Hop count based distance estimation in mobile ad hoc networks – Challenges and consequences

Sabrina Merkel *, Sanaz Mostaghim, Hartmut Schmeck

Institute AIFB, Karlsruhe Institute of Technology (KIT), 76128 Karlsruhe, Germany

Abstract

Hop Count based distance estimation is an important element for localization of devices in mobile ad hoc networks. Deriving distance estimates from hop counts is prone to error, especially in networks with low density. This paper shows that mobility can affect the accuracy of hop count based distance estimation. Two types of error are defined to describe and analyze the source of underestimation and overestimation of distances in a mobile ad hoc network. Different movement patterns are examined to get an insight of their impact on the hop counts and the estimated distances accordingly. Our experiments and analysis indicate that mobility can have a positive effect on the accuracy of distance estimates which results from a combination of asynchronous computation of hop counts and mobility of the nodes. At the same time, this positive effect can turn into a negative one with increasing mobility. Therefore, we determine characteristics, such as direction, speed, and similarity in movements of neighbors which are responsible for the disparity in the influences of the investigated mobility patterns. A study of these properties is presented and their individual effect is explained in detail. The difference between mobility and density induced error is discussed and their individual adverse effect is weighted against each other. In addition, we introduce a modified algorithm to determine hop counts which is designed to mitigate the effect of mobility. Two indicators are presented to identify and characterize the mobility of devices in a decentralized way.

1. Introduction

In many applications, such as geographic monitoring or target tracking, a large number of possibly mobile devices is used to accomplish a specific goal. In general, such devices have limited resources and wireless communication range to exchange short messages with other nearby devices. To send messages to remote devices in the network all nodes act as relay nodes and forward messages from other devices. A network of such devices is called mobile ad hoc network (MANET) as the network’s connectivity is dynamic and formed ad hoc. MANETs are widely studied in literature and there is a working group founded by the Internet Engineering Task Force (IETF) to investigate related issues [1]. As bulks of such devices might be necessary to perform a task, they are generally assumed to be mass-produced, therefore inexpensive, and of small sizes. In such networks, there is often a trade-off between reliability and cost. Additionally, to avoid further expenses, the devices usually are not equipped with any localization technique. This makes it hard to realize applications which depend on the location of each device such as the allocation of event reporting in a monitoring sensor network [2,3], location dependent routing [4,5], and many more. Localization can also support several MANET characteristic tasks, such as optimal area coverage in routing [6], assistance of network querying [7,8], or security [9,10]. Therefore, alternative localization techniques were proposed for ad hoc networks (cf. [11–14] for an overview). Many

* Corresponding author. Tel.: +49 721 608 4 4556.
E-mail address: sabrina.merkel@kit.edu (S. Merkel).

1570-8705/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.08.013

Please cite this article in press as: S. Merkel et al., Hop count based distance estimation in mobile ad hoc networks – Challenges and consequences, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.08.013
of these algorithms rely on the estimation of the distances between each node and a small number of so called anchor nodes. Anchor nodes are assumed to know their own coordinates either through a GPS-receiver or a priori configuration. There are several ways to estimate distances. In this paper, we consider distance estimation based on hop counts, i.e. the minimum number of relay nodes needed for communication, and analyze the effects of mobility on this estimation technique.

For our investigations, we focus on mobile devices which are not autonomous and cannot move by themselves like robots. Instead, they are passive, i.e. they can be moved by people, animals, or nature. This implies that the distance estimation algorithm has no information about whether or to which position a device has been moved. As the movement of devices in a mobile ad hoc network highly depends on the application and the environment, a large spectrum of mobility models is analyzed here. Two different error types are identified caused by either the distribution of the devices or mobility. Our goal is to quantify the error of such hop count based distance estimation in a dynamic network and to identify the main influencing factors by comparing various mobility patterns. Our observations and analysis indicate that mobility positively influences the error rate, counteracting the error induced by low density. Nevertheless, a high mobility can also increase the error, turning a natural overestimation into an underestimation of the respective distances. We identify various characteristics of the applied mobility, such as speed, direction and similarity of moves in a neighborhood, which have a different impact on the height of the mobility induced overestimation. Furthermore, we quantify, compare, and explain their individual impact on the hop counts and suggest two indicators, which can be computed in a decentralized way and are able to give information about the characteristic of the mobility exerted on a device.

This paper is structured as follows. Section 2 describes the related work. In Section 3, the network model and basic algorithmic concepts are introduced. Section 4 presents the investigated mobility models and Section 5 addresses the error model. Section 6 introduces the experiment settings, their results, and interpretation. The paper is concluded in Section 7.

