Trajectory Based Forwarding and Its Applications

Dragos Niculescu
dnicules@cs.rutgers.edu
Dataman Lab,
Rutgers University

Badri Nath
badri@cs.rutgers.edu
Dataman Lab,
Rutgers University

Abstract

Trajectory based forwarding (TBF) is a novel method to forward packets in a dense ad hoc network that makes it possible to route a packet along a predefined curve. It is a generalization of source based routing and Cartesian forwarding in that the trajectory is set by the source, but the forwarding decision is based on the relationship to the trajectory rather than the final destination. The fundamental aspect of TBF is that it decouples path naming from the actual path, thereby providing a common framework for applications such as: flooding, unicast, multicast and multipath routing, and discovery in ad hoc networks. TBF requires that nodes know their position relative to a coordinate system. While a global coordinate system afforded by a system such as GPS would be ideal, in this paper we propose Local Positioning System (LPS), a method that only positions the nodes along the trajectory, by making use of other node capabilities, such as angle of arrival or range estimations, compasses and accelerometers. We explore several forwarding strategies that are appropriate for these node capabilities.

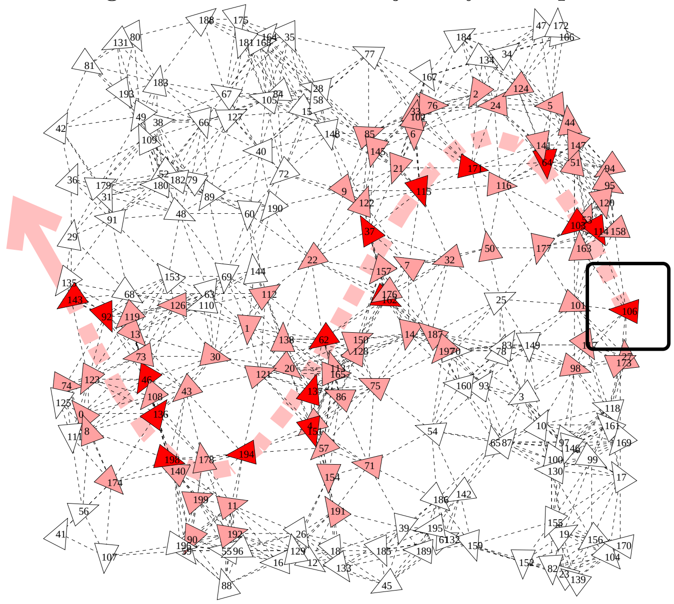
Keywords: trajectory, forwarding, routing, flooding, discovery, positioning, local positioning system

1 Introduction

The main features of new adhoc networks include large number of unattended nodes with varying capabilities, lack or impracticality of deploying supporting infrastructure, and high cost of human supervised maintenance. What is necessary for these types of networks is a class of algorithms which are scalable, tunable, distributed, easy to deploy, and most importantly easy to maintain. These large networks of low power nodes face a number of challenges: cost of deployment, capability and complexity of nodes, routing without the use of large conventional routing tables, adaptability in front of intermittent functioning regime, network partitioning and survivability. In all these networks, both basic network operations (routing, forwarding), and higher level applications (multicast, resource discovery) impose a tradeoff between communication overhead and infrastructure support. For example, a network may have high powered basestations to hold large routing tables, or it has to use flooding to discover routes on demand; it may have a GPS[1] receiver in each node, or it has to spend some energy running a positioning algorithm[2]. Another tradeoff is encountered in route management - either proactively maintain routes to all possible destinations, or reactively discover them when needed. Both approaches prove better than the other one under different mobility and communication conditions.

In this paper, we propose a new forwarding paradigm, Trajectory Based Forwarding (TBF), which aims to directly eliminate these tradeoffs by providing solutions that require neither infrastructure support, nor communication overhead for route maintenance. The central argument for using trajectories is that in ad hoc networks the topography is a good indication of topology. In this case, trajectories are a natural namespace to describe route paths. For example, an obstacle in the topography, which is mirrored into a hole in the logical graph, can be avoided with a detour trajectory. In figure 1, a node on the right of the network initiates a trajectory with the shape of a sine wave in order to avoid obstacles or congestion in the middle of the network. The dark colored nodes are directly touched by the forwarding process, while the light colored ones overhear the packet from the wireless medium. The idea is to embed the trajectory

Figure 1: Parametric trajectory example



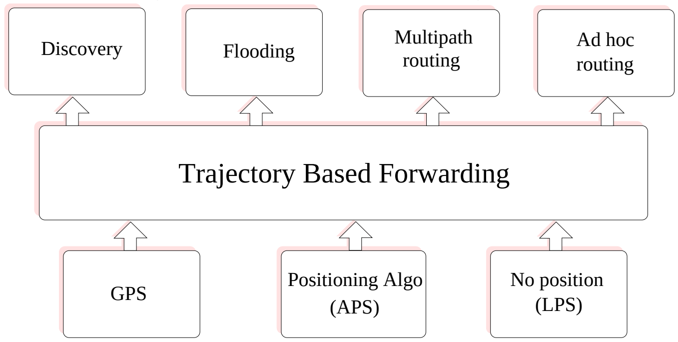
in each packet and let the intermediate nodes take the forwarding decisions. TBF is a generalization of source based routing[3] and Cartesian forwarding[4]. Like in source based routing, the path is indicated by the source, but without actually specifying all the intermediate nodes. Like in Cartesian forwarding, decisions taken at each node are greedy, but are not based on distance to destination - the measure is the distance to the desired trajectory.

TBF has a number of features that make it an ideal candidate for a low level primitive in any ad hoc network. First, it decouples the the path name from the path itself. This is the most critical aspect in a dense network, where intermediate nodes between source and destination might move, go into doze mode or fail, thereby rendering a source based path useless. Second, the specification of the trajectory is independent of the destination. This makes TBF usable both as a routing support, when the destination is indicated, as a discovery support primitive, when the destination is not known, or as a flooding replacement. Third, it may be assisted by various functionalities available in the nodes. Ideally, each node would be equipped with a GPS receiver, case in which nodes closest to the indicated trajectory will forward the packet. If, however, GPS is not available (such as non line of sight scenarios, or lack of sufficient precision) we propose LPS (Local Positioning System), a method to position only the nodes along the trajectory, making use of nodes' abilities to sense their neighbors (ranging, angle of arrival, compass).

Inherent problems in any ad hoc network, mobile or fixed include the size of the routing tables and the energy spent to find the routes. In highly mobile scenarios, much of the energy is spent on route maintenance. These problems are likely to get worse in large scale dense networks that are being envisioned for embedded computing, smart dust environments, and sensor networks. Discovery is another factor that greatly increases communication, as it is most of the time based on flooding schemes. The same is true about finding disjoint alternate paths for resilient communication. But in many cases, it is possible to route packets to a given destination without finding an exact route; or it is possible to discover a resource without flooding the entire network; disjoint paths can be generated on demand, without additional maintenance or searching. These cases require some special conditions, which are satisfied by many ad hoc networks - relatively high density and some positioning capability.

Besides simple unicast, trajectory routing and forwarding have significant advantages for many other important network functions such as flooding, discovery, multicast and broadcast, path resilience and even positioning. In this paper, we focus on issues related to trajectory forwarding in networks with and without

Figure 2: Trajectory routing layer



the availability of node positions, and identify a number of research challenges related to trajectories in ad hoc networks.

