Relative Localization with 2-Hop Neighborhood

Christopher J. Mallery
School of EECS, Washington State University
Pullman, WA 99164-2752
cmallery@eecs.wsu.edu

Sirisha Medidi and Muralidhar Medidi
Dept. of CS, Boise State University
Boise, ID 83725-2075
{sirishamedidi,MMedidi}@boisestate.edu

Abstract

Localization is the process in which nodes in a wireless sensor network self-determine their positions in the network. While there are many effective mathematical techniques for solving the problem of localization, most are not suitable for the resource-constrained distributed environment of sensor networks. We propose ANIML an iterative, range-aware relative localization technique for wireless sensor networks that requires no anchor nodes. ANIML restricts itself to the use of only local 1- and 2-hop neighbor information, avoiding the need for information flooding and thus controlling cascading ranging errors that bedevil other localization techniques. While least-squares minimization is a mathematically simple constraint optimization technique, utilizing 1- and 2-hop neighbor information as constraints, ANIML provides better localization without the need for more sophisticated error control and/or global information. We implemented ANIML in ns-2 and conducted extensive experimentation to evaluate its performance. Experimental results show that ANIML provides robust localization and scales well.

1 Introduction

Wireless sensor networks (WSN) are application-specific wireless ad-hoc networks populated with small, low-cost, resource-constrained immobile nodes equipped with one, or more, external sensors [10]. Regardless of specific application, WSN are envisioned to be densely deployed over large monitoring areas where knowledge of the deployed topology becomes critical for effective data dissemination. Many proposals sprang from the need for low-cost, lightweight and effective localization for WSN is important. The prohibitive cost of equipping sensors with GPS is the reason many localization techniques restrict GPS to only a small subset of the total network nodes, called anchors [10]. Considering the cost increase of equipping just a single node with GPS, localization techniques that minimizes the use of anchors become critical [14].

In most WSN applications, the precise location of each sensor node in the network (absolute localization) is not strictly necessary; instead overall topology identification (relative localization) is critical for sensor identification, routing, data fusion and data analysis [13]. Towards this goal, we propose a straightforward, iterative, anchor-free, range-aware relative localization technique for wireless sensor networks, called Anchor-free, local Neighborhood-based, Iterative MultiLateration (ANIML). Using only distance estimates between a node and its 1- and 2-hop neighbors, ANIML is capable of providing robust relative localization through least-squares multilateration without explicit error control or costly anchors. Also by restricting distance estimates to only 1- and 2-hop neighbors, instead of global information, we reduce the effects of cascading ranging errors; such cascading errors significantly affect the accuracy of many range-aware localization techniques [15].

2 Related Work

Previous attempts at localization in WSN fall into two groups: range-aware and hop-based. In range-aware techniques, such as SPA [2] and ILS [6], the calculation of node position estimates use inferred distances. A common assumption, which can provide more accuracy, is estimation of the distance between a normal sensor node and (one or more) beacons or (three or more) anchor nodes, which are nodes that can self-determine their own absolute positions. Hop-based methods, such as DV-Hop [8] and HCRL [14], require no ranging hardware, with the distance between nodes often simplified to the hops in the shortest path. Both range-aware and hop-based approaches often employ traditional methods, such as triangulation or optimization, to calculate node positions. However, often overburdened by constraints, most techniques reduce the problem such that traditional mathematical techniques can solve the problem. Most localization techniques for WSN provide absolute localization, however there are some techniques that do provide relative localization, such as MDS-MAP [11], SPA [2], the convex optimization technique in [3] and the distributed

978-1-4244-2100-8/08/$25.00

Kalman filter approach [12].

