Computer Networks 57 (2013) 1529–1544

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

OFA: An optimistic approach to conquer flip ambiguity in network localization

Xiaoping Wang [a,*], Yunhao Liu [b,c], Zheng Yang [b,c], Kai Lu [a], Jun Luo [a]

[a] School of Computer Science, National University of Defense Technology, ChangSha, China
[b] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong
[c] TNLIST, School of Software, Tsinghua University, BeiJing, China

ARTICLE INFO

Article history:
Received 26 July 2012
Received in revised form 28 December 2012
Accepted 5 February 2013
Available online 13 February 2013

Keywords:
Flip ambiguity
Localization
Optimistic design
Wireless sensor networks

ABSTRACT

Accurate self-localization is a key enabling technique for many pervasive applications. Existing approaches, most of which are multilateration based, often suffer ambiguities, resulting in huge localization errors. To address this problem, previous approaches discard those positioning results with possible flip ambiguities, trading the localization performance for the result robustness. However, the high false positive rate of flip prediction incorrectly rejects many reliable location estimates. By exploiting the characteristics of flip ambiguity, which causes either huge or zero error, we propose the concept of optimistic localization and design an algorithm, OFA, that employs a global consistency check and a location correction phase in the localization process. We analyze the performance gain and cost of OFA, and further evaluate this design through extensive simulations. The results show that OFA obtains robustness with extremely low performance cost, so as to reduce the requirement on the average degree from 25 to 10 for robustly localizing a network.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Location awareness is critical for both data interpretation and network services in wireless sensor networks [1]. Nevertheless, it is impractical to manually configure node positions or equip all nodes with specialized positioning devices (e.g. GPS receiver) in a large-scale cost-sensitive network. Hence, it is of great importance to investigate the self-localization problem: determining node locations through a set of inter-node distance measurements as well as a set of nodes (called anchors) with pre-known locations.

Trilateration (or multilateration) is a widely used method for network localization [2]. The basic idea of trilateration is to position an object using distance measurements to at least three nodes at known locations. Trilateration-based approaches, however, are not robust under noisy distance measurements. Among the problems brought by ranging errors, flip ambiguity is one of the major challenges [3], as it causes huge errors in the location estimation.

Flip ambiguity refers to the case in which a to-be-located node has two possible positions corresponding to a "reflection" across a set of mirror nodes. As illustrated in Fig. 1, the nodes b, c, and d are anchors with known positions and the node a computes its position through measurements d_{ab}, d_{ac}, and d_{ad}. Each measurement defines a ranging circle centered at the reference node with a radius of the corresponding measured distance, as shown by the dashed circles. Owing to measurement errors, three ranging circles do not exactly intersect at a common point. Hence, it is ambiguous to determine whether the position a or a' is the ground-truth location of node a. In this case, an incorrect determination will lead to a huge localization error. Besides single-node flip, a more serious problem is the network-wide flip: a part of the network flips over another part. As shown in Fig. 2, we sequentially localize the

[*] Corresponding author.

E-mail addresses: xiaopingwang@nudt.edu.cn (X. Wang), liu@cse.ust.hk (Y. Liu), yangzh@cse.ust.hk (Z. Yang), kailu@nudt.edu.cn (K. Lu), junluo@nudt.edu.cn (J. Luo).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.02.008

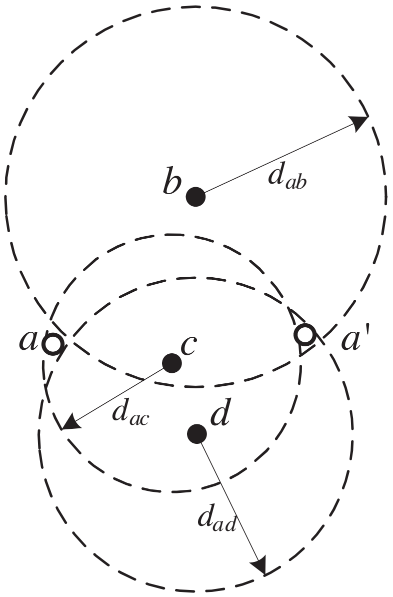


Fig. 1. Illustration of flip ambiguity.

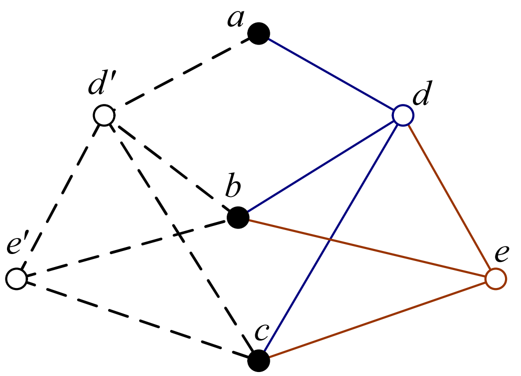


Fig. 2. A network-wide flip.

nodes *d* and *e* by three anchors *a*, *b*, and *c*, which are approximately collinear. If node *d* is wrongly localized to the flipped position *d*′, node *e* will also be flipped to the position *e*′. Eventually, a whole part of the network might be flipped across anchors *a*, *b*, and *c*.

Existing algorithms often address the problem by discarding the location estimates that may suffer from flips [2]. When locating a node, those approaches evaluate the risk of flip and discard the suspicious location estimates. As a result, the performances largely depend on the accuracy of flip prediction. Here, the word "performance" refers to the proportion of nodes that can be successfully localized out of all nodes in a network. Unfortunately, there are no such predictors that accurately identify flips. Imperfect predictors lead to: (1) false positive result: a predictor wrongly reports a correct result as flipped; (2) false negative result: a predictor wrongly reports a flipped result as correct. The aggressiveness of predictors determines a tradeoff between the false positive rate and the false negative rate. By this inherent tradeoff, we have to carefully balance two critical requirements: robustness and performance. To compensate the degradation of performance, we need high network density (over 25 neighbors on average [2]) to fully localize a network, which is practically difficult [4,5]. Clearly, simple discarding is conservative and pessimistic.

In this study, we propose an **O**ptimistic localization scheme with **F**lip **A**voidance (OFA) to achieve both performance and robustness. This design is motivated by the following observations. First, flip ambiguity causes either huge error or zero error. We classify the error sources of a localization result into two kinds: the ranging error and the flip ambiguity. Clearly, the former error cannot be eliminated. The latter error, however, depends on whether a flip is triggered or not. That is, flip ambiguity causes either huge error or no error at all. Second, anchors in a network provide a reliable skeleton to check whether the position assignment suffers possible flips. In contrary to pessimistic designs, OFA assumes that the position assignment for each node is robust, and checks the correctness of the position assignment by the anchor skeleton. If current location estimates match the skeleton, the robustness of the estimates is confirmed; otherwise, OFA will correct the flipped nodes. The advantages of OFA are twofold: first, to localize a superset of nodes in contrast to pessimistic designs; second, to guarantee robustness for flips.

Major contributions of this work are as follows.

1. We propose the concept of optimistic localization to avoid flips and design a distributed localization algorithm.
2. We introduce two key mechanisms to support OFA design: the consistency check and the error correction, which process position estimates falling into false positive and false negative predictions.
3. We conduct extensive simulations to evaluate the performance of OFA design. The results show that OFA obtains robustness with extremely low performance cost, so as to reduce the basic requirement on average degree from 25 to 10 for robustly localizing a network.

The rest of this paper is organized as follows. We discuss the related work in Section 2. Next, we present the basic idea of OFA in Section 3 and describe the details of the OFA protocol in Section 4. In Section 5, we analyze the performance and cost of OFA. Experimental studies are presented in Section 6. Finally, we conclude the work in Section 7.

2. Related work

Flip ambiguity is first defined in rigidity theory for the graph realization problem [4], which is closely related to the network localization problem. A number of works employ rigidity theory to investigate network localizability using the connectivity property [5,6]. As those designs are based on the assumption that the distance measurements are accurate, they face difficulties when ranging errors exist.

Under noisy distance measurements, many algorithms aim at minimizing the localization errors. Moore et al. introduce the concept of robust quadrilaterals to avoid flip ambiguity [2]. They build up robust positioning result by the local maps that pass the check for flip prediction. Kannan et al. extend the work to a more general case [7,8]. As a typical pessimistic design, however, their method requires an extremely high network density to localize a whole network. Liu et al. [9] and Yang and Liu [3] propose to estimate and control errors step by step in a sequential localization process. They track the error in each step and select part of the reference nodes to minimize the expected error of the result. Nevertheless, we cannot avoid flip ambiguity by error control, because flip ambiguity can be triggered by even tiny errors. In addition, as flip ambiguity is a kind of faults in localization, we can detect and correct the flipped nodes. In contrast, error-control-based algorithms neglect such faults.

Basu et al. investigate the localization with both distance and angle measurements [10]. They relax the problem to the convex form and use linear programming to solve the relaxed problem. Their design can avoid flip ambiguity and provide error estimation. However, the algorithm cannot work well when either distance or angle measurement errors do not have a clear bound. Moreover, since it relies on the knowledge of both distance and angle measurements, the design is not always practical.

Goldenberg et al. propose Sweeps algorithm [6] by tracking all possible results of bilateration to improve the performance of localization in sparse networks. Bilateration causes another kind of "flip", which target node has two possible locations by two distance measurements. Nevertheless, we solve a completely different problem compared with Sweeps. First, we abstract problems from different backgrounds. Sweeps algorithm addresses the flip issue caused by bilateration, thus requires the distance measurements to be accurate. In contrast, OFA handles flip ambiguities caused by the ranging errors themself. Second, OFA guarantees to terminate in polynomial time cost, while Sweeps cannot. OFA does not track all possible results, so that it avoids the exponential explosion of the state space.