The rest of the paper is organized as follows: the next section reviews related work, section 3 details our proposed approach, the network model assumptions and the various forwarding methods; section 4 presents Local Positioning System (LPS), a method to position only the nodes involved in forwarding the trajectory. Section 5 discusses simulation results, while 6 mentions some challenging issues and possible future work, and we summarize with some concluding remarks in section 7.

2 Related Work

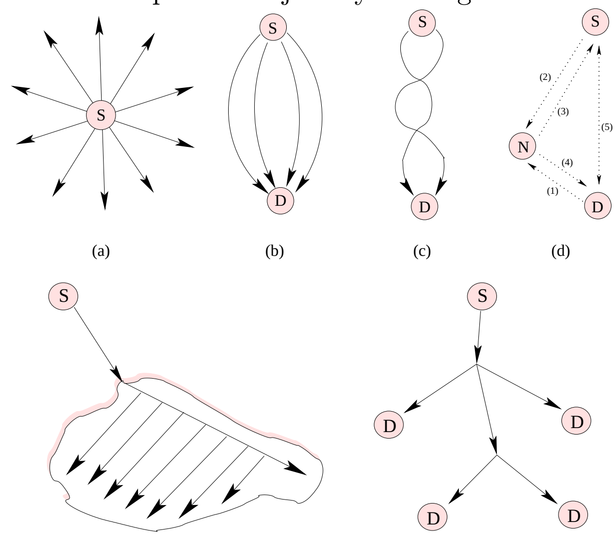
There have been significant efforts to improve routing in both fixed and mobile networks when position is available. Geographic routing[5] is a hierarchical scheme where each router is responsible for a polygonal region possibly subpartitioned into disjoint polygons assigned to other routers. This routing scheme provides an infrastructure that can be embedded in IP, and can deliver messages to specific geographic regions. Cartesian routing[4] is a greedy method that chooses a next hop that provides most progress towards the destination. DSR (Dynamic Source Routing)[3] is a form of source routing used in MANET, featuring a route discovery phase based on flooding and routes completely specified in packet headers. LAR (Location Aided Routing)[6] implements restricted area flooding in order to reduce the cost of discovery when the uncertainty about a destination is limited. It uses a phase of source based routing and a phase of controlled flooding. In a sensor network, multipath routing may be useful in providing resilience[7]. In order to use geographic or cartesian routing , node positions are necessary [1, 2], but a locations service is also necessary to translate node addresses in coordinates. GLS (Geographic Location Service)[8] implements a naming service that allows node centric applications to run on top of geographic and cartesian routing. A source can find the coordinates of the destination node from the location service and then use geographic or cartesian routing to route to that destination. TBF can be used to enhance or complement all these mechanisms, or to replace expensive energy-wise parts of them, such as flooding based discovery.

3 Trajectory based forwarding

Source based routing has the advantage that intermediate nodes are relieved of using and maintaining large forwarding tables, but it has the disadvantage of the packet overhead increasing with the path length. Cartesian routing uses positions to get rid of the routing tables, but defines one single forwarding policy: greedy, along a straight line.

TBF combines the best of the two methods: packets follow a trajectory established at the source, but each forwarding node takes a greedy decision to infer the next hop based on local position information, while the overhead of representing the trajectory does not depend on path length. In a network where

Figure 3: Examples of trajectory routing and forwarding



(a) (b) (c) (d) (e) (f)

node positions are known, the packet may be forwarded to the neighbor that is geographically closest to the desired trajectory indicated by the source node. If the destination node is known, the trajectory followed by the packet might be a line, and the method reduces to cartesian forwarding. In the general case, however, one can envision a larger array of applications (fig 2). TBF can be seen as a middle layer between position providing services such as GPS and APS[2] and many applications, such as flooding and discovery, or routing schemes, such as path resilience, unicast and multicast ad hoc routing.

3.1 Applications of TBF

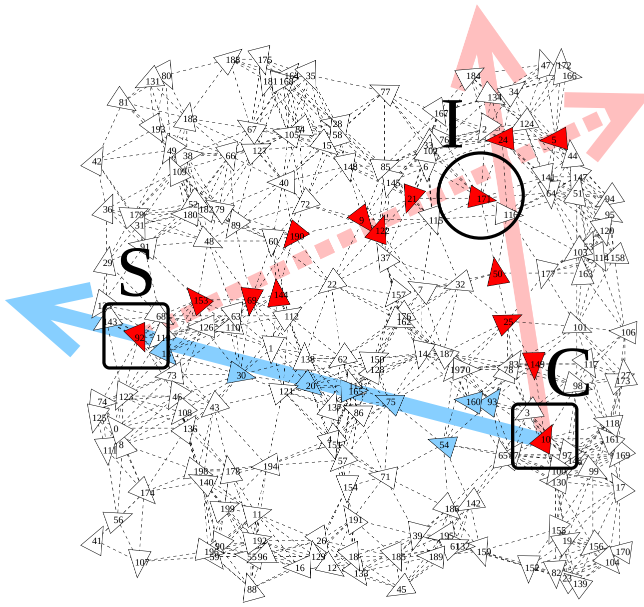
In this section, we explore some of the applications that would benefit from an implementation under TBF framework. One possible classification of these applications is based on whether they require locations disseminated through the network, or not. Flooding and discovery can work with local coordinate systems of the source nodes, while routing generally needs agreement in positions of source, destination and intermediate nodes.

Using TBF, flooding can be replaced with a number of radial outgoing lines that are reasonably close to each other to achieve a similar effect without all the communication overhead involved by receiving duplicates in classical flooding (figure 3a). More generally, a source would indicate the directions and the lengths of the lines that would achieve a satisfactory coverage (figure 3e). The coverage relies on the typical broadcast property of the wireless medium, in which several nodes overhear the packet being forwarded.

Recovery from failure often involves multipath routing from a source to a destination. In a sensor network, both disjoint (figure 3b) and braided (figure 3c) paths are useful to provide resilience[7]. Using TBF, the source may generate either disjoint paths as disjoint curves, or braided paths as two intersecting sine waves. In networks with a low duty cycle, such as sensor networks, longer alternate paths might actually be more desirable in order to increase the resilience of the transmitted messages (concept similar to Fourier decomposition). Since there is essentially no route maintenance, each packet can take a different trajectory, depending on its resilience requirements (similar to different FEC requirements).

If unicast communication is modeled by a simple curve, multicast is modeled by a tree in which each portion might be a curve. Distribution trees are used for either flooding (figure 3e), or multicast routing (figure 3f). A source knowing the area to be flooded can generate a tree describing all the lines to be followed by packets in order to achieve complete distribution with minimal broadcast communication overlap. A multicast source knowing positions for all members of a group may generate a spanning tree built of linear

Figure 4: Discovery example



trajectories to be followed by packets. There is an overhead to be paid in describing the tree in each packet, but the solution saves in route maintenance. The branching points are responsible for duplicating the packets and pruning the trees that are forwarded downstream.

Many algorithms use initial discovery phases based on flooding[3, 9] in order to find a resource or a destination. A replacement scheme using trajectories is as follows: possible destinations advertise their position along arbitrary lines and sources will replace their flooding phase with a query along another arbitrary line which will eventually intersect the desired destination's line. The intersection node then notifies the source about the angle correction needed to contact the destination directly (figures 3d and 4).