For the sake of space we now focus only on *iterative multilateration* localization techniques that directly relate to ANIML, interested readers should refer to and [5] and [7] for a more complete review of related works. Iterative multilateration localization approaches iteratively converge on a network topology [1]. These techniques are primarily range-aware, but can also be hop-based. Capkun *et al.* [2] have shown that nodes in MANETs without any anchor nodes can estimate positions by means of iteration, using their range-aware Self-Positioning Algorithm (SPA), to facilitate geographical routing. Using only local neighborhood (both 1- and 2-hops), each node builds a local map of its entire neighborhood and then aligns these individual maps. However, facing node mobility typical in MANETs, the estimated positions attempt to preserve inter-node distances in the local neighborhood and need not correlate with the physical network topology. Robinson and Marshall [9] present a distributed iterative multilateration approach of nodes guessing and re-guessing their position estimates. This series of guesses, by means of linear regressions, will converge to a topology that satisfies all distance estimates measured within the network and requires global network knowledge and a subset of anchors. Another iterative multilateration approach, although computationally expensive, uses distributed Kalman filtering and a subset of anchor nodes to handle localization [12]. Doherty *et al.* [3] proposed a centralized localization through convex optimization on local neighborhood geometric constraints. Liu *et al.* have recently proposed ILS which is a 1-hop neighborhood-based, iterative least-squares localization technique, which controls cascading ranging errors by scoring distance estimates [6]. This allows only known good estimates to be used for localization and the "bad seeds" to be filtered out. While ILS and ANIML have much in common, ILS strictly requires anchors in order to perform its localization. Recently proposed Sweeps [4] also is similar to ILS, designed specifically for sparse networks and uses graph theoretical methods instead of least-squares optimization.

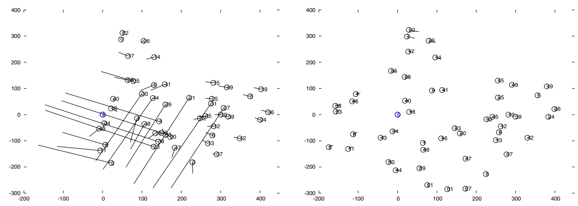
3 The ANIML Technique

The basic idea behind our iterative localization technique is for nodes to expand their positions outward, from their starting positions at the origin, towards their actual relative positions in the network, with each iteration. Since there are no anchors to provide absolute positions in the network, the localizing sensor nodes have no predefined coordinate system available on which they can converge. ANIML handles this by choosing a single node, the *reference node*, to remain "stationary" at the origin through all iterations. This gives nodes a common "absolute" position from which to expand outwards. Other than remaining at the origin, the reference node is identical in capability to all other sensor nodes in the network. We assume that the sensors are equipped with ranging hardware to estimate, from received transmissions, the distances to the direct 1-hop neighbors: thus, ANIML is a range-aware localization technique.

Node k:
while termination condition not met **do**
  BroadcastMessage()
  collect messages from neighbors
  for each message *rcvd* from a node i **do**
    $d_{k,i} \leftarrow$ measured distance estimate from node i
    update stored information for node i
    for each node j in *rcvd* list of i's neighbors **do**
      $d_{k,j} \leftarrow d_{k,i} +$ *rcvd* dist of node j from i
      update stored information for node j
    end for
  end for
  RecalculateCoordinates()
end while

Figure 1. Iterative ANIML Technique

The underlying multilateration technique in ANIML is the well-known least-squares constraint optimization. Fig. 1 outlines ANIML's iterative localization process, run independently on each node. ANIML iterations do not require any tight synchronization. In each iteration, every node calculates an updated position estimate, $\mathbf{x}$, given the received position estimates from its 2-hop neighborhood and associated distance estimates, using least-squares multilateration. Initially, every node will only be aware of its own estimated position, making it unable to recalculate a new estimated position. In which case, it will broadcast its current estimated position to its 1-hop neighbors. In the next iteration, each node will be aware of its estimated position, those of its 1-hop neighbors and the estimated distances of its 1-hop neighbors made through direct ranging. This information allows a node to begin recalculating its own position estimate. Since each node's initial position is the origin, this first recalculation will place a node roughly the average estimated distance it is from all of its 1-hop neighbors away from the origin in an arbitrary direction. Each node then broadcasts its new position estimate, in addition to the position estimates it has received from its 1-hop neighbors and the distance estimates it has made for its 1-hop neighbors. In the third, and subsequent iterations, each node is aware of its own estimated position, those of its 1- and 2-hop neighbors, the estimated distance of its 1-hop neighbors made through direct ranging and the estimated distances of its 2-hop neighbors. ANIML infers a node's distance from a 2-hop neighbor by adding the received distance estimate between the intermediate 1-hop neighbor and the 2-hop neighbor to the directly calculated distance estimate of the intermediate 1-hop neighbor. Now every node is fully able to



(a) 1-hop (b) 2-hop

Figure 2. ANIML with 1- and 2-hops

take advantage of least-squares multilateration to recalculate a more accurate position estimate, because those neighbors' estimated positions have spread apart. Additionally, the availability of 2-hop neighbor information allows the nodes of a neighborhood to begin moving closer towards their actual distance away from the reference node and any adjoining 1-hop neighborhoods.