Some researchers use the Cramer-Rao lower bound (CRLB) to investigate the error characteristics of network localization [11,12]. CRLB provides a lower bound on the variance achievable of an unbiased location estimator [11]. However, CRLB cannot characterize flip ambiguity. When there are chances of flip ambiguity, the estimator can be biased.

Recently, many studies introduce the SDP to solve the localization problem by centralized optimization method [13–16]. SDP-based algorithms can also estimate the error of the localization result [13]. However, the bound of error is much relaxed, so that using this bound for error prediction may result in high false positive rate. Besides, the computation organization of SDP is inherently centralized, which is not much efficient in terms of energy and communication.

Some researches focus on flip ambiguity under the global view [1,17–20]. Lederer et al. attempt to figure out the correct global layout by purely connectivity information [1,18]. They utilize Delaunay complex to resolve flip ambiguity in global layout construction. Other studies address flip ambiguity problem in stitching the local maps [19,20]. Unfortunately, these designs cannot avoid the flip of a single node. Kannan et al. investigate the error caused by flip ambiguity in the global location estimation by simulated annealing [17]. They identify possible flips in the result of simulated annealing, and then refine the result of the identified nodes. As the identification is based on the ranging model and boundary check, it may not identify all flips. As a result, it cannot guarantee robustness.

3. Design overview

In this section, we first formally analyze the gain of optimistic design, and then demonstrate the optimistic algorithm on a simple network topology.

3.1. Problem formulation

Given a network and corresponding distance measurements between neighboring nodes in the network, we use a distance graph $G = (V, E)$ to present the network, in which the vertices in V denote the nodes in the network and each edge $(i,j) \in E$ denotes node i and node j can measure the mutual distance. The corresponding measurement value are presented by a functions: $d(i,j): E \to R$. We suppose a small portion of nodes, called anchors, know their locations in advance. Without loss of generality, m anchors are labeled from 1 to m, together with $n - m$ ordinary nodes labeled from $m + 1$ to n, where n denotes the total number of nodes in the network.

Traditional trilateration-based localization of the target network constructs a trilaterative ordering of the vertices in the corresponding distance graph. A trilaterative ordering for a graph G is an ordering of the vertices $1, 2, \ldots, k$, $(m \leqslant k \leqslant n)$, such that, the first m vertices are anchors, and every vertex i $(i > m)$ is adjacent to at least three distinct vertices earlier in the sequence. The ordering progress terminites when there is no vertex fulfill the above condition. For robust localization, the aim is twofold: to avoid flip ambiguity and to maximize the number of successfully located node out of all non-anchor nodes. Hence, the result without robustness guarantee, i.e. k in this formulation, is the upper bound of the node number.

A pessimistic design generates a robust trilaterative ordering that all vertices in the ordering are robust. A robust trilaterative ordering for a graph G is an ordering of the vertices 1, 2, …, k', $(m \leqslant k' \leqslant k)$, such that, the first m vertices are anchors, and from every vertex i $(i > m)$, (1) there are at least three edges to vertices earlier in the sequence, and (2) vertex i does not suffer flip ambiguity. The pessimistic method only focuses on the robustness of every single vertex, hence it has to exclude the vertices which may suffer flip ambiguity. These vertices are the potential gain of optimistic design.

To highlight the difference, we make an extra assumption that we start with the result of pessimistic design, i.e. vertex ordering $1, 2, \ldots, k'$, $(m \leqslant k' \leqslant k)$. Note that this is not necessary in algorithm implementation. Optimistic method continues add the rest $k - k'$ vertices in the ordering. The ordering itself implies dependencies between vertices. From every vertex i $(m \leqslant i \leqslant k)$, there is a set of prior vertices that must be added to the ordering before it. We define such vertex set as reference set. Such dependency relationship indicates that the correctness of one vertex also reflects the correctness of its reference set. Based on this fact, optimistic method computs the location and checks the correctness of each vertex. If the result is confirmed to be correct, optimistic method then marks the reference set of the vertex as robust. On contrary, if the result is confirmed to be incorrect, there must be at least one error in the reference set. Then, optimistic method tries to find and correct the source of the error, thus to convert the incorrect results into correct ones. In a word, optimistic design outperforms pessimistic ones with the cost of tracking and attempting. Considering the fact that localization only need one-time run for providing continuous service in a static network, this cost is acceptable in most networks.

3.2. Optimistic mechanism

We make two basic assumptions in OFA design. First, the locations of the anchors are reliable. Second, the distance measurements do not have huge errors. That is to say, if an embedding of the network cannot match the anchor distribution and all the distance measurements simultaneously, then the position assignment of the to-be-located nodes contains errors. Current localization systems typically use manual configuration or GPS to assign anchor locations, and adopt TDOA [21] or TOA [22] of acoustic signal to measure distances. Clearly, such systems obey our assumptions.

To better explain the idea of OFA design, we demonstrate OFA execution on a simple topology shown in Fig. 3, where solid squares denote anchors, soft circles denote to-be-localized nodes, solid circles denote localized nodes, and dashed circles denote ranging circles.

Fig. 3a shows the initial state of the network, where two non-anchor nodes n_1 and n_2 have five neighboring anchors $a_1 - a_5$. The localization process contains two steps: (1) localizing node n_1 by referring to nodes a_1, a_2, and a_3; (2) localizing node n_2 by referring to nodes a_4, a_5, and n_1. Unfortunately, nodes a_1, a_2, and a_3 are approximately collinear. In this case, pessimistic approaches will discard the localization result of node n_1. Then, node n_2 cannot be localized either. In contrast, OFA maintains one of the possible location estimates of n_1 and checks the consistency with anchor skeleton. We separately discuss the cases based on the two possible location estimates of node n_1:

If OFA correctly determines the position of n_1 as shown in Fig. 3b, n_2 can be accordingly located, as shown in Fig. 3c. As the Euclidean distances between neighboring nodes are close to the measured distances, the estimated positions of n_1 and n_2 are consistent with the anchor skeleton. Consequently, these results are confirmed as shown in Fig. 3f.

On the other hand, suppose OFA estimates node n_1 to a flipped position, as shown in Fig. 3d. Then, the position of n_2 is by no means consistent with the anchor skeleton, as the resulting inter-node distances are much larger than the measured distance, as shown in Fig. 3e. Such inconsistency triggers an error correction process that re-assigns correct locations to nodes n_1 and n_2 as shown in Fig. 3f.

To summarize, no matter whether OFA selects the correct location estimate for node n_1 initially, it can always locate the nodes n_1 and n_2 correctly. This example only shows 1-hop case, which is the simplest case. In a practical network, node n_2 may be several hops away from node n_1. This introduces several issues for implementing the optimistic mechanism, listed as follows:

1. In a practical network, the optimistically located node, i.e. node n_1 in the example, may be confirmed by another node, i.e. node n_2 in the example, that is several hops away from the node, thus we need to track whether the location estimate of a node is robust or not. We introduce a confidence mechanism (discussed in Section 4.1) to quantitate the robustness of a location estimate.
2. If node n_2 is collinear with anchors $a_1 - a_3$, it cannot confirm the location estimate of node n_1. We design a reverse update mechanism to address this issue, as discussed in Section 4.2.
3. When OFA detects an error, the source of the error, i.e. the first flipped node causing the error, may be several hops way from the node detecting the error. We need to correct all nodes that influenced by the error-source node. We propose an error correction procedure to do this work as discussed in Section 4.3.
4. Finally, not all optimistically located nodes can be confirmed by the anchor skeleton, we should prune the unreliable results to preserve the robustness of the final result. We show this procedure and discuss the gain and the cost of OFA in Section 4.4.

