Competitive Time and Traffic Analysis of Position-based Routing using a Cell Structure

Stefan Rührup and Christian Schindelhauer

2006
Competitive Traffic and Time Analysis of Position Based Routing using a Cell Structure

Stefan Rührup∗ Christian Schindelhauer

Abstract

We present a strategy for organizing the network communication in wireless ad hoc networks based on a cell structure. We use the unit disk graph model and assume positioning capabilities for all nodes. The cell structure is an abstract view on the network and represents regions where nodes reside (node cells), regions that can be used for the communication flow (link cells) and regions that cannot be bridged due to the restricted transmission range (barrier cells). Each node can establish a cell classification of its neighborhood based on the position data which is announced by all nodes.

The cell structure has two advantages for applying position-based routing: It helps to determine local minima for greedy forwarding and improves recovery from such minima, because in recovery mode all edges can be used in contrast to other topology-based rules that can be applied only on a planar subgraph.

For the analysis of position-based routing algorithms the measures for time and traffic can be defined based on the cell structure. The size of the barriers (measured by the number of cells in the perimeters) provides a measure for the difficulty of exploring the network, which is unknown to the routing algorithm. Exploration can be done in parallel, but with increasing traffic consumption. We propose a comparative measure to assess both time and traffic, the combined comparative ratio, which is the maximum of the ratio of routing time and optimal time and the ratio of traffic consumption and the minimum exploration costs. While either flooding or common single-message strategies have a linear ratio, we present a simple algorithm that has a sub-linear combined comparative ratio of \(O(\sqrt{h})\), where \(h\) is the minimal hop distance between source and target.

1 Introduction

For position-based routing nodes are identified by their unique positions. The task is to deliver a message from a source node to a target node identified by its position in an unknown wireless ad hoc network. We try to optimize the number of messages and the time to perform this task in a worst case setting. As communication model we use the unit disk graph model.

One obstacle for efficient position based routing is the lack of knowledge about the network structure available at the beginning. In particular, reactive routing protocols that do not know any network structure in advance fail to solve this problem efficiently. We show that in the unit disk graph model even for a constant hop distance between source and target all nodes of the network need to send at least one message. Yet, if some local pro-active communication is allowed to structure the basic network, the situation becomes much easier.

It is a common technique to build special subgraphs like Delaunay triangulations to simplify communication [GGH+01]. Other approaches rely on planarized graphs where no communication links may cross each other [JT92, ALW+03]. These graph topologies can be built by some local communication protocol without much overhead. In this paper we strive for the simplest topology possible which establishes a grid cell structure and show that for appropriate choice of the cell size routing is as efficient as in the unit disk graph with only a constant factor loss in performance.

∗Institute of Computer Science. {sr, schindel}@uni-paderborn.de. Partially supported by the DFG-Sonderforschungsbereich 376 and by the EU within 6th Framework Programme under contract 001907 “Dynamically Evolving, Large Scale Information Systems” (DELIS).
Figure 1: Unit disk graph with transmission radius $r$ (left) and grid subdivision (right) of a node set.

Such topologies are not built on the fly during the routing, yet built in advance by providing some local information to neighboring nodes. We formalize this pre-structuring of the network. If no communication is done without a demand, i.e. a message to be routed through the network, we call the corresponding routing reactive. Then, a topology discovery has to be done during the routing. Here, we show that in a worst case setting this additional overhead cannot be neglected. If in advance the complete network structure can be exchanged between all reachable participants of the network in advance, we call this communication pattern pro-active.

All these routing approaches have their benefits. In this article we try to keep proactive communication at a minimum. For this, we measure the impact of information by the number of hops $k$ it has to be transmitted, denoted as $k$-hop proactive protocols. We establish our basic communication network by a 2-hop-proactive communication pattern providing the coordinates of the neighbors and the coordinates of the neighbors’ neighbors at each node.

One of the main ideas in this paper is to use the position information to define a global grid. The length of a grid square $\ell$ is a constant fraction of the transmission range $r$. We will call such grid squares cells. Each mobile host uses its positioning capabilities to determine to which cell it belongs. It can also assign its neighbors to the cells within its transmission range, based on their position coordinates. Thus, we can abstract from the view of a mobile host’s environment by classifying the cells. In Section 4 we will see how this classification leads to a formal description of barriers.

