantee packet delivery by exploring the network connectivity information, it is difficult to balance in-network load traffic and to alleviate the congestion for the nodes along a given routing path. Moreover, most of existing protocols for bypassing coverage holes focus only on discovering a single route between a given *(source, sink)* pair.

* Corresponding author.
E-mail addresses: fan.zhou@uestc.edu.cn (F. Zhou), goce@eecs.northwestern.edu (G. Trajcevski), rt@cs.brown.edu (R. Tamassia), besim@eecs.northwestern.edu (B. Avci), ashfaq@uic.edu (A. Khokhar), peters@eecs.northwestern.edu (P. Scheuermann).

http://dx.doi.org/10.1016/j.adhoc.2017.03.002
1570-8705/

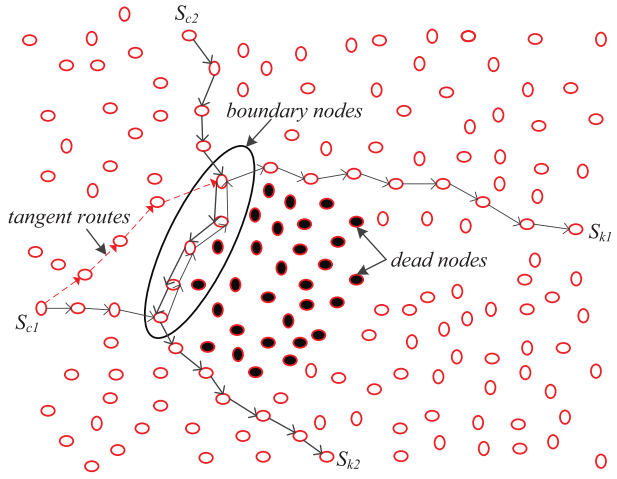


Fig. 1. Motivational scenario: two routes bypassing the same hole.

The motivation for this work is based on the following observations:

1. In scenarios in which there are multiple *(source, sink)* pairs, relying on some (e.g., GPSR-like [13]) routing protocol to bypass a given hole, their combined-effect may cause the operational nodes along the boundary of the hole to be depleted faster than their neighbors (and the rest of the network). As a consequence, this could cause a more rapid "expansion" of the hole itself.
2. Routing a particular packet around the boundary of the hole imposes an extra latency on that packet's delivery to the sink. Namely, as a straightforward consequence of the triangular inequality, if one could route the packets directly along the tangent from the source to the boundary of the hole (and vice-versa for the sink), the source-to-sink delay will be smaller than the one incurred via greedy geographic algorithms combined with face routing.

An illustrative example of the observations above is presented in Fig. 1, showing two *(source, sink)* pairs (S_{c1}, S_{k1}) and (S_{c2}, S_{k2}), that need to bypass a given hole depicted as a region containing dark disks (corresponding to dead nodes). The shaded ellipse covering part of the boundary, contains a sequence of nodes that will be used for routing twice, once for each pair (S_{c1}, S_{k1}) and (S_{c2}, S_{k2}). Clearly, this will double the rate of energy resources depletion inside the shaded ellipse, compared to the rest of nodes along hole's boundary (not to mention the rest of nodes in the WSN). As for the second observation – routing along the tangent to the hole starting from S_{c1} reduces the hop-count by 3. Both effects will be further amplified in applications in which long-running periodic transmissions are required for sampled values from a given source towards the corresponding sink.

To address these issues, we introduce two flexible routing algorithms for bypassing coverage holes, which provide a balance between communication latency and the load distribution. Specifically, the main contributions of this work can be summarized as follows:

1. We present methods for compact representations of the hole-boundary, as well as a simple but efficient propagation scheme for this information;
2. We provide two heuristic-based routing protocols aiming at balancing the load vs. latency trade-offs:
   - RCF (Ring-Constraint Forwarding) utilizes the hole bounding shape information to construct the almost-shortest paths while bypassing the hole, mainly aiming at reduction of the communication delay;
   - WCF (Wedge-Constraint Forwarding) improves the uniformity of the rate of energy-depletion in network by exploring more available paths than RCF, however, trading off an increase on length of the routing path, thereby inducing a longer delay.
3. We present an analysis on benefits of the proposed hole approximation approaches, demonstrating that the upper bound on the communication delay, with respect to the shortest-path route, for the respective hole's boundary approximation are: 0.1665 for a circle, $1/\sqrt{2}$ for a rectangle, and within (0.1665, 0.2929) for an ellipse (depending on its eccentricity).

Circle	Rectangle	Ellipse
0.1665	$1/\sqrt{2}$	(0.1665, 0.2929)

4. We conducted extensive experimental evaluations of the proposed algorithms, and the results demonstrate that significant improvement on both: (1) load balancing (maximum 24%), expressed as standard deviation of the energy reserves, and (2) extension on network lifetime (maximum 68%) can be achieved by our methods. Moreover, the proposed approaches can reduce the communication latency (maximum 11%), compared to existing works.

This paper extends our previous work [11], by presenting alternative minimum bounding shapes and a comparative analysis of the benefits of each. In addition, we provide theoretical analysis and some practical considerations, along with more detailed experimental evaluation on the proposed approaches, especially related to the impact of each approach on the latency.

The rest of this article is structured as follows. In Section 2 we introduce the basic definitions and recollect the necessary background. Section 3 presents the three algorithms for approximating the boundary of a given hole. Section 4 presents the details of the corresponding routing algorithms. Theoretical analysis on the benefits of the proposed approaches is presented in Section 5. The experimental observations regarding the benefits of proposed schemes are presented in Section 6. In Section 7, we review the relevant existing literature and position our work in that context. We summarize the paper and outline directions for future work in Section 8.

2. Preliminaries

We now introduce the basic terminology and definitions used in the rest of this paper, including the basic ideas behind the methodologies for generating the hole representation that are at the core of our proposed approaches.

We assume a dense WSN with N nodes, where the sensor nodes $\{S_1, S_2, \ldots, S_N\}$ are randomly deployed over the geographic area of interest. Each node S_i has a unique, fixed physical location represented via coordinates in a given reference system. Nodes are assumed to have the capability of determining their locations at runtime, either by means of a location hardware, such as a GPS device, or by implementing a location discovery algorithm [12,13]. Each node S_i has information about the position and state of its one-hop neighbors $NB(S_i)$. In addition, all nodes have identical transmission range denoted as R_t.

In many (applications) settings it may happen that the corresponding sensor network may have a hole(s) – i.e., a region(s) with insufficient number of nodes to provide sensing coverage. The main reason is two-fold: (1) The very process of deployment – e.g., dropping them from unmanned aerial or terrestrial vehicle may yield uncovered areas. Similarly, even with a uniform deployment, the slope of the terrain or other geo-properties (e.g., lakes) may yield significant area without a proper coverage. (2) Even more

often, the source for the lack of coverage is node failures and/or depletion of the energy of the existing nodes [3]. A hole H is determined by a set of nodes $B = \{S_1^b, \ldots, S_n^b\}$ along its boundary. To identify the nodes in the set B, we rely upon the distributed algorithm presented in [14] for correctly detecting nodes along the boundary and connecting them into meaningful boundary cycles. By "identifying boundary nodes" we mean: (a) for every point on the boundary of a hole we expect the algorithm to mark a sensor node nearby; and (b) every sensor node marked by the algorithm lies near a boundary of a hole [15].

Our first step of approximating the (boundary of) a given hole is to generate the convex hull $CH(B)$. Given a set of point-locations $P = \{P_1, P_2, ..., P_M\}$ in the Euclidian 2D space, its convex hull $CH(P)$ is defined as the smallest convex polygon whose vertices are $V_{CH(P)} \subseteq P$ and all the other points from $P \setminus V_{CH(P)}$ are inside $CH(P)$ [16]. Our main heuristics are built by subsequently enclosing the convex hull $CH(B)$ within its Minimum Bounding Rectangle(MBR), Minimum Bounding Circle(MBC) or Minimum Bounding Ellipse(MBE). We note that, regardless of the selection of a bounding shape (MBR, MBC or MBE), it will always contain (possibly in its interior) every vertex $S_i \in V_{CH(B)}$. Moreover, due to the convexity of $CH(B)$, every point from the edges and the interior of $CH(B)$ will also be contained in (or on the boundary of) the selected bounding shape.

As we will demonstrate, the hole refinement not only brings benefits in reducing the communication overhead incurred when propagating the hole information in the WSN, but it also improves the performance of the routing algorithms in terms of both: (a) decreasing the latency of the packets delivery, and (b) balancing the distribution of the traffic load.

Upon completion of the construction of the minimum bounding shape, that information is embedded in a corresponding packet (P_{MBR}, P_{MBC}, or P_{MBE}) and broadcasted to all the nodes in the WSN what are within distance D from the approximated boundary.[1]

3. Approximating the hole boundary

This section discusses in detail the hole shape approximation methods and the propagation of the information regarding the corresponding shape(s) throughout the WSN.

Initially, we rely upon the algorithm presented in [17] for the purpose of selecting one of the nodes along the boundary, say, $S_f^b (S_f^b \in B)$ to serve as a temporary fusion center. S_f^b stores the geometry-information (i.e., locations and the unique IDs) for the respective boundary nodes. The message complexity of this aggregation is $O(n \log n), n = |B|$ (cf. [17]).

We now proceed with presenting the convex hull construction, followed by the three proposed methods for further approximating the hole boundary (MBR, MBC and MBE).

3.1. Convex hull construction

To generate the convex hull, S_f^b relies on the incremental hull-construction algorithm [16] which we describe here for completeness. S_f^b sorts the nodes in B based on the ascending order of their y coordinates and, in case there are multiple nodes with the same y coordinate, the order is determined by the ascending order of their respective x coordinates. Specifically, assuming that the list of the boundary nodes B contains a sorted sequence $S_1^b, \ldots, S_n^b$ (in counter-clockwise order of traversal – cf. Fig. 2), each of the nodes S_1^b and S_n^b starts with the list $CH(R) = \{S_1^b, S_2^b\}$ and $CH(L) = \{S_n^b, S_{n-1}^b\}$. In the subsequent steps, S_1^b (respectively, S_n^b) traverses around the nodes on the *Right* (respectively, *Left*) side of the line segment $\overline{S_1^b S_n^b}$ in clockwise order. In the j^{th} iteration ($3 < j < n$), S_1^b (respectively, S_n^b) adds a node into $CH(R)$ (respectively, $CH(L)$), and checks whether the last 3 nodes in the sequence make a right turn. If this is the case, the algorithm continues; otherwise, $CH(R)$ ($CH(L)$) deletes the middle one of the last 3 nodes and checks the right turn again for the last 3 nodes of the thus updated list. The process continues until the sequence is obtained in which the last 3 nodes make a right turn. After traversing all the nodes in B, the convex hull $CH(B)$ is obtained by concatenating lists $CH(R)$ and $CH(L)$ – eliminating the duplicates (the head and rear nodes in $CH(L)$).

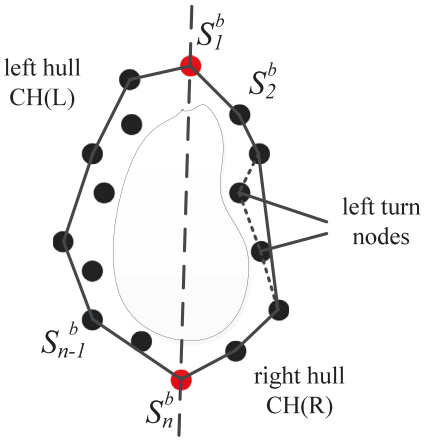


Fig. 2. Illustration of convex hull construction.

In this particular convex hull construction algorithm, the message overhead is bounded by $O(n \log n)$, where n is the cardinality of set B. The time for sorting the nodes (performed in S_f^b) is, once again, $O(n \log n)$. Both the left and the right hull are generated in linear time $O(n)$. Thus, the total time complexity for constructing convex hull is bounded by $O(n \log n)$.

3.2. Approximating a hole with MBR

The Minimum Bounding Rectangle (MBR) is the rectangle which bounds a given set of nodes and has the smallest area (note that we do not necessarily rely upon an axis parallel rectangle). To construct the MBR for approximating $CH(B)$, we adopt the rotating calipers algorithm [18]. The pseudo-code of the algorithm is specified in Algorithm 1, and we use Fig. 3 as an example to explain it.