4. OFA Protocol

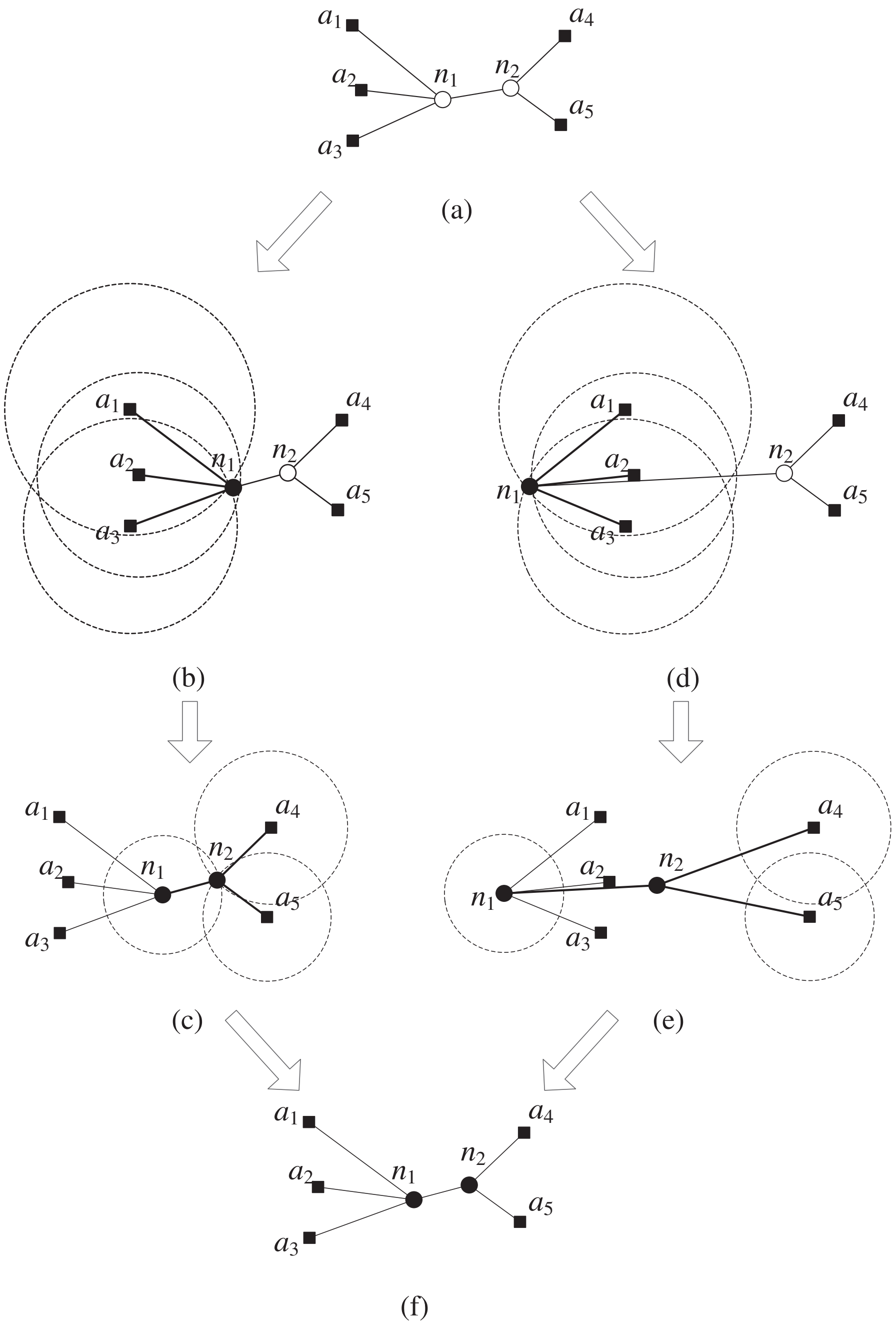
OFA includes four major components: (1) *optimistic execution*, (2) *result confirmation*, (3) *error correction*, and (4) *final result set determination*. Optimistic execution iteratively localizes nodes by multilateration, and records the location estimates as the *temporary position assignment* (TPA). Then, OFA checks whether there are errors in current TPA by the anchor skeleton. If the TPA passes the check, OFA adopts result confirmation procedure to validate the TPA. Otherwise, OFA adopts error correction procedure to correct flips in current TPA. Finally, final result set determination procedure prunes unconfirmed location estimates in the result set and reports the final result.

4.1. Optimistic execution

Optimistic execution improves the traditional multilateration by integrating the procedures of confidence estimate and consistency check. This extension is called multilateration with flip ambiguity estimation (MFAE).

4.1.1. Confidence estimation

OFA adopts the concept of *confidence* to quantify how much a location estimate can be accepted. Confidence is often defined as the probability of the location of a node is correctly estimated without flips. MFAE can adopt any



(a)

(b)

(d)

(c)

(e)

(f)

Fig. 3. OFA overview. (a) The distance graph of the network. (b) OFA takes the correct location estimate as the result. (c) The correct result matches the distance measurements, when node n_2 is located, and this result passes the consistency check. (d) OFA takes the flipped location estimate as the result. (e) The flipped result does not match the distance measurements, so that the consistency check fails. (f) The final results for nodes n_1 and n_2.

off-the-shelf flip estimators [2,3,9] to evaluate the confidence. Here, we adopt a geometric method [23]. The rationale of this method is that the target node is more likely to be flipped if the reference nodes are approximately collinear, where reference nodes refer to the nodes used to determine the location of the target node in a multilateration. MFAE uses the width of the reference node set to estimate the confidence of multilateration, and we define the width of a point set as the minimal distance of two parallel lines that contain the point set. Here, we linearly map the width to confidence value into range $[0,1]$. For zero mean Gaussian noise $N(0,\sigma^2)$, we set the coefficient of the mapping to the reciprocal of three deviations, i.e. $(3\sigma)^{-1}$.

As shown in Fig. 2, in a network-wide flip, nodes referring to flipped nodes may also be flipped. Hence, the confidence of a node is determined by both the confidence of the multilateration and the confidences of the reference nodes. Let *m_confidence* and *r_confidence* represent the

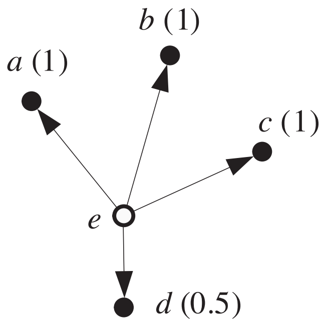


Fig. 4. Confidence computation.

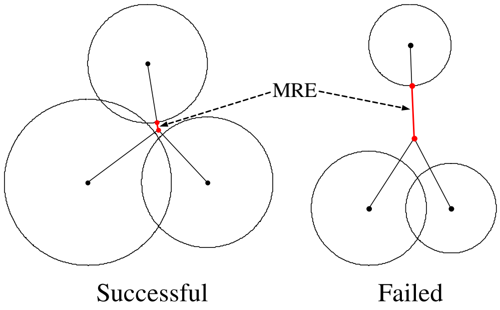


Fig. 5. Consistency check.

confidence of the multilateration and the minimum confidence of the reference nodes, respectively. We define the confidence of the location estimate as *min*(*m_confidence*, *r_confidence*). By this definition, the selection of reference nodes influences the confidence of the target node. Taking Fig. 4 as an example, node *e* is localized by the reference node set $\{a, b, c, d\}$ and the corresponding confidence *m_confidence* = 1. If the reference node set is $\{a, b, c\}$, we define *m_confidence* to be 0.7. In addition, the confidences of a, b, c, and d are set to be 1, 1, 1, and 0.5, respectively. If we adopt all the reference nodes, the confidence of node *e* is *min*(1,0.5) = 0.5. If we remove node *d* from the reference node set, then *m_confidence* accordingly reduces to 0.7. Under this configuration, as *r_confidence* is 1, the confidence becomes *min*(0.7,1) = 0.7, which is better than the previous case. In general, for a reference node set, if we remove the reference node with lowest confidence, the value of *r_confidence* increases and the value of *m_confidence* decreases. Repeating this operation can yield the maximum confidence of the target node.

4.1.2. Consistency check

We propose to use the *maximum residual error* (MRE) as the indicator of the result consistency. MRE is defined as the maximum deviation between the embedded distances (derived from multilateration result) and the estimated distances used for multilateration.

Typically, multilateration estimates the position of a target node by minimizing the squared error between the estimated distance and the measured distance through a least square estimator:

$$p = \arg\min_{p} \sum_{i=1}^{k} (\|p - p_i\| - \tilde{d}_i)^2,$$

where p is the estimated position of the target node, k is the total number of the reference nodes, p_i, $1 \leqslant i \leqslant k$, denotes the position of reference node i, and $\tilde{d}_i$, $1 \leqslant i \leqslant k$, denotes the measured distance between the target node and reference node i. Then, we define the MRE by

$$MRE = \max\{|\|p - p_i\| - \tilde{d}_i|\}, 1 \leqslant i \leqslant k,$$

which is the maximum error between the estimated distances and the measured distances, as shown in Fig. 5.

OFA adopts a threshold to determine the consistency. If MRE is below the threshold, we say the corresponding multilateration is successful; otherwise, it is failed. We in detail analyze the threshold in Section 5.2.

OFA further determines what to do next according to the result of the consistency check:

1. If the result is successful, OFA checks whether it should confirm the temporary position assignment and execute result confirmation (described in Section 4.2).
2. If the result is failed, OFA will jump to the error correction procedure (described in Section 4.3).

We show the main procedure of OFA, MFAE, and confidence computation in Algorithms 1–3, respectively.

Algorithm 1. OFA

```
Input: the distance graph G and the anchor node set Anchor
Output: the position estimates
Mark all anchors as localized nodes
  Localized = Anchor
while existing a node Node_N in G that has at least three reference nodes Reference in Localized do
  ans = MFAE (Reference, distance_measurement(Reference, Node_N))
  switch ans.state
    case success
      Node_N.position = ans.position
      Node_N.reference = Reference
      Node_N.anchors = OR(Reference.anchors)
      Node_N.multilateration_confidence = ans.confidence
      Node_N.confidence = Confidence(Reference)
      for each ReferenceNode_r ∈ Reference do
        if Node_N.anchors ≠ ReferenceNode_r.anchors then
          Result_Confirmation(ReferenceNode_r, Node_N)
      break
    case failed
      Error_Correction(Node_N, Reference)
Localized = Localized ∪ Node_N
for each Node_N ∈ Localized do
  if Node_N.confidence < 1 then
    Localized = Localized\Node_N
return Localized
```

Algorithm 2. MFAE

```
Input: the reference node set Reference and
    the corresponding distance set
    Distance_Measurement
Output: the position estimate, the consistency check,
    and the confidence
Call traditional multilateration to estimate the
    position of the node
    $Result.position = Multilateration(Reference, Distance\_Measure\ ment)$
$MRE = 0$
for each $ReferenceNode\_r \in Reference$ do
    $MRE = max(MRE, abs(\|Result.position - ReferenceNode\_r.position\| - Distance\_Measurement\ ReferenceNode\_r))$
if $MRE < THRESHOLD$ then
    $Result.state = success$
else
    $Result.state = failed$
$Result.confidence = min(1, Width(Reference)/RATIO)$
return $Result$
```

Algorithm 3. Confidence

```
Input: the reference node set Reference and the
    corresponding confidence of
    the nodes ref_confidence
Output: the confidence of the target node,
    and the corresponding reference node set
$m\_confidence = min(1, Width(Reference)/RATIO)$
$r\_confidence = min(ref\_confidence)$
$confidence = min(m\_confidence, r\_confidenc)$
while $m\_confidence > r\_confidence$ do
    $Index\_r$= the index of the node with
        minimum confidence value in $ref\_confidence$
    $Reference = Reference \backslash Index\_r$
    $ref\_confidence = ref\_confidence \backslash Index\_r$
    if $|Reference| < 3$ then
        break
    $m\_confidence = min(1, Width(Reference)/RATIO)$
    $r\_confidence = min(ref\_confidence)$
    $confidence = max(confidence, min(m\_confidence, r\_confidence))$
return confidence and the
    corresponding reference node set
```

4.2. Result confirmation

Result confirmation validates the temporary position assignment, provided that the assignment passes consistency check. As OFA uses confidence to quantify correctness, result confirmation is accomplished by increasing the confidence values of the located nodes.

