Chapter 4

Distributed Location Identification

4.1 Introduction

Most location estimation systems in mobile networks utilize the fundamental method of using trilateration and triangulation of received signals to obtain an estimate of the receiver’s position. As mentioned earlier, the Global Positioning System (GPS) does not scale well in dense urban areas or in indoor locations. Modeling of the radio propagation environment helps in providing a more accurate location estimate by mitigating the effect of NLOS errors. As mentioned in chapter 2, while reasonably accurate radio propagation models exist for outdoor conditions, unfortunately there are no such unanimously accepted models for indoor environments. Hence, in the absence of a suitable model for predicting the location of a mobile terminal, it is possible that the node may be far away from the estimated point.

In this chapter, rather than doing a location estimation, we consider the location discovery problem in terms of finding the region where a node is guaranteed to be found. The objective of the location identification process then becomes minimizing the size of such a region. We assume that a small percentage of the terminals (nodes) in the network know their locations with a high degree of accuracy - possibly through GPS access or by some other means. We
term such nodes as reference nodes. We propose a distributed algorithm using computational geometric techniques to compute the smallest region of residence where a node is guaranteed to be found, for all non-reference nodes in the network. A unique feature of our algorithm is that the location regions of the nodes in the network are improved through the exchange of location information between the neighbors in $O(nD)$ time, where $n$ and $D$ are the number of nodes and diameter of the network respectively. Simulation results also demonstrate that our algorithm succeeds in finding reasonably small stable regions of residence for all non-reference nodes.

Section 4.2 is devoted to the description of the system model. Section 4.3 outlines the preliminaries ideas and the proposed location identification algorithm is presented in section 4.4. Simulation results are then presented in section 4.5 followed by conclusion in section 4.6.

### 4.2 System Model

We model an ad hoc network scenario as a graph $G = (V, E)$, consisting of $n$ nodes. $V$ is the set of all nodes, $|V| = n$ and $E$ is the set of edges in the graph $G$. The nodes may be either stationary or mobile. All communication links are assumed to be bi-directional. We say node $v$ is a neighbor of $u$, if they are within each other’s hearing zone. The neighborhood, $N(i)$, of a node $i$ consists of all nodes that are within its transmission range. A small percentage of the nodes are assumed to know their individual locations with high precision, either through GPS or some other means. These nodes serve as the reference nodes (RN) in the network. Initially, nodes other than the RNs do not possess any knowledge of their location. The reference nodes are assumed to possess point locations (zero area regions) while the non-reference nodes are initially assumed to reside in a region of infinite size. $Ref_n = \{ u : u \in V, u \text{ is a reference node} \}$ denotes the set of reference nodes.

Considering thermal noise at the receiver, NLOS errors and channel characteristics, the mea-
sured range between two nodes $u$ and $v$ is given by equation 3.1.

We define the set $RR$ to be the set of all such measured ranges for all node-pairs in the network, i.e., $RR = \{r_{ij} : r_{ij} \in E, \forall i, j \in V \}$. Also, $RR_i = \{r_{ij}, j \in N(i) \}$.

### 4.3 Preliminaries

Our proposed algorithm is based on the triangulation technique to compute the region where a node is guaranteed to be found.

**Definition 4.1** The *region of residence*, $R_i$, of a node $i$ is defined to be the region where $i$ is guaranteed to be found.

The region of residence of a reference node is assumed to be a point location of zero area. All other nodes have a non-zero finite area region of residence. Our objective is to find the minimum region of residence of a node $i$.

**Lemma 4.1** *The range measurements obtained for a node $u$, from a neighbor $v \in N(u)$ will always be greater than or equal to the Euclidian distance between $u$ and $v.*

**Proof**: Follows directly from equation 3.1. \(\square\)

Given two nodes $u$ and $v$ and a range measurement $r_{uv}$ from $v$ to $u$, the region of residence of $u$ in the view of node $v$ is the region formed by extending $v$’s region residence $R_v$ in every direction by the measured range value, $r_{uv}$. We denote this operation by the operator $\oplus$, whose left operand is a region of residence and the right operand is a range value. Thus the region of residence of $u$ in the view of node $v$ is $R_{uv} = R_v \oplus r_{uv}$. We call $R_{uv}$ as the *viewed region of residence* of node $u$. 