By restricting distance estimates to only 1- and 2-hop neighbors, instead of globally propagated information, such as the positions of anchors, we reduce the effects of cascading ranging errors; such cascading errors significantly affect the accuracy of many range-aware localization techniques [15]. To control the message and computation complexity, we would have preferred to restrict ANIML to use only 1-hop neighbor information. However, we found that while this can provide accurate localization in some cases, often individual neighborhoods localize too rapidly based on only their own 1-hop neighborhood's information, fold onto each other, and get stuck at a local optimum. This problem is also encountered in ILS [6] and other techniques [8, 12]. Such folding of neighborhoods cannot be either detected or rectified with only 1-hop neighbor information. Fig. 2(a) shows the localization of a network by ANIML using only 1-hop neighbor information (estimated positions are denoted by circles with the arrows pointing to the true positions). The accuracy of the localization is poor with an average positioning error of 90 meters; however the average pairwise distance error is only 21 meters. However, by basing nodes' position calculations on 1- and 2-hop information, ANIML can prevent the folding of neighborhoods and from getting stuck at a local optimum. Two-hop neighbor information acts as a natural dampener to the localization process, slowing down the changes of nodes in each iteration. This allows neighborhoods that would otherwise rapidly reach a local optimum extra time to receive additional information that could prevent it from getting stuck. Fig. 2(b) shows the same topology as Fig. 2(a) localized by ANIML using 1- and 2-hop neighbor information. The localization has an average localization error of only 8 meters with an average pairwise distance error of 3 meters.

Even after introducing significant ranging error in the 2-hop neighbor distances, due to triangular inequality, the iterative nature of ANIML naturally places a node into its correct position when its neighborhood is well distributed around it. Problems occur when a node's neighbors are biased in one direction from the node (i.e. corner and edge nodes). Corner and edge nodes can end up estimating their position on the "wrong" side of their 1-hop neighborhood. These inappropriately placed nodes appear "flipped" into their 2-hop neighbors, towards the center of the network. In order to combat the problem of anomalous flipped nodes we extended ANIML with a simple sanity check technique to detect a flip and correct it if necessary.

4 Performance Evaluation

We implemented ANIML in ns-2 and compared its effectiveness to APS (DV-Hop) [8], a popular technique for baseline comparisons. Since the authors' own results show that DV-Hop outperforms DV-Distance, we compare against DV-Hop instead of the range-aware DV-Distance. The simulation environment for ANIML uses 802.11 MAC. We obtained all DV-Hop data by replicating the experiments using the DV-Hop authors' CAML implementation of APS. We used both 5% and 10% anchor distributions in the DV-Hop experiments. We generated topologies in four different sizes (250 x 250, 500 x 500, 750 x 750 and 1000 x 1000 m^2) and two different node densities (400 and 800 nodes/km^2) in order to investigate ANIML's scalability. The maximum transmission range of each sensor is 250 meters, although our presented results scale to smaller transmission ranges. Distance estimates for ANIML are obtained by adding a uniformly distributed error (0-90%) to the true distance between two neighboring nodes to mimic experiments reported in [8]. Each data point presented in our plots is the average of ten runs with differing random seeds, with no discarding of outliers.

The metric for localization effectiveness, used in the literature, is the average distance away their estimated positions are from the nodes' actual positions in the network. We give the measurement of effectiveness as a percentage of the transmission range of the sensor nodes in the network.

Since ANIML produces relative localization, the determined network coordinates may have undergone a global flip, rotation and/or shift making direct comparisons to the actual coordinates difficult. Therefore, comparisons are done post localization by (i) shifting the real coordinates by the difference between the origin and the reference node's true position, (ii) globally rotating both the real and relative coordinates to place node 1 on the y-axis and (iii) if needed, flipping the relative set of coordinates to place them in the same coordinate space. No scaling of position estimates is involved in this transformation.