To check consistency, OFA records all the anchors that a node directly or indirectly refers to. That is, the anchor dependence set of a target node is the union of all the anchor dependence sets of its reference nodes. If a node is successfully localized by multilateration and the anchors which it depends on are different with that of any reference node, the temporary position assignment is consistent with the anchor skeleton.

To efficiently validate the optimistically localized nodes, OFA records the reference node set on each node, which forms a directed acyclic graph, called distributed dependence graph (DDG). DDG is the key structure for result confirmation and error correction.

Result confirmation recursively updates the confidence of nodes along DDG. Fig. 6a shows an example. Node c is localized based on nodes a, b, and anchor A. As a, b, and A are approximately collinear, the confidence of c is low. Then, node d is successfully localized based on anchors B, C, and node c, i.e. $d.anchor = c.anchor \cup B \cup C$. Considering $B, C \notin d.anchor$, we have $d.anchor \neq c.anchor$. Then, OFA confirms the location estimates of c.

To update the confidence of c, OFA temporally adds an edge $\langle c, d \rangle$ into the DDG and assumes that the position estimate for node d is correct in the next step. Taking node d as another reference node, OFA updates the confidence of node c, as shown in Fig. 6b. Note that the update does not actually change the DDG of the network, and the edge is added conceptually to explain how the confidence is updated. After node c is updated, this procedure recursively operates on its reference nodes (parent nodes), such as nodes a and b. In addition, the children nodes will also update their confidences, if their confidences are dominated by the updated node, such as node d and e in this example. We show the details of confidence updating procedure in Algorithm 4.

Algorithm 4. Result_Confirmation

```
Input: the node Node_n to be updated and the parent
    node ParentNode_s
Output: the new confidence of the processed nodes
$Reference = Node\_n.reference \cup ParentNode\_s$
$confidence = Confidence(Reference)$
if $confidence = Node\_n.confidence$ then
    Return
$Node\_n.confidence = confidence$
$update\_list$ = nodes depending on node $n$
for each node $Node\_i$ in $update\_list$ do
    $confidence = Confidence(Node\_i.reference)$
    if $confidence > Node\_i.confidence$ then
        $Node\_i.confidence = confidence$
        append nodes depending on node $i$ to the
            $update\_list$
for each $Node\_i \in Node\_n.reference$ do
    $Result\_Confirmation(Node\_i, Node\_n)$
return updated nodes and corresponding
    confidence values
```

4.3. Error correction

Error correction identifies and corrects flipped nodes, when an error is detected in consistency check. A general case of error correction is the network-wide flip: a single node flip causes a part of the network flipping over another part. In this case, the source of the error, i.e. the first flipped node, may be several hops away from the node detecting the error. To eliminate such an error, OFA adopts a depth-first search on DDG for locating the source of the error. First, OFA generates a list of nodes to indicate the order of attempts, defined as *attempt list*. Second, OFA flips the position estimate of the first node in the attempt list. Third, OFA recursively updates the position estimates of the nodes along DDG. When the update process raises an error in consistency check, OFA will terminate this attempt and start another process by flipping the next node in the attempt list. If no error occurs until the update process reaches the node detecting the inconsistency, the error is eliminated. Then, the updated nodes are informed to accept the new position assignment. Finally, OFA also checks the anchor dependence to validate the temporary result, and goes on to localize other nodes.

Error correction must address two main issues. First, considering the computational and communicational costs, OFA must locate the source of error efficiently. Second, when locating a node as the source of error, OFA must flip its position estimate efficiently.

OFA adopts confidence-based node selection to locate the source of error. As locating a network by trilateration is an iterative process, we can list a sequence of the localized nodes in order of being located by multilateration. For the error-detection-node, there is a set of nodes must be located before the error-detection-node can be located. Such node set is the search space of the error correction procedure. OFA utilizes the confidence of the multilateration as indicator, because it shows the probability whether the node is flipped. Thus, OFA sorts the nodes in the reference set by their multilateration confidences, and processes the aforementioned attempt in this order. By the definition of confidence, the multilateration confidence of a node is not less than its confidence. Hence, we can easily locate the node with the lowest multilateration confidence in the search space, which is the node with equal confidence and multilateration confidence. Our experiments show that this node is the source of the error in most cases.

It is computation intensive to flip a node by searching the whole solution space of the multilateration. For the sake of efficiency, OFA flips the position estimate of a node by the following steps: first, selects a line; then, reflects the position estimate against the line; finally, uses the reflected position as initial position and refines this result by multilateration. The key of this procedure is how to define the line for reflecting the position estimate, called the *representing line*. As shown in Fig. 7a, the solid dots indicate the reference nodes and the vertical dashed line denotes the representing line of the reference nodes. However, the traditional linear regression of the reference nodes is not suitable for this requirement. Take Fig. 7a as an example, if we compute the linear regression of the reference node, we will get the result as shown in Fig. 7b, in which the line is orthogonal to the distribution of the reference nodes. This problem is due to the estimate function of linear regression:

$$Q(\beta_0, \beta_1) = \sum [y_i - (\beta_0 + \beta_1 x_i)]^2$$

which only considers the difference of y-value.

OFA makes two departures from the traditional linear regression. First, OFA adopts the general form $Ax +$

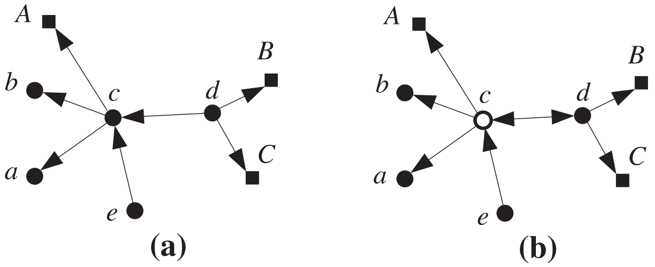


Fig. 6. Confidence update.

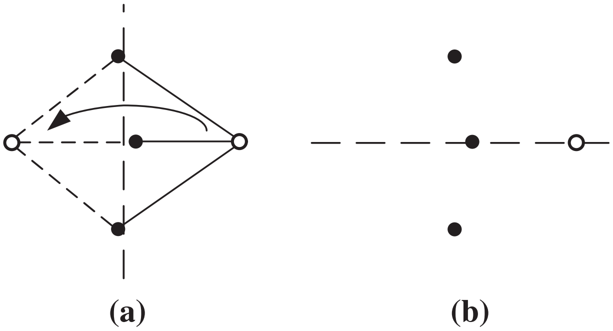


Fig. 7. The representing line of the reference nodes.

$By + C = 0$ to present an arbitrary line. Second, OFA adopts a coordinate-system-free estimate function as follow:

$$Q(A,B,C) = \sum \frac{(Ax_i + By_i + C)^2}{A^2 + B^2}.$$

This formula is the sum of squared distances between the nodes and the estimated line, which can better represent the distribution of the nodes.

We thus obtain the following simultaneous equations:

$$\begin{cases} \frac{\partial Q(A,B,C)}{\partial A} = \sum 2\frac{(Ax_i+By_i+C)(B^2x_i-ABy_i-AC)}{(A^2+B^2)^2} = 0 \\ \frac{\partial Q(A,B,C)}{\partial B} = \sum 2\frac{(Ax_i+By_i+C)(A^2y_i-ABx_i-BC)}{(A^2+B^2)^2} = 0 \\ \frac{\partial Q(A,B,C)}{\partial C} = \sum 2\frac{Ax_i+By_i+C}{A^2+B^2} = 0 \end{cases}$$

Simplifying the equations, we obtain two independent equations:

$$\begin{cases} A\bar{x} + B\bar{y} + C = 0 \\ S_{xy}A^2 - (S_{xx} - S_{yy})AB - S_{xy}B^2 = 0 \end{cases},$$

where $\bar{x} = \frac{1}{k}\sum x_i,\ \bar{y} = \frac{1}{k}\sum y_i, S_{xy} = \sum(x_i - \bar{x})(y_i - \bar{y}), S_{xx} = \sum(x_i - \bar{x})^2$, and $S_{yy} = \sum(y_i - \bar{y})^2$ for k nodes.

If $S_{xy} = 0$, the result is unique; If $S_{xy} \neq 0$, there are two possible solutions. In this case, we need to check the value of $Q(A,B,C)$ and take the solution with smaller $Q(A,B,C)$ value.

Note that the error handling procedure can only correct the flip of one node. It will fail in handling the errors which are caused by multiple flips, defined as the *combined flip*. As shown in Fig. 8, the error on the node marked by diamond is caused by the combination of two flips. In this case, flipping a single node cannot eliminate the error. OFA ignores the combined flip for the following reasons. First, in an actual network, the probability of forming a combined flip is very low. Second, to handle the combined flip will incur huge computational cost, which is not practical. Third, as OFA can prune the non-robust nodes, ignoring the combined flip will not decay the robustness of the final result. Finally, we show the details of error correction procedure in Algorithm 5.