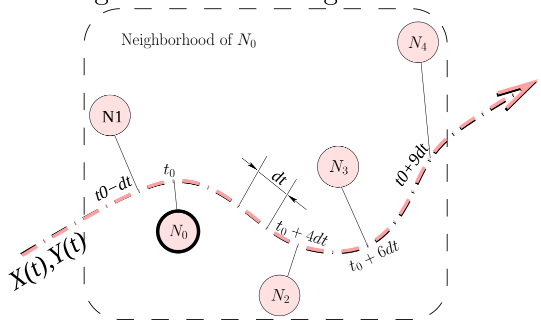
In a mobile ad hoc network, route maintenance for trajectory routing comes for free since all that is needed is the position of the destination. This is especially true when only the intermediate nodes or the source are moving, and the destination of the packet remains fixed. When the destination is moving, a location service[8] may be used, or the source may quantify its uncertainty about the destination by using a localized flooding around the destination (figure 3e).

In a network where positions are not known, positioning itself is an application that can be achieved using trajectory routing. If a master node knows its position (by GPS) it can generate a line of a known equation and forward control packets along that line. Nodes encountered can infer their own position by knowing the equation of the line in the global coordinate system, and the distance to the master node, along the line. By sweeping the control line in a radar fashion, a single master node can help all nodes in the network find their positions.

3.2 Trajectory specification and encoding

There are a number of choices in representing a trajectory: functional, equational, or a parametric representation. Functional representation (e.g. $Y = f(X)$) cannot be used to specify all types of curves (for example vertical lines). Equational representation (e.g. $X^2 + Y^2 = R^2$) requires explicit solution to deterimine the points on the curve. Parametric representation (e.g. $X = X(t)$, $Y = Y(t)$) is ideally suited for the purpose of forwarding. The parameter t of the curve is a natural metric to measure the forward progress along the path and can be linked to either length travelled on the curve, or hop count.

Figure 5: Forwarding on a curve



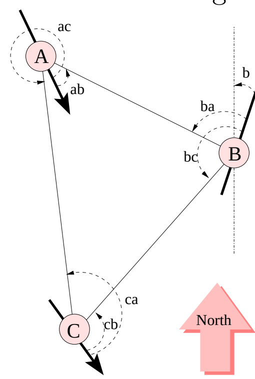
The next issue is how are the trajectories encoded. Trajectories can have several parameters and each node needs to correctly interpret the these parameters. One approach is to have a convention where the nodes know how to interpret the fields given a well know set of trajectories. Complex trajectories can have multiple components or a given trajectory can be specified as a number of simple component such as Fourier components. The more Fourier components are specified in the packet, the better the accuracy of the trajectory is. There is an intresting tradeoff between the accuracy of the curve and the overhead of specifying the components and intrepreting them. Other possibilities of encoding of the parametric curve include compiled form (ready to be executed, as in active networking), or reverse polish notation (ready to be interpreted).

3.3 Forwarding methods

The trajectory is usually decided by the source and we will assume that it expressed in parametric form $X(t), Y(t)$. The meaning of parameter t is also decided by the source, as well as the resolution at which the curve will be evaluated, dt. It is convenient, for the simplicity of the explanation to assume that t indicates the distance on the curve. The neighborhood of a node N_0 (figure 5) is defined as the portion of the curve and the nodes that are within a certain distance form N_0, shown by a dashed rectangle in the figure. In the simplest case, the neighborhood could be the smallest rectangle enclosing all N_0's one hop neighbors. In a network in which node positions are known, the main question is how to choose a next hop that best approximates the trajectory. Assume node N_0 receives a packet with the trajectory indicated by the curve $X(t), Y(t)$ and the value t_0 that corresponds to the point on the curve that is closest to N_0. Using sampling of the curve at dt spaced intervals, indicated by dots in the dashed trajectory curve in figure 5, N_0 can compute all the points of the curve that reside inside its neighborhood. For all neighbors $N_1..N_4$, their corresponding closest points on the curve are $t_0, t_0 + 4dt$, $t_0 + 6dt$, $t_0 + 9dt$. When referring to curve fitting, these values are called residuals. In fact, the mentioned method computes an estimation of the residuals, instead of the true ones, which would require either infinite resolution ($dt \to 0$), or usage of higher derivatives of $X(t)$ and $Y(t)$. Since choosing a next hop for the packet should be towards advancement on the trajectory, only portion of the curve with $t > t_0$ is considered. For this reason, node N_1 receives t_0 as the closest point, instead of $t_0 - dt$, which would be closer to the perpendicular from N_1 onto the curve. Several policies of choosing a next hop are possible:

- node closest to the curve, with the minimum residual. This policy would favor node N_2 and would tend to produce a lower deviation from the ideal trajectory;
- most advancement along the curve, choosing N_4. This policy should also be controlled by a threshold of a maximum acceptable residual, in order to limit the drifting of the achieved trajectory. It would produce paths with fewer hops than the previous policy, but with higher deviation from the ideal trajectory;

Figure 6: Nodes capabilities: measuring angles, ranges, orientation



- centroid of the feasible set, favoring N_3: the centroid is a way to uniformly designate clusters along the trajectory, and a quick way to determine the state of the network;
- randomly choose between best three: useful when node positions are imperfect, or when it may be necessary to route around obstacles;
- in mobile networks a forwarding policy that might provide better results would be to choose the next hop which promises to advance along the trajectory, or one that is expected to have the least mobility in the future.

4 Local Positioning System (LPS)

Trajectory routing is simpler when all nodes positions are known relative to a reference coordinate system by use of a global capability such as GPS, or running a self positioning algorithm such as [2, 10, 11, 12]. In this paper, we will show that TBF can be implemented even when such a global or ad hoc capability is not available due to place of deployment (e.g. indoors), high communication costs, or additional infrastructure requirements [1, 13, 11].

We extend an idea proposed in [14], and develop a method for nodes to use some capabilities (ranging, AOA, compasses) to establish local coordinate systems in which all immediate neighbors are placed. It is then possible to register all these coordinate systems with the coordinate system of the source of the packet. Local Positioning System (LPS) is a method to achieve positioning **only for the nodes along the trajectory**, with no increase in communication cost, as if all node positions were known. Instead, each node touched by trajectory will spend some computation to position itself in the coordinate system of the source of the packet, trading off some accuracy of the trajectory.

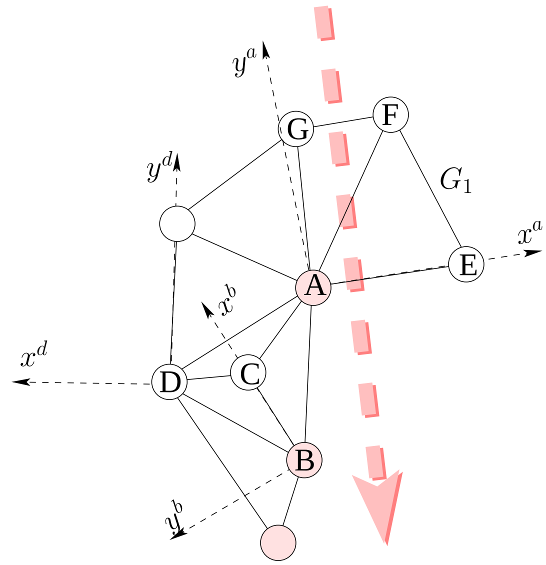
4.1 Node capabilities

The network is a large scale, dense, ad hoc set of nodes such that any any node can only communicate directly with its immediate neighboring nodes within radio range. In the ideal case, when radio coverage of a node is circular, these networks are modeled as fixed radius random graphs, or unit graphs.