At the heart of rotating calipers algorithm is the observation that the MBR of a convex hull has a side collinear with at least one edge of the hull. Consequently, the search for the MBR can be limited to the ones whose orientations are given by the edges of the $CH(B)$. Specifically, given a convex polygon $CH(B)$, the four extreme points in x and y coordinates are computed first. These points define two calipers – each has a pair of parallel lines, as shown by (l_1, l_2) and (l_3, l_4) in Fig. 3. Then the calipers rotate clockwise until one of them coincides with an edge of $CH(B)$. At each such point, the algorithm computes the area of the rectangle defined by the current calipers Φ, and compares it to the current minimum area Φ_{min}. If $\Phi < \Phi_{min}$, the list V which contains the nodes determining the calipers is updated correspondingly. The above process continues until the calipers have rotated at least 90° ($\Theta \geq 90°$) and the MBR is determined by the nodes in list V.

MBR refinement requires $O(n)$ time to find each of the four extreme points, and $O(n)$ time to scan the possible rectangles, where n is the number of vertices in $CH(B)$. Thus, for a given a convex

[1] The optimal value of D for the three minimum bounding shapes is discussed in greater detail in Section 5.

Algorithm 1 Algorithm of MBR Construction.

Input: Convex hull $CH(B)$ of set B, $|CH(B)| = n$.
Output: The Minimum Bounding Rectangle MBR($CH(B)$).
1: Set the minimum rectangle area $\Phi_{min} \leftarrow +\infty$;
2: A set $V \leftarrow \emptyset$ of vertex nodes defining the rectangle;
3: Find the extreme points xMin, xMax, yMin, yMax;
4: Construct two "calipers" containing two pairs of parallel lines$\ell_a = (l_1, l_2)$ and $\ell_b = (l_3, l_4)$;
5: Accumulated rotating degree of calipers $\Theta \leftarrow 0$;
6: Temporal rotating degree of calipers $\theta \leftarrow 0$;
7: **while** $\Theta < 90^\circ$ **do**
8: **if** $\exists\ \ell \in \{\ell_a, \ell_b\}$ coincides with an edge of $CH(B)$ **then**
9: $\Phi \leftarrow$ the area of the calipers-defined rectangle;
10: **if** $\Phi < \Phi_{min}$ **then**
11: $V \leftarrow$ the nodes determining the calipers;
12: **end if**
13: Compute θ of calipers;
14: $\Theta \leftarrow \Theta + \theta$;
15: **else**
16: $\theta \leftarrow 0$;
17: Rotate the calipers clockwise;
18: **end if**
19: **end while**
20: **return** MBR($CH(B)$) defined by nodes in V.

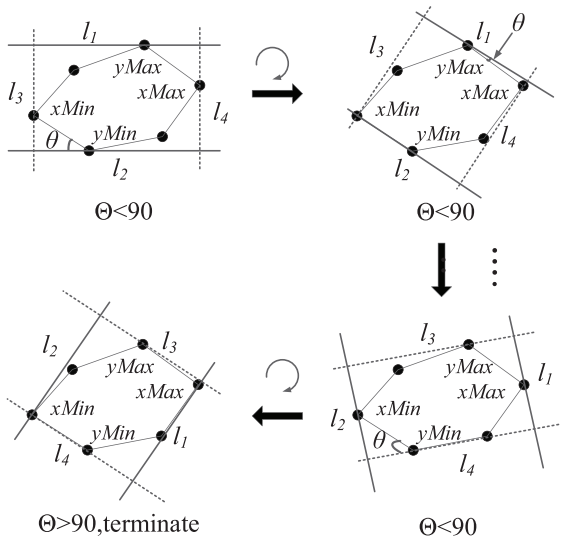


Fig. 3. Illustration of minimum bounding rectangle construction.

hull, the total time complexity of MBR construction is bounded by $O(n)$. Note that this particular MBR construction applies to a convex polygon (which, as mentioned, requires the time complexity of $O(n\log n)$ for computing the convex hull given n number of boundary nodes).

3.3. Approximating the hole with MBC

The next approximation used to represent a given hole is based on the Minimum Bounding Circle (MBC). This, in turn, relies upon the computation of the *smallest enclosing disk* that can be obtained using the *randomized linear programming* algorithm [19]. The algorithm works in an incremental way: it generate a permutation $P\_CH(B)$ of the nodes[2] in $CH(B)$ and starts by constructing a circle containing the first two points in $P\_CH(B)$. As the subsequent nodes are being considered, the circle may grow in size to ensure that each added node is in its interior or boundary. The pseudo-code of the MBC algorithm is presented in Algorithm 2.

Algorithm 2 Algorithm of MBC Construction.

Input: Convex hull $CH(B)$ of set B, $|CH(B)| = n$.
Output: The Minimum Bounding Circle MBC($CH(B)$).
1: $C_2 \leftarrow$ the MBC for $\{S_1^b, S_2^b\}$;
2: Compute a permutation $P\_CH(B) = \{S_1^b, \ldots, S_n^b\}$.
3: **for** $i = 3 \rightarrow n$ **do**
4: **if** S_i^b is contained in C_i **then**
5: $C_i \leftarrow C_{i-1}$;
6: **else**
7: $C_1' \leftarrow$ the MBC for $\{S_1^b, S_i^b\}$;
8: **for** $j = 2 \rightarrow n$ **do**
9: **if** S_j^b is contained in C_{j-1}' **then**
10: $C_j' \leftarrow C_{j-1}'$;
11: **else**
12: $C_0'' \leftarrow$ the MBC for $\{S_i^b, S_j^b\}$;
13: **for** $k = 1 \rightarrow n$ **do**
14: **if** S_k^b is contained in C_k'' **then**
15: $C_k'' \leftarrow C_{k-1}''$;
16: **else**
17: $C_k'' \leftarrow$ the unique circle withS_i^b, S_j^b and S_k^b on its boundary;
18: **end if**
19: **end for**
20: $C_j' \leftarrow C_n''$;
21: **end if**
22: **end for**
23: $C_i \leftarrow C_n'$;
24: **end if**
25: **end for**
26: **return** MBC($CH(B)$) $\leftarrow C_n$.

To explain its execution: firstly, after generating the random permutation $P\_CH(B) = \{S_1^b, \ldots, S_n^b\}$ of $CH(B)$, the smallest circle C_2 containing nodes S_1^b and S_2^b is constructed. The rationale behind Algorithm 2 is that at each iteration i, if the location of the current node S_i^b is contained inside (or on the boundary) of the smallest circle from the previous iteration, MBC C_{i-1}, then the solution does not change (line 5). Otherwise the current MBC C_i should be modified so that S_i^b is locating on its boundary of C_i (lines 6–23). To compute the latter case, the algorithm maintains the MBC C_i for $S_1^b, \ldots, S_i^b$ which limits S_i^b on the boundary of C_i, and uses the same approach that adds the nodes in $P\_CH(B)$ one by one, checking for the location of the node S_j^b, $j \in [1, n]$ – if S_j^b lies in the current MBC C_{j-1}', it remains (line 10); otherwise, C_j' should have both S_i^b and S_j^b on its boundary. Lines 12–20 finds the solution under the constraint that both S_i^b and S_j^b should lie on the boundary of the optimal MBC, where the similar methodology is applied – the MBC C_k'' remains if it contains S_k^b or it is determined by nodes S_i^b, S_j^b and S_k^b – all of which are on the boundary of C_k''.

Obviously, in all the cases in which the "else" part (lines 6–23) is not executed, the running time of MBC is $O(n)$ ($n = |CH(B)|$). Otherwise, a "backward analysis" can be used to compute the probability of executing its execution (cf. [16]). Assume that we have already constructed the current MBC C_i which passes through node S_i^b. Since C_i has at least three nodes, including S_i^b, on its boundary, C_i changes only when we remove one of those three points. The probability that S_{i-1}^b is one of the three nodes is $2/i$,

[2] We observe that the smallest enclosing disk for a set of point B is equivalent to the smallest enclosing disk of the convex hull of that set of point, $CH(B)$.

since S_i^b is already one of them. Thus, the expected computation time for n nodes is $O(n) + \sum_{i=2}^{n} O(i)\frac{2}{i} = O(n)$.

3.4. Approximating the hole with MBE

Similar to MBC, Minimum bounding Ellipse (MBE) is the smallest area ellipse that contains a set B of n boundary nodes. Algorithm 3 depicts the process of constructing MBE using a *recursive function* mbe(Θ, Δ), which computes the smallest enclosing ellipse containing nodes in set Θ with the ones in set Δ on its boundary.

Algorithm 3 Algorithm of MBE Construction.

Input: Convex hull $CH(B)$ of set B, $|CH(B)| = n$.
Output: The Minimum Bounding Ellipse MBE($CH(B)$).

```
1: Compute a permutation P_CH(B) = {S_1^b, ..., S_n^b}.
2: if Θ = ∅ ⊕ |Δ| = 5 then
3:   return mbe(∅, Δ);
4: else
5:   Select S_i^b ∈ P_CH(B);
6:   e ← mbe(P_CH(B) \ {S_i^b}, Δ);
7:   if S_i^b is contained in e then
8:     return e;
9:   else
10:    return mbe(P_CH(B) \ {S_i^b}, Δ ∪ {S_i^b}).
11:  end if
12: end if
```

The idea behind Algorithm 3 is the same as MBC construction, except for the fact that an ellipse is determined by 5 points on its boundary. Thus, if Θ is empty or $|\Delta| = 5$, the MBE is defined by mbe($\emptyset$, Δ) (Line 2–3). Otherwise, it selects a node $S_i^b \in P\_CH(B)$ and recursively determines whether S_i^b is contained in e (Line 5–6) – the smallest ellipse containing nodes in $P\_CH(B) \setminus \{S_i^b\}$. If this is the case, then we get mbe($P\_CH(B) \setminus \{S_i^b\}$, Δ) as the MBE; otherwise, the MBE must has S_i^b on its boundary, and the algorithm proceeds with computing mbe($P\_CH(B) \setminus \{S_i^b\}$, $\Delta \cup \{S_i^b\}$) (Line 10). The expected run-time complexity for constructing the MBE is also (upper)bounded by O(n) – however, the proof is a bit more involved. For details, we refer the interested reader to [20].

3.5. Propagating the hole information

After calculating the above three minimum bounding shapes, the one with smallest area is selected as the approximation of the (convex hull of the) hole. Once that is determined, a broadcast packet can be sent containing the corresponding description of the P_{MBR}, P_{MBC} or P_{MBE} to propagate the information about the type, as well as the parameters, representing the hole's approximation. For example, the packet representing the MBE approximation will contain the coordinates of the ellipse center, along with the values of the (semi)major axis and (semi)minor axis.

When propagating this information, an additional parameter is added into the corresponding packet–the propagating factor $p_f(>0)$. This value is used to control the distance D of how far the information should be propagated, with respect to the original boundary of the approximation used. For example, in the case of MBE approximation, if the MBE is specified by the equation $(x-x_{h_c})^2/a^2 + (y-y_{h_c})^2/b^2 = 1$, the P_{MBE} packet will be propagated to the nodes within the ellipse $(x-x_{h_c})^2/(a*p_f)^2 + (y-y_{h_c})^2/(b*p_f)^2 = 1$. A detailed analysis of the criteria for selecting the value of p_f is presented in Section 5.

We note that the hole information is only broad-casted once, and the nodes within the propagation area would cache that information after receiving the corresponding packet. The size of the message-content of the packet is small – and the message complexity of propagating the hole information is bounded by $O(m)$, using the dissemination techniques proposed in [21], where m is the number of nodes in the propagation area, minus the boundary nodes. We also note that the temporary fusion center performs more computing than other nodes. However, the energy spent on constructing boundary approximation is far less than that consumed at routing stage, and thus is ignored here. We also note that a hole grows with time as the boundary nodes deplete their energy. However, we do not need to update the hole information before the growing hole reaches the boundary of its original propagation area (specified by parameter p_f). The reason is that a hole would grow in accordance with its bounding shape, e.g., a MBC hole would continue to grow/expand in circular fashion, due to the intrinsic properties of the routing algorithms we developed, which are introduced in next section.

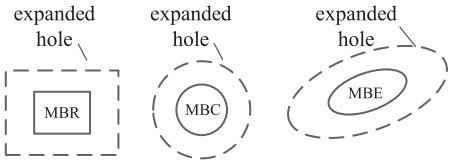


Fig. 4. Expanding the approximation of a hole.

4. Virtual hole expansion and bypassing

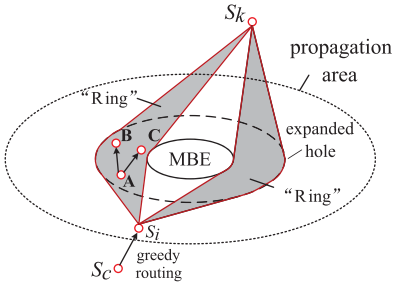
In this section, we present the details of the two routing protocols, each of them aiming at balancing the trade-offs between two complementary desiderata: reducing the latency of source-to-sink communication, and balancing the load among the nodes participating in the transmission, especially considering the ones along the boundary of the hole. Essentially, the proposed algorithms create certain "permissible areas" within which the possible paths for a given *(source, sink)* pair are selected, keeping the latency within acceptable bounds. The construction of a particular route builds upon the Trajectory Based Forwarding (TBF) [22] paradigm.