35
Example 4.1 Suppose node $v$ has a triangular region of residence $\triangle ABC$, and the range of $u$ measured by $v$ is $r_{uv}$ as shown in Figure 4.1. We draw a line $A'B'$ parallel to $AB$ at a distance $r_{uv}$ from $AB$ and on the opposite side of the node $C$ such that $AA'B'B$ forms a rectangle. Similarly, we draw a line $B''C'$ parallel to $BC$ on the side opposite to that of $A$ and distant $r_{uv}$ from $BC$ so as to form a rectangle $BB''C'C$, and also a line $C''A'$ parallel to $CA$ on the side opposite to that of $B$ and distant $r_{uv}$ from $CA$ so that $CC''A''A$ is a rectangle. Now from the point $A$, draw a circular arc of radius $r_{uv}$ so as to cut the lines $A'B'$ and $A''A'$ at $A'$ and $A''$ respectively. Similarly, draw two other circular arcs of radius $r_{uv}$: i) from $B$ to cut the lines $A'B'$ and $B''C'$ at $B'$ and $B''$ respectively, and ii) from $C$ to cut the lines $C''B''$ and $C'A'$ at $C'$ and $C''$ respectively. The closed convex region $A'B'B''C''A''$ is the resulting $R_{uv}$.

It may be mentioned here that the region $R_{uv}$ can also be viewed as the Minkowski’s sum [16] of the region of residence $R$ of the node $v$ and a circle of radius $r_{uv}$ centered at origin. We assume that the initial regions of residence of all nodes are bounded either by straight line segments or by circular arcs. Hence, the region $R_{uv}$ will also be bounded by straight line segments and/or circular arcs only. We now have the following result.
Lemma 4.2  
Node $u$ is guaranteed to be found at some location inside $R_{uv}$.

Proof: Follows from the construction of $R_{uv}$.

Theorem 4.1  The current minimum region of residence, $R_u$, of a node $u$, based on the information from its neighbors is the region formed by the intersection of the viewed regions of residence $R_{ui}$’s, $i \in N(u)$, i.e., $R_u = \bigcap_{i \in N(u)} R_{ui}$.

Proof: The proof follows from lemma 4.2 as the common intersection region is the smallest region that satisfies lemma 4.2 for all neighbors $i \in N(u)$.

Note that this current minimum region of residence may subsequently get refined (contracted in size) by improved viewed regions of residences from its neighbors.

Theorem 4.2  The minimum region of residence of a node $u$, $R_u$ (based on the information from its neighbors) can not subsequently be made larger by an altered viewed region of residence, $R_{ui}$ from any neighbor $i$.

Proof: The proof follows directly from theorem 4.1.

To find the minimum region of residence, our algorithm proceeds in two steps:

- In the first step, every node $u$ in the network determines its current region of residence by ranging with each of its neighbors.

- Once $u$ has determined its current minimum region of residence, it attempts to improve the regions of residence of each neighbor, using its own region of residence and the range measurements that it obtained from the respective neighbors.

Example 4.2 below illustrates the working of our algorithm:
Example 4.2 Consider a node $u$ with three neighbors $i$, $j$ and $k$. Figure 4.2 demonstrates a probable situation where the triangles $\triangle ABC$ and $\triangle DEF$ define the region of residence of nodes $i$ and $j$ respectively. $PQRS$ is the region of residence of node $k$. For simplicity, we assume the regions of residence as polygonal. Let $r_{1u}$, $r_{2u}$ and $r_{3u}$ be the range measurements that $u$ obtains by ranging with $i$, $j$ and $k$ respectively. According to the view of node $i$, $u$ lies in the region $R_{iu}$, dictated by the shape $A'B'B''C'C''A''$ as demonstrated in example 4.1. Similarly, $D'E'E''F'F''D''$ and $P'Q'Q''R'R''S'S''P''$ define the region of residence of $u$ in the views of the nodes $j$ and $k$ respectively. Following theorem 4.1, the shaded region $LMN$ defines the minimum region of residence of node $u$, where $u$ is guaranteed to be found.