2. Related work

Many localization algorithms rely on special anchor nodes, i.e. nodes which know their own location, to derive the location of all other nodes in the network. For example, a node’s position can be estimated to be in the centroid of the closest surrounding anchor nodes, such as proposed in [15–23]. The main drawback of this approach is the need for numerous anchor nodes to be able to distinguish the positions of all other network nodes. To avoid this issue, there are algorithms, e.g. [24] which rely on the estimation of the distance between the node of which the location is needed and a small number of anchor nodes. Anchor nodes can be sensor nodes in the network [24] nearby cell-towers [25] or even wireless local area network routers [26,27].

The computation of distance estimates has been widely studied in literature. The most commonly described methods to assess the distance between two devices are called range-based approaches. In range-based distance estimation the communication signal is analyzed using special hardware. This is mostly done by evaluating its strength on reception [12,25–36], its transmission time [37–41], or its angle of arrival [42,43]. Another way to estimate distances without relying on hardware is called range-free distance estimation. Range-free distance estimation only works for multi-hop networks. There are two types of range-free distance estimation algorithms. The first one uses the number of shared communication partners to approximate the surface of the common communication range between two nodes and deduces the distance from it [44–47]. The second kind of range-free distance estimation is based on hop counts [24,30,48–54]. To estimate the distances between the nodes in the network and the anchor nodes, all nodes count the communication hops between themselves and the anchor nodes. This value is called the hop count and is subsequently multiplied with an estimate for the physical length of one hop to compute the distance estimate. Various methods are proposed to estimate the length of one hop. The most basic idea is to use the communication range \( r \) as an approximation for the length of a hop because in a dense and uniformly distributed network, the nodes with the same hop count are located in rings around the anchor node and each ring has approximately the width of the communication range \( r \) [55]. Nevertheless, in a network which is not perfectly dense this assumption is rarely fulfilled. Fig. 1 shows the perfect gradient rings with black lines and it is easy to see that the gradient rings formed by the GA differ from the perfect gradient rings. Due to this density problem, a more sophisticated estimation method for the length of one hop is needed. In [55], the expected length of a hop is given for uniformly random distributed networks, depending on the local neighborhood density. A similar principle is used in [50] for networks with varying density. Here, density dependent reduction rates are chosen empirically. Another method to estimate the hop length is provided by the DV-HOP approach [30,48,49,51] which uses the known Euclidean distances between anchor nodes and divides them by the hop counts determined with GA.
to calculate the average length of a hop. Besides improving the hop length estimate, different methods have been developed to determine the position of a node within its own gradient ring. For example in [24], an average of all hop counts in a node’s neighborhood is calculated before multiplying it with the hop length estimate. In [54] this approximation is further refined by using the exact proportions of neighbors with lower, equal, or higher hop counts to improve the estimate even further.

The previously described methods to estimate distances based on hop counts mostly consider static networks. While [56] investigates the effect of mobility on a network protocol in terms of network connectivity and communication link breakages, little is done to examine the effects of mobility on localization in MANETs. Some research was performed on improving localization using mobile anchor nodes [57–60]. In [61], mobility of some nodes in the network is identified to have a positive effect on distance estimation. However, the underlying assumption, here, is that a node knows when it is moved and, if so, immediately refreshes its own distance estimation. This leads to a general improvement of distance estimation within the network as it resembles a scenario in which a new node is placed at the target position and, therefore, the network density is increased. In [54], a method is presented to account for mobility when calculating the distance estimates from hop counts. However, the expected moving distance of the applied mobility model has to be known to adapt the estimation. Similar knowledge is required in [41] where a directional localization algorithm is proposed for mobile networks. For the localization algorithm, the nodes are supposed to know the direction and distance of their last movement and exchange this information with their neighbors to calculate their neighbors’ new relative locations. In [62], the authors assume that mobility in a network is usually not completely random but follows a certain pattern and they propose a dead reckoning model to disseminate both location and the movement pattern, which is then used to predict or track future movements. In [63] a learning based classification method is used to recognize the mobility model in raw mobility traces collected from the nodes’ motions. Similar location prediction is done in [64,65] where a combination of sensors or the average walking speed are used for a dead reckoning model to track locations of mobile devices. In [66], a sequential Monte Carlo method is proposed to probabilistically determine locations. The disadvantage is its need for numerous anchor nodes to narrow down the possible locations. In [67], a signal strength based distance estimation technique is evaluated for dynamic networks. The authors detected that Rayleigh fading may introduce significant errors due to the motion of the sensor. Similar to these findings, we expect passive mobility to have an impact on range-free distance estimation methods. However, the influence is not originated in hardware related issues and, therefore, has to be analyzed independently. In [68] this suspicion is reinforced by the results of a preliminary study. Subsequently, we give a detailed analysis of the effects of mobility and provide some guidance about how to detect and handle passive mobility when using the GA for distance estimation.