This cell structure is a tool for describing efficient algorithms, not merely an end in itself. We can show that one loses only (small) constant factors in resources if one uses this cell structure. Furthermore, we show how using local information this cell structure can be established. Basically we have to distinguish only between three types of cells: node cells, link cells and barrier cells. This cell structure gives an abstract view on the network and represents regions where nodes reside (node cells), regions that can be used for the communication flow (link cells) and regions that cannot be bridged due to the restricted transmission range (barrier cells). The cell sizes are chosen such that locality, i.e. two nodes are located in vertically or horizontally neighbored cells, ensures communication. Under this assumption we can show that the cell path using node cells and link cells is equivalent to the optimum path connecting two nodes up to a constant factor of performance factor.

This leads to our second main contribution. The complexity of the network scenario can be described by the sum of perimeters of barrier cells. If this barrier perimeter size is small then routing is easy, which means it can be achieved using greedy algorithms with a recovery strategy in the case a small barrier is hit. If this barrier size is large, then flooding is the best possible algorithm.

We formulate this problem as an online routing problem where we measure the messages as well as the time needed to deliver the message. As complexity measures we consider time and traffic for delivering the message from source to target cell. Time is the number of rounds (measured in cell hops) until the message reaches the destination if the node is accessible. Traffic is the number of messages sent between cells. We investigate the time under a competitive measure, i.e. we compare the time needed by the message delivery algorithm with the length $h$ of the shortest cell path using only node and link cells between source and target. This so-called competitive analysis is well known in the field of online algorithms [BE98]. For the traffic we define a comparative measure, which is the number of messages divided by the perimeter $p$ of all barriers formed by barrier cells plus the minimum hop-distance $h$. This perimeter size denotes the length of all right-hand traversal paths surrounding all sets of barrier nodes. For some instances this perimeter size gives a lower bound on the traffic.
There are single message strategies that can route a message in such a network with optimum comparative traffic ratio of 1. However, for such strategies the competitive time ratio is up to $\Omega(h)$. The best time behavior is provided by broadcasting a messages, aka. as flooding. This method has optimum competitive time ratio 1, yet an arbitrarily bad comparative traffic ratio. One can improve this method by the expanded ring search scheme, giving a competitive time ratio of $O(1)$, yet the competitive traffic ratio of $\Omega(h)$ for some instances remains rather high. A detailed description of these ratios is given in Subsection 5.2.

At a first look it seems that for routing in an unknown environment there is a trade-off between time and traffic with respect to the comparative ratio. In this paper we focus on this question and introduce the combined comparative ratio which is the maximum of the competitive time ratio and the comparative traffic ratio. Note that all position based routing algorithms for ad hoc networks considered so far have a combined competitive ratio of $\Omega(h)$. In this paper we present as our third main contribution a simple online routing algorithm with a sub-linear combined comparative ratio of $O(h^{3/2})$.

2 Related Work

There are many algorithms for routing in mobile ad hoc networks. Some rely on the maintenance of routing tables, others do the routing completely reactive which leads to flooding algorithms. If the nodes known their position, then single-message routing can be done without establishing routing tables.

Position-based routing (also called geometric or geographic routing, see [MWH01, GS04] for a survey) uses positioning capabilities of the nodes for routing packets in the network. It is based on the assumptions that (1) nodes can determine their own position, (2) know the position of their direct neighbors and (3) that the source node knows the position of the destination.