4.1. Expanding the hole

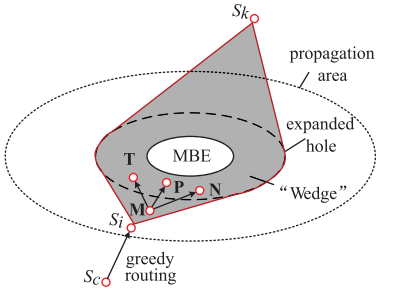
Unless there is a direct "line of sight" route that does not intersect a given hole and with enough active sensors to transmit the packets between a given *(source, sink)* pair, the shortest path when bypassing a hole consists of:

1. The tangent from the source to the hole's boundary;
2. The tangent from the sink to the hole's boundary;
3. The portion of the hole's boundary in-between the two tangent points above.

However, as we observed in Section 1, routing continuously along such shortest paths may rapidly deplete the energy of a subset of the nodes, i.e., those ones that are located along the boundary of the hole. Thus, rather than routing constantly along the shortest path, we utilize the sub-optimal paths to share the traffic which can be explored by expanding the hole with a factor d_f. Similarly to the hole propagation, hole expansion is controlled by another constant factor $d_f \geq 1$. We define $d_f = r'/r$ for MBC, $d_f = w'/w$ for MBR and $d_f = a'/a$ for MBE, where r', w', and a' is the expanded radius, width and semi-major axis, respectively. We note that larger value of d_f may significantly detour the packets, thereby introducing unacceptable delays. The main idea of expanding the approximation of a particular hole is illustrated in Fig. 4.



(a) Ring-Constrained Forwarding (RCF).



(b) Wedge-Constrained Forwarding (WCF).

Fig. 5. Illustration of RCF and WCF routing.

4.2. Routing protocols

We now focus on the two routing approaches presented in this work – *Ring-Constrained Forwarding (RCF)* and *Wedge-Constrained Forwarding (WCF)*. As an initial observation, we note that while RCF aims at reducing the communication delay, WCF achieves better load balance.

Each of the two routing protocols has a common (partial) behavior, specified as follows (Fig. 5): the source node S_c starts with forwarding along the shortest path towards the sink S_k in a greedy manner, e.g., following the $\overline{S_cS_k}$ line segment in a TBF-like manner [22]. If no hole is encountered to be bypassed, the protocol completes the transmission with greedy routing. Otherwise, a particular (intermediate) node S_i determines whether its location falls within the propagation area which is determined by the hole information and the parameter p_f. If this is not the case, it also forwards the data packets using greedy routing. Otherwise, S_i triggers RCF or WCF routing protocol.

Before delivering any data packet, the source S_c selects the value of the expanding factor d_f and augments the packet-content with that value, which will be used for the purpose of establishing the subsequent hops.

4.2.1. Ring-Constrained Forwarding (RCF)

The first algorithmic details that we present pertain to the RCF (*Ring-Constrained Forwarding*) protocol, which explores the almost-shortest paths for routing packets between given *(source, sink)* pair impeded by a coverage hole.

Specifically, RCF routes the packets in following way:

Every subsequent routing node randomly designates one of its neighbors as the next-hop, that is:

- closer towards S_k than itself;
- is within the zone bounded by the *tangents to the inner and outer bounding shape*, as constructed both from S_c and S_k, and the portion of the "ring" bounded by the *"arcs"* from the boundary in-between the tangents (gray area).

The first constraint guarantees that the packets is forwarded in a greedy manner towards destination. Under the second condition, the routes scheduled in RCF approaches the tangent line when node density is sufficient high.

The main ideas behind the RCF protocol are illustrated in Fig. 5(a). Note how the sensor node denoted as **A** in Fig. 5(a) can select any of its neighbors **B** or **C**, if they both satisfy above two criteria.

The advantage of RCF mechanism is that it reduces the communication latency when compared to greedy routing and disperses the data over multiple paths in the ring area. Theoretical analysis on improvement in terms of communication latency is investigated in Section 5. We also observe through simulation (discussed in Section 6) that the load-balancing achieved by RCF increases with parameter d_f. Increasing d_f, however, also decreases the performance of RCF on transition time.

4.2.2. Wedge-Constrained Forwarding (WCF)

While our first heuristic RCF focus on reducing transition latency while allowing smaller variations within some bounds from the (near-)shortest path around (minimum bounding shapes of) the hole, the second heuristic that we considered offers a wider choice of selecting next-hop routing nodes. The main difference is in the forwarding policy:

Every subsequent routing node randomly selects one of its neighbors as the next-hop that is:

- closer towards S_k than itself;
- is within the zone ("wedge") bounded by the *tangents to the outer bounding shape*, as constructed both from S_c and S_k, and the portion of the *outer boundary of the ring* in-between the tangents (gray area).

Fig. 5(b) illustrates the specifics of the WCF heuristics. As shown, the node **M** can select any of its neighbors **N, P**, or **T** – exemplifying the basic trade-offs between RCF and WCF:

1. WCF offers wider flexibility of paths to distribute the traffic, thereby providing better load-balancing;
2. RCF, on the other hand, restricts the amount of next-hop selections; however, it provides a smaller latency of the transmission between S_c and S_k.

The process of RCF and WCF, as well as the criteria for selecting nodes are summarized in Algorithm 4. We note that both RCF

Algorithm 4 Algorithm of RCF and WCF Routing.

Input: S_c, S_k, geometric space $\mathcal{G}$ of a minimum bounding shape and its expanded space $\mathcal{G}^+$.
Output: Connectivity of network.

```
Compute intersecting points set p(S_c, S_k) = \overline{S_cS_k} ∩ G;
if |p(S_c, S_k)| ≤ 1 then
    Forwarding packets with greedy routing;
else
    Random d_f, d_f ∈ [0, ξ];
    Compute tangent points set P_t^- to G;
    Compute tangent points set P_t^+ to expanded space G^+;
    Compute path distribution area in RCF or WCF;
    Deliver packet to node S_{i+1} selected by RCF or WCF.
end if
```

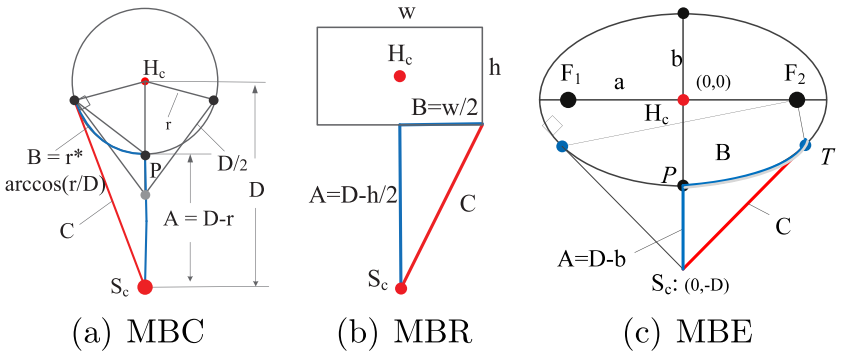


Fig. 6. Illustration of shortest paths of MBC, MBR and MBE.

and WCF choose the nodes in a random manner instead of greedy strategy, which may introduce extra hop(s) during routing. However, the reason behind this choice is that it allows an exploration of more available routes. This, in turn, will reduce the rate of the "expansion" of the hole, which is one of the motivations of this work.

Another observation is that RCF and WCF – since they do behave in a greedy manner – cannot always guarantee that the route will not get "stuck" in some local minimum (in terms of distance towards the sink). Such is the case, for example, when a node that was selected as a next hop, happens to be on the boundary of the hole, and all of its neighbors are further away from the sink than that node itself. This is illustrated with the node labeled P in Fig. 5(b). We note, however, that this case can be reduced to the traditional hole-bypassing scenarios – a problem which has already been investigated in the literatures [5,23,24]. Thus, throughout this work we assume an existence of a backtracking mechanism "embedded" in RCF.

5. Trade-offs analysis and practical considerations

In this section, we firstly analyze the worst case behavior of the proposed techniques and its implications on the latency, followed by the derivation of the respective upper bounds for each of the proposed hole-approximating approaches. Subsequently, we analyze two issues of practical relevance: (1) the aspect of load-balancing; (2) broader pragmatic considerations about applicability of the proposed techniques.

5.1. Latency impact

Communication latency depends on several factors, ranging from the MAC protocol, through links quality, to the semantics of the particular application [25,26]. Since throughout this work we rely on TBF [22], for a given source-sink pair (S_c, S_k), the *communication latency*, denoted $\chi(S_c, S_k)$, is estimated via the length of the geographic route used. Apparently, $\chi(S_c, S_k)$ consist of 3 parts: (1) the route from sink to the hole boundary; (2) the route for bypassing the hole boundary; and (3) the distance from boundary to the destination. Unless stated otherwise, the discussion below models previous 2 parts for convenience since the last one can be derived naturally.

Let H_c denote the hole center and S_c denote the source node. Without loss of generality, we fix the hole size with radius r and vary the distance, D, from source node to the hole center, i.e., the length of segment $\overline{H_cS_c}$. As shown in Fig. 6(a), the length of routes in greedy routing can be expressed as $A+B$, where A is the path from S_c to hole boundary $\overline{S_cP}$, and B is the path along hole boundary, which can be expressed as $D-r$ and $r\cdot\arccos(\frac{r}{D})$, respectively.

With the hole geometric information regarding the hole – location, shape and boundary, the tangent path C can be calculated as $C=\sqrt{D^2-r^2}$.

Definition 1. The *shortest path gain* $\pi(D)$ to greedy routing is defined as the ratio of :

$$\pi(D)=\frac{A+B-C}{A+B}=1-\frac{C}{A+B} \tag{1}$$

5.1.1. Minimum bounding circle

For a minimum bounding circle of the hole, we have following theorem:

Theorem 1. *The upper bound on shortest path gain* π_{max} *for a Minimum Bounding Circle approximation of hole is 0.1665, which is achieved when* $D=1.4498r$.

Proof. We are interested in maximizing the shortest path gain $\pi(D)$, i.e., $\arg\max_{D>r}(\frac{A+B-C}{A+B})$. This problem is equal to minimize $\frac{C}{A+B}(D>r)$. Then we have following objective function $f(D)$:

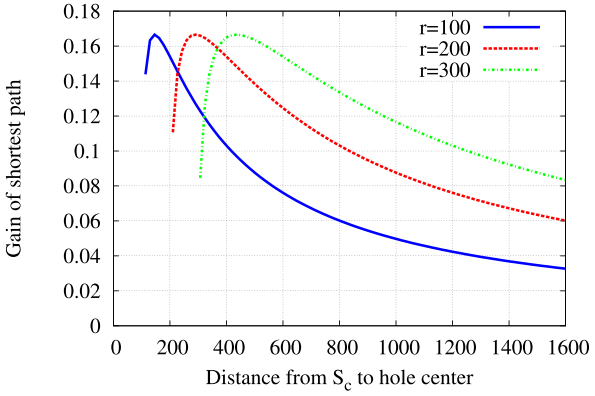
$$\begin{aligned} f(D)&=\frac{C}{A+B}\\ &=\frac{\sqrt{D^2-r^2}}{D-r+r\arccos(\frac{r}{D})}(D>r) \end{aligned} \tag{2}$$

By differentiating function $f(D)$ with respect to D we get:

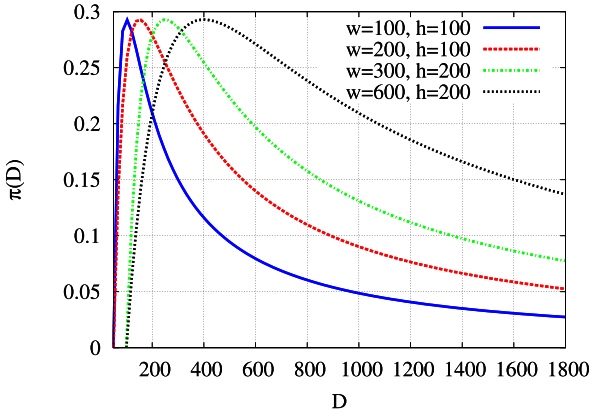
$$f'(D)=\frac{\frac{D\left(D-r+r\arccos\left(\frac{r}{D}\right)\right)}{\sqrt{D^2-r^2}}-\sqrt{D^2-r^2}\left(1+\frac{r^2}{D^2\sqrt{D^2-r^2}}\right)}{(D-r+r\arccos(\frac{r}{D}))^2} \tag{3}$$

Since the denominator of equation is greater than zero, we only consider the numerator, denoted as $h(D)$, of Eq. (3). By solving function $h(D)=0$, we have $D\approx 1.4498r$. Namely, the maximum value of $\pi(D)$ in Eq. (1) is achieved when $D\approx 1.4498r$, and the upper bound π_{max} is 0.1665. □

Fig. 7(a) plots function $\pi(D)$ by varying the parameter D with different radii of circular boundary of the hole, r. This upper bound is important because it reveals the optimal improvement the tangent path can achieve. The routing algorithm can also be adaptive to the distance D based on this characteristic. In some applications, the routing algorithm may adopt a threshold ϵ based on the hole size to control the propagation range of the hole information, where ϵ describes the ideal latency the routing algorithm is able to achieve. For example, when the hole radius is 100 distance units and $\epsilon=0.1$, the hole information only needs to be propagated to the nodes within about 400 distance units away from the hole



(a) The shortest path gain $\pi(D)$ for MBC hole.



