Localised Convex Hulls to Identify Boundary Nodes in Sensor Networks

Marwan Fayed*
School of Information Technology and Engineering,
University of Ottawa, Canada
email: *mmf@site.uottawa.ca*
*Corresponding author

Hussein T. Mouftah
School of Information Technology and Engineering,
University of Ottawa, Canada
email: *mouftah@site.uottawa.ca*

Abstract: Intuitively, identification of nodes close to the network edge is key to the successful setup, and continued operation, of many sensor network protocols and applications. Many virtual coordinate constructions rely on the furthest set of nodes as beacons, and sensing applications may find useful the knowledge of the network edge. In this paper we propose local convex view (*lcv*) as a means to identify nodes close to the network edge. It is motivated by the hypothesis that some structural information relevant to the network is buried within view of many nodes. The *lcv* differs from most previous methods in that it is a localized algorithm. Nodes using *lcv* may establish neighbourhood coordinates if no location information is available a priori. In those cases where needed information is missing, we adopt a simple probabilistic model to decide the boundary status of a node. We identify two metrics for evaluation and compare via simulation the performance of lcv against two methods with similar properties. Further simulations reveal the surprising observation that *lcv* seems unaffected by position estimation error. We enumerate and analyse a complete set of of node configurations seen by *lcv*. We conclude that the geometric properties underlying *lcv* are responsible for its resilience to error.

Keywords: sensor networks; boundary detection; edge detection; localisation; convex hulls.

1 Introduction

A growing number of projects are deeply engaged in research to provide context-awareness to nodes in wireless networks. An appropriate level of context-awareness is a key property for the success of many wireless applications.

Intuitively, pure sensing applications may benefit from knowledge of the network boundary (see Ben-Chen et al. 2006, Cao & Abdelzaher 2004, Fang et al. 2005, Fonseca et al. 2005, Langendoen & Reijers 2003, Rao et al. 2003, Shang et al. 2003, Zhao et al. 2005). For example, nodes close to the network boundary are often assumed to provide the best candidates for beacon nodes in virtual coordinate constructions. The assumption is that the set of beacons furthest apart produces the finest resolution in coordinates. Moreover, awareness of the network boundary prevents the processing of requests that are impossible to satisfy when the request lies outside of the network space. To date, identification of nodes along the network edge remains a challenging problem.

In this paper we introduce the local convex view (*lcv*) as a heuristic approach to identify nodes close to the network edge. The local convex view is defined as the convex hull of the nodes within range. (Thus, as many local convex views exist as nodes in the network.) In the simplest terms a node decides it is close to the network boundary by asking the following question: Am I in my own local convex view? It is inspired by the knowledge that the convex hull of a set of points consists of the outermost points in the set. Furthermore, it is motivated by the hypothesis that within view of many nodes there exists structural information relevant to the network.

For those contexts where position information is unavailable *lcv* assigns local coordinates using its 1-hop distance measurements and the 1-hop measurements of its neighbours. Each node constructs a Cartesian space by plac-

ing itself at the origin and its furthest neighbour along the horizontal axis. All remaining nodes in the neighbourhood are assigned coordinates relative to established coordinates. The success of this triangulation approach relies on the assumption that a node's neighbourhood is 2-connected, ie. removing the node in question leaves no disconnected components. Where this assumption fails, we show the maximum number of possible disconnected components is 5. Furthermore, when a neighbourhood is not so connected, we suggest a simple probabilistic model to determine boundary status according to the criteria set in *lcv*.

In our evaluation we simulate networks of varying size and topology. Topologies vary by generating networks where node locations are selected from uniform, normal, and skewed (Pareto) distributions. In our investigation we have identified two metrics for comparison. The *edge proximity* measures the likelihood of boundary-node declarations that occur relative to the network edge. The *regional proportionality* measures the proportion of boundary- versus regular-nodes relative to the network edge.

We first compare *lcv* against the 2-hop method from Rao et al. (2003) and an adaptation of the tent-rule in Fang et al. (2004), whose localised operations and limited messenging complexity make them suitable for comparison. Results indicate that the tent-rule is unsuitable for network edge detection. The underlying cause is in its design which is to identify nodes that abut *any* unreachable region, including those that are 'inside' the network. The 2-hop method, which gathers some global information, generally returns the most accurate results. While perhaps unsurprising, this observation is misleading. Further investigation shows that the 2-hop method reveals clusters of boundary nodes as the network grows dense. This leaves some regions of the network edge over-represented while others are left under-represented. We find this is an artifact of the way in which the 2-hop method gathers information. By contrast, *lcv* reveals a reasonable set of boundary nodes. Unlike the 2-hop and tent-rule methods, *lcv* is fairly resilient to changes in topology.

Our evaluation is further propelled by the growing number of studies that suggest error in position estimation is currently unavoidable (see Niculescu & Nath 2004, Savvides & Garber 2005, Whitehouse et al. 2005). We insert position error to the system by blurring neighbour positions stored at each node. In doing so, a pair of neighbouring nodes is unlikely to agree on the relative position of any third node in common view.

The results are surprising and counter-intuitive: The difference in accuracy between the *lcv* in a perfect environment and one with position estimation error are statistically insignificant. Motivated by this observation we enumerate a complete base set of node configurations that may be seen by *lcv*. Our analysis reveals that *lcv* is immune to two of these configurations. Further simulations show the frequency of false-positives and false-negatives imposed by a third, ambiguous, configuration is low. We conclude that the geometric properties underlying *lcv* are responsible for its resilience to error.

In summary, in this paper we develop a boundary node selection method. It uses only nodes in range and resolves any needed information that is missing. We study our method alongside two methods with similar properties, and in environments where position estimation is erroneous. Our simulations and analysis reveal that *lcv* is resilient to the impediments faced by competing methods, as well as errors in position estimation.

2 Context and Related Work

Our work appears amid a growing body of research on boundary detection. We focus on works that are distributed or localized in nature. Existing work may be classified according to the taxonomy presented in Wang et al. (2006) as either geometric, statistical, or topological in nature.

Geometric solutions to the boundary identification problem use the positions associated with each node. Our work falls into this category. This class of solution is simple and localised but also most vulnerable to position errors. To our knowledge the work by Fang et al. (2004) is the first such work. In it the authors find those nodes abutting any 'hole' in the network as defined by greedy routing techniques. Their methods detect such nodes using local information. A network node is found to abut a local minima if its neighbourhood reflects a geometry as defined by a tent-rule. Since their method is both geometric and local, we have selected the tent-rule method to compare against our own. For this reason we reserve the details of the tent-rule for discussion later in this section.

Probability distributions underlying network deployments have been used to formulate statistical solutions. One solution, proposed by Fekete et al. (2004), relies on the idea that nodes close to network boundaries have fewer incident edges in the network graph than internal nodes. The authors use statistical methods to derive suitable thresholds to separate edge nodes from internal nodes using the node degrees. In S.P. Fekete and M. Kaufmann and A. Krller and N. Lehmann (2005) a similar statistical separation is proposed. Boundary nodes are separated from internal nodes by using a 'centrality' measure which counts the number of shortest paths that pass through a node. A higher centrality value occurs among internal nodes. Statistical solutions generally hinge on uniformly distributed networks and exceedingly high densities. Also, it is unclear if or how statistical methods might suffer in the presence of estimation error. Our method shares in the view that nodes at the boundary exhibit unique characteristics. Unlike statistical methods, our approach is localized, is shown to be resilient to the underlying distribution, and performs well in lower densities environments.