Algorithms for trajectory based forwarding require that each node knows its position relative to a reference coordinate system. If positioning is not available, or it is not practical to deploy a GPS receiver in every node, it is still possible to implement trajectory based forwarding by making use of the nodes' capability of positioning relative to each other. We present several methods that would enable trajectory based forwarding, based on relative positioning that use ranging and angle of arrival (AOA) measurements.

Range based estimation of distance between two nodes has been previously used for positioning purposes [2, 10, 12] even with the high measurement error involved. In figure 6, node A would have estimations

Figure 7: Local coordinate systems



of distances to its neighbors AB and AC, but could also find BC after communicating with B or C. In most implementations, ranging is achieved either by using an estimate from the strength of the signal (unreliable) or using time difference of arrival (TDOA).

In AOA approach, each node in the network is assumed to have one main axis against which all angles are reported and the capacity to estimate with a given precision the direction from which a neighbor is sending data. After the deployment, the axis of the node has an arbitrary, unknown heading, represented in figure 6 by a thick black arrow. When interacting with two neighbors, a node can find out the angle between its own axis and the direction the signal comes from. Node A "sees" its neighbors at angles $\widehat{ac}$ and $\widehat{ab}$, and has the possibility of inferring one angle of the triangle, $\widehat{CAB} = \widehat{ac}$-$\widehat{ab}$. For consistency all angles are assumed to be measured in trigonometric direction. AOA capability is usually achieved by using an antenna array, which might be prohibitive in size and power consumption. A small form factor node that satisfies conditions outlined has been developed at MIT by the Cricket Compass project[15]. Its principle of operation is based on both time difference of arrival (TDOA) and phase difference of arrival. Other node capabilities that might be available in small form factors include accelerometers and compasses. The accelerometer's main use is to indicate pose, while compass indicates the absolute orientation of each node.

4.2 Local coordinate systems

In figure 7, if node A is able to measure distances to all its neighbors, via a ranging method, it can compute the sides and the angles for all triangles created with all pairs of neighbors which are also neighbors of each other. This would enable A to place itself in $0, 0$ of its local coordinate system and all its immediate neighbors at positions that satisfy all the range requirements known to A, ranges indicated by continuous lines in the figure. To establish its coordinate system, A randomly chooses E as an indicator for its x^a axis and F as an indicator for its y^a axis. All the other neighbors are then placed in this coordinate system such that all the range readings are respected. Most nodes can be unambiguously positioned if they are neighbors with two previously positioned nodes. Node G for example is positioned immediately after E and F are positioned by choosing between two possibilities. It can be placed where it is on the figure, or on the position of G_1. But if G were G_1, it would have been close enough to E to have a direct link, so this possibility is eliminated. In the same fashion, all nodes around $A(0, 0)$ are positioned. Node B sets up a similar local coordinate system by initiating its axes using neighbors C and D. Since a condition of global connectivity in a random ad hoc network is for the average degree to be at least 6 [16], most nodes will succeed in placing all their neighbors in their local coordinate systems.

The registration between two local coordinate systems is the process that computes a transformation which will transform any point from one coordinate system to the other. The input to this process are points for which the coordinates are known in both coordinate systems with some accuracy. If perfect ranging were used in creating the local coordinate systems, the registration would produce a rigid transformation. In the example in figure 7, a translation and a rotation are enough to overlap the local system of B over the local system of A. In the case of D, however, the first two neighbors randomly chosen as axes indicators produce a coordinate system that cannot be overapped over A's using only translation and rotation. In this case, due to $D's$ original, localized and independent choice of axes, a mirroring transformation will also be produced by registration. The main cause for these mirroring transformations is the fact that ranging information does not provide a sense of direction (at least in the first phase, before the local coordinate systems are set).

Using AOA, node A knows all angles to neighbors reported against its main axis (see section 4.1), and therefore has already an established set of axes. In triangle AFE for example, A knows all angles but no sides. Assuming that the range $AE = 1$, it can then find the sizes AF and FE which can then be propagated to all other triangles in order to get coordinates for all points. This coordinate system can be registered with B's using a scale transformation (in addition to translation and rotation), because B will probably choose a different edge as a unit for its local coordinate system. If a compass is available in each node, all the reference systems have parallel axes, and the rotation is eliminated from the registration. A mirroring transformation will occur when using AOA only in the case in which a node is deployed upside down.

It is possible to use both AOA and ranging in creating local coordinate systems, possibly enhanced with local compasses. The following table indicates all the possible combinations of node capabilities, and the transformations involved in the registration process (T=translation, R=rotation, S=scaling, M=mirroring).

Capability	Transformations
Range	T, R, M
AOA	T, R, S, (M)
AOA+Compass	T, S, (M)
AOA+Range	T, R, (M)
AOA+Range+Compass	T, (M)

When mirroring is indicated in parenthesis, it can only happen as a result of a node being deployed upside down, not from the randomness in starting the local coordinate system. When only ranging is used, mirroring is possible regardless of the pose of the node, depending on the nodes chosen as indicators for local axes. In all the other cases, since AOA is assumed to report angles in the same (trigonometric) direction for all nodes, mirroring between two local coordinate system appears only when one node is flipped, situation which can be robustly detected by a digital accelerometer. The general transformation matrix is

$$H = \begin{bmatrix} sr_1 & sr_2 & t_x \\ sr_3 & sr_4 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $R = \{r_i\}$ is the rotation transformation and possibly mirroring, s the scaling factor, $\{t_i\}$ the translation. R is an orthonormal matrix with the following properties:

$$\begin{cases} |r_1| = |r_4| \\ |r_2| = |r_3| \\ |R| = -1\, for\, mirroring,\, 1\, otherwise \end{cases}$$

If a transformation is not present, its parameters will be set to neutral values to preserve the properties of H. For example, when ranging is available and no scaling is involved, $s = 1$, when compases are available and there is no rotation, $R = I_2$ (the identity matrix).