3. Network model and algorithmic concepts

The network considered in this study contains mobile devices which are randomly distributed on a two-dimensional plane. The mobile devices do not have knowledge of their topology or locations. Each device can be moved and can only communicate with the devices in its neighborhood. The devices do not know if or where they have been moved. The plane does not have any obstacles and collisions between devices are not considered. We are aware that these simplifications are not consistent with most real environments of MANETs. Nonetheless, these simplifications make sure that the observed results are triggered solely by the exerted mobility and that the movement is not influenced by other devices or obstacles on the plane. We define the neighborhood of a device as a physical neighborhood on the plane within a fixed Euclidean distance $r$ from the device. We do not consider shadowing and fading effects when moving the devices and assume that all the devices have homogeneous properties except for one anchor device located at the top-left corner of the environment (cf. Fig. 1). The anchor device is not mobile, but has the same communication radius of $r$.

3.1. Gradient Algorithm (GA)

In the Gradient Algorithm proposed by Nagpal et al. [24], the anchor device initiates a gradient wave by sending a message including an integer value of zero, called hop count, to its neighbors. Each neighbor takes the minimum hop count it receives, increments it by one, and propagates it to its neighbors. This process is constantly repeated in order to adapt to changing hop count values due to mobility. Fig. 1 shows an example network performing a GA. All devices with the same hop count value are colorized in the same shade of Grey.

4. Mobility

In this section, we first briefly explain the mobility models we implemented for our analysis and additionally present new mobility models which we call Angle Mobility. Similar to [56], we categorize the mobility models into individual and group mobility models [68].

4.1. Individual mobility models

In individual mobility models, a node defines its next position independently from any other node in the system. Fig. 2 shows trajectories of a node for the different moving models.

In the Random Walk mobility model [56], every node selects a random direction and a random speed within allowed ranges and keeps on moving until a predefined distance is traveled. In Chaos Move [68], we slightly modify this model and let the nodes select a new random direction and speed “at each time step”. This mobility model is supposed to keep the mobile devices in a small area around their starting position compared to the Random Walk model.
We design Random Direction Walk [56] so that a node starts moving like Random Walk until it reaches the border of the simulation area. There, it pauses for some cycles before it changes its direction. Bounded Random Walk [56] is similar to Chaos Move with the difference that a node changes its speed and direction within a small range around its former values. Another variant is the Gauss Markov Move [56] in which a node selects the next direction and speed according to the following equations:

\[
s_t = \alpha \cdot s_{t-1} + (1 - \alpha) \cdot \mu_s + \sqrt{(1 - \alpha^2)} \cdot s_g
\]

\[
d_t = \alpha \cdot d_{t-1} + (1 - \alpha) \cdot \mu_d + \sqrt{(1 - \alpha^2)} \cdot d_g
\]

where \(s_t\) and \(d_t\) are new values for speed and direction, \(\alpha\) is a random parameter \((0 \leq \alpha \leq 1)\), and \(s_g\) and \(d_g\) are chosen from a random Gaussian distribution with zero mean and a standard deviation of one. \(\mu_s\) and \(\mu_d\) are constant values such as 0.03 and 0. In Probabilistic Random Walk [56], three possible states are defined separately for the movement in y- and x-axis directions. For the y-axis the node is moving backward, forward or it stands still. For the x-axis the node is moving left, right, or stands still. There are fixed probabilities to transit from one state to the other which emphasize continuous moves in the same direction [56].

4.2. Group mobility models

The group mobility models are characterized by mutual influences as one node’s locomotion affects other nodes’ next positions. In the Column Mobility model [56], all nodes move randomly within the environment except for a configurable set of nodes which consists of a group leader and its followers. All followers move more or less in a row behind the leader, simulating children walking in a line following their parent. In the Nomadic mobility model [56] a group of leaders is defined which move according to the Chaos Move and collect followers. Whenever a node is within the signal range of a leader, it starts following the leader and can move randomly within the communication range of the leader. When a leader has more than a predefined number of followers (20 nodes), a randomly chosen node leaves the group. In the Reference Point mobility model [56], each node has a virtual reference point and moves randomly within the communication range of its reference point. At each step, all reference points are moved according to Chaos Move, but with the same direction and speed like they are connected to each other. In [68], we introduced the Stream Mobility model to simulate a streaming movement of devices, as if they were put into water or moved by wind. Each node remembers the last move of its neighbors and calculates its next move by modifying the observed moving direction by adding or reducing a randomly chosen degree between 0 and 30. This way, the nodes in a neighborhood, move like in a stream. All nodes which are not involved in the defined group mobility patterns are moved according to the Chaos Move.

Fig. 2. Trajectories. Top row from left to right: Chaos Move, Random Direction Walk, and Bounded Random Walk. Bottom row from left to right: Random Walk, Gauss Markov Move, and Probabilistic Random Walk.

4.3. Angle mobility models

We propose Angle Mobility in which nodes move with a specific predefined angle and speed around the anchor node. We propose this mobility model to be able to regard the effects of movement direction, speed and similarity between neighbors in isolation. With the angle mobility model, each of these three characteristic settings can be controlled separately while keeping the value for the others constant. In accordance to the previous categorization, we distinguish between individual and coupled angle mobility.