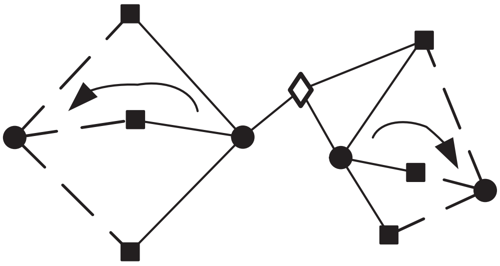


Fig. 8. Combined flip.

Algorithm 5. Error_Correction

```
Input: the node Node_n that detects an error
Output: the new position assignment of nodes
attempt_list = sort the nodes in the reference set of
  Node_n by their local confidence values
for each node Node_s in attempt_list do
  Node_s.attempted_position = the flipped position of
  Node_s
  update_list = nodes depending on node Node_s
  for each node Node_i in update_list do
    ans = MAFE(Node_i.referece, distance_
  measurement
  (Node_i.reference, Node_i))
    if ans.state = failed then
      break
    else
      Node_i.attempted_position =
       ans.position
      Node_i.attempted_
      multilateration_confidence =
      ans.confidence
      Node_i.attempted_confidence = Confidence
      (Node_i.reference)
      update_list = append nodes depending on
      node Node_i to the update_list
      if Node_i = Node_n then
        Notify all the attempted nodes to accept
        the new position assignment
        for each Node_r $\in$ Node_n.reference do
          if Node_n.anchors $\neq$ Node_r.anchors then
          Result_Confirmation(Node_r, Node_n)
    return true
return false
```

4.4. Final result set determination

OFA repeats the above procedures until no node can be optimistically localized in the network. As not all the flipped nodes can be detected by the anchor skeleton, OFA needs to prune non-robust location estimates in the result set. That is, the confidence value must be above a predefined threshold. We set the threshold to 1 for robustness. Generally, the nodes on the brim of the network have fewer chances to be validated by anchors and are prone to be excluded in the final result set.

To summarize, through precious computation and checking, OFA can locate a larger number of nodes and yield more robust localization results simultaneously than existing approaches. According to the experience on real-world large-scale networks [24,25], it is highly worthy to trade computational costs for location accuracy. First, many location-based services, for example geographic routing [26], cannot work properly on the nodes without locations or with incorrect locations. Hence, accurate localization is more essential than efficiency related issues. Second, as localization only needs one-time-execution for static networks during the setup phase, it is profitable to spend more resource in this step, for better quality of service in the rest of the network lifetime.

5. Analysis

In this section, we formally analyze the expected performance gain and the cost of OFA, as well as the threshold used in consistency check step.

5.1. Performance analysis

1. P_{FN} denotes the proportion of flips in all location estimates, if we do not adopt any flip avoidance in localization. Hence, P_{FN} is equal to the overall proportion of false negative predictions.
2. P_{RT} denotes the probability that a node passes the robust test, when we adopt a flip predictor. As the predictor is reliable, it requires two conditions for a node to pass the test: (1) robust to flip ambiguity; (2) not excluded by the false positive predictions. Let $\overline{P}_{RT} = 1 - P_{RT}$ to denote the probability that a node fails in the robust test.
3. R_v denotes the proportion of nodes confirmed by anchor skeleton out of all the optimistically localized nodes. It is approximately equal to the ratio of the area of the convex hull of all anchors to the total area of the network. With the increase of anchor numbers, this ratio can continuously approach 1.
4. C_{EH} denotes the mean cost to correct a flip by OFA. Here, we use the number of multilaterations to evaluate the cost. C_{EH} can be reduced by increasing the number of anchors.

Theorem 1. *The performance gain of the optimistic scheme to pessimistic scheme is at least* $1 + R_v\overline{P}_{RT}/P_{RT}$.

Proof 1. Suppose we localize a network with large number of nodes, so that we ignore the boundary of the network. We analyze the state that the pessimistic method finishes the localization procedure.

Suppose the pessimistic method robustly localizes N_p nodes. Locating these N_p nodes will test N_p/P_{RT} nodes. If we localize the same node set by the optimistic method, the N_p nodes will also be localized robustly. The remaining $\overline{P}_{RT}N_p/P_{RT}$ nodes will be optimistically localized, and $R_v\overline{P}_{RT}N_p/P_{RT}$ of them will be validated by anchors. Hence, in current state, optimistic method will successfully localize $(1 + R_v\overline{P}_{RT}/P_{RT})N_p$ nodes. This is the lower bound of nodes that the optimistic method can localize, because the optimistic method does not finish the localization procedure in this state and may further localize other nodes in the network. Hence, the lower bound of the performance gain is $1 + R_v\overline{P}_{RT}/P_{RT}$. □

Theorem 2. *When we use the optimistic method, the expected number of multilaterations to localize a node is:*
$$\frac{1 + R_vP_{FN}C_{EH}}{P_{RT} + R_v\overline{P}_{RT}}.$$

Proof 2. The cost for the optimistic method comes from two aspects: the costs for optimistically localizing nodes and for error handling. Suppose we have successfully localized N_o nodes. Let N denote the total number of optimistically localized nodes. Thus, N is the total number of multilaterations for optimistic localization. As discussed in Theorem 1, the localized nodes are either directly localized as robust, or the nodes that are optimistically localized and validated by anchors. Hence, we have $N_o = P_{RT}N + R_v\overline{P}_{RT}N$. That is $N = N_o/(P_{RT} + \overline{P}_{RT}R_v)$. Locating N node will introduce$P_{FN}N$ errors, among which $R_vP_{FN}N$ will be detected and corrected. Hence, the cost of error handling is $R_vP_{FN}NC_{EH}$. In sum, the mean number of multilaterations to localize a node by optimistic method is given by:
$$\frac{1 + R_vP_{FN}C_{EH}}{P_{RT} + R_v\overline{P}_{RT}}.$$
□

For example, a network with average degree 9 and 10% anchors may typically have the following parameters: $P_{FN} = 0.1$, $P_{RT} = 0.7$, $R_v = 0.8$, and $C_{EH} = 1.3$. Then, the lower bound of the performance gain would be 1.34, and the cost would be 1.17. Actually, our experiment results show that OFA can perform far better than that lower bound when the target network is sparse.

5.2. Threshold analysis

In consistency check step, OFA adopts the MRE to determine the consistency of current position assignment. Actually, when there is a flip, we can observe a huge increase of MRE. Hence, OFA can accept a wide range of threshold. In this appendix, we analyze the threshold selection of OFA.

We use a probabilistic model to present the uncertainties of the distance measurements and position estimates. Many researches [9,11] show that the distance measurements are roughly Gaussian distributed. As the systematic deviation can be subtracted out by calibrating the measurement results, we assume the measurement errors are zero mean Gaussian distributed: $f(e_r) = N(0, \sigma_r^2)$, where e_r denotes the measurement error and σ_r denotes the standard deviation of the error. We model the location estimation error of node i as a random variable e_i. The physical meaning of e_i is the error of distance estimation caused by the localization error in multilateration. We assume e_i is Gaussian, and let $f(e_i) = N(0, \sigma_i^2)$, where σ_i denotes the standard deviation of e_i.

We analyze σ_i and the bound of MRE with no flip ambiguities. Suppose we localize a node by measurements with k neighboring nodes $n_1, n_2, \ldots, n_k$. Following the same notations, we use n_{k+1} to denote the target node. Let σ_i denote the standard deviation of node n_i, $1 \leqslant i \leqslant k + 1$. The error of the target node, presented by σ_{k+1} is related to the error of each reference and the error of distance measurements e_r. For each reference node i, $1 \leqslant i \leqslant k$, the effective error E_i for localizing node n_{k+1} is the sum of e_i and e_r, i.e. $E_i = e_i + e_r, f(E_i) = N(0, \sigma_r^2 + \sigma_i^2)$. In the location estimation phase, the errors from the reference nodes counteract each other. Hence, the error of the result is the average of all the effective errors, that is

$$\sigma_{i+1}^2 = \frac{1}{k}\left(\sigma_r^2 + \overline{\sigma_i^2}\right),$$

where $\overline{\sigma_i^2} = \sum_1^k \sigma_i^2/k$. This formula shows the way of updating the standard deviation σ_i for each node.

Since the error of the result is the average of the effective errors, the residual error on each measurement is denoted by $RE_i = E_i - \overline{E}_i, 1 \leqslant i \leqslant k$, where $\overline{E}_i = \sum_1^k E_i/k$.