Topological solutions appear in Funke (2005), Kröller et al. (2006), Rao et al. (2003). Kröller et al. (2006) propose a combinatorial approach; it is the only deterministic work of which we are aware to produce correct re-

sults without relying on the unit disc graph model (where all communication ranges are normalized). This solution comes at a high cost: It deals with complex combinatorial structures in a distributed manner. Rao et al. (2003) suggest a single-beacon broadcast solution. We compare this method with our own since its total messaging cost is similar to localized approaches such as ours. A more detailed discussion appears later in this section. Funke (2005) later proposed extensions to a similar idea. The idea relies on the observation that 'rings' of the network, described by hop-distances to a beacon, are broken when encountered by a network boundary. This project was later refined in Funke & Klein (2006). Finally, topological inference is used by Wang et al. (2006) to detect internal boundaries. Their detection method works by identifying the distinct portions of similar paths that span the network. We expect that topological methods' use of connectivity would render them immune to estimation errors. However, topological solutions are generally expensive and centralised in nature.

To our knowledge this is the first study that also deals explicitly with edge detection in the presence of position estimation error. We emphasize that our goal, different from most boundary detection methods, is to identify a set of nodes that lie close to the network edge. Needing only 1-hop information, our method is localized and converge quickly. The cost of localization is reduced accuracy. To properly evaluate our algorithm we have selected the following two approaches that function under similar computational and messenging constraints.

The 2-hop method (Rao et al. 2003). Stemming from a need to identify a set of furthest beacons, the authors inject an anchor into the network. Each node in the network records and transmits its hop distance to the anchor. Any node that finds itself the furthest of all its 2-hop neighbours from the anchor becomes a beacon. On account of the focus of this work being not the beacon selection but an ensuing coordinate construction, no evaluation of the quality of selection was provided.

Tent-rule (Fang et al. 2004). The tent-rule, so named for its appearance when represented diagrammatically, identifies all unreachable regions as defined by greedy routing. It works by sorting neighbours angularly about a node. If the bisectors of edges to contiguous neighbours intersect outside of range of the node, then the node abuts an unreachable region. Given that the network boundary itself delineates an unreachable region, the adaptation of the tent-rule for our investigation is appropriate.

3 Local Convex View

In this section we describe the man contribution of our paper, an autonomous method for selecting a subset of nodes to describe the network edge. Our method relies on the hypothesis that within the local view of each node some structural information exists. Our algorithm makes only the assumptions that generated or assigned node IDs within each neighbourhood are unique, and that distance measurements are available. We consider a deployment of a large wireless network where, initially, nodes lack any knowledge of their positions.

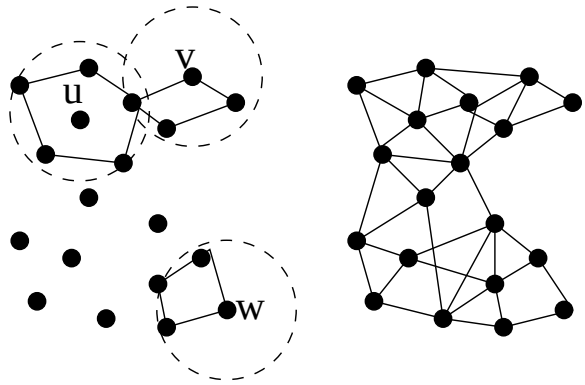


Figure 1: (Left) Nodes u, v, w compute their local convex hull; only v and w declare boundary status. (Right) The collection of local convex hulls closely resembles the α-hull.

3.1 Local Boundary Node Identification

A set of nodes far apart may be obtained by finding the convex hull[1] of a set of nodes. There are potential drawbacks to choosing the convex hull of the network: Too few boundary nodes may be chosen, the shape of irregular networks may fail to appear and, finally, we know of no methods by which an approximation of the convex hull may be localized.

By contrast we can localise the convex hull computation to capture the 'shape' of a sensor network. The idea is illustrated in Figure 1 where we compute the local convex view for each node. A node's local convex view consists of the set of nodes that comprise the convex hull of the neighbourhood in view. (Thus there are as many local convex views in a network as there are network nodes.) Observe in Figure 1 that by taking the set of "outside" nodes from the set of local convex views, an $\alpha-$hull-like structure begins to emerge. We use this observation to motivate our method for boundary node identification.

In the simplest terms a node decides if it lies on the network boundary by asking the following question: Am I in my own local convex view? With the return of a positive answer, a node declares itself to be a boundary node. Referring again to Figure 1 we focus on the nodes labelled u, v, w. Using the local convex view criteria nodes v and w declare themselves to be boundary nodes while node u sits idle. We evaluate the accuracy of this criteria in Section 4 and proceed with a discussion of the algorithm and its merits.

The complete procedure appears in Algorithm 1. For each node u the boundary node declaration process consists of three steps. Steps 1 and 2 remedy the initial lack

[1] In two-dimensions, a set S of points is defined as *convex* if for every $x \in S$ and $y \in S$, the segment $xy \subseteq S$ (O'Rourke 1998).

Algorithm 1 Boundary Node Identification Algorithm at any node u.

1: Share the distance measurements to single hop neighbours.
2: Set u as the origin of, and construct local coordinate system.
3: Compute the local convex view (lcv).
4: **return** $u \in lcv$

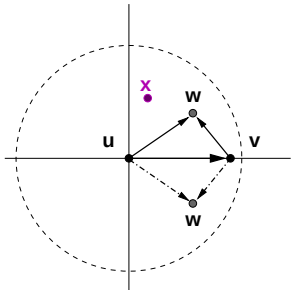


Figure 2: Neighbourhood coordinate establishment.

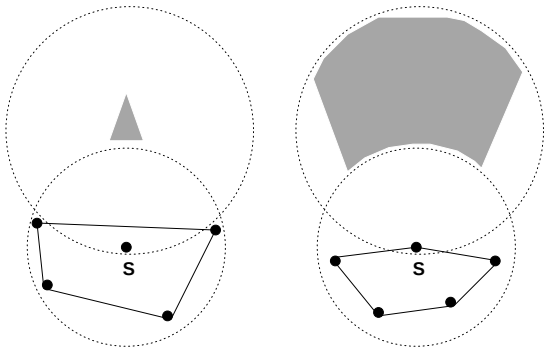
of position awareness. Recall that network nodes lack any knowledge of their positions, information that is necessary to compute a convex hull. We obtain this information by constructing a local coordinate system as this is sufficient to compute the local convex hull.

Each node in the network begins by sharing its 1-hop distance measurements with its neighbours. Once obtained, each node constructs a local coordinate system by placing itself at the origin and, in a depth-first manner, places remaining neighbours relative to those whose local coordinates have been established. We demonstrate this idea from the perspective of node u in Figure 2. Node u places itself at the origin of a Cartesian space and sits neighbour v on the horizontal axis. (For our purposes we use the furthest neighbour.) The next node, w, may be placed on either side of the horizontal axis since convex hull computations are unaffected by rotations and translations. Remaining nodes are placed similarly and with respect to established coordinates.

In the final step a node computes the convex hull over the neighbourhood in view. If it sits within its local convex view (ie. in the set of points describing the convex hull of the local neighbourhood) then it declares itself to be a boundary node.

3.2 Correctness of Local Convex View

It is impossible for a node to declare boundary status unless it sits on a network boundary, as demonstrated in Figure 3. We can see that node s may border an unreachable region according to two definitions: Figure 3a demonstrates the case in which we are uninterested since s has neighbours that lay qualitatively closer to the unreachable region (neighbours of s sit closer to the greyed region than the tangent at s); by contrast Figure 3b demonstrates that, within view, s is closest to the unreachable region. This implies that false positives may be returned (eg. routing 'holes' inside the network), and that false negatives are impossible.



(a) The 'concave' case. (b) The convex case.

Figure 3: A node may abut unreachable regions in two ways. The lcv detects regions that are outwardly unreachable.

The cost of the algorithm is one transmission to share distance vectors, and $O(d \log d)$ computation where d is the number of nodes in the neighbourhood. The local convex view avoids broadcasts and inter-node cooperation. It is both protocol and architecture independent.