(b) The shortest path gain $\pi(D)$ for MBR hole.

Fig. 7. Illustration of shortest path gain.

center. From the perspective of routing data packet, the algorithm needs only switch (from original schemes, say greedy routing) to *WCF* or *RCF* when packets enter the zone where nodes cache the hole information, which is controlled by propagation factor p_f.

5.1.2. Minimum bounding rectangle

In the case of bounding a hole using MBR, the percent of latency improvement $\pi(D)$ is:

$$\pi(D) = 1 - \frac{C}{A+B} = 1 - \frac{\sqrt{(\frac{w}{2})^2 + (D-\frac{h}{2})^2}}{D - \frac{h}{2} + \frac{w}{2}} (D > h/2) \quad (4)$$

where w and h denote the width and height of the hole, and D is the distance from S_c to hole center H_c, as illustrated in Fig. 6(b).

Theorem 2. *The upper bound on shortest path gain π_{max} for a Minimum Bounding Rectangle approximation of hole is $\frac{1}{\sqrt{2}}$, which is achieved when $D = \frac{1}{2}(w+h)$.*

Proof. Similar to proof in Theorem 1, by solving derivative $\pi'(D) = 0$ w.r.t. D we have that when $D = \frac{1}{2}(w+h)$, the maximum gain π_{max} is achieved and $\pi_{max} = 1 - \frac{1}{\sqrt{2}} \approx 0.2929$. □

Fig. 7(b) plots a numerical study of $\pi(D)$ for different sizes of rectangle-based approximation of the hole.

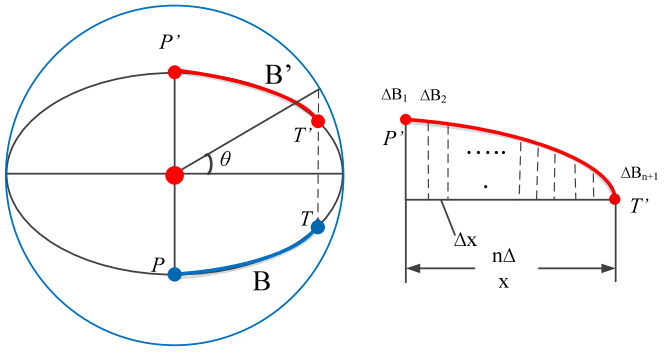


Fig. 8. Calculation the length of an arc segment with Simpson's Rule.

5.1.3. Minimum bounding ellipse

We now consider a hole bounded by an ellipse which is defined by function $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$, where a and b is the major and minor semi-axis, respectively. Assume that the center of the ellipse approximating the hole coincide with the origin of the coordinate system. Then, the coordinates of S_c are $(0, -D)$, and the coordinates of the tangent point T can be expressed as $(a\sqrt{1-(\frac{b}{D})^2}, -\frac{b^2}{D})$, as illustrated in Fig. 6(c). The gain $\pi(D)$ is given by:

$$\pi(D) = 1 - \frac{C}{A+B} \quad (5)$$

where $A = D - b$, $C = \sqrt{a^2(1-(\frac{b}{D})^2) + (D - \frac{b^2}{D})^2}$, and B is the length of arc $\widehat{PT}$. Without loss of generality, we map the arc $\widehat{PT}$ to the first quadrant and compute the arc length $\widehat{P'T'}$. As illustrated in Fig. 8, θ can be expressed as $\arccos(\sqrt{1-(\frac{b}{D})^2})$. Since accurately calculating the arc length of ellipse requires elliptic integral which cannot be obtained in a closed analytical form, we use Simpson's rule to approximate the length of segment $\widehat{PT}$ and have:

$$B = a\int_{\theta}^{\frac{\pi}{2}} \sqrt{1 - \left(\frac{a^2-b^2}{a^2}\right)\sin^2(\phi)}\, d_\phi \approx \frac{\Delta\theta}{3}\left(\Delta B_1 + \sum_{i}^{n/2}\Delta B_{2i} + \sum_{i}^{n/2}\Delta B_{2i+1} + \Delta B_{n+1}\right) \quad (6)$$

where $\Delta\theta = \frac{\frac{\pi}{2} - \arccos(\sqrt{1-(\frac{b}{D})^2})}{n}$, and ΔB_j is given by: $\sqrt{a^2\sin^2\theta_j + b^2\cos^2\theta_j}$ $(\theta_j = \theta + (j-1)\Delta\theta,\ 1 \le j \le n+1)$.

Due to the lack of a closed-form solution – i.e., since we are dealing with approximate values – we state the following property:

The upper-bound on the shortest path gain π_{max} when a Minimum Bounding Ellipse is used to approximate the hole is in the range of [0.1665, 0.229).

To justify this property, we use the observations that:

- When the value of the eccentricity e of the ellipse is very small (i.e., approaches 0), the value of the gain is approaching the one obtained when the Minimum Bounding Circle is used. As an illustration, in Fig. 9 we see that the value of π_{max} is 0.16647 when $e = 0.01$.
- When $e \to 1$, the maximum gain is very close to that of a rectangle-based approximation of the hole, e.g., the rightmost elliptical approximation ($e = 0.99999$ has $\pi_{max} = 0.29282$), which approaches the maximum gain for a rectangle-based approximation of the hole ($\pi_{max} = 0.2929$). Thus, as illustrated, the maximum gain π_{max} for an ellipse-based hole approximation is distributed in the range of [0.1665, 0.2929), and

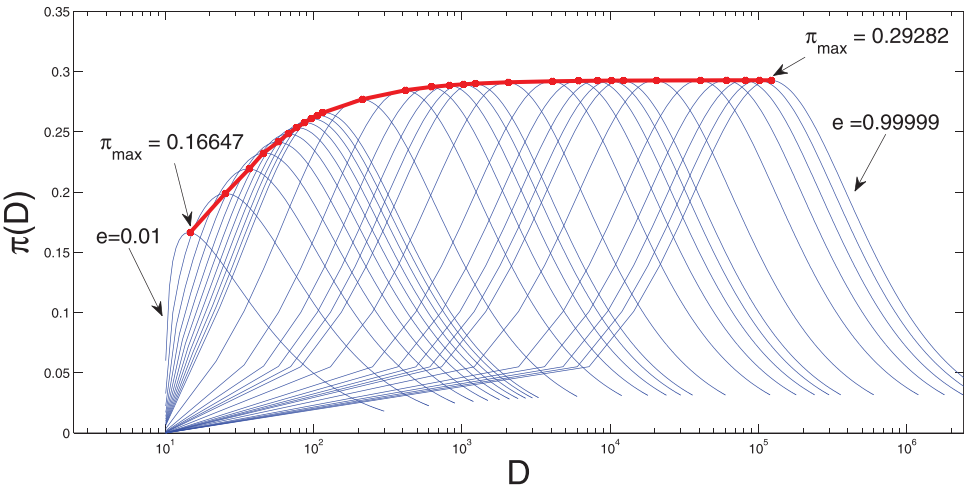


Fig. 9. The shortest path gain $\pi(D)$ for MBE hole. We increase the distance $D(D > b)$ from S_c to H_c (x-axis), and plot $\pi(D)$ for different ellipse holes, indicated as the blue lines, by varying major semi-axis a. Red bold line connects the points of the maximum values π_{max} for different hole size. Note the logarithmic scale of x-axis. Parameters: minor semi axis is fixed as $b = 10$, major axis $e \in [0.01, 0.99999]$), and $n = 20$ intervals for approximating arc length $\widehat{P'T'}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

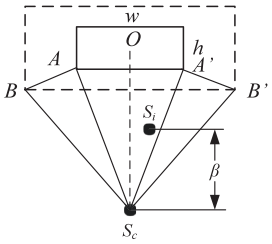


Fig. 10. Path distribution in RCF and WCF.

its value increases with eccentricity e – depicted as the asymptotic line (the red one) of the maximum value of π_{max}.

5.2. On load balancing

One important goal of using multi-path routing mechanism is to spread the traffic among the nodes in the network, thereby to improve load-balancing. Also it may eliminate or at least alleviate the congestion at the hot-spot areas. In RCF, the paths for routing are distributed in the ring area determined by the coverage hole and expanding factor d_f. Compared to RCF, WCF consists of more paths distributed in the wedge areas enclosed by the ring of RCF. Thus, WCF provides better load-balancing than RCF.

Let t_r, I and D_p denote the nodes' transmission rate, transmission interval and the data size of each packet, respectively. Then the time spent for data transmission at each node is D_p/t_r, and there are T/I packets sending from the source node if the total simulation time is T. The total data size D_t transferred from a source node over the network is therefore $T \cdot D_p/I$. In addition, we assume that there is no congestion in all nodes, i.e., I is set to satisfy $I > D_p/r$. Then we have following results.

Theorem 3. *The traffic of nodes is inversely proportional to the area of routing paths distributed, and WCF routing always achieves better load-balancing than RCF.*

Proof. We now consider the case for rectangular hole approximation, as depicted in Fig. 10. It is easy to obtain that the area of paths distributed in WCF, say A_{WCF}, equals to the area A_{RCF} that used in RCF ($\triangle ABS_c$ and $\triangle A'B'S_c$), plus a wedge area A_{Wedge} ($\triangle AA'S_c$). With simple geometric computing we have:

$$\begin{aligned} A_{WCF} &= A_{RCF} + A_{Wedge}, \\ A_{RCF} &= \frac{w \cdot D \cdot (d_f - 1)}{4}, \\ A_{Wedge} &= \frac{w(D - h/2)}{2}. \end{aligned} \tag{7}$$

where w, h and d_f is the width, height, and expanding factor of MBR, respectively. Therefore, the number of nodes in area A_{RCF} and A_{WCF} is $\lambda \cdot A_{RCF}$ and $\lambda \cdot A_{WCF}$, respectively, where λ is the node density. If the nodes are distributed uniformly and independently in the deployed area with a Poisson point process, the probability that there are n nodes residing in an area A is:

$$Pr(N_A = n) = \frac{e^{-\lambda A} \cdot (\lambda A)^n}{n!} \tag{8}$$

The average traffic of nodes in RCF and WCF routing can be respectively computed as $T_{RCF} = \frac{T \cdot D_p}{I \cdot \lambda \cdot A_{RCF}}$ and $T_{WCF} = \frac{T \cdot D_p}{I \cdot \lambda \cdot A_{WCF}}$. Since A_{WCF} is always greater than A_{RCF}, we have $T_{RCF} > T_{WCF}$, i.e., WCF achieves better load-balancing than RCF, which is proved in the experiments (Section 6).

However, the accumulated traffic in both methods is not uniformly distributed among the intermediate nodes. According to the characteristics of both WCF and RCF routing, the closer is a node to the source node (or sink node), the more flow it participates in relaying. For example, in the WCF routing, the traffic of a node S_i, located a distance β from the line orthogonal to $\overline{OS_c}$ at point S_c, can be mathematically approximated as:

$$T(S_i) = \frac{T \cdot D_p \cdot (D - \frac{h}{2})}{I \cdot k \cdot \beta} \tag{9}$$

where we assume there are k disjoint paths employed for routing and the throughput is evenly distributed over the k paths.

Our interest is to minimize both $T(S_i)$ and communication delay $\chi(S_c, S_k)$ simultaneously. Eq. (9) reveals that with fixed other parameters, $T(S_i)$ is inversely proportional to the number of paths k and distance β. □

Since k depends on the area of path distribution A, increasing which, however, may also detour the paths and thus increase $\chi(S_c, S_k)$. As we will observe in the experiments, there is a trade off between $T(S_i)$ and $\chi(S_c, S_k)$ and should be balanced when bypassing holes, which is our goal in this work.