Once the minimum region of residence of node $u$ is found, node $u$ then tries to refine the minimum region of residence of a neighbor $v$, \( \forall v \in N(u) \), using $R_u$ and the corresponding measured range $r_{uv}$ from $v$. The new minimum region of residence of node $v$, $R'_v$ is defined as the intersection of the viewed region of residence of $v$ by $u$, $R_{vu}$ and the current minimum region of residence of $v$, $R_v$. In Figure 4.3, region $LMN$ defines the minimum region of...
residence of node $u$. $r_{3u}$ is the measured range from $k$ to $u$. The region defined by the dotted lines, $L'M'N'$ defines the viewed region of residence, $R_{ku}$ of $k$ by $u$. The region $UN'VQ$ is the intersection region of the current minimum region of residence of $k$, $PQRS$ and $L'M'N'$. Following theorem 4.1, the new minimum region of residence of node $k$ is the region $UN'VQ$. Node $u$ tries to similarly improve the regions of nodes $i$ and $j$ using the measured ranges $r_{1u}$ and $r_{2u}$ and its minimum region of residence, $\mathcal{R}_u$.

A careful scrutiny of figure 4.3 in the above example reveals that the part of the boundary of the region of residence of $k$ which causes a computation (improvement) in the region of residence for $u$, and the part of the boundary of the region of residence for $k$ which is refined (improved) due to this computed part of the region of residence for $u$, are mutually disjoint. This observation holds even if the node $k$ would have an initial region of residence of a different shape.

**Lemma 4.3** Given a minimum region of residence of a node $u$, $\mathcal{R}_u$ and a measured range...
from a neighbor \( v \), the improved minimum region of residence of \( v \), \( \mathcal{R}'_v \), is given by the intersection of the viewed region of residence of \( v \) by \( u \), \( R_{vu} \) and the current minimum region of residence of \( v \), \( \mathcal{R}_v \).

\[
\mathcal{R}'_v = \mathcal{R}_v \cap R_{vu}
\]  

(4.1)

**Proof**: The proof follows directly from lemma 4.2 and theorem 4.1.

From lemma 4.3, we get, \( R'_v \subseteq R_v \). Note that \( R'_v \) is generated by introducing some extra arc and/or straight line segments on \( R_v \) due to the region computation initiated by the node \( u \). Let us denote these set of new arcs and straight line segments by \( E'_v \). Elements of \( E'_v \) are either parallel to some boundary edge of \( R_u \) or a circular arc of a circle with radius \( r_{vu} \), centered at some vertex on the boundary of \( R_u \).

Suppose there is a path in the network starting from a node \( u_0 \) to some node \( u_k \), given by \( u_0 \ u_1 \ u_2 \ \cdots \ u_k \). If \( u_0 \) initiates its region computation by its neighbors and gets its region computed by these neighbors as \( R_{u_0} \), then \( R_{u_0} \) may cause an improvement in the region of node \( u_1 \). This, in turn, may cause an improvement in the region of the node \( u_2 \), and so on, so that the process of region refinements may successively follow through the nodes \( u_1, u_2, \ldots, u_k \). In particular, if we now assume that \( u_k = u_0 \), i.e., \( u_0 \ u_1 \ u_2 \ \cdots \ u_k \) is a cycle, then we claim that this process of successive region refinements will not be able to further refine the region \( R_{u_0} \) of \( u_0 \) after a finite number of steps. To establish this claim, we proceed as follows.

Let \( E_{u_0}^{u_1} \) denote the set of newly introduced lines and/or arcs on the boundary of the minimum region of residence of \( u_1 \) due to the region computation of \( u_0 \) caused by all the immediate neighbors of \( u_0 \). The changed region \( R'_{u_1} \) of \( u_1 \) due to \( E_{u_0}^{u_1} \) may cause a change in \( R_{u_2} \) by introducing some new lines and/or arcs which we denote by the set \( E_{u_0}^{u_2} \). In general, we denote the set of newly added lines and/or arcs in the region of \( u_j, 1 \leq j \leq k \), by \( E_{u_0}^{u_j} \). Because of the properties of Minkowski’s sum of \( R'_{u_{j-1}} \) and a circle of radius \( r_{u_{j-1}, u_j} \) (range
value between nodes \( u_{j-1} \) and \( u_j \) with center at the origin, we see that for any \( j, 1 \leq j \leq k \), two possible cases may arise:

Case 1: A line segment (arc) in \( E_{u_0, u_1 \ldots u_{j-1}} \) is parallel to some line segment (arc) in \( E_{u_0, u_1 \ldots u_{j-2}} \) (for \( j > 1 \)) or in \( R_{u_0} \) (for \( j = 1 \)).