Karp and Kung [KK00] and Bose et al. [BMSU01] have proposed the idea to combine a greedy forwarding strategy with a recovery strategy that is used when the greedy forwarding rule has guided the message into a local minimum. Greedy forwarding is based on a local decision of a node to pass a packet to a neighbor node that is nearer to the destination. Local decision strategies are: MFR (most forward within the transmission radius) [TK84] tries to minimize the number of hops, NFP (nearest with forwarding progress) [HL86] tries to minimize energy consumption, and compass routing [KSU99] tries to minimize the path length by choosing the node with the minimal angular distance to the direction of the destination. Since a packet could get stuck in a local minimum, greedy forwarding needs a recovery strategy like perimeter routing [KK00], where a packet is passed along the edges of the face incident to the dead end node until a forwarding progress according to the greedy strategy can be made. Another strategy is to traverse the incident face in case of a local minimum (“face routing”), that has been proposed by Kranakis et al. [KSU99] (see also [BMSU01]) as a routing strategy (not only for recovery). They consider a line segment between the source and the destination and then traverse the faces of the network graph that are intersected by the line segment. Using position information the nodes determine the intersection points and know where and when to switch to the next face. For perimeter routing and face routing the so-called right-hand rule is used for traversing a face. To prevent messages from getting into a loop while applying this rule the network topology must be a planar embedding of a graph. There are various planar graphs that can be used for constructing a planar subgraph of the unit disk graph, e.g. the Delaunay triangulation or relative neighborhood graphs [JT92]. The construction can be done locally by position information and communication in the local neighborhood. These subgraph constructions and variants have been proposed by Gao et al. [GGH+01] and Alzoubi et al. [ALW+03] as topologies for routing in mobile ad hoc networks. The corresponding planarization strategies eliminate advantageous long edges and preserve short ones. This leads to an increased hop distance of the shortest path in the planarized graph compared to the original communication graph. In our construction we allow all edges to be used in recovery mode.

The proposed topologies for planarization have been analyzed with respect to their spanning properties.\footnote{The right-hand rule is a well-known rule for traversing a maze: By keeping one hand (either the left or the right hand) in touch of the wall, it is guaranteed to find a way out of a maze.}

\footnote{A subgraph of the unit disk graph is a spanner, if the length (Euclidean length, hop distance or power distance) of a path between two nodes is only a constant factor larger than the distance of the nodes in the original graph}
Often the desired properties (e.g. constant stretch factor) are only valid if a constant minimum distance between the nodes is assumed (“civilized graphs” [WL02], $\Omega(1)$-model [KWZ02]). To take advantage of such properties in scenarios with unbounded node density, where the minimum distance assumption may be violated, a clustering approach can be applied. Clustering is used in [GGH+01] or [ALW+03] to form a backbone network connecting the clusterheads which has the desired graph properties.

We use an implicit clustering based on a cell structure, which is based on the positions of the nodes in the neighborhood. We need no additional communication for leader election etc. The advantage of the cell structure over the planarization strategies is that communication to all the neighbors is allowed. Recovery becomes easy because the right-hand rule traversing a barrier can be done easily in the cell structure. Topology-based rules (e.g. for applying the right-hand rule on the edges on the graph), that need the planarization are no longer necessary.

Furthermore we use the cell structure to identify barriers, which are local minima to a greedy forwarding strategy. Fang, Gao and Guibas [FGG04] proposed an algorithm for identifying local minima and the regions behind, which they call “routing holes” and which are similar to our notion of barriers. The algorithm uses local rules, but the nodes on the border of the barriers need to communicate so that traffic used for traversing the barriers is shifted from the routing algorithm to the pro-active part of the communication.

3 Learning the Neighborhood

We start our discussion with the question how much communication is reasonable to learn enough about the neighborhood to establish a mesh structure. For some applications, this step is not necessary. In particular, if the nodes are already placed on a grid like structure and the position is given by two integer coordinates. Now, if the grid size is small enough we can skip some parts of the following protocol and continue with routing in the mesh network.

Here, we discuss the issue how far location information must be proactively distributed until a routing protocol can deliver a message. As a simplification we consider the unit disk graph model in this section combined with a round model. Every node has its exact, unique location information. In one time step at most one message can be received by a node. In this paper we use the terms message and data packet interchangeably. This means, a message can be successfully received, if the distance to the sender is at most $r$ and there are no other nodes within the radius $r$ sending in the same time step. Note that in such a round model, the beacon-less routing protocol [HB03] cannot be applied. We motivate the round model by the fact, that the response time of ad hoc nodes is limited, and therefore in practice a continuous medium access protocol relying on arbitrarily small time differences does not work.