Individual angle mobility: In the individual angle mobility model, a node moves with a predefined speed in
the direction of $\alpha$ with respect to the anchor node (cf. Fig. 4).

**Coupled angle mobility**: The coupled angle mobility model is similar to the individual angle mobility. A node moves together with a fixed number of its neighbors in the same direction (cf. Fig. 4).

In both angle mobility models, we define the same angle $\alpha$ for all the devices which move. An Angle of $\alpha = 0$ indicates that the mobile nodes move away from the anchor where $\alpha = 180$ means that the mobile nodes move towards the anchor. Therefore, a move with $\alpha = 90$ means that the mobile nodes move along the same gradient ring. Fig. 5 illustrates example trajectories of these movements. From this, we observe that the density of nodes in the environment for individual mobility is more regular than for the coupled mobilities. This will be further analyzed in the experiments.

### 5. Error in Gradient Algorithm

As stated in Section 2, many localization algorithms rely on distance estimation derived from hop counts. Although there are various refinement techniques to improve distance estimation (cf. Section 2), the main principle is to multiply the GA derived hop count with an estimated length for one hop. To evaluate the robustness of hop count based positioning concepts under mobility, we look at the deviation of calculated hop counts using GA from ideal hop counts $h^{\text{ideal}}$, which corresponds to the ordinal number $i$ of the perfect gradient ring in which a node is located. This allows it to make a general statement about the influence of mobility on any hop count based distance estimation and not just a specific method. Considering a perfectly dense and evenly distributed ad hoc network, each device $n \in N$ with $N$ being set of all devices in the network, would have a hop count corresponding to $h^{\text{ideal}}(n) := \left\lceil \frac{d(n, \text{anchor})}{r} \right\rceil$, where $d(n, \text{anchor})$ indicates the Euclidean distance between the node $n$ and the anchor. The area in which $h^{\text{ideal}}(n)$ would return the same value for all $n$ corresponds to the rings shown in Fig. 1. Such a ring is called gradient ring $i$, where $i$ denotes the common value for $h^{\text{ideal}}$ and corresponds to the ordinal number of the gradient ring, starting at 0 for the gradient ring which is equal to the communication range of the anchor node. Usually, the devices are not perfectly distributed and, therefore, we can compute the deviation from the ideal case for each node $n$ as the difference between the ideal hop count and the one obtained by the GA ($h(n)$):

$$E(n) = h(n) - h^{\text{ideal}}(n)$$

This deviation is called “hop count error”, aware of the fact that it is not really an error but rather a deviation from an ideal case. Since it makes a difference if the hop count assigned by the GA is higher or lower than the ideal hop count, a quadratic error is not considered. If $h^{\text{ideal}}(n)$ and $h(n)$ match for all $n$, the hop count error is zero and the length of each hop equals $r$. As a result, the distance estimation is straight forward and has only little room for error. The remaining source of error arises from the approximation of the node’s position within its gradient ring and can be minimized using techniques like [24] or [54]. Therefore, the hop count error is a useful indicator to evaluate the robustness of any hop count based positioning concept. Similar to the computation of the hop...
count error, the average error in a gradient ring $i$ can be computed as

$$E_i = \frac{1}{|J|} \sum_{j \in J} (i - h(j))$$

with $J = \{ j \in \mathcal{N} | d(j, \text{anchor}) = i \}$ referring to all the nodes which are physically located in the gradient ring $i$.

**Definition 5.1 (Positive Error).** Node $n$ is defined to have a positive hop count error if it holds that $E(n) > 0$.

This means that the hop count resulting from GA ($h(n)$) is larger than it must be ($h^{\text{ideal}}(n)$). We call this error an overestimation because it usually leads to overestimated distances. Fig. 6(a) illustrates a simple example. The values for $h(n)$ are depicted within the dots representing the devices, while the value for $h^{\text{ideal}}(n)$ corresponds to $i$ and is depicted above the respective gradient ring.

The error occurs in the gradient ring 2 due to the gap in ring 1. The node physically located in ring 2 does not have any member of ring 1 in its neighborhood and, therefore, its hop count is computed as 3. The error value in hop 2 is $E_2 = \frac{1}{4}$. In general, this error occurs due to low density in the smaller gradient ring.

The positive-error has a cumulative character because not only the node with low density in its neighborhood is affected but also subsequent nodes which use this node as basis to calculate their hop counts and so on. In mobile networks, the positive error can dynamically occur and disappear as the gaps between nodes can become larger or smaller respectively.

**Definition 5.2 (Negative Error).** Node $n$ is defined to have a negative hop count error if it holds that $E(n) < 0$.

Due to asynchronous computation throughout the network and lack of knowledge about the movements, another type of error can happen: When a node is moved away from the anchor into a gradient ring with a larger hop count value, its new neighbors might adapt their hop counts to the arrived node before the moved node can update its own hop count. This way, an underestimation of the hop count can occur among the devices in the gradient ring with higher ordinal number. This phenomenon helps to reduce the density-induced natural overestimation of hop counts by causing a counteracting negative error. Fig. 6(b) illustrates this effect. In this example, the negative error happens in the gradient ring 3 with $E_3 = -\frac{1}{5}$.