Then, the MRE is denoted by max$\{|RE_i|, 1 \leqslant i \leqslant k\}$. By the distribution of E_i, we obtain

$$f(RE_i) = N\left(0, \left(1 - \frac{1}{k}\right)\sigma_r^2 + \left(1 - \frac{2}{k}\right)\sigma_i^2 + \frac{1}{k}\overline{\sigma_i^2}\right), 1 \leqslant i \leqslant k.$$

Using the approximation $\sigma_i^2 \approx \overline{\sigma_i^2}$, we get the identical distribution of each RE_i:

$$f(RE_i) = N\left(0, \left(1 - \frac{1}{k}\right)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)\right), 1 \leqslant i \leqslant k.$$

Using the inequality $\max(|E_i - \overline{E}_i|) \leqslant \sum |E_i - \overline{E}_i|/2$, we obtain an upper bound of MRE, denoted by $MRE_{\max}$:

$$MRE_{\max} = \frac{1}{2}\sum_{i-1}^{k}|RE_i|.$$

By the distribution of RE_i, we obtain

$$\begin{cases} E(|RE_i|) = \sqrt{\frac{2}{\pi}}\sqrt{\left(1 - \frac{1}{k}\right)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)} \\ D(|RE_i|) = \left(1 - \frac{2}{\pi}\right)\left(1 - \frac{1}{k}\right)\left(\sigma_r^2 + \overline{\sigma_i^2}\right) \end{cases}, 1 \leqslant i \leqslant k.$$

By the central limit theorem, we obtain an approximate distribution of $MRE_{\max}$:

$$f(MRE_{\max}) = N\left(\frac{1}{2}\sqrt{\frac{2}{\pi}}\sqrt{k(k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)}, \frac{1}{4}\left(1 - \frac{2}{\pi}\right) \times (k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)\right.$$

Under less than 1% chance faults, we obtain an upper bound of$MRE_{\max}$ as well as MRE:

$$\begin{aligned} MRE &\leqslant \frac{1}{2}\sqrt{\frac{2}{\pi}}\sqrt{k(k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)} + \frac{3}{2} \\ &\quad \times \sqrt{\left(1 - \frac{2}{\pi}\right)(k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)} \\ &= \frac{1}{2}\sqrt{k(k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)}\left(\sqrt{\frac{2}{\pi}} + \frac{3}{\sqrt{k}}\sqrt{1 - \frac{2}{\pi}}\right) \end{aligned}$$

For the second part, as the number of references $k \geqslant 3$ is necessary for multilateration, we get the following relaxations:

$$\begin{aligned} \sqrt{\frac{2}{\pi}} + \frac{3}{\sqrt{k}}\sqrt{1 - \frac{2}{\pi}} &\leqslant \sqrt{\frac{2}{\pi}} + \sqrt{3}\sqrt{1 - \frac{2}{\pi}} \\ &< \sqrt{2}\sqrt{\frac{2}{\pi} + 3\left(1 - \frac{2}{\pi}\right)} < 2. \end{aligned}$$

Hence, we obtain an upper bound of MRE:

$$MRE < \sqrt{k(k-1)\left(\sigma_r^2 + \overline{\sigma_i^2}\right)}.$$

In optimistic execution procedure, if the MRE breaks the upper bound, OFA detects an error of current position assignment.

6. Performance evaluation

We conduct extensive simulations to evaluate OFA.

6.1. Experiment setup

We randomly distribute 200 nodes in a square region, with a certain percentage of them as anchors. The mean degree of the network instances are controlled by the distance measurement range. The distance measurements between neighboring nodes are corrupted by zero mean Gaussian noises [9]. For each set of simulations, we take multiple runs and report the average.

We evaluate OFA by comparing with the state-of-the-art design, robust quadrilaterals (RQ).[2]. RQ adopts robust local structures to avoid flip ambiguity, in which it sets a bound on the geometric element to achieve the robustness of the four-node local maps. We implement full RQ algorithm with cluster optimization to mitigate error accumulation.

We use three metrics in our simulations: performance, accuracy, and cost. The proportion of robustly localized nodes shows performance of each algorithm. The standardized position estimation error (SPEE) indicates the localization accuracy of each algorithm, defined as the percentage value between the mean position estimation error and the maximum distance-measurement range:

$$SPEE = \frac{1}{nR_{\max}}\sum_{i=1}^{n}\|p_i - \hat{p}_i\| \times 100\%,$$

where n is the total number of robustly localized nodes, $R_{\max}$ is the maximum range of distance measurement, p_i and $\hat{p}_i$ are the ground truth position and the estimated position of node i, respectively. If $n = 0$, we define SPEE as Not-a-Number and do not count such values in the final result. We evaluate the cost by the mean number of multilaterations for localizing a node by each algorithm. Moreover, we conduct the experiments by controlling the following parameters:

1. the mean degree of the network instances;
2. the proportion of anchors in the network;
3. the standard deviation of the ranging noise.

In addition, we also evaluate OFA by comparing with a SDP-based algorithm [27], named SDP for short. We implement the whole algorithm by using SDPT3 toolbox. As SDP requires high computational cost, we adopt 50 nodes in this experiment. Besides, SDP is designed for accurate distance measurements. Our experiment results show that SDP cannot work well when the maximum error is higher than 1%. Hence, we set the maximum error to be 0.001%, 0.01%, 0.1%, and 1%, respectively, and show the results in logarithmic scales.

6.2. The impact of average degree

We first examine the performance and cost of OFA and RQ when the node average degree varies. We fix the anchor proportion to 10% and set the errors to be at most (three deviations) 10% of distance measurements. The average

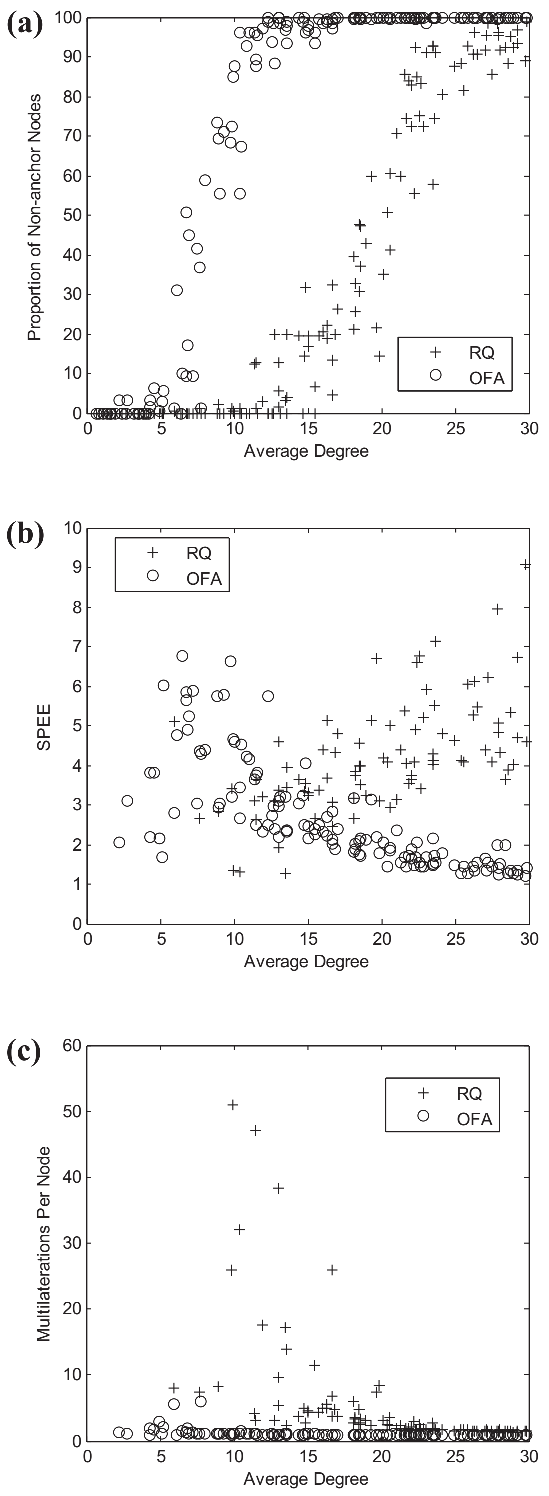


Fig. 9. The impact of average degree. (a) The number of robustly located nodes. (b) The average positioning error. (c) The average cost per locating a node.

degree of each instance varies from 0 to 30 with step length about 0.2.

Fig. 9a plots the proportion of robustly localized nodes against average degree. When the average degree increases, both algorithms localize more nodes. OFA can localize the entire network when the average degree is greater than 10. The requirement is nearly the same as

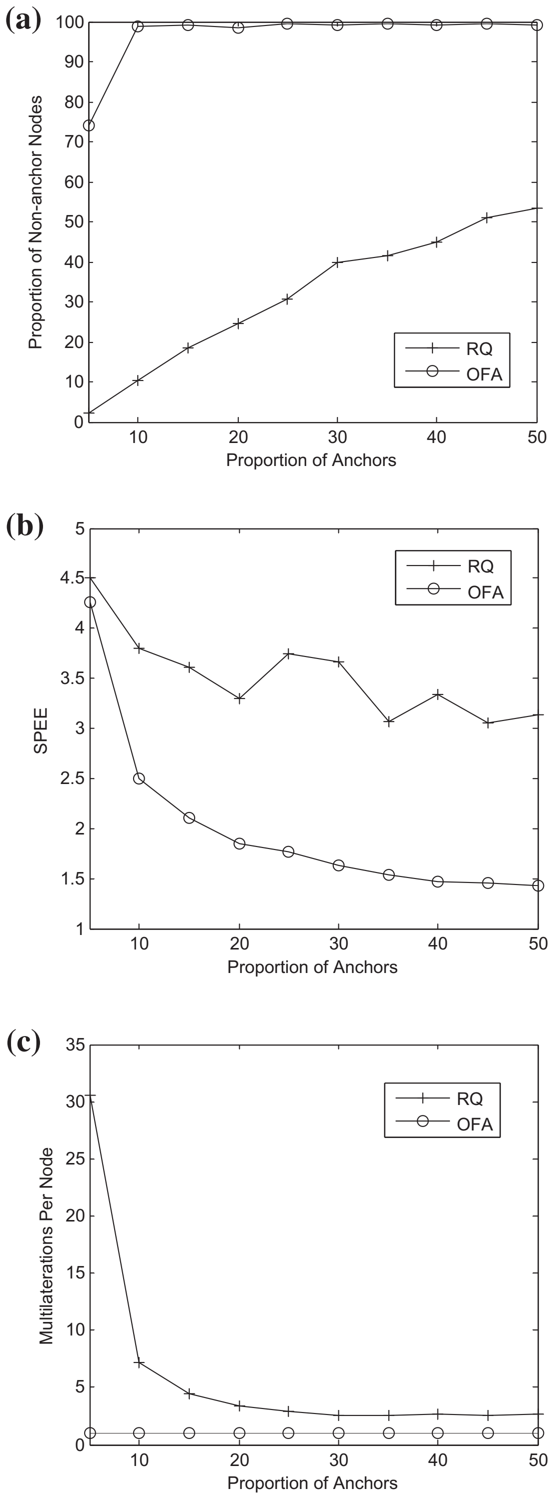


Fig. 10. The impact of anchor proportion. (a) The number of robustly located nodes. (b) The average positioning error. (c) The average cost per locating a node.

traditional multilateration [2], which means that OFA obtains robustness by extremely low performance cost. In contrast, RQ requires the average degree over 25 for localizing the entire network. In fact, RQ needs to generate uniformly overlapped local clusters to form global clusters. However, the flip ambiguity predictor of RQ drops a large proportion of local maps in the generation step, so that RQ demands high network density to compensate the false positive predictions.