We model the network as a graph $G = (V, E)$ with $|V| = n$, where the $n$ nodes $V$ are the participants of the network and the edges $E$ are the communication links. The nodes are placed in the Euclidean plane and have a fixed transmission radius of $r$. Because of the bounded transmission range, we refer to unit disk graphs which correspond to a normalized transmission radius ($r = 1$):

Proactive routing protocols are often referred to as table-driven protocols, whereas reactive routing is also called on-demand routing. In position-based routing proactive communication is often used to exchange position information among neighbored nodes. We call a routing algorithm proactive if a communication pattern includes a periodic exchange of control messages even if there is no communication demand. If communication is triggered only by demand, we call this routing algorithm reactive.

Here, we define a parameterized notion of $k$-hop-proactive routing. A communication protocol is $k$-hop proactive, if proactively transmitted information of a node are available only in nodes in a hop-distance of $k$. Such a constant-hop routing algorithm can adapt quickly to network changes, while reducing the proactive traffic. According to this notion 0-hop-proactive routings refers to reactive routing, while $n$-hop-routing denotes classical routing.

In [FGG04] a similar approach is used. In their routing protocol a message is sent around the routing holes in the network (here called barriers). If $p$ is the maximum hop distance such a message travels around a barrier, then this routing protocol is $p$-hop proactive. In this paper we reduce the proactive hop distance even further and present a 2-hop proactive communication pattern establishing a mesh network structure (with empty mesh cells).
4 Establishing the Cell Structure

In this section we describe how we establish the grid structure from the proactively distributed information. We choose a cell structure based on a grid subdivision with $\ell \times \ell$ squares, called cells, where $\ell \leq \sqrt{3}/20 r$, where $r$ is the radius in the unit disk model. Any smaller cell size can be used, too. Yet, it increases the constant factors involved in the routing algorithm. Each square, called cell, defines a cluster of all nodes lying inside. A node cell contains at least one node.

Two node cells are connected directly, if at least a pair of nodes from each cell is connected. We present a scheme for each node of a cell for answering the questions: Which cells are directly connected to my cell? How can I deliver a message to such a cell?

The first question can be solved by using the 2-hop information about the coordinates of the neighbors’ neighbors. This follows by the fact that all nodes of the cells are within reach (since $\ell \leq r/\sqrt{2}$). Now, all neighbored cells have at least one node which is neighbored to a neighbor. Clearly, then the second question can be answered from the same information. Therefore we can route a message to each neighbored node cell within two hops.

Note that for providing this information it suffices to transmit even less information according to the following scheme: 1st information hop: Every node knows the coordinates of its neighbors. 2nd information hop: Every node knows all neighbors of the neighbors and their coordinates.
This communication pattern reduces the information size in the second hop by replacing continuous coordinates of arbitrary many nodes with a constant number of discrete cell addresses.

We introduce three classifications for such a cell, see Figure 3: A node cell contains at least one node. A link cell does not contain any node. It satisfies one of the following geometric properties: It is intersected by the line between two nodes with (transmission) distance \( r \). All points of this cell have distance at most \( r/2 \) to a node. Every other cell is called a barrier cell.

Barrier cells can obstruct the position routing, especially if they come in a larger group.

**Definition 1** A barrier is a set of barrier cells, which are connected orthogonally or diagonally. The border cells of a barrier is the set of barrier cells having cell or link nodes as orthogonal or diagonal neighbors. The perimeter is the number of border cells.

Note that all border cells of a barrier are connected orthogonally. One of the main contributions of this paper is that paths in the cell structure using only node and link cells are equivalent paths in the communication network. We define such cell-based paths formally as follows.

**Definition 2** A cell path \((c_1, \ldots, c_m)\) consists of node or links cells, such that \(c_i\) and \(c_{i+1}\) describe the same cell or orthogonally neighbored cells.

We call an edge the implicant for a link cell if it intersects with the link cell. Similarly we call a node cell the implicant of a link cell, if all points in the link cell have maximum distance \( r/2 \) to this point. We show that paths in the network and cell routes are essentially equivalent. The crucial point for this equivalence is the connectivity property of a link cell \(C\):

- If two edges are implicants for the link cell, then there is a direct connection between at least one node of each edge.
- If an edge and a node are implicants of a link cell, then the node is connected to at least one of the nodes of the edge.
- If two nodes are implicants of a link cell, then these are directly connected.