### 5.1. Various effects of mobility on the Gradient Algorithm

There are three scenarios in which mobility influences the GA:

1. Scenario 1: A node has been moved to another area and has not yet updated its hop count. The node’s hop count, in this case, can be either higher or lower than it should be.
2. Scenario 2: A node from a small gradient ring is moved to a gradient ring with higher hop count values. Because the GA calculates the hop counts using the smallest available hop count in a node’s neighborhood, the moved node reduces the hop count of its new neighbors. This leads to underestimation and is an infectious error because it can spread through the network.

3. Scenario 3: A node is moved to a low density area and helps to correct the hop counts of badly connected nodes. This has a positive effect on the hop count error.

The first scenario is negligible because it is only temporary until the next update of the hop count and the error concerns just the moved node itself. The second scenario, on the other hand, can have considerable impact due to its infectious character. Here, the error is not limited to one node or the node’s neighborhood but can spread through the whole network. It has a cumulative character similar to the density induced overestimation. The third scenario has a positive effect on the hop count distribution as higher density always works in a correcting way. On the other hand, it is just a shift of density and the original region of the moved node gets sparser at the same time, increasing the error there. The only situation where this really improves the overall hop counts is when the nodes move closer to the anchor because, due to the cumulative character of density induced overestimation in a network, a lower error near the anchor has a positive effect on all subsequent nodes in the network. Fig. 6(c) shows an example for the third scenario. From the standpoint of a positioning algorithm, the most inconvenient error is the infectious error described in scenario 2.

5.2. Overestimation vs. underestimation

In order to analyze the effects of mobility on the GA, we consider overestimation and underestimation separately. In general, overestimation and underestimation of distances are both undesired for localization algorithms. According to that, the best case would be given when both effects compensate each other and the distance estimates become accurate. In case of controllable mobility, the right settings could be used to reduce the distance estimation error. Since the optimal parameter settings vary for any mobility model they have to be found empirically for each model. However, this is not the focus of this study since we consider passive mobility. In many applications the characteristic of the mobility is either unknown or varies over time. This leads to an unpredictable underestimation which can result in a negative error. The refinement techniques to improve hop count based distance estimation which are presented in Section 2 are all designed for static networks and their objective is to deal with the problem of low density. In dynamic networks with uncontrollable mobility, however, they cannot be applied without further ado. When used in a mobile network, the refinement techniques might not only fail but even worsen the distance estimation error leading to higher underestimations. We conclude that trying to counteract density induced overestimation with mobility is not recommendable unless the mobility can be controlled. Therefore, it seems straightforward to try and contain mobility induced underestimation, especially the infectious type described in scenario two, in order to be able to apply the refinement techniques presented for static networks. A distance estimation algorithm on a device should be able to recognize mobility online during its operational time, analyze the mobility’s characteristic influence on the hop count, and eliminate this effect by adapting its own hop count accordingly.

5.3. Decentralized mobility recognition and treatment

As concluded before, the infectious mobility underestimation of Scenario 2 in Section 5.1 is the most harmful error for localization. Hence, to avoid infection, we define a slight variation of GA, where the nodes only update their hop count when they not only find a new minimum hop count in their neighborhood but also a new maximum hop count. The consideration behind this is that it is more likely for both values to change when the node itself is dynamic. Hence, the adaption of infected nodes to lower hop counts can potentially be avoided. We call this variation Maximum oriented Gradient Algorithm (MoGA). The computational complexity of MoGA is comparable to that of GA, since the search for a maximum and a minimum of a list can be done simultaneously. Only the number of comparisons is doubled.

As a second proposal to tackle the mobility in GA, we present two indicators which are designed to recognize and characterize mobility. In this paper, we consider passive mobility which means that the nodes do not know whether or where they are moved. This makes it challenging to adapt the GA to the exerted mobility. The only way for nodes to realize mobility is when a change in their neighborhood can be observed. One consequence is that nodes are not readily able to differentiate between their own movement and the movement of their neighborhood. To integrate mobility in the GA, a decentralized online mobility recognition is needed such that each node is enabled to decide locally whether and how to adapt its hop count. As stated before, the only available information about dynamism are changes in a node’s neighborhood. Therefore, we propose two indicators which we call ID-change and HC-change and are designed to capture these changes in the neighborhood.