Fig. 9b plots the SPEE against average degree. The SPEE of OFA decreases when the average degree increases. Nevertheless, the SPEE of RQ is quite stable over all the tested ranges. RQ can hardly benefit from the increase of average degree, because RQ always generates four-node local maps, i.e. localizing a node by three measurements only. In contrast, OFA performs a more accurate estimate through all available distance measurements, leading to a better error control with the increase of the average degree.

Fig. 9c plots the mean number of multilaterations needed for localizing a node. Note that we do not show the results that no nodes are localized, so the left most of this graph is blank. With the increase of the average degree, the cost of each algorithm decreases. The cost for OFA includes the attempt of localizing nodes that are not validated in the final step as well as error handling. For RQ, it must blindly generate all possible local clusters, no matter whether they are finally localized or not. Such a procedure introduces many multilaterations that do not contribute to the final result set. As shown in Fig. 9c, even in the worst case, OFA seldom localizes a node by over two multilaterations, and the cost of OFA is always lower than RQ.

6.3. The impact of anchor proportion

We further examine the performance and cost when the anchor density varies. We fix the average degree about 12 and set the errors at most 10% of distance measurements. We report the mean result of 50 network instances in Fig. 10, while the proportion of anchors varies from 5% to 50% with step length 5%.

Fig. 10a plots the proportion of localized nodes against anchor density. OFA can successfully localize most nodes when the proportion of anchors is over 10%, because OFA only requires the network to have enough anchors for the consistency check. On the other hand, RQ performs linearly with anchor density. As RQ discards many local maps by the flip predictor, it cannot generate a network-wide global map. In this case, the chance of localizing the small-scale clusters is linear with anchor density.

Fig. 10b shows the SPEE against anchor density. The SPEE of OFA decreases when more anchors exist, as more anchors will help to make a better position estimation as well as to diminish the error accumulation. Based on the small-scale clusters, RQ cannot be aware of the density of anchors in cluster generation and merging steps. Hence, more anchors cannot restrain the error accumulation of RQ. As a result, RQ does not benefit much from the increase of anchor density.

Fig. 10c shows the number of multilaterations needed to localize a node for each algorithm. OFA can achieve almost the ideal execution cost, because the false negative rate for flips is naturally very low in actual networks. For RQ, the overall execution cost of cluster generation is independent of the anchor density. Hence, the mean cost of localizing a node is dominated by the total number of nodes that are successfully localized. As a result, the mean

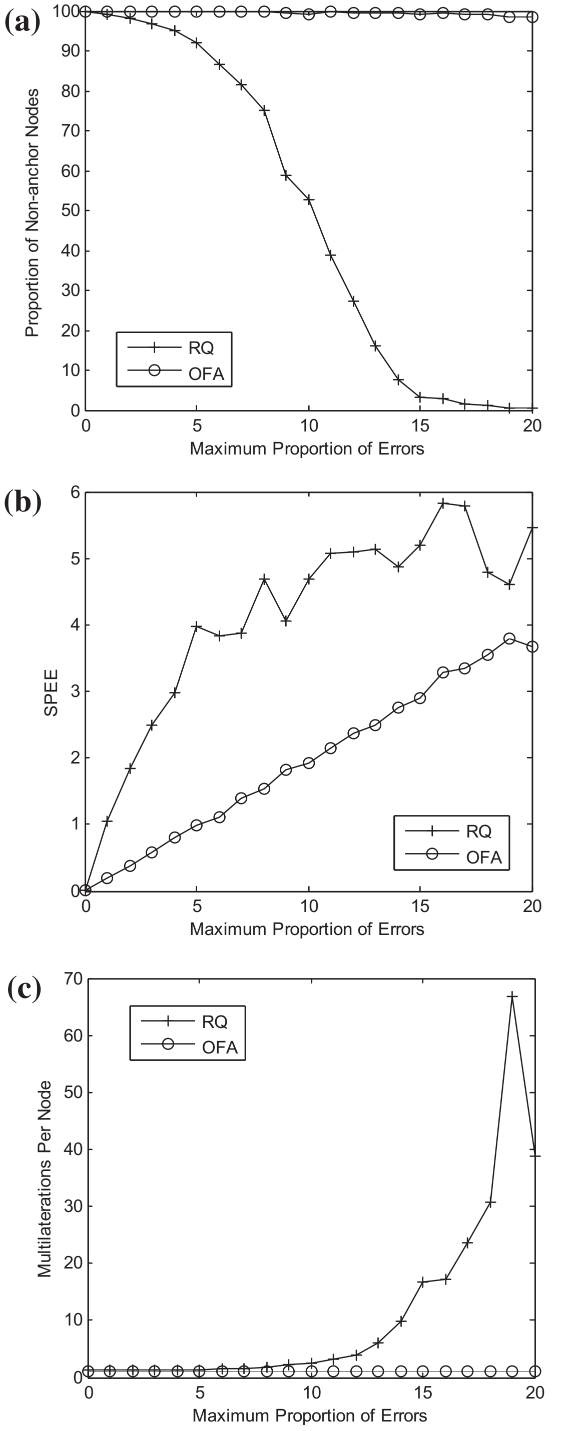


Fig. 11. The impact of error magnitude. (a) The number of robustly located nodes. (b) The average positioning error. (c) The average cost per locating a node.

cost is inversely proportional to the number of localized nodes.

6.4. The impact of measurement error

In this section, we evaluate the performance and cost when the standard deviation of the errors varies. We fix the anchor proportion to 10% and set the average degree at around 20. We report the mean result of 50 network instances in Fig. 11, while the maximum proportion of error varies from 0 to 20 with step length 1.

Fig. 11a plots the proportion of localized nodes against the error magnitude. The performance of OFA decreases slightly with the increase of errors, because the optimistic mechanism of OFA can properly compensate the impact of the increased errors. In contrast, the number of localized nodes for RQ decreases rapidly when the error magnitude enlarges. Larger errors lead to severer false positive predictions in the cluster generation of RQ. Hence, the performance of RQ degrades sharply, when the cluster generation step cannot produce adequate overlapped local clusters.

Fig. 11b plots the SPEE against error magnitude. The SPEE of OFA is approximately linear to the error configuration. The SPEE of RQ is a bit complicated. When the proportion of error is less than 8%, the SPEE of RQ is also approximately linear to the error magnitude. Then, the SPEE of RQ starts to increase slowly and fluctuate more. The reason is that RQ does not locate the full network in this stage. When an algorithm locates only a few nodes, it suffers less error accumulation, so as to decrease the overall error.

Fig. 11c shows the mean number of multilaterations needed to localize a node for each algorithm. The cost of OFA is quite stable over the tested range. For RQ, the cost is in the same level of OFA when the error magnitude is low, then it increases sharply when the error magnitude increases. This is because that the mean cost is highly related to the total number of successfully localized nodes, as discussed before.

6.5. Evaluation with SDP scheme

In this section, we evaluate OFA by comparing with a SDP-based method, which is also aimed to solve flip ambiguity. As SDP is computational intensive, we deploy 50 nodes in total, among which 10 nodes are anchors. The average degree of the network instances is about 10. The SDP-based method is designed for accurate distance measurements. When the maximum error is beyond 1%, it fails to locate most network instances. Hence, we restrict the maximum ranging error to be 0.001%, 0.01%, 0.1%, and 1%, respectively. We report the mean result of 50 network instances in Fig. 12 with diversified error magnitude.

Fig. 12a plots the proportion of successfully localized nodes against the error magnitude. The performance of OFA decreases slightly with the increase of errors, because the increased error results in higher drop rate in the final step of OFA. In contrast, the number of localized nodes for SDP decreases rapidly when the error magnitude enlarges. As SDP formulation is based on precise inter-node distances, the increased measurement errors make it hard to obtain the global optimum result. This then causes higher rejection rate when we evaluate the localization errors.