If for example, $v_C^a = \begin{bmatrix} x_C^a & y_C^a & 1 \end{bmatrix}^T$ designates the coordinate vector of C in the coordinate system of A, and v_C^b in coordinate system of B, then

$$v_C^a = H v_C^b \tag{1}$$

4.3 Registration

If two nodes, such as A and B in figure 7 wish to agree their coordinate systems, they must implement a registration procedure. They reach this agreement by using points that are common in the two coordinate systems, in our example A, B, C and D. In practice, since ranging does not provide perfect distances and AOA perfect angles, the two coordinate systems will not overlap in all points if we restrict H to the type of transformation described above. The error in registration is defined as the sum of squared distances between corresponding points after registration. If B has the two sets of coordinates of nodes A, B, C and D in both coordinate systems, it computes the transformation H that minimizes the error:

$$\sum \left\| v_i^a - H v_i^b \right\|^2,\ i = A, B, C, D \tag{2}$$

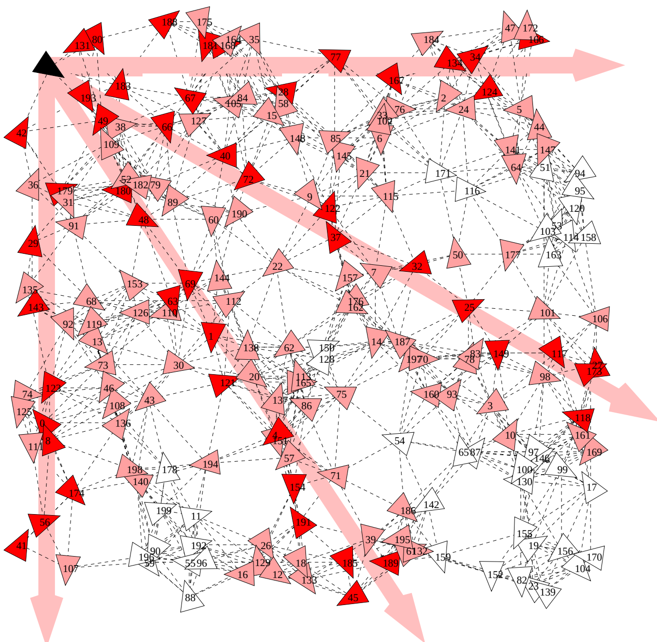
Fortunately, there is a closed form solution to this optimization, described in [17], that runs in a time linear in the number of common points. A short summary of the method is presented in appendix A.

4.4 Forwarding using LPS

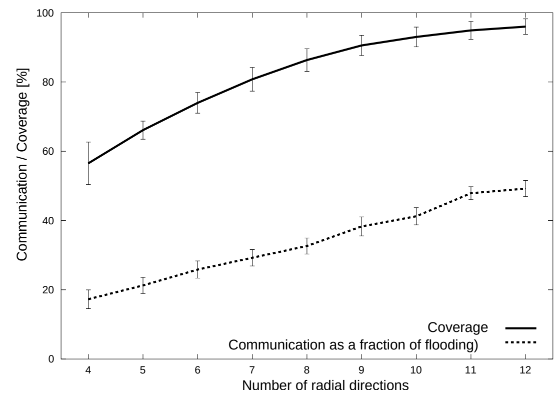
The aim for LPS is to make forwarding along the trajectory similar to the procedure followed in a network where node positions are available. The key idea is that the only nodes that are positioned are the ones involved in forwarding along the trajectory. Positioning is done in a hop by hop fashion, in the coordinate system chosen by the initiating node - the source of the packet. The forwarding procedure works with a node selecting the next hop based on the proximity to the desired trajectory, or any of the other possible policies. In figure 7, the ideal trajectory is shown as a thick dashed arrow. Assume that A knows the equation of the trajectory in its own coordinate system, which has been already registered to the coordinate system of the packet. If the next node to be selected along the trajectory is B, it will receive from A v_i^a, so that B, using v_i^b, can register its own coordinate system to A's by solving the optimization problem (2). Once H is obtained at B, all the neighbors of B are evaluated in the coordinate system of the source, by using transformation (1). Node B is then able to select one of its own neighbors that is closer to the trajectory, in order to continue the process.

What is in fact achieved by LPS is the registration of all coordinate systems of visited nodes to the coordinate system of the initiating node, which achieves positioning of all these nodes in the coordinate system of the source. This positioning system has a number of advantages: first, it is localized to the nodes actually touched by the trajectory. Unlike a network wide positioning algorithm, such as [2, 10], which involves collaboration and coordination of a large number of nodes, LPS involves only the nodes "touched" by the desired trajectory. Second, the size of packet only depends on the number of common neighbors two consecutive nodes have, and this is upper bounded by the maximum degree of the nodes. In the forwarding packet, the equation of the trajectory is also included, but this is a fixed size, so the packet does not increase with the length of the trajectory. Third, LPS can make use of any of the two previously used localization capabilities - AOA or range estimation, possibly enhanced with local compasses.

Figure 8: Trajectory based flooding



(a) Approximating flooding with spokes



(b) Probability of reaching the destination

5 Simulation

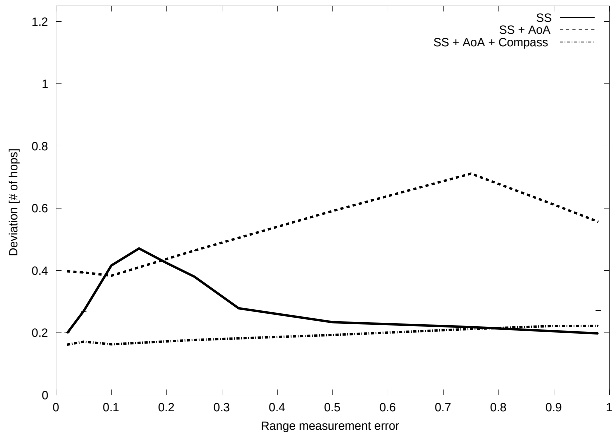
We simulated an isotropic[1] map (average degree=9.5), with 200 nodes each having a random, but unknown heading, depicted in figure 1. This is the case in which ranging/angular measurements are error free, so the forwarding follows the path that best approximates the desired trajectory. When using perfect measurements, LPS would obtain perfect positions, thus generating as good as an approximation to the ideal trajectory as it would be possible using GPS enabled nodes. The trajectory is initiated by a node on the left of the map (106) and is encoded parametrically as a sine wave in the approximate direction of node 143, on the other side of the map. The dark shaded nodes are the ones directly involved in the forwarding process, while the light shaded ones are nodes in the wireless range of the trajectory. The white colored nodes never receive or overhear the message. For the latter ones, positions are also computed, as part of LPS, and they overhear the packet being forwarded, even if they are not selected to continue forwarding.

In another simulation, trajectory routing was used as a replacement for flooding. In the same network, a node starts radial trajectories in order to distribute a packet to as many node as possible. In figure 8a, a node in the corner sends 12 radial trajectories, only 4 of which are actually propagated in the map. In order to evaluate the possibility of replacing flooding with radial trajectories we counted the ratio of nodes which receive the message and the amount of communication used. The measure of energy used is one unit for transmission and one unit for reception of the message. In figure 8b, coverage represents the percentage of nodes which receive the message, and communication is shown as a percentage of the communication used by flooding. On the horizontal axis, we vary the number of radials used by the originating node. Each point also shows the standard deviation after 1000 runs with random starting angles. While the communication is reduced by more than half, trajectory based flooding can achieve almost full coverage. The performance graph used an originating point in the middle of the map (157).