Providing only 1-hop neighbor information to ANIML can lead to poor overall localization due to neighbors get-

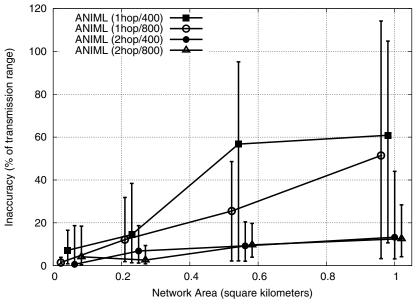


Figure 3. 1-Hop ANIML vs. 2-Hop ANIML

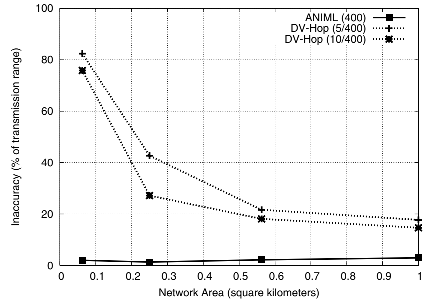


Figure 4. Localization in Uniform Topologies

ting stuck at local optima. Fig. 3 shows the localization effectiveness of ANIML using 1-hop information compared to ANIML using 1- and 2-hop information. The data points are shifted slightly left and right of the real values (0.0625, .25, .5625, 1) on the x-axis to allow for clear presentation of the error bars. Fig. 3 shows that ANIML using 1- and 2-hop information provides better overall localization accuracy than ANIML using only 1-hop information. More importantly, the results also show that the problem with using only 1-hop information is not necessarily the accuracy, but the consistency of localization. By simply adding 2-hop information, ANIML is able to significantly improve the consistency of its localization results, as can be seen by the much tighter error bars.

Fig. 4 shows the localization effectiveness of ANIML and DV-Hop in uniform topologies. For clarity we only present the 400 nodes/km^2 results, naturally as density increases so does accuracy. The results show ANIML provides more accurate localization than DV-Hop. As a comparison, ILS's localization effectiveness, based on the authors' results presented in [6], is approximately 20%. More importantly, the results show ANIML does not need any anchors to provide accurate localization.

5 Conclusions & Future Work

We have presented ANIML, an iterative, anchor-free, range-aware relative localization technique for wireless sensor networks that requires no explicit error control or global information. By explicitly basing a nodes positioning off of its 1- and 2-hop neighborhood, instead of just its 1-hop neighborhood, ANIML is capable of providing accurate localization, even in the presence of packet losses, using nothing more than simple iterative least-squares. While adding 2-hop information does slightly increase the computational complexity of ANIML's least-squares multilateration by a constant factor, using 1- and 2-hop information allows ANIML the ability to provide resilient localization. Through simulation, despite using a non-idealized MAC, we showed that ANIML provides good relative localization and better accuracy than DV-Hop in uniform topologies. Our future work on ANIML focuses on further reducing the message complexity without sacrificing accuracy or convergence time and implementing it on actual sensor motes. We are also adapting ANIML to address localization in three dimensional sensor networks.

References

[1] T. Adebutu, L. Sacks, and I. Marshall. Simple position estimation for wireless sensor networks. In *Proc. of LCS 2003*.

[2] S. Capkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *HICSS-34*, 2001.

[3] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. of INFOCOM 2001*, volume 3, pages 1655–1663.

[4] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse, and Y. R. Yang. Localization in sparse networks using sweeps. In *MobiCom 2006*, pages 110–121.

[5] J. Hightower and G. Borriello. A survey and taxonomy of location systems for ubiquitous computing. Technical report, Univ. of Washington.

[6] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *Proc. of MobiHoc 2006*, pages 250–261.

[7] M. Medidi, R. Slaaen, Y. Zhou, C. Mallery, and S. Medidi. Scalable localization in wireless sensor networks. In *Proc. of HiPC, LNCS 4297*, pages 522–533, 2006.

[8] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *GLOBECOM 2001*, volume 5, pages 2926–2931.

[9] D. P. Robinson and I. W. Marshall. An iterative approach to locating simple devices in an ad-hoc network. In *Proc. of LCS 2002*.

[10] Y. Sang and W. Ruml. Improved MDS-based localization. In *Proc. of INFOCOM*, volume 4, pages 2640–2651, 2004.

[11] Y. Sang, W. Ruml, and Y. Zang. Localization from mere connectivity. In *Proc. of MobiHoc*, pages 202–212, 2003.

[12] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proc. of ACM WSNA*, pages 112–121, 2002.

[13] M. Wong and D. Aksoy. Relative accuracy based location estimation in wireless ad hoc sensor networks. In *Proc. of ICC 2007*.

[14] S. Yang, J. Yi, and H. Cha. HCRL: A hop-count-ratio based localization in wireless sensor networks. In *SECON 2007*.

[15] J. Yi, S. Yang, and H. Cha. Multi-hop-based monte carlo localization for mobile sensor networks. In *SECON 2007*.