ID-change (cf. Eq. (5)) reflects the percentage of neighbors that has changed since the last update and HC-change (cf. Eq. (6)) computes the percentage change of the average hop count in the neighborhood since the last update:

\[
\text{ID-change} = \frac{|ID_t \setminus ID_{t-1}| + |ID_{t-1} \setminus ID_t|}{|ID_{t-1} \cup ID_t|} \quad (5)
\]

With \(ID_t\) denoting the set of IDs in the node’s neighborhood at time \(t\). The HC-change is computed as follows:

\[
\text{HC-change} = \frac{HC_t - HC_{t-1}}{HC_{t-1}} \quad (6)
\]

where \(HC_t\) denotes the average hop count in the node’s neighborhood at time \(t\). Experiments show that both values are suitable to provide information about whether a node has been moved and even some indication about...
6. Experiments

In order to quantify the impact of passive mobility on the hop count distribution in a network and to identify the influence of various mobility patterns and their responsible characteristics, the following experiments are performed. For the experiments a simple Java-based simulation of a mobile ad hoc network is used. The GA is initiated from one static anchor node such that all nodes determine their hop counts with respect to this anchor node. The average hop count error is calculated for each gradient ring separately as described in Section 5. In the first set of experiments, the different mobility models presented in Section 4 are investigated. The second set of experiments provides some insights into the impact of moving direction, speed, and coupled movements of neighbors. In the third set of experiments, the effectiveness of the MoGA is evaluated. In addition, the ID-change and HC-change indicators are assessed and their usage for mobility recognition and classification is tested.

6.1. Experiment settings and mobility parameters

For simulation, we select a 2-dimensional rectangular environment with no obstacles and normalize its size to 1.0 × 1.0 units. Hence, we can denote all sizes used in the simulation, for example the signal range, as a ratio of the field width and the results can be transferred to any testbed size easily. 1000 nodes are positioned in the environment using a random uniform distribution. The anchor node is placed in the top-left corner of the field and the communication radius is set to 0.07 units. This value for the signal radius is selected as it corresponds to an average neighborhood of 15 for distance estimation. The implemented mobility models are selected from Section 4.

Table 1

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
<td>0.4</td>
<td>0.6</td>
<td>0.9</td>
<td>1.1</td>
<td>1.4</td>
<td>1.7</td>
<td>1.8</td>
<td>2.1</td>
<td>2.3</td>
<td></td>
</tr>
</tbody>
</table>

maximum distance for moving into the same direction is selected as 0.6 units. In Random Direction Walk a move is paused for 10 cycles. For Gauss Markov Move $z$ is set to 0.75, and an average angle of $0^\circ$ measured from the $x$-Axis of the environment (bottom border) is selected. Angle tolerance for Bounded Random Walk is set to $30^\circ$. In the Column Move, there are 10 leaders which are each followed by 10 nodes. The angle tolerance for Stream Move is set to $30^\circ$.

6.2. Analysis of different mobility patterns

In the first set of experiments, the error is calculated for 100 cycles starting in cycle 30. Hence, we ignore the phase in which the GA has to be propagated through the network for the first time. In one simulation cycle, 1000 nodes are executed in a random order, i.e. they first update their hop count and then move with a probability of $p_m = 0.5$.

6.2.1. Overestimation of hop counts in a static network

First, we look at a static network and calculate the error value $E_i$ for all 21 gradient rings. The results are shown in Table 1. We obtain only positive values which intensify as the index of the gradient ring increases. This shows that the overestimation accumulates with increasing distance from the anchor node, which confirms the error model in Section 5.1. In a static network, the error is positive for all gradient rings, i.e. the hop count in all gradient rings is overestimated.

6.2.2. Influence of individual mobilities

In order to analyze the effect of mobility on the overestimated hop counts of a static network, we take the mobility models from Section 4.1. At first, we look at the individual mobility models where the next position of a node does not rely on the movement of other nodes in the network. Fig. 7 shows the average error for the individual mobility models in gradient rings 10 – 15. As one can
see, all the individual mobility models overcome the positive error of a static network. Furthermore, the error is quite constantly near zero for all six gradient rings. This shows that the compensation of positive and negative error works for each hop independently, which is an important observation that guarantees an improvement in a hop count derived distance estimation throughout the network. When looking at Fig. 7, the Probabilistic Random Walk (PR) shows a noticeably higher underestimation than the other mobility models. This can be explained by the character of this model as all nodes are moved almost into the same direction which leads away from the anchor. As an underestimation can only occur with movements in the opposite direction of the anchor, the negative error introduced by this move is higher than when using other mobility models.

Among the other mobility models, Bounded Random (BR), Gauss Markov (GM), Chaos Move (CM), Random Direction (RD) and Random Walk (RW) show very similar behavior in terms of error values. Random Direction exhibits a slightly more positive behavior. We explain this by the reduced mobility, which is accompanied by a reduced underestimation, during the observed period of time thanks to the pauses at the border.

From these results, we conclude that the error rate of GA can be reduced in scenarios with independent individual movements. Our experiments reveal that underestimation does not get dominant under the selected settings, such as a movement probability of $p_m = 0.5$ and a speed between $\frac{\omega}{x} - 0.001$ and $\frac{\omega}{x} + 0.001$ units per cycle, unless the movements mostly lead away from the anchor.