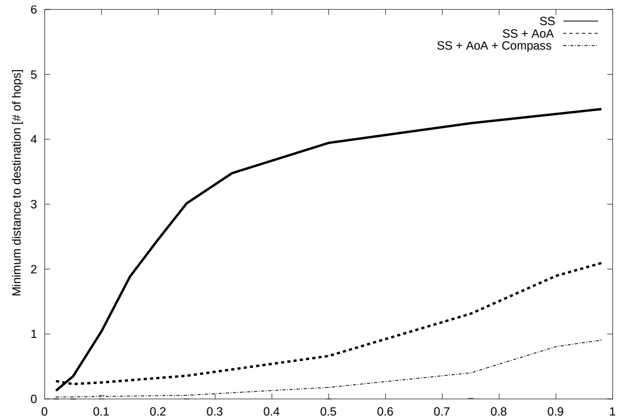
In order to evaluate TBF for the purposes of forwarding, routing, and discovery we tested various linear

[1] *isotropic* = having the same physical properties in all directions (connectivity, density, node degree)

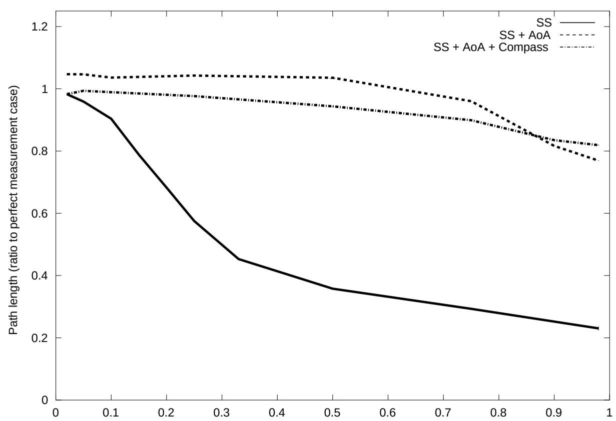
Figure 9: Range, Range + AOA, Range + AOA + Compass



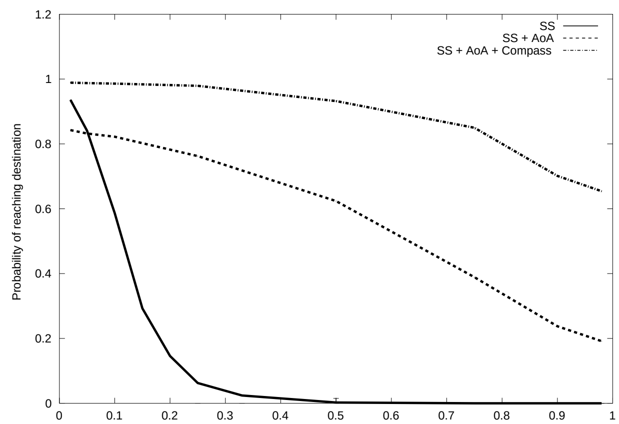
(a) Deviation from ’perfect’ trajectory



(b) Minimum distance to destination



(c) Path length - relative to ’perfect’ trajectory



(d) Probability of reaching the destination

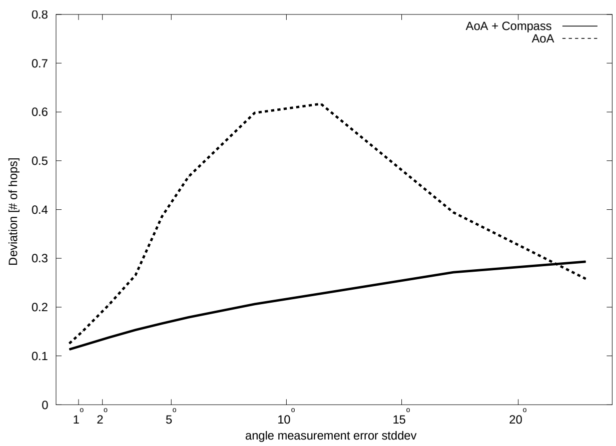
trajectories in the same 200 node network, and used four metrics to compare the performance of forwarding in the presence of ranging errors and angular errors. For each of 30 pairs of source-destination nodes situated near the sides of the map (North-South and East-West), we compared the trajectory obtained when ranging or angle measurement is imperfect with the trajectory obtained when these measurement are perfect. In the latter case all nodes register to their true positions, as would be obtained from GPS. Therefore, the “perfect trajectory” is the one obtained with perfect locations, which is still different from the ideal trajectory, as described by the parametric curves.

- **Deviation** of a trajectory is computed as the mean distance between a point on the obtained trajectory and the closest point on the perfect trajectory. The value of this deviation is then normalized to the maximum hop size to be expressed as a hop count.

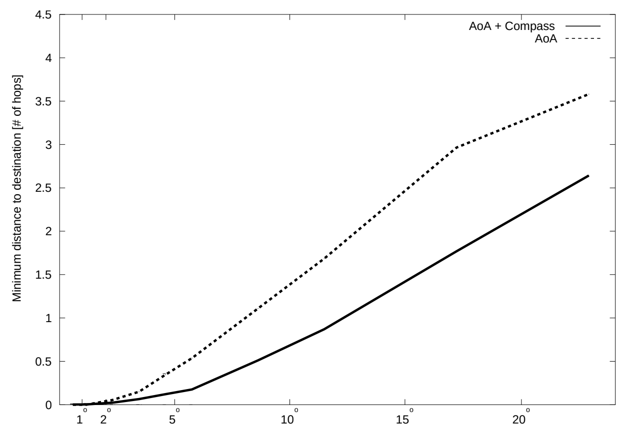
- **Path length** is the number of hops achieved by a trajectory, relative to the perfect trajectory. The forwarding process may be stopped when the trajectory exits the map, or when packets cannot be forwarded anymore due to local aberrations in registration, obstacles, faulty or inexistent local coordinate systems.

- **Minimum distance to destination** shows how close a trajectory gets to the desired destination.