**Lemma 1** If \( \ell \leq \sqrt{\frac{3}{8}} r \) then all link cells fulfill the connectivity property.

**Proof:** In the case of two nodes causing the link cell, this follows by the triangle inequality, since all points of the link cell have maximum distance \( r/2 \) to the nodes. In the case of an node and an edge as a implicant, there is a point in the link cell, which has at most distance \( r/2 \) to one of the nodes of the edge and the connectivity property follows.

If the two causing edges of the link cell cross each other in the link cell. Then this crossing point has at most distance \( r/2 \) to one of the nodes each. Again the triangle inequality proves the claim.

If the link cell was originated by two link cells without an intersection in the cell, then we have two lines of maximum length \( r \) where two points on the lines exist with maximum distance \( \sqrt{2} \ell \). Given two lines with these properties the question is how far can be the minimum distance of pairs of nodes of different edges. The minimum distances between points of two different edges occurs for at least one end point. Then, the worst case is that this end point form a isosceles triangle with the other edge as base line and height \( \sqrt{2} \ell \). This leads to the maximum distance \( \sqrt{2\ell^2 + \frac{r^2}{4}} = r \) which leads to the bound \( \ell \leq \sqrt{\frac{3}{8}} r \).

It remains to consider the case when a node and an edge are implicants for the link cell. Then, the maximum distance between a node of the edge and the implicant node occurs when the edge is a base line of an isosceles triangle, where the height of this triangle is the diagonal of the cell. Then, we get the inequality \( 2\ell^2 + \left(\frac{r}{2}\right)^2 \leq r^2 \) which leads to \( \ell \leq \sqrt{\frac{1}{3}} r \).

**Theorem 2** If \( \ell \leq \sqrt{\frac{3}{20}} r \), then paths in the network and cell routes in the cell structure are equivalent up to a constant factor.
1. Every path \( p \) of the network can be replaced with cell path \( p' \) of length at most \( 2\left\lceil \frac{r}{\ell} \right\rceil |p| + 1 \), where the node cells of the cell path \( p' \) contain all nodes of the path \( p \).

2. For every cell path \( p \) there exists a path of length at most \( 2|p| \).

Proof:

1. We substitute the path \( p = (u_1, \ldots, u_n) \) as follows. For each edge \( e_i := (u_i, u_{i+1}) \) we replace every node by its node cell. So, we observe pairs of connected node cells which are connected via link cells.

2. For each cell we assign one or two responsible node. For a node cell this is one responsible node within the cell. For a link cell there are two responsible nodes being the ends of an implicant edge of the link cell. We now route the message by replacing the node and the link cells with the responsible nodes as follows. We consider all three possible transitions between orthogonally neighbored cells:

- Node cell to node cell: The responsible nodes are uniquely determined. We can route directly, since \( \ell \leq \frac{1}{4}r \leq \frac{1}{\sqrt{5}}r \) and therefore even the farthest nodes of the two node cells are connected.

- Node cell to link cell: Because of the connectivity property all implicants for the link cell are connected. Since \( r \leq \frac{1}{4} \leq \frac{1}{\sqrt{5}}r \) all nodes in the node cell are also implicants for the link cell. So, we route from the responsible node for the node cell to any of the two responsible nodes of the link cell, or vice versa.

- Link cell to link cell: Note that for implicants \( e_1 = (u_1, u_2) \) and \( e_2 = (v_1, v_2) \) for each two link cell that there is always a pair \( (u_i, v_j) \) such that the distance \( |u_i, v_j| \leq r \). This follows from a careful extensive geometric analysis for \( \ell \leq \sqrt{\frac{3}{20}}r \). For the path we choose such a pair. Now if the message is in \( u_{3-i} \), then we add the edge \( (u_{3-i}, u_i) \) to the path and route the message to \( v_j \), which is a responsible node for the neighbored link cell.