Case 2: An arc in \( E_{u_0, u_1 \ldots u_{j-1}} \) is

i) not parallel to any arc in \( E_{u_0, u_1 \ldots u_{j-2}} \) (for \( j > 1 \)) or in \( R_{u_0} \) (for \( j = 1 \)),

ii) but is an arc of a circle with radius \( r_{u_{j-1}, u_j} \), having center at one point on the region \( R'_{u_{j-1}} \) which is the point of intersection of two different arcs or two different line segments or an arc and a line segment, at least one of which must be in \( E_{u_0, u_1 \ldots u_{j-2}} \) (for \( j > 1 \)) or in \( R_{u_0} \) (for \( j = 1 \)).

This fact is illustrated in Figure 4.4 where the arcs \( \alpha \) and \( \beta \) on \( R'_{u_{j-1}}, R'_{u_j} \), respectively are parallel to each other, while the arc \( \gamma \) on \( R'_{u_j} \) is derived from the point \( Q \) on \( R'_{u_{j-1}} \) (with \( Q \) as center and a radius equal to \( r_{u_{j-1}, u_j} \)). We also see from Figure 4.4 that for every point on \( R'_{u_j} \), there exists a unique point on \( R'_{u_{j-1}} \) from which this point was derived. Thus, for the point \( T \) on \( R'_{u_j} \), the corresponding point on \( R'_{u_{j-1}} \) is \( Q \), which is transitively derived from a point \( P \) on \( R_{u_0} \).

**Lemma 4.4** Let \( T \) be any point on \( E_{u_0, u_1 \ldots u_{j-1}} \), and \( P \) be the corresponding point on \( R_{u_0} \) from which \( T \) was derived. The Euclidean distance \( PT \) is always greater than or equal to the maximum of all \( (r_{u_{j-1}, u_j}, \forall j, 1 \leq j \leq k) \).

**Proof**: We prove this by induction on \( j \). Our claim is trivially true for \( j = 1 \). Suppose the claim is true for \( j = 1, 2, \ldots, j-1 \). For \( j \geq 1 \), referring to Figure 4.4, the Euclidean distance \( PQ \) is then greater than or equal to the maximum of \( (r_{u_0, u_1}, r_{u_1, u_2}, \ldots, r_{u_{j-2}, u_{j-1}}) \). Now, if the point \( T \) is on an arc or line segment in \( E_{u_0, u_1 \ldots u_{j-1}} \) parallel to an arc or line segment in
Figure 4.4: Refinements of regions of successive neighbor nodes

If a node \( u \) initiates its region computation with the help of range readings
from all of its neighbors, the computed region \( R_u \) of \( u \) may cause refinements of the successive neighbors through the whole network, but it will never be able to further refine \( R_u \) of \( u \) itself.

**Definition 4.2** The stable region of residence of a node \( u \) is the minimum region of residence of \( u \) which can not be further improved upon using the current global set of range readings for all node pairs in the network. We denote such a region by \( S_u \).

**Theorem 4.4** A node \( u \) can compute its stable region of residence once it gets the range readings of all possible directly communicating nodes in the network along with the initial region information of all nodes.

**Proof :** The minimum region of residence of a node \( u \) may get refined only when at least one of its neighbors, say \( v \) is able to improve its region of residence. This, in turn, depends on whether any neighbor of \( v \) excluding \( u \) can improve its region of residence, and so on. The process can go on until all the nodes in the network are tried with for any improvement in their respective regions of residences. Hence effectively, if \( u \) has all the necessary information from all of these updates, \( u \) can eventually compute its stable region of residence.

**Theorem 4.5** The computation of the stable regions of residence of all nodes in the network is functionally equivalent to an all-to-all broadcast of the range information of all node-pairs in the network (the set \( RR \)) and the set \( Ref_n \).