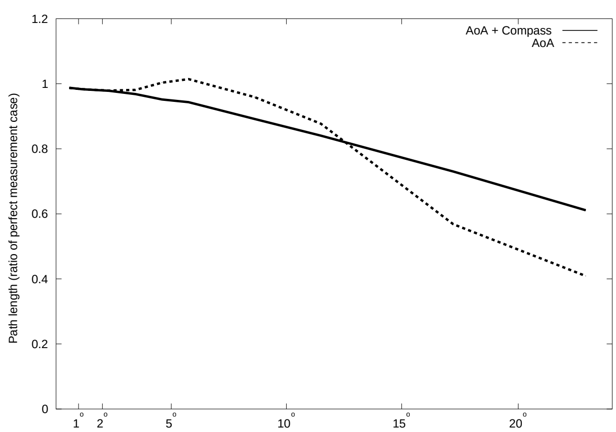
Figure 10: AOA, AOA + Compass



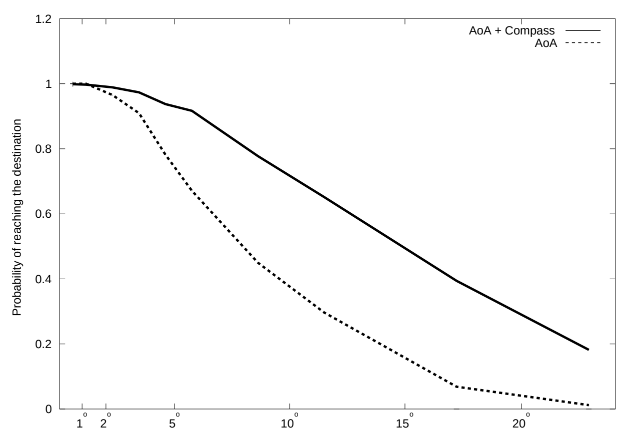
(a) Deviation from 'perfect' trajectory



(b) Minimum distance to destination



(c) Path length - relative to perfect trajectory



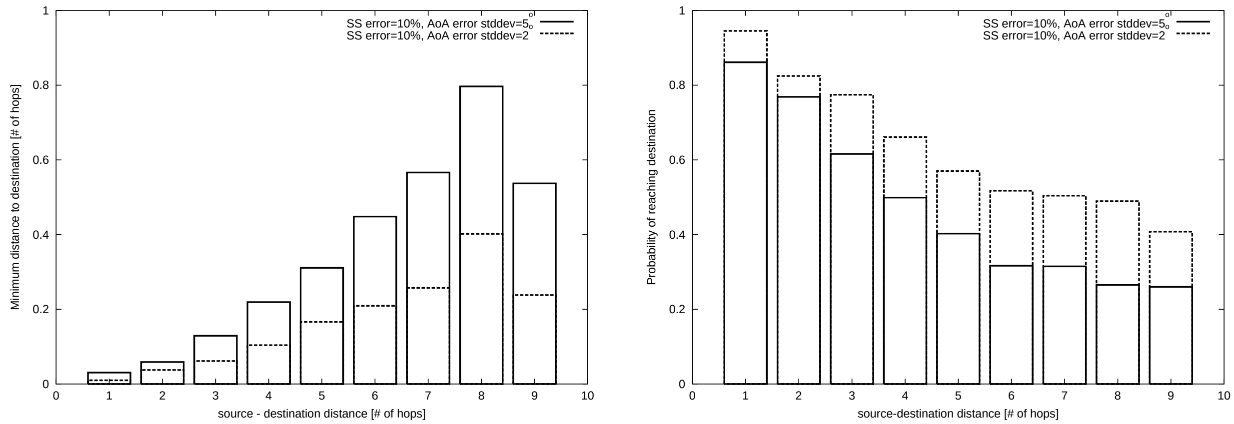
(d) Probability of reaching the destination

- **Probability of reaching destination** expresses the chances of the trajectory passing within one hop of the desired target. In this set of experiments, when the trajectory reaches the destination, or an immediate neighbor of it, the forwarding is stopped.

For the angle of arrival, we model normally distributed errors whose standard deviation is shown on the horizontal axis. Corresponding to an error of $\pm 5^o$, achieved by the Cricket compass project corresponds a deviation of 2.5^o on our graphs. For ranging measurement, we assume a linear error model where the actual measurement is is within a fixed fraction of the true range (0-100%).

Simulations are divided in two groups, based on the node capabilities used by LPS. The first group (figure 9) compares ranging (labeled as 'SS') with ranging and AOA, possibly enhanced with a compass. The AOA deviation is fixed in these experiments to 0.15 (about 8.59^o). The experiments reveal that ranging is less robust in forwarding the coordinate system, being greatly enhanced by the presence of angular measurements, even with high errors. The reason for this difference stems from the way local coordinate systems are built. Using ranging with errors, the coordinate systems are prone to false mirroring, large aberrations produced by working with small or obtuse angles, bias depending on the nodes initializing the coordinate system. We experimented with various thresholds to limit the shape of triangles resulted from ranging - eliminating very obtuse or very acute angles from being used in inferring local coordinates. Such

Figure 11: Errors compound with distance



(a) Minimum distance to destination

(b) Probability of reaching the destination

thresholds provide a tradeoff between the amount of possible misleading information that is accepted in the registration and the number of nodes which successfully get a local coordinate system. For this set of experiments, we used an angular threshold of 0.6 ($\sim 36^o$), thus eliminating from the inference the triangles with smaller angles, at the price of dropping around 3% of the nodes which weren't able to establish local coordinate systems. When AOA is available, the local coordinate system building relies on the inherent order in which angles are reported, completely eliminating false mirroring, and reducing range aberrations. Range based measurement shows a maximum in the deviation (figure 9a), explained by the sharp decrease in the path length (figure 9b) following the maximum, which produces shorter paths with a relatively good start.

The second group of experiments (figure 10) studies the behavior of angular measurements, possibly enhanced with a compass, in the absence of ranging. On the horizontal axis of all these graphs, the standard deviation of the angular measurement error is indicated. The deviation when using simple AOA (figure 10a) shows a decreasing trend that is caused by the paths getting shorter with increasing errors, but still having a good start, which yields a low deviation. The same type of behavior was encountered when a single type of sensing was employed (ranging), and is a symptom indicating short paths. For the region where the deviation is high, path length can be even longer than the perfect path, because, once it misses the destination, it can wander for 1-2 more hops before it reaches the edge of the map. Both groups of experiments show that a compass in each node brings a substantial improvement in all metrics. One of the main causes for this improvement is that rotation is eliminated from the registration process ($R = I_2$).

These two sets of simulations showed that multimodal sensing (AOA + ranging) performs much better than its components taken separately, under the same error conditions. A compass attached to each node brings significant improvement in all the metrics considered.

In order to evaluate the amount of degradation that occurs in the iterative registration process, we conducted a set of experiments in which source and destination nodes are chosen randomly on the map, with uniform probabilities for X and Y coordinates. In figures below, minimum distance and probability of reaching the destination are plotted as a function of the initial distance between source and destination, normalized by the maximum hop size. The local positioning method employed ranging with an error of 10% and AOA measurements with deviations of 2^o and 5^o. As expected (figure 11), the error is compounding with distance yielding decreasing probabilities of reaching the destination and increasing distances to destination. An interesting side effect is that for far apart source - destination pairs, which will usually be placed on the opposite sides of the map, the minimum distance between the trajectory and destination decreases, as the trajectory may shortly wander left or right, searching for a possible continuation of the map, and thus reducing the distance to the destination.

The main conclusions revealed by the simulations are that, although error compounds with distance, it can be countered efficiently by using additional hardware available in small form factors. An accelerometer present in each node could detect node flipping, eliminating false mirroring, while a digital compass could eliminate rotations from registrations process, when angular measurements are used. With AOA error range in the range $2^o - 5^o$, which has already been realized by other projects, TBF maintains a low deviation, making it usable for discovery purposes, and reaches the destination with very high probability, making it usable for routing. When used as a replacement for flooding, it can achieve high coverage, with only a fraction of the communication spent by classical flooding.

6 Research issues

Several issues related to determining, specifying and modifying the trajectory are still under consideration. While any trajectory in the plane can theoretically be expressed as a parametric curve, many applications, such as discovery, broadcast or multicast may require description of large trees, or of large parametric expressions. Here we intend to explore efficient representation methods for tree representation, such as fractals, or other recursive methods. Concepts from active networking might also be viable in sending a function in a compact compiled form. Modification of the trajectory might be decided by intermediate nodes in order to impose local detours, without involving all the upstream or downstream nodes. Trajectory might also be modified globally if better information about the destination becomes available.

In many cases, the trajectory is not available in parametric form at the source, but as a list of points. Depending on the requirements of the application and the pattern described by the curve, the possibilities for curve encoding range from an anchor list of all points (which will amount to a form of source based routing, with cartesian forwarding between intermediate points), to a curve fitting using a function base (Fourier or polynomial). Curve fitting is attractive in cases where the number of anchors is large and the requirement of touching all anchors is not strict.

Although TBF is characterized as a generalization of source based routing, it is not clear who should specify the actual trajectory. For example, if a sink is collecting data from several sources, it can specify trajectories that would achieve optimal aggregation and minimum overall communication, based on information which is not available at each individual source. Also, when a destination has more information about the network, it might provide better routing around obstacles, or a single route that would serve multiple close destinations.