For routing messages we abstract from the actual network topology and consider only the cells. Passing messages from one cell to another is implemented on a lower layer using the responsible nodes. If these responsibilities are optimized, most of the time the routing between cells does not imply actual message transmission. A responsible node can forward a message (virtually) from cell to cell until a barrier prevents further forwarding or his visibility range is exceeded.

Remember that barrier cells are deserted cells which obstruct routing. One may extend this notation also to barriers in the original meaning like walls obstructing radio signals. Our concept of barrier cells is compatible to this notion as long as the connectivity property of link and node cells is satisfied. In Figure 4 an example is given where this property is violated by barriers obstructing radio signals. In such a case the communication graph is not planar anymore and position based routing techniques cannot be applied, in general.

5 Cell-based Routing

Position-based routing is usually done by a combination of greedy forwarding and recovery strategies (cf. Section 2) These strategies can be applied on the cell structure. Considering routing based on the cell structure is more convenient, because we do not have to care about graph topology issues like crossing edges, that require applying a planarization of the network graph. The disadvantage of planarization strategies like constructing relative neighborhood graphs, is that long edges are removed, if small edges can be used instead. This increases the number of hops. In the cell structure every connection between the nodes responsible for cells can be used. The virtual cell-based routing provides an implicit planarization that is required for loop-free paths in recovery mode (i.e. if a barrier is traversed).
5.1 Time and Traffic

Routing time and the total traffic are important performance measures for any routing algorithm. From now on, we see time as the number of time steps according to the cell structure needed to deliver a message from the source node to the destination. As we do not consider multiple concurrent routing demands and delays induced by switching, the number of time steps correspond to the length of the route a message takes in the network. Regarding a cell path instead of counting the hops between the nodes responsible for cells causes only a constant factor change (cf. Theorem 2). As a reference, we consider the length $h$ of the shortest barrier-free path between source and target. This gives the lower bound for the time to deliver a message.

The traffic is the number of messages produced while performing the routing task (only the reactive part of the protocol). This can also be measured in terms of cells: then traffic is the total number of times when some cell is entered. Traffic shows the amount of parallelism used in the routing algorithm. Obviously, the length $h$ of the shortest barrier-free path is also a lower bound for traffic. But if we only look at this optimal solution that can be found with global knowledge, we disregard the cost for exploring the network which is unknown to the routing algorithm. So we have to regard routing as an online problem. This leads to comparative performance measures which are introduced subsequently. The exploration of the network is difficult if there are many barriers, because in a worst case setting the barriers have to be traversed like a maze. Therefore, we define the total perimeter size $p$ as the sum of the perimeters of all barriers (cf. Def. 1) and use it as a measure for the exploration costs.

5.2 Comparative Ratios

The performance of online algorithms is often evaluated by a comparison of the online solution and the optimal solution, that can be found by an offline algorithm. We use this so-called competitive ratio [BE98] for measuring the time efficiency of a routing algorithm.

**Definition 3** Let $h$ be the length of the shortest barrier-free path between source and target. A routing algorithm has competitive time ratio $R_t := T/h$ if the message delivery is performed in $T$ rounds.

Regarding traffic, the optimal offline solution needs $h$ messages, because the structure of the complete network is known to the offline algorithm. In this case the costs for the exploration of the network are neglected. But any online algorithm can be forced to pay these exploration costs in a worst case scenario by a proper placement of barriers. Therefore, we consider the quotient of the traffic produced by the online algorithm and the online lower bound for traffic. We call this measure comparative ratio. This term was introduced by Koutsoupias and Papadimitriou [KP00] in a more general context.

**Definition 4** Let $h$ be the length of the shortest barrier-free path between source and target and $p$ the total perimeter size in the network. A routing algorithm has comparative traffic ratio $R_T := M/(h + p)$ if the algorithm needs $M$ messages to deliver a message to the target.