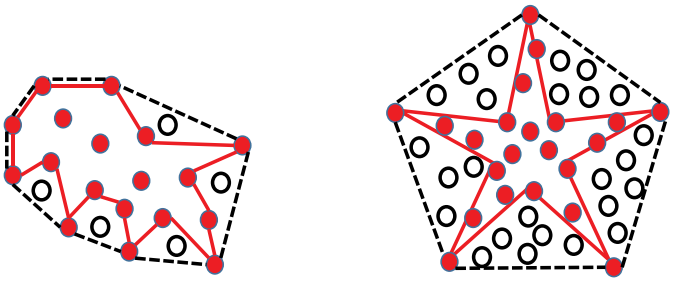


Fig. 11. Convex vs. concave hull and hole boundary.

5.3. Further practical consideration

So far, we have focused on the various benefits (respectively, trade-offs) when analyzing the proposed approximation approaches – *modulo convex hull* of the corresponding holes in the sensor networks. In other words, we have implicitly assumed that the convex hull is a "reasonably good" approximation of the hole, and then worked towards its more compact representation. However, in practice there may be cases when this premise is not quite appropriate for a different reason. Namely, when approximating the boundary of a given hole with its convex hull *CH*, the interior of *CH* may contain a non-negligible amount of "healthy" nodes (i.e., nodes that could still be used for both sensing and transmission) that will be treated as dead ones for all practical purposes. Thus, in this section we compare the convex hull approximation of the (boundary of the) holes with the one using *concave hull* (denoted CH^C). Concave hulls of point sets have been used in both computational geometry and image processing and the first such approximation were the, so called, α-shapes [27]. Subsequently, the χ-shapes were introduced as a more desirable formalism in the sense of better efficiency and avoiding some scenarios where α-shapes could yield a non-closed polygonal representation of the boundary of a given point-set. In our recent work [28] we provided a distributed algorithm for estimating the boundaries of shapes in sensor networks – although the work was tackling a complementary problem, i.e., detecting a boundary of a contiguous shape in which every sensor's measurements of a particular phenomenon exceed a given threshold-value.

The issues are illustrated in Fig. 11. Each part of Fig. 11 shows a setting consisting of both dead nodes, represented as solid (red-colored) disks; and operational nodes, represented as empty disks. In the left portion of Fig. 11 we see that the CH^C of the dead nodes is a pentadecagon, whereas the *CH* is an octagon. In turn, the right portion shows a scenario in which the CH^C is a decagon whereas the *CH* is a pentagon. While in each setting the "description cost" of *CH* is approximately 50% of the one using CH^C – there are other important differences. Namely, the number of operational nodes that will be rendered "unavailable" in the left portion of Fig. 11 is <20% of the number of dead nodes. However – the number of operational nodes that will be declared dead for all practical purposes, because they are inside *CH* in the right portion of Fig. 11, is >50% of the *total* number of nodes – i.e., it is greater than the number of nodes defining the hole. As much as this may be a pathological case unlikely to occur in practice – the example depicted in the right portion of Fig. 11 brings another aspect to light: it is not only the ratio of description-size of the boundaries [3] – or, for that matter, the relative ratios of the respective areas or perimeters of *CH* and CH^C – but one may need to also consider the number of operational nodes that may be declared useless, when approximating a boundary of a hole.

While a detailed formal study of this type of trade-offs is beyond the scope of this article, in Section 6.4 we present certain quantitative observations that shed a light on the issue of ignoring nodes that could still be used for sensing and/or communication.

6. Experimental evaluation

We compare our proposed hole-bypassing routing heuristics with the face routing algorithm [23] in several settings.

The experiments were performed on the open source simulator for WSN, SIDnet-SWANS [11]. Each run simulates 1200 homogeneous sensor nodes configured as: (1) 20 kbps radio data rate on the MAC 802.15.4 protocol; (2) 5 s idle-to-sleep interval of inactive nodes to preserve battery power, and 2-s interval of data transmission of source node; (3) power consumption characteristics meet the specifications of Mica2 Motes; (4) fully-charged battery with initial capacity 25 mAh.

We evaluate over the lifetime of sensor nodes, in network load-balancing, and the communication latency. Specifically, the quality of the load-balancing is measured by the standard deviation of the distribution of energy consumption.

Each setting was tested for: (1) 2 node densities ($\lambda \in 12, 24$ is the average neighbors per node); (2) 3 hole expanding factors ($d_f = 1.25$, $d_f = 1.50$ and 1.75); (3) 3 different hole sizes (Hole = 1%, 5% and 15% of field area); (4) Random choice between 1 up to 4 *(source, sink)* pairs unless otherwise specified. The holes were generated using pentagons and hexagons which were "deformed" by moving the mid-point of the edges towards the interior of the polygon for a randomly selected factor of 20%–60% of the edge's length, and then randomly perturbing the initial location of the vertices within a disk of size 5% of the edge's length. The experiments provide 72 distinct configurations, each of which has 10 runs. In the sequel, we present the averaged observations of all the runs unless otherwise specified.

6.1. Minimum bounding on real GIS data

To demonstrate the effects of minimum bounding algorithms proposed in Section 3, we first construct Convex Hull, MBR, MBC and MBE using Minnesota Department of Natural Resources(MDNR) GIS data deli.[4]

Fig. 12 shows the minimum bounding shapes (MBS) on Minnesota state, which is outlined by blue dots. In this case, the difference between (minimum) bounding ellipse and circle is not obvious due to the eccentricity e of the MBE is very close to 0 (ref. Section 5.1.3).

We also construct minimum bounding shapes using a Minnesota Lake (plotted with blue dots) GIS data, which are illustrated in Fig. 13. In this case, MBE is a more "tight" approximation than MBC (and also MBR) because of higher eccentricity of the bounding ellipse, which may incur a larger values of shortest path gain π_{max} (ref. Section 5.1.3 and following experiments).

6.2. Results on MBC approximation

This section reports the experimental observation when the hole is approximated with a circle.

Figs. 14–16 plot the standard deviation of 3 approaches under different settings, which explain the network load-balancing. When *time* < *3 h*, all of the three methods have (almost) the same load-balancing because no nodes or few nodes die in this

[3] We are trying to avoid the use of the phrase *complexity of the description* because that falls in the realm of Kolmogorov complexity [29] and is way beyond the scope of the current work.

[4] http://deli.dnr.state.mn.us.

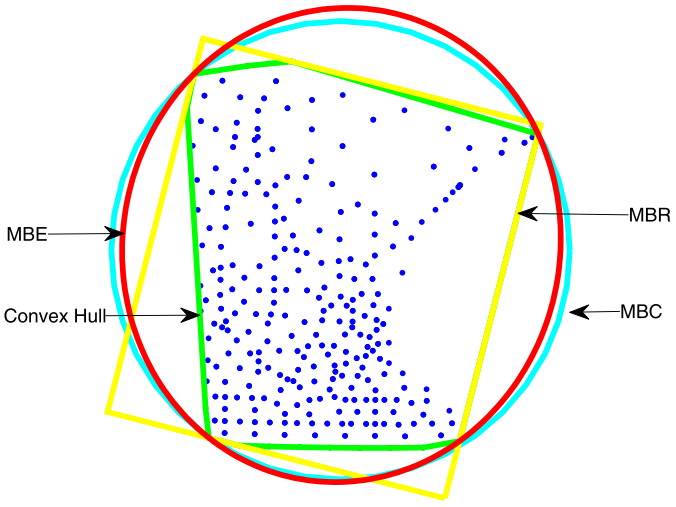


Fig. 12. MBS Construction on Minnesota state.

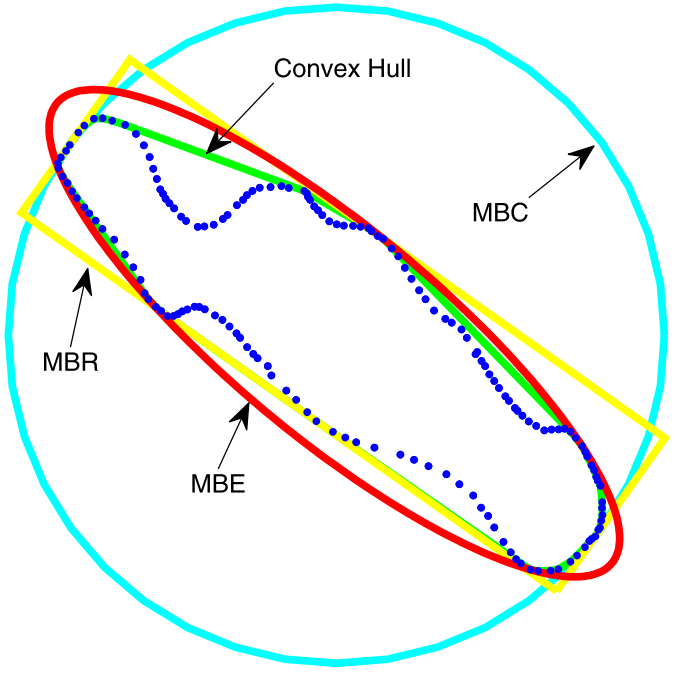


Fig. 13. MBS Construction on a Minnesota lake.

period of time. As the time increase, RCF and WCF achieves better load-balancing (smaller standard deviations) due to employing more paths for routing. Finally ($time > 12\ h$), the stand deviation of all approaches would merge. The reason is that after depleting all available paths, the network load-balancing in all methods should return to the same level.

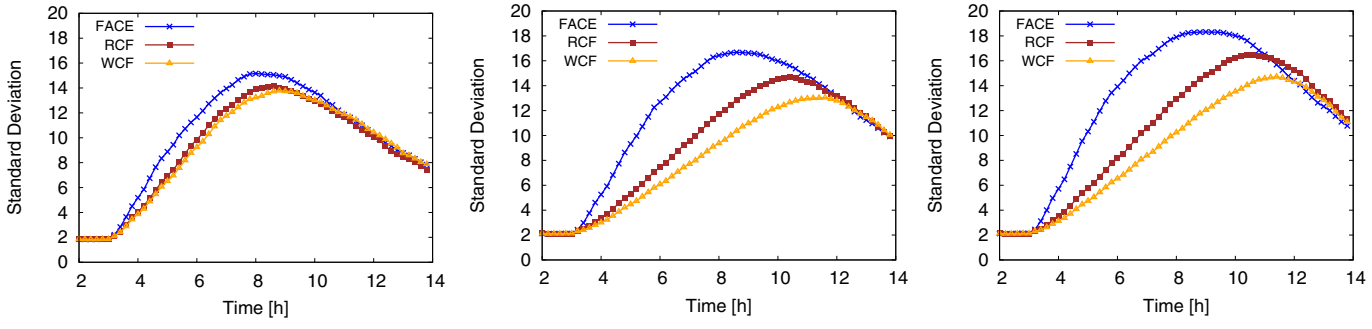
As shown in Fig. 14, when the hole size is small, e.g., 1% of field area, the effects of the three approaches are very similar (Fig. 14(a)). Also, the standard deviation will increase with hole size because more relay nodes are required to bypass the coverage hole. However, face routing is more sensitive to the hole size than RCF and WCF – due to the lack of path diversity. While the length of routing paths in RCF and WCF also increase with the hole size, more available routing paths can be explored and will reduce the standard deviation of the energy consumption.

Fig. 15 shows for $\lambda = 12$, RCF and WCF respectively achieves 13% and 24% improvement on the load-balancing over face routing. For $\lambda = 24$, the improvement decreases to 11% (RCF) and 20% (WCF). Note that in both experiments, WCF outperforms RCF, due to employing more available routing paths.

Fig. 16 shows the impact of expanding factor d_f on the performance of routing schemes RCF and WCF (for convenience of comparison, we also plot face routing although it is not impacted by d_f). As illustrated, increasing d_f affects the performance of both RCF and WCF in similar manners. First, increasing d_f (e.g., $d_f = 1.5$) explores more permissible paths for both RCF and WCF, which may utilize a larger fraction of the nodes to share the communication costs. However, increasing d_f (e.g., $d_f = 1.75$), especially for larger hole size, also increases the path-length of both RCF and WCF routing schemes, which may incur more energy consumption, and consequently compensate for the load-balancing gain of exploring more available routing paths.