A question that arises in forwarding is how to recover from dead ends. This happens when obstacles in the graph are non monotonous along the direction of the trajectory. [18] proposes the use of the right hand rule to route around obstacles. What also could be done, using TBF, is to allow a limited backtrack when advancement is not possible, trading away delivery guarantees for reduced complexity. That would mean that if a node cannot forward a packet anymore, the packet would go in reverse gear for just a couple of hops in order to get around the obstacle.

TBF can be used as a low level primitive in implementing many network protocols in ad hoc networks. In static networks it can be used for end to end routing, either coupled with a discovery phase, or with a location service. In mobile networks, TBF is imune to source mobility and intermediate node mobility, since the trajectory does not explicitly encode intermediate members of the path. The research issue remains the mobility of the destination, which may require a solution based on LAR[6] and TBF.

7 Conclusions

We presented Trajectory Based Forwarding (TBF), a method to forward packets in an ad hoc network where certain capabilities (GPS, AOA, ranging, compass) are present. The method is a generalization between source based routing and cartesian forwarding in that the trajectory is set by the source, but the forwarding decision is local and greedy. When GPS is not available, TBF uses Local Positioning System (LPS), a positioning method that is localized, requiring only the collaboration of nodes along the trajectory, which does not incur the overhead of a network wide positioning algorithm. The main applications of TBF include routing - unicast, multicast, multipath, flooding, discovery, and positioning.

References

[1] B. Parkinson et. al. *Global Positioning System: Theory and Application*, volume I. Progress in Astronautics and Aeronautics, Volume 163, 1996.

[2] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM, San Antonio*, November 2001.

[3] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[4] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks, March 1987. University of Southern California.

[5] Julio C. Navas and Tomasz Imielinski. Geographic addressing and routing. In *MOBICOM'97, Budapest, Hungary*. ACM, September 26-30 1997.

[6] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *MOBICOM'98*, October 1998.

[7] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly resilient, energy efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review (MC2R)*, 1., No. 2, 2002.

[8] J. Li et. al. A scalable location service for geographic ad hoc routing. In *ACM MOBICOM, Boston, MA*, August 2000.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *ACM MOBICOM, Boston MA*, August 2000.

[10] A. Savvides, C.-C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM MOBICOM, Rome, Italy*, July 2001.

[11] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *6th ACM MOBICOM, Boston, MA*, August 2000.

[12] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building rf-based user location and tracking system. In *IEEE INFOCOM 2000, Tel-Aviv, Israel*, March 2000.

[13] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine, Special Issue on Smart Spaces and Environments*, October 2000.

[14] S. Capkun, M. Hamdi, and J.P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Hawaii International Conference On System Sciences, HICSS-34*, January 3-6 2001. Outrigger Wailea Resort.

[15] N.B Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *6th ACM MOBICOM, Rome, Italy*, July 2001.

[16] B. Krishnamachari, S.B. Wicker, and R. Bejar. Phase transition phenomena in wireless ad-hoc networks. In *GLOBECOM, San Antonio*, November 2001.

[17] B.K.P. Horn, H. M. Hilden, and S. Negahdaripour. Closed form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society*, 5, No. 7, July 1988.

[18] B. Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *6th ACM MOBICOM, Boston, MA*, August 2000.

A Closed Form Solution for Registration

Given the coordinates of two sets of n points in two reference systems – v_i^a in system A and v_i^b in system B, $1 \leq i \leq n$, the problem is to find the transformation H minimizing the error:

$$\sum_{i=1}^{n} \left\| v_i^a - H v_i^b \right\|^2 \tag{3}$$

The general transformation matrix should have the form

$$H = \begin{bmatrix} sr_1 & sr_2 & t_x \\ sr_3 & sr_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $R = \{r_i\}$ is the rotation transformation and possibly mirroring, s the scaling factor, t the translation. R is an orthonormal matrix. Since the two sets of points might not overlap perfectly when restricting H's shape to (4), relation (3) in fact minimizes the distance between the points in set A and the images of the corresponding points from B through transformation H. The centroids of the two sets are defined as

$$r^a = \frac{1}{n}\sum_{i=1}^{n} v_i^a \; and \; r^b = \frac{1}{n}\sum_{i=1}^{n} v_i^b$$

Both sets of points are translated in their centroids:

$$\bar{v}_i^a = v_i^a - r^a \, and \, \bar{v}_i^b = v_i^b - r^b$$

The error term to be minimized in (3) can be written as

$$e = v^a - sR(v^b) - t = \bar{v}^a - sR(\bar{v}^b) - \bar{t}$$

where

$$\bar{t} = t - r^a + sR(r^b)$$

therefore the square error becomes

$$\sum_{i=1}^{n} \left\| \bar{v}_i^a - sR(\bar{v}_i^b) - \bar{t} \right\|^2 \tag{5}$$

which is minimized with $\bar{t} = 0$, or

$$t = r^a - sR(r^b) \tag{6}$$

This formulation of the error term leads to an asymmetry in determination of the scale factor - the "optimal" transformation from A to B is not the exact inverse of the "optimal" transformation from B to A. Adopting a symmetrical expression for the error term

$$e = \frac{1}{\sqrt{s}}\bar{v}^a - \sqrt{s}R(\bar{v}^b)$$

allows finding the scaling transformation without the need to find rotation as

$$s = \sqrt{\frac{\sum_{i=1}^{n} \|\bar{v}_i^a\|^2}{\sum_{i=1}^{n} \|\bar{v}_i^b\|^2}} \tag{7}$$

For rotation, we have to find a matrix R that maximizes the scalar product between points in set A and the rotated points from B:

$$\sum_{i=1}^{n} (\bar{v}_i^a)^T R \bar{v}_i^b$$

since $a^T Rb = Trace(R^T ab^T)$ the expression to maximize is

$$Trace(R^T \sum_{i=1}^{n} \bar{v}_i^a (\bar{v}_i^b)^T = Trace(R^T M)$$

where

$$M = \begin{pmatrix} \sum_{i=1}^{n} \bar{x}_i^a \bar{x}_i^b & \sum_{i=1}^{n} \bar{x}_i^a \bar{y}_i^b \\ \sum_{i=1}^{n} \bar{y}_i^a \bar{x}_i^b & \sum_{i=1}^{n} \bar{y}_i^a \bar{y}_i^b \end{pmatrix} \tag{8}$$

M is orthonormal, and therefore it can be decomposed in $M = US$, where U is orthonormal and S is positive semi-definite (when M is nonsingular). The matrix S is always uniquely determined $S = (M^T M)^{\frac{1}{2}}$ and

$$U = M(M^T M)^{-\frac{1}{2}} \tag{9}$$

It turns out that $Trace(R^T M)$ is maximized when $R = U$, therefore computing U will yield the desired rotation transformation (power $-\frac{1}{2}$ of a matrix can easily be computed using the eigenvalues and the eigenvectors).

In order to compute the transformation matrix H, it is necessary to use relation (7) to find s, (8) and (9) to find the rotation R, and finally, the translation is given by (6). Since we are looking at the 2D case, computing the eigenvalues and eigenvectors does not take a considerable amount of time, therefore the main complexity factor comes from computing the centroids and the matrix M, which provide a linear running time for the entire registration.