**Proof :** To reconstruct the ad hoc network graph centrally, two pieces of information would be required: (i) the measured ranges of all node pairs and, (ii) the information as to whether an individual node is a reference node or not. From theorem 4.4, we see that if a node possessed the range values of all node-pairs in the network (the set \( RR \)) and the set \( Ref_n \), it could locally construct the network graph and then compute the stable regions of residence of all nodes. Since the possession of the set \( RR \) and the set \( Ref_n \) by a node in the network
effectively implies a broadcast of these two sets, if every node were to locally compute the stable regions of residence, the problem maps to that of an all-to-all broadcast of the set \( RR_i \) and status (whether its a reference node) of each node \( i \) in the system. Each node on receiving this information from a neighbor would attach its own \( RR_i \) set and its status and broadcast the message again.

\[ \square \]

### 4.4 Proposed Location Identification Algorithm

Every node in the network maintains a local variable \( status \), which is set to 1 if the node is a reference node, zero otherwise. Initially, the minimum regions of residence of all non-reference nodes are assumed to be infinity. Each node \( i \) does a ranging with its neighbors to obtain a set of measured ranges, \( RR_i \). \( i \) then computes the viewed region of residence for every node \( j \in N(i) \). Node \( i \) then exchanges three pieces of information with each neighbor \( j \in N(i) - i \) i) value of the status variable, \( status_i \), ii) viewed region of residence of \( j, R_{ji} \), iii) area of the current minimum region of residence of \( i, A_i \).

Let \( RR_i = \{ r_{ij} : j \in N(i) \} \), \( T \) = set of viewed regions of residence, \( R_{ij} \). Once node \( i \) has its viewed region of residence, \( R_{ij} \) from its every neighbor \( j \), it computes its current minimum region of residence using the following algorithm:

**Function compute_region : Boolean**

\[ \text{var } A_{old}, A_i : \text{Real}; \]

\[ \text{begin} \]

\[ \text{for each } R_{ij} \in T \text{ such that } A_j \neq \infty \text{ do} \]

\[ \text{/* Compute the minimum region of residence of node } i \text{ from the viewed regions */} \]

\[ R_i \leftarrow R_i \cap R_{ij}; \]

\[ \text{endfor;} \]

\[ A_i \leftarrow \text{Area of } R_i; \]

\[ \text{if } A_i < A_{old} \text{ then return true; } /* R_i improved */ \]

\[ \text{end} \]
else return false; /* No improvement in $R_i$ */
end.

Once node $i$ computes its minimum region of residence, it tries to improve the minimum region of residence of each of its neighbors as follows:

**Procedure improve_region**

begin
  for each $j \in N(i)$ such that $status_j \neq 1$ do
    /* Construct $R_{ji}$, the viewed region of $j$ */
    $R_{ji} = R_i \oplus r_{ij};$
    Transmit $R_{ji}$ to node $j$;
  endfor;
end.

The following location identification algorithm is executed by each node $i$ until the node attains its stable region of residence, $S_i$.

**Algorithm location_region_identify**

var region_change_flag : Boolean;
begin
  while (true)
    Get neighbor set $N(i)$;
    Generate $RR_i$ : measure range with every neighbor $j \in N(i)$;
    region_change_flag = false;
    repeat
      Get $T$ : viewed regions of residence $R_{ij}$ from every neighbor $j$;
      if $status_i = 0$ then region_change_flag = compute_region($i$, $T$);
improve_region(i, N(i), RR_i);

until region_change_flag = false; /* iterate until \( R_i = S_i \) */

endwhile;

end.

4.4.1 Complexity Analysis

The operations `compute_region` and `improve_region` both need \( O(\Delta) \) time in the worst case, where \( \Delta \) is the maximum node degree in the network. Also, the operation `Get T` needs \( O(\Delta) \) time. If we assign a distinct time slot to each node depending on its unique id number [15] to avoid collision, then each iteration of the `repeat` loop would need \( O(n) \) time slots. From theorem 4.5, the algorithm terminates when an all-to-all broadcast of the sets \( RR \) and \( Ref_n \) is achieved. Assuming that the effect of the viewed range information of all node pairs can be transmitted in \( O(1) \) time slots, the whole communication process will be completed in \( O(D) \) rounds, \( D \) being the diameter of the network (each round consists of \( O(n) \) time slots for message communication from all nodes in a layer to the nodes in the next layer [15]). Hence, the algorithm `location_region_identify` will need \( O(nD) \) time.