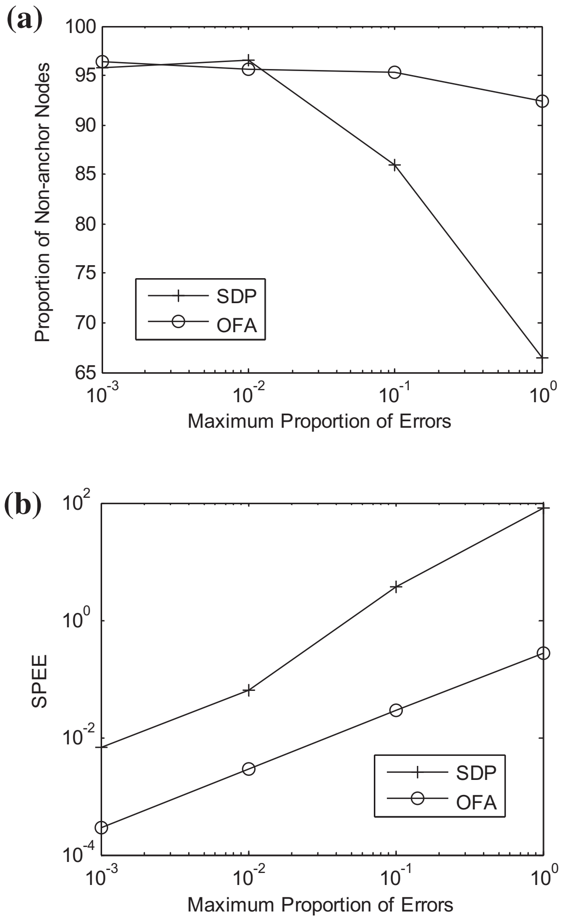


Fig. 12. Evaluation with SDP. (a) The number of robustly located nodes. (b) The average positioning error.

Fig. 12b plots the SPEE against error magnitude. Both SDP and OFA perform approximately linear error increase in the log–log scales. The actual error of SDP is much higher than that of OFA. As we have discussed, the measurement error may mislead the optimizing process of SDP to converge at a local minima. Unfortunately, this error influences all nodes, because SDP compute the locations of all nodes simultaneously. In contrast, OFA can build high accurate results step by step. As a result, OFA achieves fairly more accurate results than SDP does.

7. Conclusions and future work

By exploiting the characteristics of flip ambiguity, which causes either huge or zero error, we propose the concept of optimistic localization and design an algorithm, OFA, that employs a global consistency check and a location correction phase in the localization process. OFA can locate a larger number of nodes and yield more robust localization results simultaneously than existing approaches.

The future work leads into two directions. First, we will investigate the characteristics of combined flip, as naïve eliminating combined flip may introduce exponential cost with the number of flips. Second, to further improve the performance, we will apply the concept of clusters in OFA design. Currently, we are implementing OFA in our ongoing projects.

Acknowledgments

This work is supported in part by NCET, NSFC/RGC Joint Research Scheme N_HKUST602/08, National Basic Research Program of China (973 Program) under Grants No. 2011CB302705, the National High-tech R&D Program of China (863 Program) under Grants No. 2012AA01A301, 2012AA010901, and 2010CB328004, and NSF China 61170261, 61272142, 61272483, 60970118, 61272056, and 61272482.

References

[1] S. Lederer, Y. Wang, J. Gao, Connectivity-based localization of large scale sensor networks with complex shape, in: IEEE INFOCOM, Phoenix, Arizona, USA, 2008, pp. 789–797.

[2] D. Moore, J. Leonard, D. Rus, S.J. Teller, Robust distributed network localization with noisy range measurements, in: ACM SenSys, Baltimore, MD, 2004, pp. 50–61.

[3] Z. Yang, Y. Liu, Quality of trilateration: confidence-based iterative localization, IEEE Transactions on Parallel and Distributed Systems (TPDS) 21 (2010) 631–640.

[4] T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, P.N. Belhumeur, Rigidity, computation, and randomization in network localization, in: IEEE INFOCOM, Hong Kong, China, 2004, pp. 2673–2684.

[5] D.K. Goldenberg, A. Krishnamurthy, W.C. Maness, Y.R. Yang, A. Young, Network localization in partially localizable networks, in: IEEE INFOCOM, Miami, FL, 2005, pp. 313–326.

[6] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A.S. Morse, Y.R. Yang, Localization in sparse networks using Sweeps, in: ACM MobiCom, Los Angeles, CA, 2006, pp. 110–121.

[7] A.A. Kannan, B. Fidan, G. Mao, Robust distributed sensor network localization based on analysis of flip ambiguities, in: IEEE GlobeCom, 2008, pp. 1–6.

[8] A.A. Kannan, B. Fidan, G. Mao, B.D.O. Anderson, Analysis of flip ambiguities in distributed network localization, in: IEEE Information, Decision and Control, 2007, pp. 193–198.

[9] J. Liu, Y. Zhang, F. Zhao, Robust distributed node localization with error management, in: ACM MobiHoc, 2006, pp. 250–261.

[10] A. Basu, J. Gao, J. Mitchell, G. Sabhnani, Distributed localization using noisy distance and angle information, in: ACM MobiHoc, Florence, Italy, 2006, pp. 521–524.

[11] N. Patwari, J.N. Ash, S. Kyperountas, A.O.H. III, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, IEEE Signal Processing Magazine 22 (4) (2005) 54–69.

[12] A. Savvides, W. Garber, R. Moses, M.B. Srivastava, An analysis of error inducing parameters in multihop sensor node localization, IEEE Transactions on Mobile Computing 4 (2005) 567–577.

[13] R. Sugihara, R.K. Gupta, Sensor localization with deterministic accuracy guarantee, in: IEEE INFOCOM, Shanghai, China, 2011, pp. 1772–1780.

[14] A.M. So, Y. Ye, Theory of semidefinite programming for sensor network localization, in: SODA, 2005.

[15] P. Biswas, Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization, in: IEEE/ACM IPSN, 2004.

[16] Z. Zhu, A.M.-C. So, Y. Ye, Universal rigidity towards accurate and efficient localization of wireless networks, in: IEEE INFOCOM, 2010, pp. 1–9.

[17] A.A. Kannan, G. Mao, B. Vucetic, Simulated annealing based wireless sensor network localization with flip ambiguity mitigation, in: IEEE Vehicular Technology Conference, 2006, pp. 1022–1026.

[18] Y. Wang, S. Lederer, J. Gao, Connectivity-based sensor network localization with incremental delaunay refinement method, in: IEEE INFOCOM, 2009, pp. 2401–2409.

[19] L. Zhang, L. Liu, C. Gotsman, S.J. Gortler, An as-rigid-as-possible approach to sensor network localization, ACM Transactions on Sensor Networks 6 (4) (2010) 1–21.

[20] O.-H. Kwon, H.-J. Song, S. Park, Anchor-free localization through flip-error-resistant map stitching in wireless sensor network, IEEE Transactions on Parallel and Distributed Systems (TPDS) 21 (11) (2010) 1644–1657.

[21] C. Peng, G. Shen, Y. Zhang, Y. Li, K. Tan, BeepBeep: a high accuracy acoustic ranging system using COTS mobile devices, in: ACM SenSys, Sydney, Australia, 2007, pp. 1–14.

[22] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: MOBICOM, 2000.

[23] X. Wang, Z. Yang, J. Luo, C. Shen, Beyond rigidity: obtain localisability with noisy ranging measurement, International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC) 8 (1/2) (2011) 114–124.

[24] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, G. Zhou, J. Hui, B. Krogh, VigilNet: an integrated sensor network system for energy-efficient surveillance, ACM Transactions on Sensor Networks 2 (2006) 1–38.

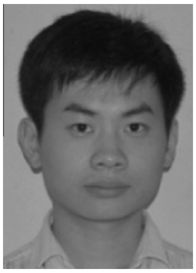
[25] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, Canopy closure estimates with GreenOrbs: long-term large-scale sensing in the forest, in: ACM SenSys, Berkeley, California, USA, 2009, pp. 99–112.

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

[27] S. Severi, G. Abreu, G. Destino, D. Dardari, Understanding and solving flip-ambiguity in network localization via semidefinite programming, in: IEEE Global Telecommunications Conference (GLOBALCOM), 2009, pp. 1–6.

Xiaoping Wang received his BS, MS, and PhD degree in the School of Computer Science from National University of Defense Technology, China, in 2003, 2005, and 2010, respectively. He is now an assistant professor in the School of Computer Science at National University of Defense Technology. His research interests include sensor networking and operating system.

Yunhao Liu (M'02-SM'06) received his BS degree in Automation Department from Tsinghua University, China, in 1995, and an MA degree in Beijing Foreign Studies University, China, in 1997, and an MS and a PhD degree in Computer Science and Engineering at Michigan State University in 2003 and 2004, respectively. He is a member of Tsinghua National Lab for Information Science and Technology, and the Director of Tsinghua National MOE Key Lab for Information Security. He is also a faculty at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. Being a senior member of IEEE, he is also the ACM Distinguished Speaker.



Zheng Yang received a BE degree in computer science from Tsinghua University in 2006 and a PhD degree from Hong Kong University of Science and Technology (HKUST) in 2010. He is currently a post-doc fellow in Tsinghua University. His main research interests include wireless ad hoc/sensor networks and pervasive computing. He has worked on location-based services for over 4 years and published a number of research papers in highly recognized journals and conference, including IEEE/ACM Transactions on Networking (ToN), IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE INFOCOM, IEEE ICDCS, IEEE RTSS, ACM SenSys, etc. He is a member of the IEEE and the ACM.

Kai Lu received his BS and PhD degree in the School of Computer Science at National University of Defense Technology, China, in 1995 and 1999, respectively. He is now a professor in the School of Computer Science at National University of Defense Technology. His research interests include high performance computing, operating system, parallel computing, and sensor networking.

Jun Luo received his BS degree in Computer School from Wuhan University, China, in 1984, and an MS degree in the School of Computer Science at National University of Defense Technology, China, in 1989. He is now a professor in the School of Computer Science at National University of Defense Technology. His research interests include operating system, parallel computing, security, and sensor networking.