The efficiency of a routing algorithm regarding both time and traffic is expressed by the combined comparative ratio:

**Definition 5** The combined comparative ratio is the maximum of the competitive time ratio and the comparative traffic ratio: $R_c := \max\{R_t, R_T\}$

5.3 A Routing Algorithm with Sub-Linear Comparative Ratio

There are two basic routing strategies: single-message or multi-message. The best single-message strategy needs $O(h+p)$ time and traffic, where the total perimeter size $p$ is bounded by an area of size $h^2$. One can easily see that in a worst case scenario any algorithm can be forced to traverse a labyrinth of barriers which contributes with a time overhead of $O(p)$. Though the single-message strategy has optimal comparative traffic ratio, the competitive time ratio is $O(h)$. The best multi-message strategy is flooding. Assumed we know the right flooding depth, this
strategy is time optimal. But we use always traffic \( h^2 \), even if there are small barriers \( (p \approx h) \), which gives a comparative traffic ratio of \( O(h) \).

In the single-message case, we allow the algorithm to spend time for exploration, in the multi-message case we do the exploration in parallel. Both strategies have a combined comparative ratio of \( O(h) \).

We will see that we can achieve a sub-linear combined comparative ratio with a strategy that combines a single-message strategy with flooding. For the single message-strategy we use a combination of greedy forwarding and recovery (for traversing barriers) which is known from position-based routing (cf. Section 2): The greedy rule is to follow a guide line connecting source and target. If this is not possible because of a barrier, then the barrier is traversed by applying the right-hand rule until the guide line is reached again at another position which is nearer to the target (progress condition). For the multi-message strategy we use flooding. Both strategies are applied with a search depth restriction (time-to-live). We start with a small search depth and repeat the greedy strategy and flooding alternatingly with increased search depth until the target can be reached (see Figure 5).

It seems that we combine the disadvantages of both flooding and the greedy strategy. We will see that this algorithm has a better combined competitive ratio.

**Theorem 3** The Alternating Algorithm needs time \( O(h^{3/2}) \) and produces traffic \( O(\min\{h^2, h^{3/2} + p\}) \).

**Proof:** We distinguish between two cases:

*Case 1:* The target is reached while applying the greedy strategy (Fig. 5, line 3). The greedy strategy uses greedy paths (segments on the guide line) and recovery paths (along the border of a barrier). As the greedy rule is only applied if a progress condition is fulfilled, the length of the greedy paths is bounded by \( h \). In recovery mode at most all the barriers have to be traversed, so the length of the recovery paths is bounded by \( p \). The target is reached in \( h + p \) steps, i.e. within \( \frac{4}{3} \log(h + p) \) rounds (because \( 2^{2i/3} \leq h + p \)). The search depth for flooding in the previous round has been \( 2^{2i/3 \log(h+p) - 1} = \frac{1}{2}(h + p)^{2/3} \). The target could not be reached while flooding, but flooding yields optimal paths. Hence, \( h \) is larger than the flooding depth: \( \frac{2}{3}(h + p)^{2/3} \leq h \Rightarrow h + p = O(h^{3/2}) \).

The traffic is bounded by \( \sum_{i=1}^{\frac{4}{3} \log(h+p)} 2^{3i/2} + 2^{2i} = O((h + p)^{4/3}) \)

*Case 2:* The target is reached while flooding (Fig. 5, line 4). It takes \( \log h \) rounds to approach the target through incremental flooding. In each round there is an additional delay caused by the greedy strategy. So, the time is bounded by \( \sum_{i=1}^{\log h} 2^{3i/2} + 2^i = O(h^{3/2}) \).

The target is only reached by flooding (with depth \( h \)), if the greedy strategy is forced to make a detour. But any barrier of with perimeter size smaller than \( h^{1/2} \) cannot prevent a message in recovery mode from returning to the guide line and then reaching the target. That means reaching the target by flooding occurs only if barriers of size \( p \geq h^{1/2} \) are present. If there are smaller barriers \( (p < h^{1/2}) \) then the target can be reached in greedy mode with traffic \( O(h + p) \) for the greedy strategy and \( O(h^{3/2}) \) for the flooding strategy. Otherwise \( (p \geq h^{1/2}) \) the traffic is bounded by \( \sum_{i=1}^{\log h} 2^{3i/2} + 2^{2i} = O(h^2) \).

**Corollary 1** The Alternating Algorithm has a combined comparative ratio of \( O(\sqrt{h}) \).

**Proof:** Follows by the definition of competitive time ratio and comparative traffic ratio and Theorem 3. •
References