To better illustrate the outcome of our method we use the three networks depicted in Figure 4. Each network consists of 3000 nodes in a 200x200 space each with a range of 8 units. Figure 4a represents a network where node locations are distributed according to a uniformly random distribution; in Figure 4b node locations are distributed according to a normal distribution; in Figure 4c node locations are distributed according to a skewed (Pareto) distribution. Within each figure we plot the complete set of network nodes on the left and the subset of nodes that declare boundary status on the right. Comparing the right and left plots we see the majority of nodes that declare beacon status lie in the outer regions of the network.

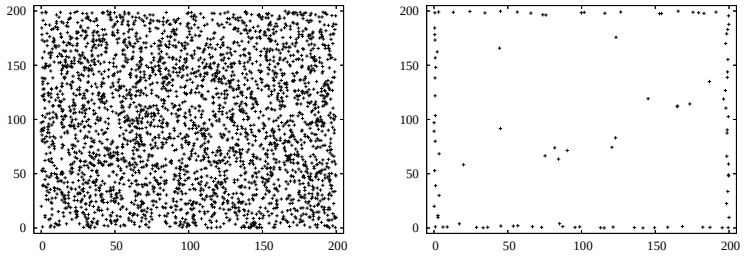
Our continued evaluation appears in Section 4. Next we construct a simple probabilistic model to deal with missing information during coordinate assignment.

3.3 Dealing with Incomplete Information

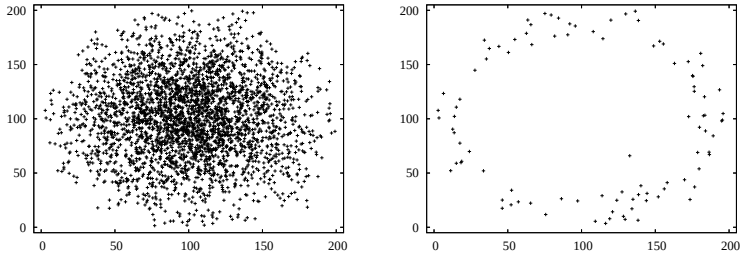
We previously described the way each node assigns coordinates to neighbours. There remains in this approach a caveat. Say node u constructs its neighbourhood coordinate system according to Algorithm 1. Coordinate assignments may only succeed if u's neighbourhood is 2-connected[2]. We can also say the removal of u must leave a single connected component. Without 2-connectedness a node will be unable to assign a coordinate to at least one neighbour. In this section we suggest a probabilistic solution.

Consider two contiguous neighbours around u sorted in angular order. By normalizing the communication

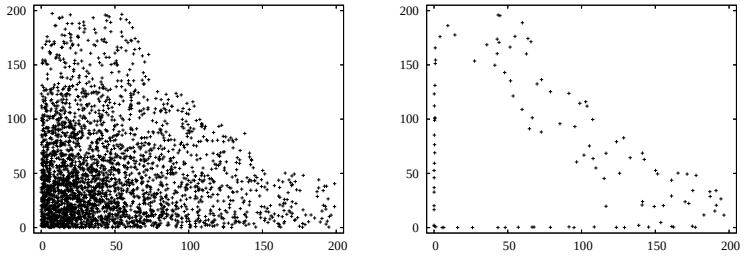
[2]A 2-connected network is one in which there are 2 disjoint paths between every pair of nodes.



(a) A uniform network.



(b) A normal network.



(c) A skewed (Pareto) network.

Figure 4: Example networks of 3000 nodes with varying topologies on the left. The corresponding *lcv* nodes for each network on the right.

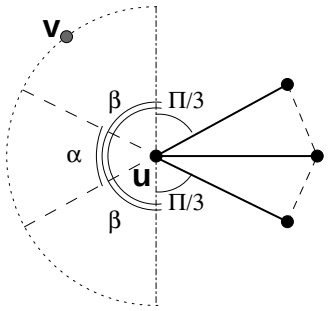


Figure 5: During coordinate assignment u finds v is disconnected from the remaining neighbourhood. Node u lies on the *lcv* only if neighbour v sits in the range denoted by β.

range to 1 and solving the associated cosine rule, $1^2 = 1^2 + 1^2 - \cos A$, we determine the maximum possible angle between two communicating neighbours to be $\pi/3$. Using this result we can show that there exists at most 5 disconnected components (ie. non-communicating neighbours) around u. Furthermore, a node cannot lie on its local convex view if it has greater than 3 such components.

In the example shown in Figure 5 node u has assigned coordinates to three neighbours in a connected component. Wishing to assign a coordinate to node v, node u sees that v communicates with no other neighbour of u and so must lie somewhere along the dotted arc. If v lies in the ranges described by angle β then u must sit on its local convex view. Hence, u sits on its *lcv* with probability

$$\mathrm{p}[u \in lcv] = \frac{2\beta}{2\beta + \alpha}. \tag{1}$$

The range of angles of α is equivalent to the range of angles covered by the neighbours with assigned coordinates. So β may be written as

$$\beta = \pi - (\alpha + \frac{\pi}{3}). \tag{2}$$

Note that the probability $\mathrm{p}[u \in lcv] = 0$ when $\alpha \geq 2\pi/3$. Then, by substituting Equation 2 into 1, node u declares it is on its *lcv* as follows,

$$\mathrm{p}[u \in lcv] = \begin{cases} \frac{\frac{4\pi}{3} - 2\alpha}{\frac{4\pi}{3} - \alpha}, \text{if } \alpha < \frac{2\pi}{3} \\ 0, \text{ otherwise.} \end{cases} \tag{3}$$

We have demonstrated a probabilistic solution for the case where there are two disconnected components. This

approach may be similarly extended for three such components. We omit the details due to space limitations.

4 Performance Evaluation

In this section we evaluate *lcv* with the tent-rule and the 2-hop methods, described in Section 2. Our comparison is followed with an evaluation of *lcv* when position estimation error is inserted into the system.

4.1 Experimental Design

To evaluate and compare the algorithms, we simulate networks of varying density and distribution. Network nodes are distributed in a 200x200 unit space, each node with a fixed range of 8 units. We vary node density by changing the network size. Note that by changing size instead of communication range we can vary neighbourhood density without affecting the maximum network diameter. Network sizes are 1500, 2500, and 3500 nodes. (In the uniform networks this results in average neighbourhood sizes of 7, 12, and 17 nodes.) To obtain more accurate results we tabulate and experiment over the largest connected component of each network. Experiments repeat over 25 runs of each network generated using non-overlapping streams.

Nodes locations are chosen from a normal or skewed (Pareto) distribution in addition to the uniform distribution traditionally used to generate wireless network topologies. Uniformly distributed networks may be sufficient to provide insight yet are poor representations of many real deployments. Normal coordinates are generated with an average of 100 (the center) and a standard deviation of 40. Skewed coordinates are chosen from the Pareto distribution with scale parameter 1.0 and shape parameter 100.5. Examples topologies appear in Figure 4.

The choice of appropriate metrics is not obvious. We have isolated two metrics we believe to be suitable for this study. Our measures of success center on locations of nodes that declare boundary status relative to the network boundary. For this reason networks are separated into partitions appropriate for each network type, described in Section 4.3.1. Measures of success are represented as follows:

- We measure the *edge proximity* as the probability that the location X of a declared boundary node lies in region x. This measures the likelihood that nodes sitting on their local convex view are good beacon candidates (ie. sitting close to the network boundary).
- We measure the *regional proportionality* as the percentage of nodes within a region that declare boundary status. This ratio should be highest towards the true network boundary. It is designed to reinforce those methods that are more likely to find nodes close to the edge of the network.