4.5 Simulation Results

The goal of the simulation was to evaluate the performance of the proposed algorithm `location_region_identify` in determining the stable regions of residence of all nodes in an ad hoc network. We experimented with various graph topologies by randomly generating static ad hoc graphs. A small percentage of the nodes in the graph were randomly chosen to be reference nodes. The reference nodes were assumed to have point locations of zero area. All other nodes were assumed to have an initial region of residence of infinite area. All communica-
tions were assumed to be symmetrical in nature. We assumed a transmission range of 30 units of distance for all nodes in the system. All range estimates are assumed to be erroneous due to NLOS propagation errors. All errors are assumed to be positive, following equation 3.1. We considered an environment with a mix of LOS and Obstructed LOS (OLOS) signals. OLOS signals represent the scenario where no LOS communication is present between two nodes and hence all received communication between them is through NLOS propagation. The LOS range errors are drawn from a gaussian distribution of mean $\mu$ and standard deviation $\sigma$, while the OLOS range errors are drawn from an exponential distribution with parameter $\lambda$ [8]. The parameters $\mu$, $\sigma$ and $\lambda$ characterize the channel characteristics and hence the amount of ranging errors. We evaluated the performance of our algorithm against varying channel characteristics, percentage of LOS/OLOS signals and percentage of reference nodes in the system. For each set of parameters, the experiments were repeated for 100 randomly generated graphs and the area of the stable region of residence of all nodes in the system were measured. Figures 4.5 to 4.8 demonstrate the results of our simulation for random graphs of 100 nodes. For example, with $\lambda = 5.3$, $\mu = 0.0$, $\sigma = 0.3$, 80% LOS signals and 25% of the nodes designated as RNs, we get a median area of 1.98 sq. units and a mean area of 7.008 sq. units. Figure 4.5(a) and figure 4.5(b) illustrate the variation of the median and mean area of stable regions of residence (in sq. units) respectively, against $\lambda$. We chose to measure the median area as it is less affected by outlier cases. The mean area, on the other hand, can get significantly affected by situations such as a non-reference node, $i$ having only one neighbor (say $j$), in which case $R_i = R_{ij}$ and the area of $R_{ij}$ can be either very small or quite large, depending on the ranged distance between the two nodes. Figures 4.6(a) and 4.6(b) show the variation of median and mean area of stable regions of residence against $\sigma$. Figures 4.7(a) and 4.7(b) demonstrate that the median and mean area of the stable regions of residence decrease exponentially with increase in percentage of LOS signals. Similar behavior is observed by varying the number of reference nodes, as seen from Figures 4.8(a) and 4.8(b).

We approximated the regions of residence of every node in the system at every step of the
algorithm with circumscribing convex polygons for the ease of implementation. Thus, the true areas of the stable regions of residence would be smaller than those presented in this chapter.

From the graphs we deduce that for any given scenario, a small increase in the number of reference nodes would significantly improve the computed regions of residences. Also, using different technologies such as Ultra-Wideband (UWB), which would generally increase the percentage of observed LOS signals can also lead to significant improvement in the regions computed by our algorithm as seen from Figures 4.7(a) and 4.7(b).

Table 4.1 shows how the median area of the stable regions of residence is affected by the node density, measured in number of nodes per square unit. The median areas depicted are for $\lambda = 5.3$, $\mu = 0.0$, $\sigma = 0.03$, 80% LOS signals and 25% of the nodes designated as RNs.
Figure 4.6: Variation of size of stable region of residence with $\sigma$

Figure 4.7: Variation of size of stable region of residence with % of LOS signals
Figure 4.8: Variation of size of stable region of residence with % of RNs

Table 4.1: Variation of median area of stable regions of residence with node density

<table>
<thead>
<tr>
<th>Node Density (nodes/sq. units)</th>
<th>% of RNs</th>
<th>Median area of stable region (sq. units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>25</td>
<td>1.980</td>
</tr>
<tr>
<td>0.0075</td>
<td>25</td>
<td>2.656</td>
</tr>
<tr>
<td>0.005</td>
<td>25</td>
<td>5.249</td>
</tr>
<tr>
<td>0.003</td>
<td>25</td>
<td>10.275</td>
</tr>
</tbody>
</table>
4.6 Conclusion

We have presented a novel approach to the problem of location discovery in an ad hoc network using computational geometric methods. In contrast to the existing approaches of estimating a point location, algorithm `location_region_identify` computes the region of residence for a node, where it is guaranteed to be found. The proposed algorithm takes only $O(nD)$ time to identify the stable regions of residence for all nodes in the network. Simulation results show that our algorithm succeeds in finding reasonably small stable regions of residence for all non-reference nodes under varying conditions.