6.2.3. Influence of group mobilities

In order to investigate these observations in more detail, we carry out experiments using the group mobility models from Section 4.2. Fig. 8 shows the hop count error values for the group mobility models. The Stream mobility model was designed to simulate similar movements amongst neighbors. When a node moves according to Stream Move (StrM), its direction and speed are similar to other nodes in the same region. Thus, the infection described in scenario two of Section 5.1 is not so high and the similarity of the moves in a neighborhood has a moderating effect on the mobility induced underestimation as can be seen in Fig. 8. The same effect can be observed with the Nomadic Move (NomM) where groups of nodes of move together through the network. Also, for this move, the underestimation seems to be less pronounced. As already shown for the Probabilistic Random Walk, another important factor for the development of the hop count error is the direction of a move relative to the anchor. Both in the Stream and the Nomadic Move the groups move quite randomly through the network. The mixture of movements leading away and towards the anchor are resulting in a less negative error as for example the movements in the Column Move (ColM). Here, the groups keep moving in almost the same direction. Next to the Column Move, the Reference Point Move (RefM) also shows a high negative error. This can be explained by the additional movement which is introduced through the dynamic reference point network supplementary to the individual moves of each node.

The examined group mobility models further emphasize the observation that the direction of the movements influences the error rate in a network. Moreover, the results for the Reference Point Move indicate that speed might be an influencing factor for the level of negative error that arises through mobility. It has been shown that the way nodes move relative to each other is another important factor. A similar movement of adjacent nodes seems to lead to a smoother compensation of overestimation.

Fig. 8. Hop count error with various group mobility models.

Fig. 9. Error with different angle and speed for individual (a) and coupled (b) mobility.
The experiments confirm that introducing certain mobility into a network has a counteracting effect on the natural hop count overestimation, but they also show that this effect can turn into a network wide underestimation. Also, the impact of mobility is not equal but strongly depending on the characteristics of the mobility pattern, making the introduced underestimation hard to predict. Both observations confirm the problems with mobility induced underestimation compared to overestimation originated in low density as discussed in Section 5.2. In addition, the direction of the movement with respect to the position of the anchor plays an important role as an underestimation can only arise from movements leading away from the anchor node.

6.3. In-depth analysis of direction, speed, and coupled movements

The investigated mobility models indicate that there are three characteristics of movements which influence the mobility induced error in the GA:

- Direction with respect to the anchor node.
- Speed, i.e. traveled distance between hop count updates.
- Similarity of movements in a neighborhood.

With the following experiments, we take a closer look at these three identified influencing factors and study their individual impact. For this, we consider the angle mobility models defined in Section 4.3 and apply them to 50 random nodes (5 leaders with 10 followers each). Fig. 9(a) shows the average error in the gradient rings 10–15 for individual movements with different angle and speed. We observe that both direction and speed are indeed relevant factors for the mobility induced underestimation in a GA. The experiment additionally shows that both, angle and speed, have to be considered together when trying to estimate the amount of underestimation induced by a movement. While speed is negatively correlated with the error value for an angle between 0° and 90° as well as between 270° and 360°, it has almost no impact when nodes move with an angle between 135° and 225°. In fact, movements towards the anchor hardly influence the GA overestimation at all, and the hop count distribution in the network is almost the same as in a static one. Similar results can be observed for coupled mobilities as shown in Fig. 9(b). A considerable difference between the coupled and individual mobilities can be observed around the angle of 180°, where coupled movements seem to decrease the error in contrast to individual movements. We explain this by a simultaneous increase in the density close to the anchor node as described in scenario 3 of Section 5.1. When nodes move in the direction of the anchor, they do not infect other nodes but provide higher density in this area and therefore moderate the natural overestimation of hop counts. As this error is cumulative, the whole network profits from a high density around the anchor node. The reason why we do not see the same effect for individual movements is that the density does not increase significantly at a specific point in time. Fig. 10 shows the average number of agents located in the first five gradient rings with individual and coupled angle mobility. We observe that in the case of 180° (nodes move towards the anchor) the density increases for both cases. This implies that the difference has to be originated in the concurrency of the density increase with coupled mobility. In general, the coupled mobility leads to a higher average density in the first gradient rings during the observed period of time compared to the individual mobility.

Fig. 10. Average number of agents located in the first five gradient rings for individual and coupled angle mobility.

Fig. 11. Error for individual vs. coupled angle mobility for different speeds (a) and angles (b).
As the error is symmetrical for the respective angles, we only consider angles with 0–180° in the following evaluations.