Our simulations implement the algorithm described in Section 3 with one addition. Nodes that observe a local convex view consisting of three or fewer nodes return negative boundary status. Our experiments show that this reduces the number of reporting nodes in dense networks to a manageable level without compromising results in sparse networks.

4.2 Performance in a Perfect Environment

Our evaluation begins with a direct comparison between the performance of *lcv*, as well as the tent-rule and the 2-hop methods described in Section 2. These methods' similarities to *lcv* in properties makes them appropriate for comparison.

4.2.1 Boundary Node Set Size

One of our goals is to identify a reasonable set of boundary nodes: too many nodes create ambiguity, while too few risk sacrificing resolution. We report the number of actual reporting nodes as the boundary node set size. The boundary node set size is largely a subjective measure we use to gain insight into measures that are later used.

We expect and confirm uniformly generated networks to be the least-well performing of the three networks we study. Tables 1, 2, and 3 list the average size of the largest connected component (lcc), and the average number of nodes that declare boundary status for each network, for all methods. All values appear with their 99% confidence intervals.

The 2-hop rule consistently returns the smallest number of nodes that declare boundary status. We believe the underlying causes are that i) the 2-hop method compiles global information and that ii) it maintains a record of its 2-hop neighbourhood, allowing decisions that are better informed. (We will discover this observation to be untrue in later sections.) Despite the increased knowledge of the 2-hop method, *lcv* remains competitive in all but the sparsest of uniform networks. Across all networks the tent-rule returns 2-3 times greater a number of nodes than the best performing method. The underlying cause is that the tent-rule is designed to report all routing holes, including holes that are 'internal' to the network.

Comparing the values from Tables 1, 2, and 3, the number of boundary-status nodes appear to be most greatly affected by density in uniform networks. For example, in uniform networks the ratio of boundary-to-regular nodes reported by *lcv* varies from 1 in 6 in the sparsest networks to 1 in 30 among dense networks. While there is some variation in non-uniform network numbers, the effects associated with increases in network size are much less pronounced.

4.2.2 Edge Proximity

The boundary node set size hides the locations of the nodes that report boundary status. In this section we measure the proximity of reporting nodes to the edge of the network. The edge proximity, depicted in Figure 6, is tabulated as the cumulative distribution over partitions of the

Table 1: No. of boundary nodes returned in Uniform networks with 99% confidence intervals.

Network Size (Neighbourhood)	Method			Largest Conn. Comp.
	lcv	tent rule	2-hop	
1500 (7)	264.4 ± 8.9	466.7 ± 8.1	130.3 ± 9.0	1490 ± 7.5
2500 (12)	140.9 ± 5.1	299.3 ± 6.0	103.6 ± 5.6	2499.8 ± 0.4
3500 (17)	100.6 ± 3.4	181.6 ± 5.6	101.4 ± 9.8	3499.9 ± 0.2

Table 2: No. of boundary nodes returned in Normal networks with 99% confidence intervals.

Network Size	Method			Largest Conn. Comp.
	lcv	tent rule	2-hop	
1500	83.9 ± 5.8	161.0 ± 6.3	63.2 ± 4.4	1406.7 ± 7.7
2500	87.3 ± 4.9	160.1 ± 4.9	68.0 ± 3.8	2433.8 ± 4.9
3500	88.0 ± 4.6	155.4 ± 5.5	69.6 ± 3.9	3450.8 ± 5.3

Table 3: No. of boundary nodes returned in Skewed networks with 99% confidence intervals.

Network Size	Method			Largest Conn. Comp.
	lcv	tent rule	2-hop	
1500	103.8 ± 7.1	168.8 ± 9.3	78.3 ± 10.2	1359.0 ± 12.9
2500	116.0 ± 5.6	216.7 ± 11.1	114.3 ± 15.4	2382.2 ± 10.9
3500	125.4 ± 6.0	244.1 ± 9.3	135.6 ± 17.4	3403.8 ± 12.3

network. Each type of network is partitioned in a fashion that is appropriate for its overall shape, according to the following criteria.

Uniform networks are partitioned into quadrilateral 'rings'. Each ring is of width equivalent to $0.25R$, where R is the communication range.

Normal networks are partitioned into rings that are 0.25 standard deviations in width. The statement "80% of reporting nodes sit outside 2σ" may be interpreted as 80% of reporting nodes sit amongst the outermost 5% of network nodes.

Skewed networks are partitioned into diagonals that span from the y to the $x-$ axis, 10 units apart. So, for example, nodes that report in the 180 region have x and y coordinates with a sum greater than 180.

The accuracy of all three methods generally follows similar trends. Among normal networks depicted in Figures 6d-6f each method reports a very small likelihood of edge proximity amongst the furthest 1% of nodes, and quickly converges to much higher values. As network size increases the curves shift to the left, indicating the point at which the cumulative distributions converge to 1 occurs further from the network center. Among the skewed networks in Figures 6g- 6i the 2-hop method reports the greatest number of boundary nodes closest to the edge, though all methods converge on 1 very quickly. As with the normal networks, convergence to 1 shifts further from the origin (a left shift in the curves) as the network increases in size. One important note: in normal and skewed networks the network edge physically occurs further from the origin as the network grows large. Therefore the conclusion that increases in network size are responsible for the increased accuracy represented by curves shifting to the left should be avoided.

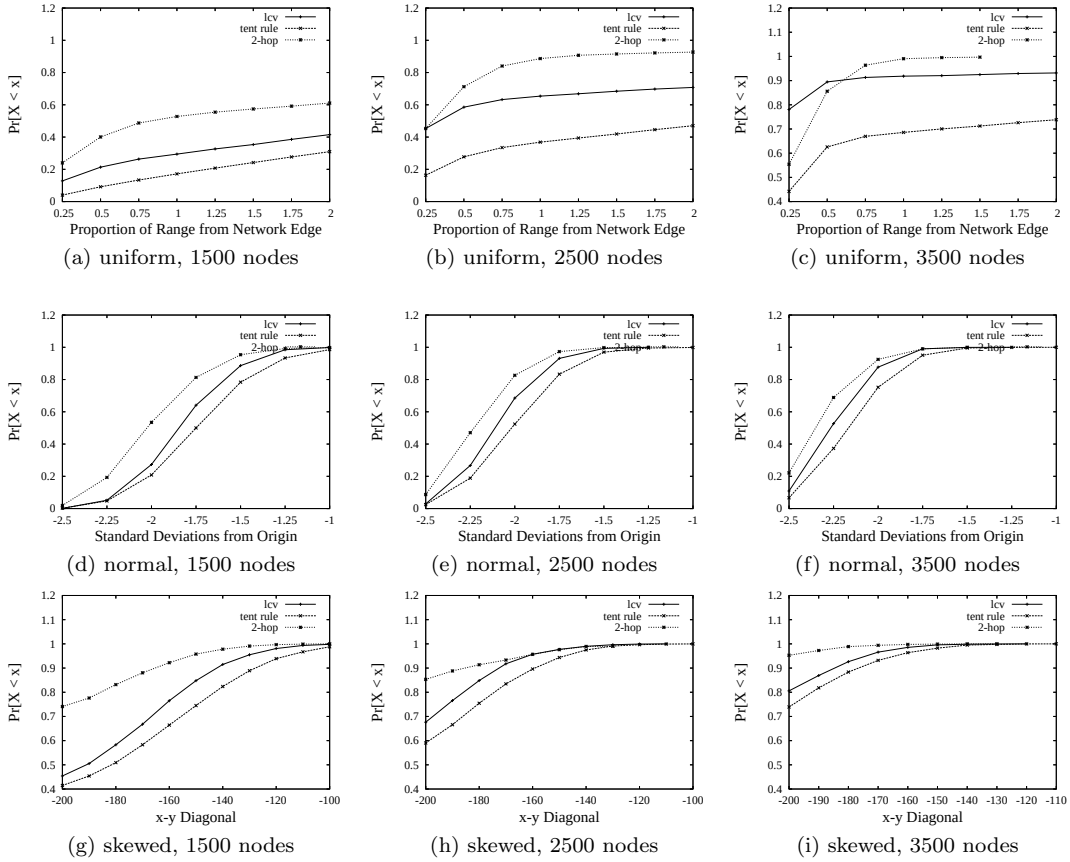
One observation requires special attention. Notice as uniformly random networks increase in size from 1500 nodes in Figure 6a to 3500 nodes in Figure 6c, the increase of boundary nodes reported by *lcv* in the outer-most $0.25R$ increases to twice that of the 2-hop method. Further investigation reveals that the 2-hop method suffers from a clustering effect that is due to the way it records hop counts. In a wireless environment where a node location is recorded as the hop-count from a beacon, many neighbouring nodes will share the same hop distance. These neighbours may span a region as wide as R units. In higher density networks the 2-hop method will heavily concentrate some boundary nodes along some portions of the network edge, while leaving other portions under-represented. This phenomenon is illustrated in Figure 8 which shows the boundary nodes as declared by the 2-hop method in networks of 3000 nodes. The conclusion to be drawn is that the 2-hop method is adversely affected by higher density environments.