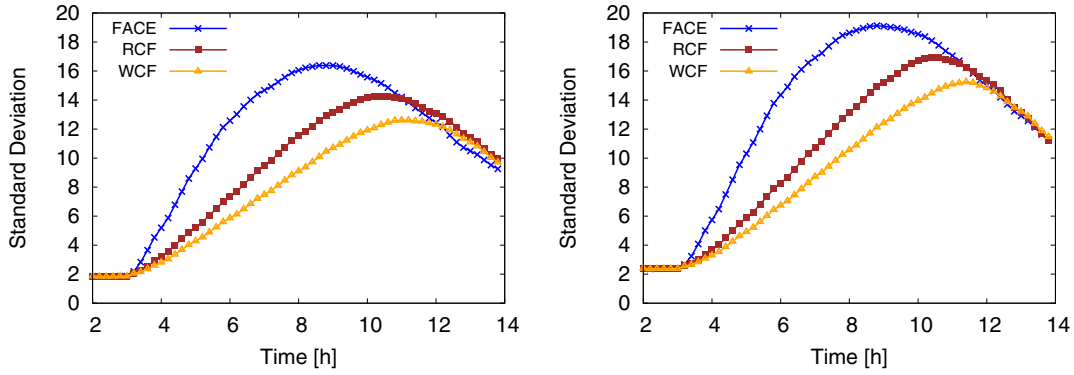
Fig. 17 compares the number of *(source, sink)* pairs on in-network load-balancing. It is obvious that a larger number of pairs may deplete the nodes faster while a smaller number of pairs can extend the lifetime of a network. For example, when $Pairs = 16$, the network becomes unavailable quickly, e.g., around 9 h in this reporting. However, the efficiency of WCF routing –we only plot the results of WCF for clarity– on balancing in-network energy is maintained no matter how to increase the burden of a WSN.

Fig. 18(a) shows the life time of nodes, averaged by all parameter settings, where the time is measured based on three different "policies": (1) first dead node; (2) 5% dead nodes; and (3) 10% dead nodes. For a lifetime metric of 15% of dead nodes, RCF and WCF achieve 1.2 h (14%) and 2.0 h (23%) of additional lifetime than face routing. When the lifetime metric is reduced to the first dead node, the improvements are even higher (2.6 h (52%) and 3.4 h (68%), respectively). This proves the effectiveness of RCF



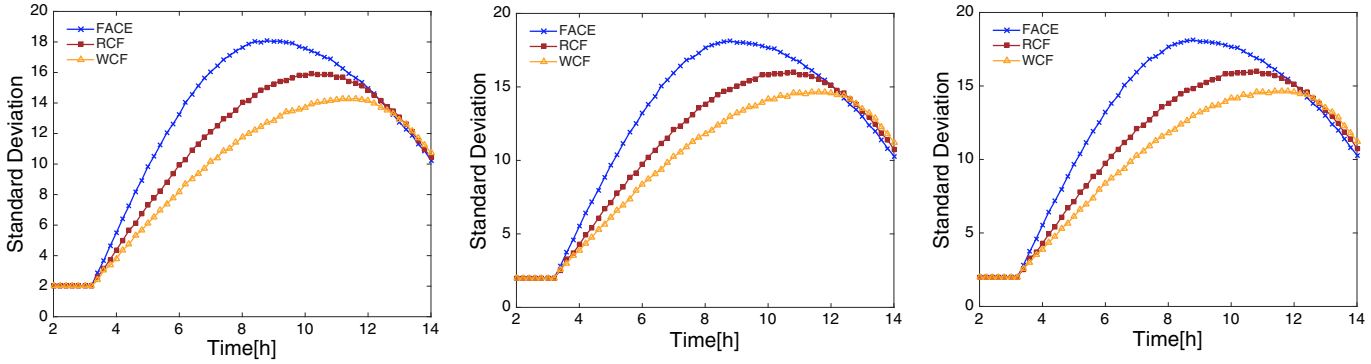
(a) $Hole = 1\%$ of field area (b) $Hole = 5\%$ of field area (c) $Hole = 15\%$ of field area

Fig. 14. Impact of hole size on the load-balancing. $\lambda = 24$ and $d_f = 1.25$.



(a) $\lambda = 12$ (b) $\lambda = 24$

Fig. 15. Impact of node density λ on the load-balancing. *hole* = 15% and $d_f = 1.25$.



(a) d_f=1.25 (b) d_f=1.50 (c) d_f=1.75

Fig. 16. Impact of expanding factor d_f on the load-balancing. $\lambda = 24$ and *hole* = 5%.

and WCF on conserving the energy and extending the lifetime of nodes. Again, WCF's effect on load-balancing turns out to be superior, now in terms of lifetime. We also found that when the dead nodes exceed 15% (16%–18% in most cases, depending on configurations), the network may get disconnected and no available paths can be leveraged.

Communication latencies are compared in terms of the time to transmit a packet from S_c to S_k. As illustrated in Fig. 18(b), RCF and WCF yield 11% and 6% improvements when compared to face routing. RCF performs better than WCF due to the use of almost-shortest paths employed to route packets. The latency of each approach increases as the time evolves, because the nodes around the coverage hole die faster, thereby expanding the hole itself, which leads to longer path lengths overall. Fig. 18 explains the trade-offs that we mentioned previously – while WCF achieves better load-balancing (in terms of lifetime), RCF incurs smaller latency.

Fig. 19 depicts the average residual energy of in-network nodes. As shown, the proposed routing approaches perform slightly better than face routing, and RCF outperforms WCF because it employs fewer hops per routing packet.

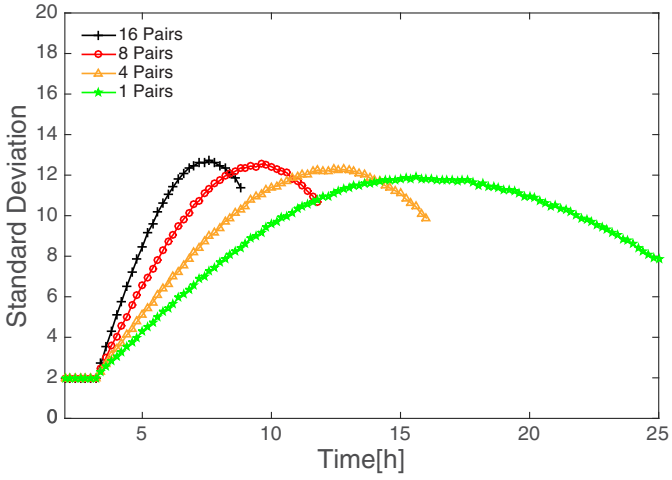


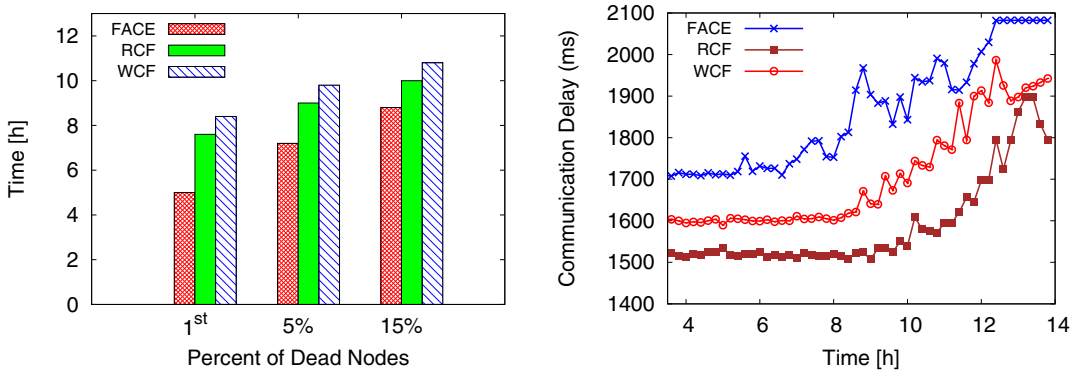
Fig. 17. Comparison of *(source, sink)* pairs using WCF Routing. $\lambda = 24$, *hole* = 15% and $d_f = 1.25$.

6.3. Results on MBE approximation

We now report the observations of bounding holes with ellipse (MBE), in contrast to MBC approximation in above results.

Fig. 20 illustrates the impact of eccentricity of the ellipse hole on the load-balancing of WSN. For $e = 0.75 (b = 1.5a)$, RCF and WCF prolong the lifetime of networks than face routing due to explore more paths. For a larger value of $e = 0.94$ ($b = 3.0a$, Fig. 20(b)), each path requires more nodes to bypass the hole, and thus increases the energy consumption of network. As it shows, the standard deviation in RCF increases slightly, while in face routing increases significantly. In addition, the effect of increasing length of routing path is compromised by incorporating more paths in WCF, which may even experience a slightly lower standard deviation in higher value of e.

Fig. 21(a) shows the communication latency for different algorithms to bypass an ellipse hole with eccentricity $e = 0.94$. From



(a) Lifetime of nodes comparison. (b) Communication latency comparison.

Fig. 18. Trade-offs between latency and lifetime.

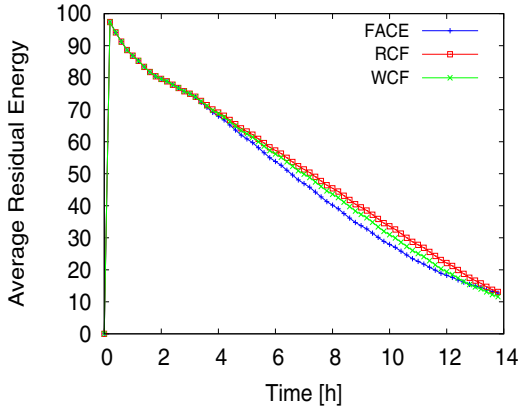


Fig. 19. Average residual network energy.

Fig. 21(a) we observe that both RCF and WCF experience increasing on communication latency which is the result of hole expansion after depleting the nodes along the original boundary and those residing in the wedge area. An interesting result is that face routing undergoes a "U-turn" on the communication latency which is an "unconscious" effect of this scheme. This is reasonable because it routes in a greedy manner at the beginning. Namely, after exhausting the energy of nodes residing in the wedge area, its routes will be "bent" to the (almost) shortest path, which results in the decrease of communication latency. However, the communication time would return to the baseline as the nodes along the shortest path dissipating its energy and become unavailable

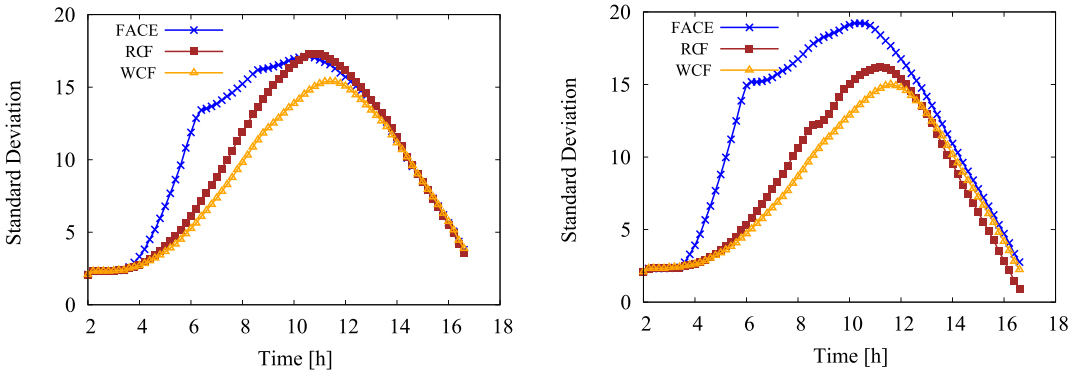
Finally, Fig. 21(b) plots the improvements of RCF and WCF over face routing on communication latency, where the horizontal line denotes the optimal (maximum) gain π_{max} one can achieve in theory. Obviously, the performances of both RCF and WCF deteriorate with time because nodes along the optimal paths become unavailable due to draining the battery.

We note that the results of MBR construction are not presented. The reason is that when $e \to 1$, the MBE is very close to MBR. Since we report the results of MBE, we ignore MBR here which has almost the same figures with MBE when $e = 0.94$.

From above presented experiment results, the proposed algorithms not only improve network load-balancing, but also reduce routing latency. For load-balancing, both algorithms achieve better results in a larger hole (Fig. 14) and/or in a sparser network (Fig. 15). On the metric of percentage of dead nodes, RCF and WCF extend the lifetime of nodes, especially for the first few dead ones (Fig. 18(a)). On communication latency, both RCF and WCF reduce packet delay (Fig. 18(b)) and their benefits depends on the distance between hole and sink nodes (discussed in Section 5).