Subsequently, we compare the individual and coupled mobilities. Fig. 11 illustrates the hop count error averaged over the angles (Fig. 11(a)) and speeds (Fig. 11(b)) respectively. These results confirm the observations made in the investigations of the group mobility models in Section 6.2.3. The same number of nodes are moving in both experiments, consequently, we can conclude that similar movements in a neighborhood indeed mitigates the effect of mobility. This results in a lower underestimation of hop counts at least for all movements which are not in the direction of the anchor node (0–90°). For directions between 0° and 90°, the coupled mobility also has a lower overestimation. We do not explain this by the effect of counteracting mobility-induced underestimation but by the effect of simultaneous density increase near the anchor (cf. Section 5.1).

6.4. Decentralized mobility recognition

With the following experiments, we intend to test the effectiveness of the MoGA for suppressing the negative-error caused by mobility. Also, we want to analyze if it is possible to recognize mobility and its characteristics solely by observing the local neighborhood. MoGA, as introduced in Section 5.3, was designed in order to avoid infection of nodes with negative-error. Hence, we let the nodes only update their hop count when they not only have a new minimum hop count in their neighborhood but also a new maximum hop count. Fig. 12 shows the results with and without this Maximum modification. In all cases, we succeeded to decrease the mobility induced underestimation in the network. Nevertheless, the decrease is almost constant for any speed or angle. This illustrates the need for a decentralized way to detect and characterize mobility to be able to adapt the hop counts dynamically at runtime.

For this purpose, we proposed ID-change and HC-change (cf. Eqs. (5) and (6)). Fig. 13(a) shows the average values of both factors in static and dynamic networks. It becomes apparent that, while there is only a small difference in HC-change between static and dynamic nodes, the ID-change is a strong indicator for whether a node has been moved or not. Although HC-change does not seem to be a good indicator to differentiate between static and dynamic nodes, it has another application. When looking at Fig. 13(b), HC-change is depicted for different angle and speed. It becomes obvious that the curve has a similar shape to the error depicted in Fig. 9(a). This suggests that HC-change can be used to determine speed and direction of a moved node in a network. Both factors can be computed online on a device with only local information about the nodes in the neighborhood. It might be possible to use...
this information in combination with an online learning mechanism to enable nodes to recognize whether and how they have been moved and to adapt their own hop count accordingly.

7. Conclusion and future work

This paper presents an experimental study about the influence of mobility on hop count based distance estimation. The goal is to obtain an understanding of the hop count distribution in the presence of various mobility models in an ad hoc network. Particularly, two error types are defined and investigated which capture mobility and density induced error leading to under and overestimated distances. The experiments and analysis indicate that with mobility of low rates, the error in a network can be reduced. Nevertheless, a high degree of mobility as well as certain characteristics of the movements can lead to an overcompensation of the natural negative error. For situations in which the mobility in a network cannot be controlled, this overcompensation might have an even worse effect on the accuracy of the distance estimates compared to the naturally existing error due to low density. Experiments revealed that direction, speed, and similarity of movements in neighboring regions affect the error in different ways. A modified algorithm to calculate hop counts was presented which aims to reduce error caused by mobility. While the results were improved, the correction was not enough to compensate all kinds of mobility. In addition, two indicators were suggested which can be calculated online in a decentralized way using only information about changes in a node’s local neighborhood. It was found that both indicators in combination are helpful in discovering mobility and even characterize its properties such as speed and the direction with respect to the anchor node. We plan to focus our future research on using these indicators to dynamically adapt the estimated distances depending on the current mobility in the network. With this measure, we expect an improvement in localization accuracy for mobile nodes.

References


Please cite this article in press as: S. Merkel et al., Hop count based distance estimation in mobile ad hoc networks – Challenges and consequences, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.08.013


Please cite this article in press as: S. Merkel et al., Hop count based distance estimation in mobile ad hoc networks - Challenges and consequences, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.08.013.
Sabrina Merkel is a PhD Student and a Research Associate at Karlsruhe Institute of Technology (KIT), Germany. She graduated from Karlsruhe Institute of Technology (KIT), Germany, with a degree in Information Engineering and Management in 2010. Her research interests are swarm intelligence, Mobile Ad Hoc Networks and distributed algorithms.

Sanaz Mostaghim is a lecturer and research assistant at Karlsruhe Institute of Technology in Karlsruhe, Germany. She received her PhD degree in electrical engineering from the University of Paderborn, Germany in 2004. After her PhD, she worked as a post-doctoral fellow at the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland until 2006 before starting her work in Karlsruhe. In 2012, she received a habilitation degree in applied informatics from Karlsruhe Institute of Technology. Her current major research interests are swarm intelligence, computational science and organic computing.

Hartmut Schmeck studied at the Universities of Kiel (Germany) and Waterloo (Canada). Since 1991 he is a Full Professor of Applied Informatics at the Karlsruhe Institute of Technology (KIT). He is (co-)author of more than 140 publications on advanced algorithms and architectures, in particular on bio-inspired methods in optimization, algorithms for reconfigurable architectures, and, more recently, on self-organizing, adaptive systems in autonomic and organic computing with a special focus on critical infrastructures like traffic or energy systems.