4.2.3 Regional Proportionality

Edge proximity reveals the position of nodes declaring boundary status relative to the edge of the network. In this section we seek further insight by evaluating regional proportionality, the proportion of nodes that declare boundary status within each partition, as described in Section 4.3.1, versus those nodes that claim not to be on the boundary. We call this the boundary- vs. regular-node ratio.

We plot regional proportionality for each algorithm, for each network, in Figure 7. From this view it appears that the tent-rule method produces the most accurate results, that the 2-hop method produces the least accurate results, and that *lcv* lies in between. This interpretation is misleading. It is inappropriate to compare the curves against each other. Rather, it is more appropriate to compare the curves against the network-wide proportion of boundary declaring nodes. The network-wide proportions may be calculated using the values reported in Tables 1- 3.

As an example, we direct our reader to the horizontal



(a) uniform, 1500 nodes (b) uniform, 2500 nodes (c) uniform, 3500 nodes

(d) normal, 1500 nodes (e) normal, 2500 nodes (f) normal, 3500 nodes

(g) skewed, 1500 nodes (h) skewed, 2500 nodes (i) skewed, 3500 nodes

Figure 6: Edge proximity distributions reveal the proximity to the network edge of nodes that declare boundary status.

plots in Figures 7a, 7d, 7g. (We have otherwise omitted these curves for clarity, and due to space limitations.) The horizontal lines represent the network-wide proportion of nodes that declare boundary status using *lcv*. This average permits a clearer interpretation of the results. For example, in Figure 7a the network-wide proportion of boundary declaring nodes by *lcv* is 0.17. From the same figure we see that the proportion of boundary declaring nodes in the outer-most $0.25R$ region using *lcv* is 0.64. We can conclude that, using *lcv*, nodes in the outer-most ring are almost 4 times as likely to identify with the edge of the network.

Taking this perspective, the reason tent-rule appears to produce more accurate results is that it yields a higher network-wide average. By taking the network-wide averages into account we can observe that sparse uniform, and normal networks in general, are best served by the 2-hop method. In the remaining networks *lcv* produces the highest proportion of boundary nodes closer to the network edge.

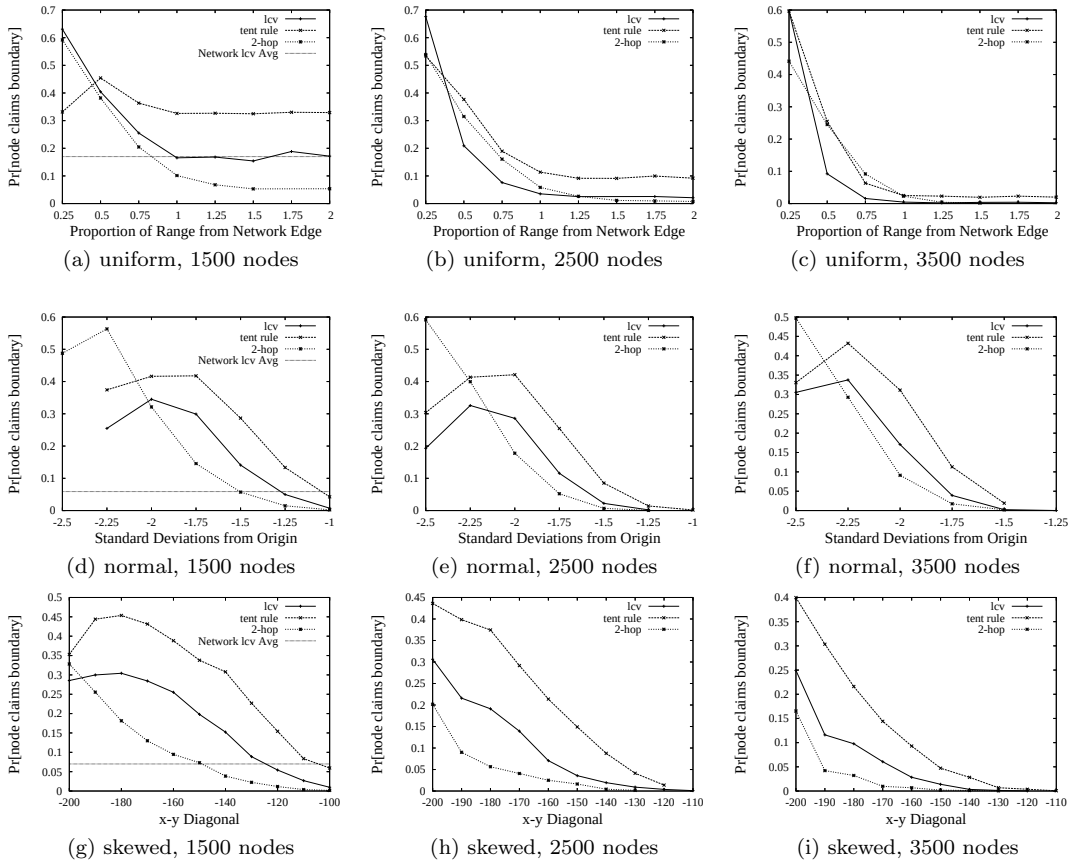
4.2.4 Summary of Comparison

In summary, the *lcv* approach performs consistently across all networks tested. The tent-rule, by design, finds all nodes that abut an unreachable region irrespective of the region's location in the network. By contrast the *lcv* reveals a smaller set of boundary nodes that describe the network boundary more concisely.

Moreover, the observation that the 2-hop approach produces more accurate results than *lcv* is misleading. Figure 8 illustrates that the 2-hop method is adversely affected by network density because the distance to the bootstrap beacon is recorded in hops. Since many neighbouring nodes record the same hop count, the 2-hop approach reveals boundary nodes that are closely clustered together. This leaves many portions of the network edge under-represented. Furthermore, the ability of the 2-hop method to reveal boundary nodes suffers if the bootstrap beacon is poorly placed (see Rao et al. 2003). The *lcv* suffers none of these drawbacks.

4.3 *lcv* and Position Estimation Error

Error is added to the system by blurring the position of nodes from their actual locations. This blurring occurs from the perspective of each node so that two nodes may see a common neighbour in two different positions. Before computing the local convex view, each coordinate is shifted. We shift coordinates by adding a vector consisting of an angle chosen from the uniform distribution, and a length chosen from a parametrized normal distribution. We use the edge proximity and regional proportionaly metrics described in Section 4.1 to evaluate the efficacy of the *lcv* method in the presence of error.