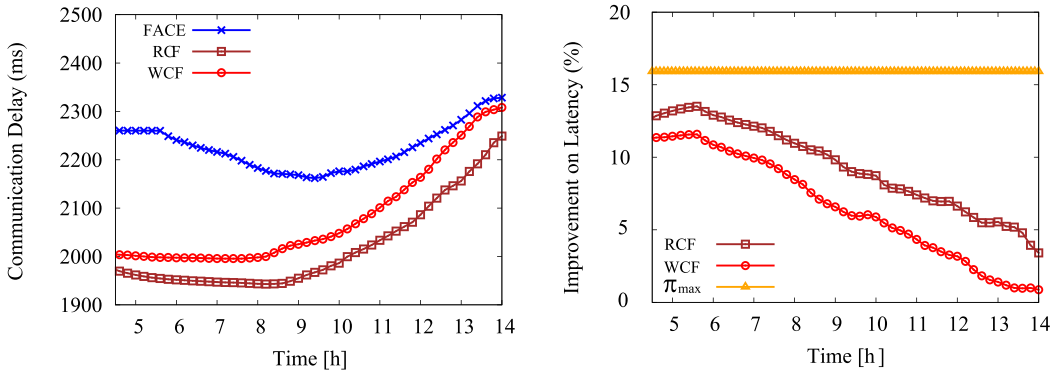
6.4. Compactness vs. loss of available nodes

As discussed in Section 5.3, using the convex hull for initial approximation of the hole boundary may yield certain compactness – however, it may incur other kinds of costs, namely, casting nodes that are actually "alive" into the category of unusable ones. We now give an empirical evaluation of the impact of the CH-based



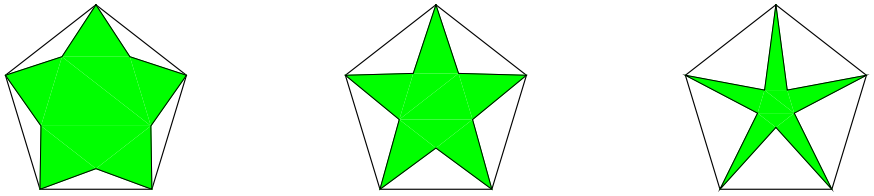
(a) $e = 0.75$ (b) $e = 0.94$

Fig. 20. Impact of eccentricity e on the load-balancing.



(a) Communication latency. (b) Improvement on communication latency

Fig. 21. Impact of eccentricity e on load-balancing.



(a) 25%. (b) 50% (c) 75%

Fig. 22. Impact of the convex hull based approximation of the holes.

approximation on the energy-waste in terms of not using otherwise available nodes. Specifically, the compactness-induced loss of a convex hull is defined as the ratio of the operational but unavailable nodes (because of CH bounding) to the all nodes residing in a CH – both operational and dead ones. As illustrated in Fig. 22, we use 3 different settings, each of which has the same "savings" in terms of the description of the hole's boundary using the convex hull vs. the one with the concave hull – however, the ratio of the actually-dead nodes vs. the alive ones (but useless) varies, to conduct the evaluation. We assume a uniform distribution of the nodes, and our running scenario was a pentagon, the area of which was 15% of the entire network. As shown, we varied the area of the CH^C, relative to the one of CH.

Fig. 23 shows the comparative observations of the impact of using different bounding shapes on the time until the depletion of 10% of the nodes. Obviously, convex hull based hole approximations (such as MBC, MBR or CH itself) achieve the relatively similar result. In comparison, if one relies on the concave hull bounding of the boundary of the hole and uses face-routing, the time until 10% of the nodes are depleted is decreased. The phenomenon is further accentuated when the ratio of the areas (respectively, the number of nodes) of the CH^C and CH is smaller. The reason for it is that the routing will attempt to use as many nodes along the boundary as possible, and as often as possible – thereby depleting those nodes sooner. Clearly, one would want to have some practically applicable criteria to balance the exploitation of the nodes inside the CH but outside CH^C – however, such study is a subject of a future work.

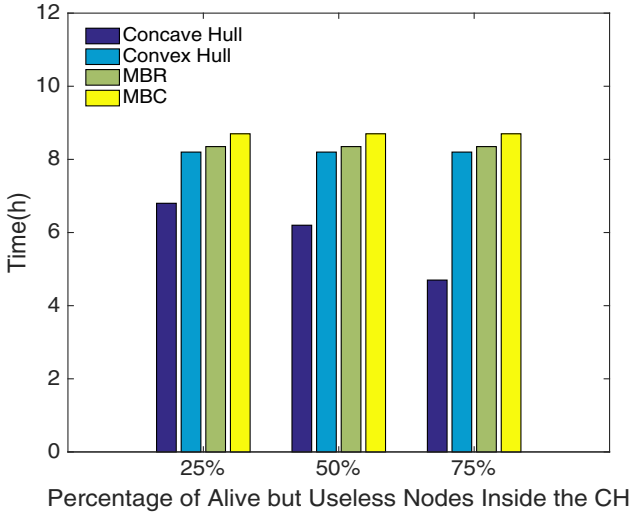


Fig. 23. Time of 10% exhausted nodes.

7. Related work

Problems related to detecting, representing and bypassing holes in WSN settings have been investigated since the emergence of the field (see [1] for a survey). Theoretical analysis on energy-hole properties in multi-hop networks have been conducted by several existing works. Analytical model for energy hole problem in WSN assuming uniform node distribution has been presented in [30]. In [31] a formal proof is presented that the occurrence of an energy hole is inevitable under certain conditions. An investigation of the energy-balance problem with the many-to-one traffic pattern undertaken in [32] has demonstrated that nearly balanced energy depletion is still possible if the number of nodes increases in geometric progression.

There have been numerous proposals for coverage hole and boundary detection based on topological or geometric techniques. Relying on homology, the problem of coverage and hole detection was addressed in [33] and, more recently, distributed algorithms for finding and patching connectivity holes were presented in [34]. In [8] a heuristic for detecting holes is proposed, based on topological properties with only connectivity information. Distributed approaches for locating and bypassing holes based on geometric techniques were used in [3], in which the communication graph follows the unit-disk graph assumption. In addition, a statistical model for boundary detection is presented in [35], which explores the property that nodes on the boundary have much lower average degrees than interior nodes. A distributed approach for boundary recognition based on communication graph is developed in [14], which identifies the homotopy types for distinct shortest paths. In addition, a distributed generation of convex hulls was used for coverage holes' detection in [7]. Our work does rely on the existing literatures for hole detection and representation techniques, however, we also tackled the problem of bypassing the hole (in a sense) "earlier" and presented two algorithms towards that end, keeping in mind the delay-overheads.

A number of geographical protocols have been proposed for circumventing holes. Among the first popular geometric routing approaches with a guaranteed packet delivery is the compassed routing II (a.k.a. face routing), presented in [36]. The key idea of face routing is to construct a planar graph locally and to forward a message along one or possibly a sequence of adjacent faces which are providing progress towards the destination node [4]. Several variants that assure bypassing local minima but differs in recovery strategies have been developed based on face routing, e.g., [5,6,23,37], although none of them outperforms the original one in the worst case. For instance, a hybrid routing algorithm called GOAFR+ was proposed in [6], which combines the greedy routing and face routing, and switches between each other. Formal comparisons and proofs of several proposed face routing algorithms on guaranteeing packet delivery in planar graph can be found in [4]. However, these protocols suffer from blind detouring problem which arises from routing along a detour path due to hole information is unknown. It has been proved in [6] that the path length can grow as much as $O(L^2)$ in a sparser deployment network, where L is the optimal path length.

In order to overcome the blind detouring problem, some geographic routing protocols, e.g., [38–40], have been proposed to shorten the path lengths. In [38], a visibility graph is constructed to find the shortest path and the hole is represented by a polygon. However, this work incurs communication overhead due to building the visibility graph. HRR protocol [39] regularizes a hole with an ellipse such that an intermediate node can decide which side of the hole has shorter path length to forward the packet. However, the path length is not guaranteed. A convex hull construction algorithm to represent the hole and GOAL routing protocol to achieve a constant path stretch was presented in [40]. However, the optimal path is also not guaranteed in GOAL because the convex hull approximation of the hole. We share the same objectives with these works, namely, to solve the blind detouring problem by exploring the hole information. However, in comparison, we propose bypassing hole algorithms that can address the issue of load-balancing, and can prolong the lifetime of nodes especially for those along the hole boundary.

Several techniques based on, so called, virtual coordinates have been proposed towards geographic routing with or without coverage holes [24,41–45]. Most of these works embeds the original network graph in Euclidean space into different metric spaces such as hyperbolic and Riemann surfaces, and then to deliver the message along the virtual coordinates in the transformed space systems. However, most of these approaches only focus on discovering a single path for delivering packet between a given source-sink pair. The approaches in [42,43] explore multiple paths by applying Möbius transformation to generate different conformal mapping; however, the quality of routing in terms of communication latency is not discussed – whereas we provide the analysis of our results, along with experimental verifications.

Multi-path routing has been extensively studied in both Internet context as well as WSNs – with the objectives of improving throughput or reducing network congestion [46–48]. Analytical models and comparisons of single path and multi-path routing in ad hoc networks are presented in [49,50]. Works such as [51,52] provided schemes on exploring multi-paths for communication between source-destination pairs and thus improving the load balancing of networks. However, how to balance load distribution and routes length while bypassing coverage hole was not explicitly addressed in these works.

8. Conclusions and future work

In this work we addressed the problem of coupling the management of the lifetime of the nodes along the boundary of a communication hole in WSNs with the routing latency. Motivated by the observation that nodes along the boundary are likely to be more frequently used during bypassing on behalf of queries from different *(source, sink)* pairs, we proposed two heuristics – RCF and WCF – for trading off the load balancing among the nodes and the communication latency. To cater to the different shapes of holes, we investigated approximate representations of it via minimum bounding circles, ellipses or rectangle, respectively. More specifically, we firstly generate the convex hull of the polygon bounding the hole and then proceed with approximating it with one of the three shapes – based on the minimum "waste" of the coverage. The source node decides which routing policy to use and our experiments demonstrated that typically RCF provides shorter delay for packets delivery, while WCF has a better load balancing. We also analyzed the upper bounds on reducing transmission latency. Our experiments demonstrated that the proposed approaches not only prolong the lifetime of the nodes along the hole boundary and decrease communication delay, but also achieves desirable load balancing, when compared against the baseline approach from [23].

Currently, we are investigating adaptations of the approaches proposed in this work towards several different contexts and settings. Our main thrust is on considering multiple holes in the WSN. Among the other few extensions that we are also working on, one is focusing on including the mobility of the sink-nodes in the model [53,54] and investigate how that would impact the balance between latency and prolonging the lifetime of boundary nodes. The other extension considers the impact of having heterogeneous nodes and location-uncertainty in tracking scenarios [55,56], and varied routing algorithms towards multiple holes situations.

Acknowledgment

This work was supported by the U.S. National Science Foundation (Grants CNS-0910952, CNS-1646107, CCF-0830149, IIS-1213038 and IIS-1212508), ONR grant N00014-14-10215, National Natural Science Foundation of China (Grant No. 61602097 and No. 61672135), Sichuan Science-Technology Support Plan Program (No.2016GZ0065), and the Fundamental Research Funds for the Central Universities (No.ZYGX2015J072).

References

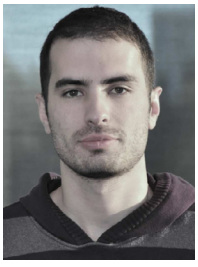
[1] N. Ahmed, S. Kanhere, S. Jha, The holes problem in wireless sensor networks: a survey, ACM SIGMOBILE Mobile Comput. Commun. Rev. 9 (2) (2005) 4–18.
[2] I. Khan, M. Javed, Hole healing energy aware algorithm for wireless sensor networks, Int. J. Comput. Theory Eng. 2 (5) (2010) 1793–8201.