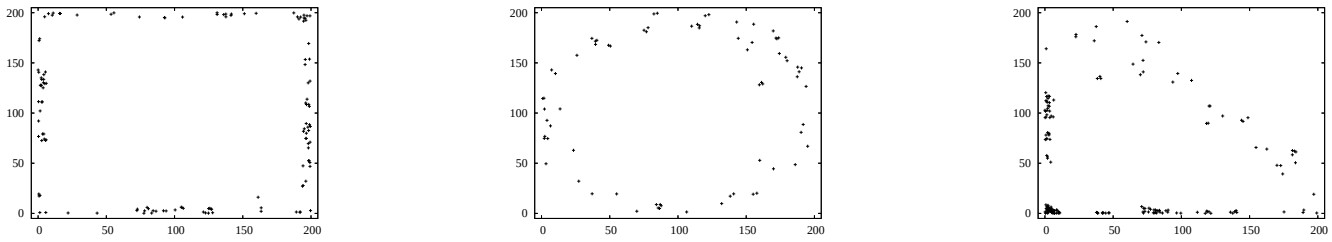


(a) uniform, 1500 nodes (b) uniform, 2500 nodes (c) uniform, 3500 nodes

(d) normal, 1500 nodes (e) normal, 2500 nodes (f) normal, 3500 nodes

(g) skewed, 1500 nodes (h) skewed, 2500 nodes (i) skewed, 3500 nodes

Figure 7: Regional proportionality reveals the proportion of nodes in each region to declare closeness to the network edge.



(a) Uniform (b) Normal (c) Skewed

Figure 8: The 2-hop method is adversely affected by increased density. Boundary declaring nodes cluster together leaving many regions of the network edge under-represented. Example networks are 3000 nodes in size.

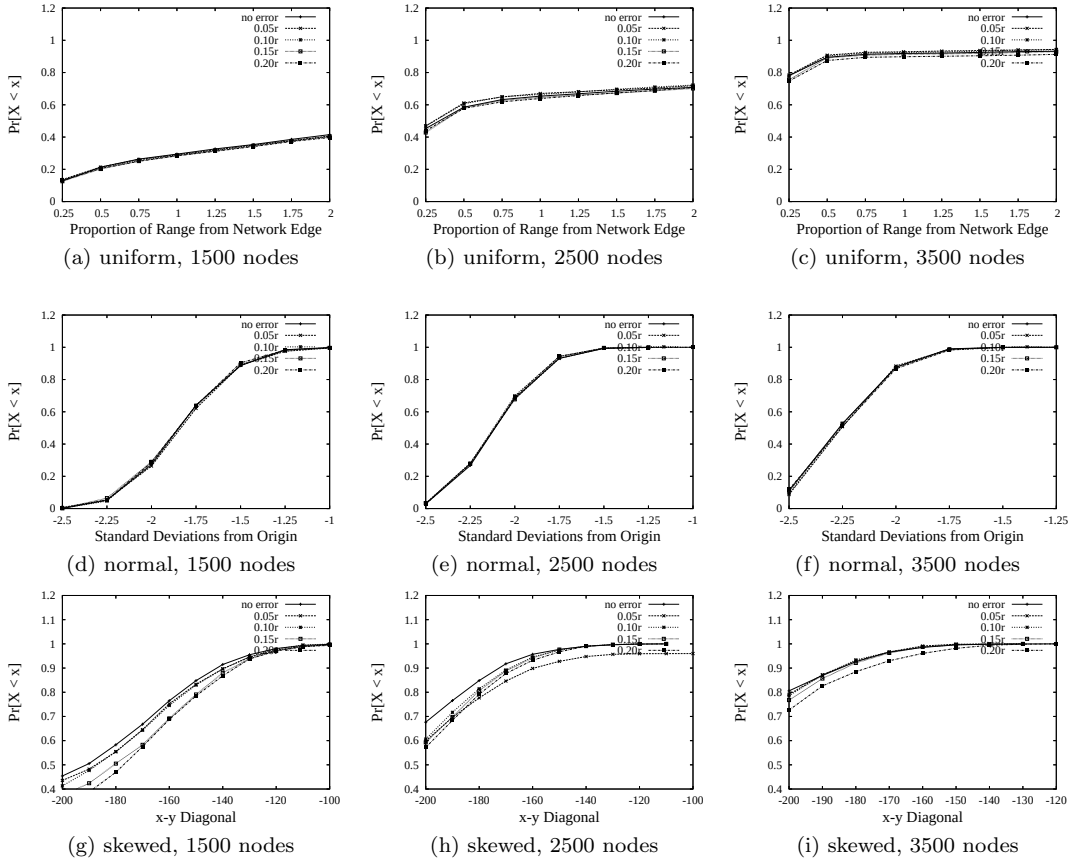
4.3.1 Edge Proximity under Error

Edge proximity is defined as the probability that the location X of an *lcv*-node lies in region x. This gives the likelihood that nodes sitting on their local convex view also sit close to the network boundary.

Edge proximity plots appear in Figure 9. Subfigures are organised such that network size and density changes across each row, while the underlying network distribution varies down each column. For each network we plot the edge proximity for varying error values, parametrized by increasing the variance from 0 to 20% of the communication range. Each plot represents the cumulative distribution over partitions of the network. Networks are partitioned in the manner described in Section 4.2.2.

With respect to the effect of error on the performance of *lcv* we find the observations to be somewhat counter-intuitive. Within each subfigure, each curve represents a different degree of error. Curves within each subfigure show identical trends with differences in accuracy that are largely statistically insignificant. (Confidence intervals have been omitted for clarity.) This would indicate that, in all tested networks, the accuracy of *lcv* is largely unaffected by error. We reserve a discussion of the causes for Section 5.

We proceed in the next section with an evaluation using



(a) uniform, 1500 nodes (b) uniform, 2500 nodes (c) uniform, 3500 nodes

(d) normal, 1500 nodes (e) normal, 2500 nodes (f) normal, 3500 nodes

(g) skewed, 1500 nodes (h) skewed, 2500 nodes (i) skewed, 3500 nodes

Figure 9: Edge proximity distributions reveal the proximity of *lcv*-nodes to the network edge. Error ranges from $0-20\%$ of communication range, r.

a second metric to confirm our observations.