[3] Q. Fang, J. Gao, L. Guibas, Locating and bypassing holes in sensor networks, Mobile Netw. Appl. 11 (2) (2006) 187–200.
[4] H. Frey, I. Stojmenovic, On Delivery Guarantees of Face and Combined Greedy–face Routing in Ad Hoc and Sensor Networks, in: MOBICOM, ACM, 2006, pp. 390–401.
[5] B. Karp, H.T. Kung, Gpsr: Greedy Perimeter Stateless Routing for Wireless Networks, in: MobiCom, ACM, 2000, pp. 243–254.
[6] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric Ad-hoc Routing: Of Theory and Practice, in: PODC, ACM, 2003, pp. 63–72.
[7] L. van Hoesel, P. Havinga, Distributed coverage area reporting for wireless sensor networks, Centre for Telematics and Information Technology University of Twente, 2009 Technical report tr-ctit-09-16.
[8] S. Funke, Topological Hole Detection in Wireless Sensor Networks and Its Applications, in: Proceedings of the 2005 joint workshop on Foundations of mobile computing, ACM, 2005, pp. 44–53.
[9] A. Basu, J. Gao, J. Mitchell, G. Sabhnani, Distributed Localization Using Noisy Distance and Angle Information, in: MobiHoc, ACM, 2006, pp. 262–273.
[10] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, R. Morris, A Scalable Location Service for Geographic Ad Hoc Routing, in: MobiCom, ACM, 2000, pp. 120–130.
[11] G. Trajcevski, F. Zhou, R. Tamassia, B. Avci, P. Scheuermann, A. Khokhar, Bypassing Holes in Sensor Networks: Load-balance vs. Latency, in: GLOBECOM, IEEE, 2011, pp. 1–5.
[12] L. Fang, W. Du, P. Ning, A Beacon-less Location Discovery Scheme for Wireless Sensor Networks, in: INFOCOM, Vol. 1, IEEE, 2005, pp. 161–171.
[13] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network, in: IPSN, Springer-Verlag, 2003, pp. 333–348.
[14] Y. Wang, J. Gao, J. Mitchell, Boundary Recognition in Sensor Networks by Topological Methods, in: MobiCom, ACM, 2006, pp. 122–133.
[15] S. Funke, C. Klein, Hole Detection or: "How Much Geometry Hides in Connectivity? in: Symposium on Computational Geometry, ACM, 2006, pp. 377–385.
[16] M. De Berg, O. Cheong, M.V. Kreveld, Computational geometry: Algorithms and applications, Springer-Verlag New York Inc, 2008.
[17] H. Attiya, J. Welch, Distributed computing: Fundamentals, simulations, and advanced topics, vol. 19, Wiley-Interscience, 2004.
[18] G. Toussaint, Solving Geometric Problems with the Rotating Calipers, in: Proc IEEE Melecon 9 (May), 1983, pp. 1–8.
[19] E. Welzl, Smallest Enclosing Disks (Balls and Ellipsoids), in: New results and new trends in computer science, 1991, pp. 359–370.
[20] B. Gärtner, S. Schönherr, Exact primitives for smallest enclosing ellipses, Inf. Process Lett. 68 (1) (1998) 33–38.
[21] W. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, in: MobiCom, ACM, 1999, pp. 174–185.
[22] D. Niculescu, B. Nath, Trajectory Based Forwarding and Its Applications, in: MOBICOM, ACM, 2003, pp. 260–272.
[23] P. Bose, P. Morin, I. Stojmenović, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Netw. 7 (6) (2001) 609–616.
[24] R. Sarkar, X. Yin, J. Gao, F. Luo, X. Gu, Greedy Routing with Guaranteed Delivery Using Ricci Flows, in: IPSN, 2009, pp. 121–132.
[25] T.-D. Hoa, D.-S. Kim, Minimum Latency and Energy Efficiency Routing with Lossy Link Awareness in Wireless Sensor Networks, in: Int.l Workshop on Factory Communication Systems, 2012, pp. 75–78.
[26] G. Lu, B. Krishnamachari, Minimum latency joint scheduling and routing in wireless sensor networks, Ad Hoc Netw. 5 (6) (2007) 832–843.
[27] H. Edelsbrunner, D.G. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inform. Theory 29 (4) (1983) 551–558.
[28] B. Avci, G. Trajcevski, P. Scheuermann, Managing Evolving Shapes in Sensor Networks, in: Conference on Scientific and Statistical Database Management, SSDBM '14, Aalborg, Denmark, June 30, - July 02, 2014, 2014. 22:1–22:12.
[29] M. Li, P.M.B. Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, Third Edition, Texts in Computer Science, Springer, 2008.
[30] J. Li, P. Mohapatra, An Analytical Model for the Energy Hole Problem in Many–to-one Sensor Networks, in: IEEE Vehicular Technology Conference, Vol. 62, Citeseer, 2005, p. 2721.
[31] S. Olariu, I. Stojmenovic, Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting, in: INFOCOM, 2006, pp. 1–12.
[32] X. Wu, G. Chen, S. Das, Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution, in: IEEE Transactions on Parallel and Distributed Systems, 2007, pp. 710–720.
[33] R. Ghrist, A. Muhammad, Coverage and Hole-detection in Sensor Networks via Homology, IPSN, IEEE Press, 2005. 34–34.
[34] J. Yao, G. Zhang, J. Kanno, R. Selmic, Decentralized Detection and Patching of Coverage Holes in Wireless Sensor Networks, in: Proc. SPIE, Vol. 7352, 2009, p. 73520V.
[35] S. Fekete, A. Kroller, D. Pfisterer, S. Fischer, C. Buschmann, Neighborhood-based Topology Recognition in Sensor Networks, in: Algorithmic Aspects of Wireless Sensor Networks, 2004, pp. 123–136.
[36] E. Kranakis, et al., Compass Routing on Geometric Networks, in: Proc. 11 th Canadian Conference on Computational Geometry, Citeseer, 1999. 34–34.
[37] B. Leong, S. Mitra, B. Liskov, Path Vector Face Routing: Geographic Routing with Local Face Information, in: ICNP, IEEE Computer Society, 2005, pp. 147–158.
[38] G. Tan, M. Bertier, A. Kermarrec, Visibility-Graph-Based Shortest-path Geographic Routing in Sensor Networks, in: INFOCOM, 2009, IEEE, IEEE, 2009, pp. 1719–1727.
[39] P. Li, G. Wang, J. Wu, H. Yang, Hole Reshaping Routing in Large-scale Mobile Ad-hoc Networks, in: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, IEEE, 2009, pp. 1–6.
[40] M. Won, R. Stoleru, H. Wu, Geographic Routing with Constant Stretch in Large Scale Sensor Networks with Holes, in: WiMob, 2011, pp. 80–88.
[41] R. Kleinberg, Geographic Routing Using Hyperbolic Space, in: INFOCOM, IEEE, 2007, pp. 1902–1909.
[42] W. Zeng, R. Sarkar, F. Luo, X. Gu, J. Gao, Resilient Routing for Sensor Networks Using Hyperbolic Embedding of Universal Covering Space, in: INFOCOM, IEEE, 2010, pp. 1–9.
[43] R. Jiang, X. Ban, M. Goswami, W. Zeng, J. Gao, X. Gu, Exploration of Path Space Using Sensor Network Geometry, in: IPSN, 2011, pp. 49–60.
[44] M. Goswami, C.-C. Ni, X. Ban, J. Gao, X.D. Gu, V. Pingali, Load Balanced Short Path Routing in Large-scale Wireless Networks Using Area-preserving Maps, in: MobiHoc, ACM, 2014, pp. 63–72.
[45] K. Huang, C.-C. Ni, R. Sarkar, J. Gao, J. Mitchell, Bounded Stretch Geographic Homotopic Routing in Sensor Networks, in: INFOCOM, 2014 Proceedings IEEE, 2014, pp. 979–987.
[46] P. Djukic, S. Valaee, Reliable Packet Transmissions in Multipath Routed Wireless Networks, in: IEEE Transactions on Mobile Computing, 2006, pp. 548–559.
[47] C. Sengul, R. Kravets, Bypass routing: an on-demand local recovery protocol for ad hoc networks, Ad Hoc Netw. 4 (3) (2006) 380–397.
[48] C. Liu, M. Yarvis, W. Conner, X. Guo, Guaranteed on-demand discovery of node-disjoint paths in ad hoc networks, Comput. Commun. 30 (14–15) (2007) 2917–2930.
[49] P. Pham, S. Perreau, Performance Analysis of Reactive Shortest Path and Multipath Routing Mechanism with Load Balance, in: INFOCOM, Vol. 1, 2003, pp. 251–259.
[50] Y. Ganjali, A. Keshavarzian, Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multi-path Routing, in: INFOCOM, Vol. 2, 2004, pp. 1120–1125.
[51] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, I. Stoica, Balancing Traffic Load in Wireless Networks with Curveball Routing, in: MobiCom, 2007, pp. 170–179.
[52] G. Trajcevski, O. Ghica, P. Scheuermann, M. Zuniga, R. Schubotz, M. Hauswirth, Improving the Energy Balance of Field-based Routing in Wireless Sensor Networks, in: GLOBECOM, 2010, pp. 1–5.
[53] J. Luo, J. Panchard, M. Piórkowski, M. Grossglauser, J.-P. Hubaux, Mobiroute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks, in: DCOSS, 2006, pp. 480–497.
[54] M.M.A. Mohamed, A.A. Khokhar, G. Trajcevski, Energy Efficient Resource Distribution for Mobile Wireless Sensor Networks, in: MDM, 2014, pp. 49–54.
[55] B. Khaleghi, A.M. Khamis, F. Karray, S.N. Razavi, Multisensor data fusion: a review of the state-of-the-art, Inform. Fusion 14 (1) (2013) 28–44.
[56] G. Trajcevski, O. Wolfson, K. Hinrichs, S. Chamberlain, Managing uncertainty in moving objects databases, ACM Trans. Database Syst. 29 (3) (2004) 463–507.

Fan Zhou received his BS degree in Computer Science from Sichuan University, China, in 2003, and his MS and PhD degrees from University of Electronic Science and Technology of China, in 2006 and 2011, respectively. He is currently an Associate Professor at the University of Electronic Science and Technology of China. His research interests include mobile computing and wireless sensor networks.

Goce Trajcevski received his B.Sc. degree from the University of Sts. Kiril i Metodij, and his MS and PhD degrees from the University of Illinois at Chicago, and is currently an Assistant Chairman of the Dept. of Electrical Engineering and Computer Science at Northwestern University. His main research interests are in the areas of Spatio-Temporal Data Management, Uncertainty and Reactive Behavior Management – in both Moving Object Databases and Wireless Sensor Networks settings. In addition to a book chapter and three encyclopedia chapters, he has co-authored over 100 publications in refereed conferences and journals, and has received two best paper awards (CoopIS 2000 and MDM 2010). His research has been funded by BEA, Northrop Grumman Corp., the NSF and the ONR. He has served in the organizing and program committees in numerous conferences and workshops, and was a General Co-Chair of ICDE 2014 as well as a PC Co-Chair of ADBIS 2014 and ACM SIGSPATIAL GIS 2016.

Roberto Tamassia is the Plastech Professor of Computer Science and the Chair of the Department of Computer Science at Brown University. He is also the Director of Brown's Center for Geometric Computing. His research interests include information security, cryptography, analysis, design, and implementation of algorithms, graph drawing and computational geometry. He has published six textbooks and more than 240 research articles and books in the above areas and has given more than 70 invited lectures worldwide. He is an IEEE Fellow and the recipient of a Technical Achievement Award from the IEEE Computer Society for pioneering the field of graph drawing. He is listed among the 360 most cited computer science authors by Thomson Scientific, Institute for Scientific Information (ISI). He serves regularly on program committees of international conferences. His research has been funded by ARO, DARPA, NATO, NSF, and several industrial sponsors. He received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1988.



Besim Avci received his B.Sc degree in computer science from Bilkent University in 2010, and his M.S. degree in computer engineering from Northwestern University in 2013. Currently, he is a Ph.D. candidate at Northwestern University under supervision of Peter Scheuermann and Goce Trajcevski. His research interests include wireless sensor networks, spatial databases, distributed systems and machine learning.

Ashfaq A. Khokhar is the Professor and Palmer Department Chair at The Department of Electrical and Computer Engineering (ECE), Iowa State University. He also holds adjunct appointment in the Departments of Computer Science, ECE, and Health Information Sciences at the University of Illinois at Chicago (UIC). Dr. Khokhar received his B.Sc. in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1985, his M.S. in Computer Engineering from Syracuse University, in 1989 and Ph.D. in Computer Engineering from University of Southern California, in 1993. He served two years as a Visiting Assistant Professor in the Department of Computer Sciences (CS) and School of ECE at Purdue University. In 1995, he joined the ECE Department at the University of Delaware, where he first served as Assistant Professor and then as Associate Professor. In Fall 2000, Dr. Khokhar joined UIC in the CS and ECE department, and served at the rank of Professor and Director Graduate Studies till Summer 2013. From Fall 2013 till Spring 2017, Dr. Khokhar served as Professor and Department Chair of ECE at the Illinois Institute of Technology, Chicago. Dr. Khokhar's research centers on high performance solutions for diverse application area including: computational biology, health care data mining and content-based multimedia modeling. Dr. Khokhar has published over 270 technical papers and book chapters in refereed conferences and journals in the areas of healthcare data mining, wireless networks, multimedia systems, data mining, and high performance computing. He is a recipient of the NSF CAREER award in 1998. He has received numerous outstanding paper awards, and has served as program chair and technical program committee members of leading IEEE/ACM conferences. He is a Fellow of IEEE for his contributions to multimedia computing and databases.

Peter Scheuermann is currently a Professor with the Department of Electrical Engineering and Computer Science, Northwestern University. He has held visiting professor positions with the Free University of Amsterdam, the University of Hamburg, the Technical University of Berlin and the Swiss Federal Institute of Technology, Zurich. During 1997–1998 he served as Program Director for Operating Systems at the NSF. Dr. Scheuermann has served on the editorial board of the Communications of ACM, The VLDB Journal, IEEE Transactions on Knowledge and Data Engineering and is currently an associate editor of Data and Knowledge Engineering. His research interests are in distributed database systems, mobile computing, sensor networks and data mining. He has published more than 120 journal and conference papers. His research has been funded by NSF, NASA, HP, Northrop Grumman, and BEA, among others. Peter Scheuermann is a Fellow of IEEE and AAAS (American Association for the Advancement of Science).