4.4 Regional Proportionality under Error

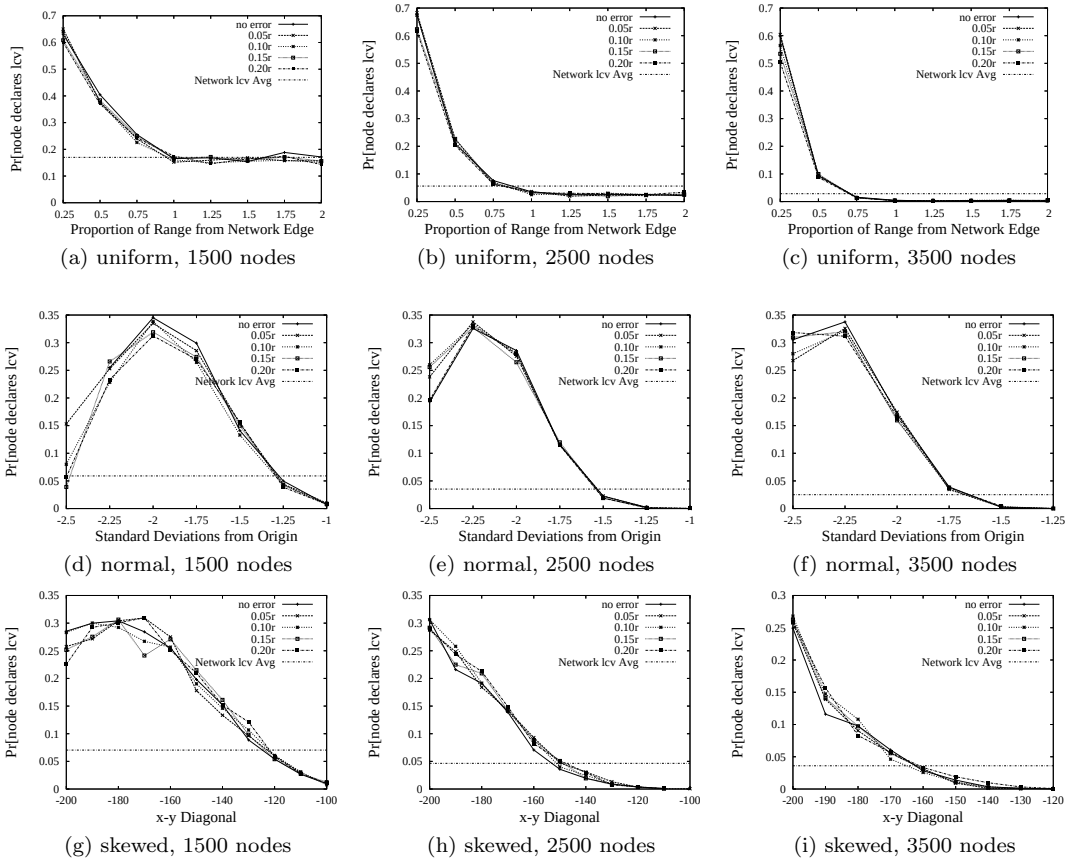
We find that, using the edge proximity metric, *lcv* seems largely unaffected by errors in position estimation. In this section we seek further insight by evaluating regional proportionality. *Regional proportionality* is the proportion of nodes that declare *lcv* status within each partition as described in Section 4.2.2, versus those nodes that claim not to be *lcv*. We call this the *lcv*- vs. regular-node ratio.

We plot regional proportionality in Figure 10. Subfigures are organised such that network size and density changes across each row, while the underlying network distribution varies down each column. For each network we plot the edge proximity for varying error values, parametrized by increasing the variance from 0 to 20% of the communication range. Note the leftward shift of curves as size increases among normal and skewed networks. Recall from previous that this is an artifact of the network's edge shifting further from the origin as the networks grow.

Within each subfigure we can compare the curves against the network-wide proportion of *lcv*-nodes, represented by the horizontal line. The network average permits a clearer interpretation of the results. For example, in Figure 10a the network-wide proportion of *lcv*-nodes is 0.17. From the same figure we see that the proportion of *lcv*-nodes in the outer-most $0.25R$ region is 0.64 when there is zero error. We can conclude that, with no error, nodes in the outer-most ring are almost 4 times as likely to identify with the edge of the network.

The regional proportionality metric seems to reinforce the observation that error, as tested, has little-to-no effect on the performance of *lcv*. Similar to the edge proximity metric in Section 4.2.2, plots within each subfigure show identical trends with differences in accuracy that are largely statistically insignificant. However, there are subtle noteworthy observations. We refer our reader first to plots derived from uniformly generated networks in Figures 10a-10c. We can see that error has a more pronounced effect on *lcv* accuracy as the network density increases, but only in the outer-most region of the network.

We emphasize that the differences in accuracy, statistically speaking, are insignificant - with one exception. A dense uniformly generated and bounded network will eventually capture the shape enforced by the bounds. In our experiments this shape is a quadrilateral. It is directly responsible for the loss in accuracy in the outer-most ring of the network, as density increases. We develop this idea next in Section 5.



(a) uniform, 1500 nodes (b) uniform, 2500 nodes (c) uniform, 3500 nodes

(d) normal, 1500 nodes (e) normal, 2500 nodes (f) normal, 3500 nodes

(g) skewed, 1500 nodes (h) skewed, 2500 nodes (i) skewed, 3500 nodes

Figure 10: Regional proportionality reveals the proportion of nodes in each region to declare edge-node status via *lcv*. Error ranges from $0 - 20\%$ of communication range, r.

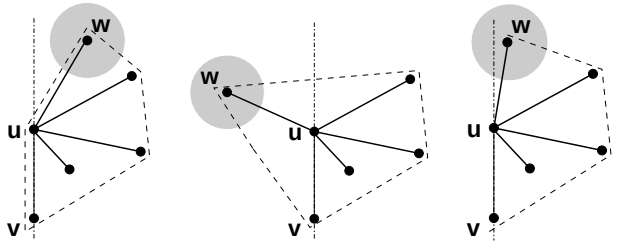
5 On the Resilience of *lcv* to Error

The previous section revealed little-to-no degradation in the performance of *lcv* in the presence of errors. This idea is counter-intuitive, and so we use this section to enumerate and discuss the scenarios faced by the local convex view method.

In our analysis we assume that the position of only a single node has been incorrectly estimated. This relatively benign assumption permits a clear demonstration of the effects of error on *lcv* without sacrificing accuracy or completness. Specifically, all remaining cases may be composed of the cases presented here.

The three cases under consideration by *lcv* are presented in Figure 11. For the purpose of demonstration, we consider the *lcv* operation at node u. In our example topologies, neighbours are joined to u with a solid line. The position of some neighbouring node v determines a dash-dot-dashed line that represents a threshold of interest. The neighbour in question has a position estimated by the node labeled w, with a true position that may exist anywhere inside the greyed region. Finally, the dashed poly-line corresponds to the local convex hull under consideration.

Figures 11a and 11b depict the two 'good' cases where the *lcv* computation is unaffected by error. In the first case, shown in Figure 11a, node u determines it is on the local convex view and declares itself close to the network boundary. Note that the local convex hull consists of the same nodes irrespective of the actual location of w anywhere inside the grey region. The second case shown in Figure 11b is of a similar theme. Node w is estimated to have a location that renders node u inside its local convex view. In fact, node w may sit anywhere in the grey region without affecting the local convex view. In both these cases the underlying geometry ensures the resilience of the local convex view method: the convex hull remains consistent so long as the error region of w remains entirely to one side or the other of the threshold determined by



(a) Consistent (b) Consistent (c) Inconsistent

Figure 11: In consistent cases *lcv* is unaffected by position estimation error.

(u, v).

Figure 11c depicts the ambiguous case. Node w is estimated to have a position close enough to the threshold that its actual location may exist on either side of the threshold. In the example shown in Figure 11c, a w is estimated to have a position that renders u on the local convex view of u. If such a w actually sits on the the other side of the threshold then u has falsely determined that it sits on its local convex view. The reverse may occur if w is estimated to sit close to, but on the other side of, the threshold.

The observation in Section 4.3 is that *lcv* seems relatively unaffected by error. From our analysis we have determined that any adverse effect of error to *lcv* is caused by the ambiguous scenario demonstrated in Figure 11c. Our hypothesis is that *lcv* is relatively unaffected by error because the ambiguous case occurs very infrequently. To test our hypothesis we evaluate the frequency of false positives and false negatives when error is added to *lcv*. For each type of network the results are partitioned as described in Section 4.3.1 so that we may observe *lcv* performance in each area of the network. We plot for all networks the worst tested case in Figure 12, where the variance parameter is equal to 20% of the communication range.

Figure 12a reveals that, in uniform networks, the error in the outer-most ring of the network hovers about 20%. Interestingly, the frequency of false positives and negatives in this region climbs as density increases. The reason is that increased densities along the network edge more closely approximate the artifical lines artificially bounding the network. This causes a greater number of w nodes which are the cause of the ambiguity that leads to false positives and negatives. As we move deeper into the network where no artificial boundaries exist, the frequency of incorrect responses drops dramatically.

In the outer-most regions of normal and skewed networks, the ambiguous case appears much less frequently. For example, among the outer-most 5% of nodes in normal networks, the rate of false *lcv* responses is approximately 10%. Similarly for skewed networks. This may be explained by the lack of artificial boundaries in normal and skewed networks (ie. unlike the uniformly generated networks, these networks fail to approximate the quadrilateral region that contains them). As we move deeper into the network the rate of false positives quickly drops in both normal and skewed networks.

6 Conclusions and Future Work

This paper has presented and compared methods to identify a subset of nodes on the network boundary (for sensing and localizing applications). Our key contribution is a heuristic algorithm that operates locally. A node decides it is close to the network edge if the node finds that it lies on the convex hull of its 1-hop neighbourhood. Where position information is unavailable *lcv* assigns local coordinates to each neighbour so that geometric relationships, and hence the convex hull, may be computed. Coordinate assignment requires that a node view its neighbourhood as 2-connected, ie. if a node removes itself then a single connected component remains. We have determined the maximum possible number of disconnected components to be 5. In such cases we have shown that it is possible for a node to sit on the local convex hull only where removal of the node leaves a maximum of 3 components. A simple probabilistic model was proposed to decide the convex hull of neighbourhoods with disconnected components.

We simulated *lcv*, tent-rule, and 2-hop methods in networks of varying density constructed from uniform, normal, and skewed (Pareto) distributions. For evaluation we identified two metrics, edge proximity, and regional proportionality. We found that the behaviour of all three methods demonstrated similar trends. According to these metrics the tent-rule was the least well-performing, while the 2-hop method performed best. However, this conclusion is misleading as the 2-hop method also reveals boundary nodes that are tightly clustered, while ignoring other boundary nodes altogether. *lcv* was found to be resilient to the qualitative drawbacks of the 2-hop approach.

In this paper we examined the ability of the local convex view (*lcv*) algorithm to identify network edge nodes in the presence of position estimation error. We engaged in extensive simulations of networks with topologies of varying size and underlying distributions. Position errors were chosen from a normal distribution with a variance up to 20% of the communication range.

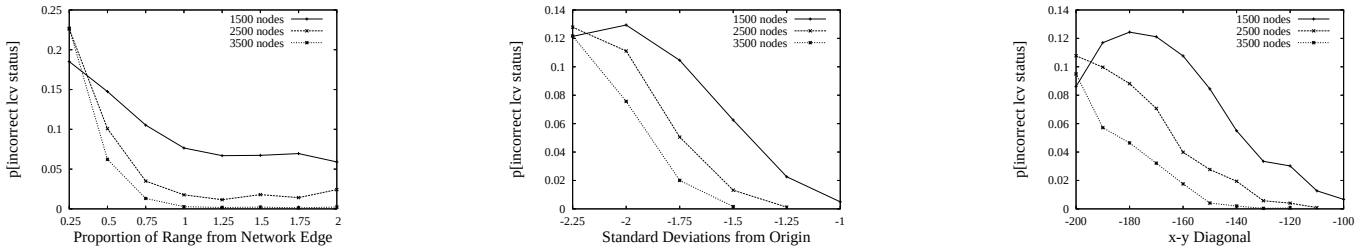
Further examination failed to reinforce the assumption that *lcv* would be adversely affected by position estimation error. To explain the disconnect between intuition and observation we enumerated and analysed the three base neighbour configurations that may be seen by a node. In two cases position estimation error changes the shape of the local convex view, but not the nodes that comprise it. In the third case position error leads to ambiguity, where the true position of a neighbour may lead to a false insertion or an omission of the node undergoing the *lcv* computation. Further simulation revealed the frequency of the ambiguous case to be very low, about 10% in the worst case for all networks tested. We conclude that the geometric properties underlying *lcv* are responsible for its resilience to error.

At present we are transforming *lcv* into a deterministic algorithm that produce a descriptive map of the network boundaries. Preliminary results are positive, and will be part of a forthcoming publication.

REFERENCES

Ben-Chen, M., Gotsman, C. & Gortler, S. (2006), Routing with guaranteed delivery on virtual coordinates, *in* 'Proceedings of the 18th Canadian Conference on Computational Geometry (CCCG'06)', pp. 117–120.

Cao, Q. & Abdelzaher, T. (2004), 'A scalable logical coordinates framework for routing in wireless sensor net-



(a) Uniform (b) Normal (c) Skewed

Figure 12: *lcv* false positives and negatives in the worst tested case, with error variance 20% of communication range.

works', *Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS)* pp. 349–358.

Fang, Q., Gao, J. & Guibas, L. (2004), Locating and Bypassing Routing Holes in Sensor Networks, *in* 'Proceedings of IEEE/ACM Infocom', Hong Kong, China.

Fang, Q., Gao, J., Guibas, L. J., de Silva, V. & Zhang, L. (2005), Glider: gradient landmark-based distributed routing for sensor networks., *in* 'Proceedings of the 24th Annual IEEE Conference on Computer Communications (INFOCOM)', Miami, FL, USA, pp. 339–350.

Fekete, S. P., Kröller, A., Pfisterer, D., Fischer, S. & Buschmann, C. (2004), Neighborhood-based topology recognition in sensor networks, *in* 'Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)', Vol. 3121 of *Lecture Notes in Computer Science*, Springer, pp. 123–136.

Fonseca, R., Ratnasamy, S., Zhao, J., Ee, C. T., Culler, D., Shenker, S. & Stoica, I. (2005), Beacon vector routing: Scalable point-to-point routing in wireless sensornets, *in* 'Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05)', Boston, MA, USA.

Funke, S. (2005), Topological hole detection in wireless sensor networks and its applications, *in* 'Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC '05)', New York, NY, USA, pp. 44–53.

Funke, S. & Klein, C. (2006), Hole detection or: "how much geometry hides in connectivity?", *in* 'Proceedings of the 22^{nd} annual symposium on Computational Geometry (SGC)', pp. 377–385.

Kröller, A., Fekete, S. P., Pfisterer, D. & Fischer, S. (2006), Deterministic boundary recognition and topology extraction for large sensor networks, *in* 'Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (SODA)', New York, NY, USA, pp. 1000–1009.

Langendoen, K. & Reijers, N. (2003), 'Distributed localization in wireless sensor networks: a quantitative comparison', *Computer Networks* **43**(4), 499–518.

Niculescu, D. & Nath, B. (2004), Error characteristics of ad hoc positioning systems (aps), *in* 'Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)', New York, NY, USA.

O'Rourke, J. (1998), *Computational Geometry in C*, Cambridge University Press, New York, NY, USA.

Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S. & Stoica, I. (2003), Geographic routing without location information, *in* 'Proceedings of ACM MobiCom', San Diego, CA.

Savvides, A. & Garber, W. L. (2005), 'An analysis of error inducing parameters in multihop sensor node localization', *IEEE Transactions on Mobile Computing* **4**(6), 567–577. Senior Member-Randolph L. Moses and Senior Member-Mani B. Srivastava.

Shang, Y., Ruml, W., Zhang, Y. & Fromherz, M. P. J. (2003), Localization from mere connectivity, *in* 'Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)', New York, NY, USA, pp. 201–212.

S.P. Fekete and M. Kaufmann and A. Krller and N. Lehmann (2005), A new approach for boundary recognition in geometric sensor networks, *in* 'Proceedings 17th Canadian Conference on Computational Geometry (CCCG)', pp. 82–85.

Wang, Y., Gao, J. & Mitchell, J. S. (2006), Boundary recognition in sensor networks by topological methods, *in* 'Proceedings of the 12th annual international conference on Mobile computing and Networking (MOBICOM)', pp. 122–133.

Whitehouse, K., Karlof, C., Woo, A., Jiang, F. & Culler, D. (2005), The effects of ranging noise on multihop localization: an empirical study, *in* 'Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN)'.

Zhao, Y., Li, B., Zhang, Q., Chen, Y. & Zhu, W. (2005), Efficient hop id based routing for sparse ad hoc networks, *in* 'Proceedings of the 13TH IEEE International Conference on Network Protocols (ICNP)'.