Geographic Routing in City Scenarios

Christian Lochert
University of Düsseldorf
Düsseldorf, Germany

Martin Mauve
University of Düsseldorf
Düsseldorf, Germany

Holger Füßler
University of Mannheim
Mannheim, Germany

Hannes Hartenstein
University of Karlsruhe
Karlsruhe, Germany

1 Introduction

Existing position based routing protocols [4] are very well suited for highly dynamic environments such as inter-vehicle communication on highways. However, it also has been discussed that radio obstacles [3], as they are found in urban areas, have a significant negative impact on the performance of position-based routing. In prior work [5] we presented a position based approach which eliminates these failures and is able to find robust routes within city environments. The algorithm needs global knowledge of the city topology which is provided by a static map. Given this information forwarding of the packets along a street is performed in a position-based fashion while the list of junctions that have to be traversed by the packet is calculated by the sender based on the static map using the Dijkstra shortest path algorithm. In this short paper we show how position-based routing can be done in a city scenario without assuming that nodes have access to a static global map.

2 Position based routing

In existing position based routing approaches an intermediate node forwards a packet to the direct neighbor which is closest to the geographic position of the destination. This is called greedy forwarding. For this task each node has to be aware of *i)* its own position, *ii)* the position of its direct neighbors and *iii)* the position of the final destination. A node determines its own position by using GPS, the position of the neighbors is received through one hop beacon messages transmitted periodically by all nodes and the position of the final destination is provided by a location service [2] or by a geocast application. Since greedy forwarding uses only local information a packet may reach a local optimum regarding the distance to the destination, i.e. there exists no neighbor which is closer to the destination than the intermediate node itself. In order to escape from a local optimum a repair strategy may be used. The general aim of a repair strategy is to forward the packet to a node which is closer to the destination than the node where the packet encountered the local optimum. Once such a node is reached greedy forwarding can be resumed. Several repair strategies have been proposed, including Greedy Perimeter Stateless Routing (GPSR) [4] and face-2 [1]. However, it has been shown [3, 5] that existing repair strategies do not perform well in city environments because they rely on distributed algorithms for planarizing graphs. In the presence of radio obstacles even in static networks the use of these algorithms frequently partitions an otherwise connected graph, making the delivery of packets impossible.

3 Greedy Perimeter Coordinator Routing

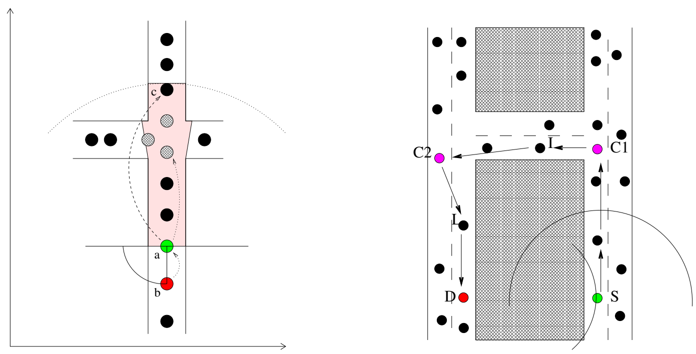
Greedy Perimeter Coordinator Routing (GPCR) is a position based routing protocol using standard greedy forwarding and a repair strategy that does not require a graph planarization algorithm. In the following we focus exclusively on the repair strategy.

The repair strategy of GPCR avoids using graph planarization by making routing decision on the basis of streets and junctions (which form a natural planar graph) instead of individual nodes and their connectivity (which do not form a natural planar graph). As a consequence the repair strategy of GPCR consists of two parts: (1) On each junction it has to be decided which street the packet should follow next. (2) In between junctions a special form of greedy forwarding is used to forward the packet towards the next junction. Given that no external map is available the key challenges are to identify nodes that are on a junction and to avoid missing junctions while greedy forwarding is used. The latter problem is illustrated in Figure 1(a) where a would forward the packet beyond the junction to node c if regular greedy forwarding were used. In the remainder of this work we call nodes that are located in the area of a junction a *coordinator*. In a first step we assume that each node knows whether it is a coordinator (i.e., located in the area of a junction) or not. We will show in section 4 how a node can learn about this information.

If the forwarding node is located on a street and not on a junction the packet is forwarded along the street towards the next junction. To achieve this the forwarding node selects those neighbors that approximately extend the line between the forwarding node's predecessor and the forwarding node itself. Out of these qualified neighbors one has to be selected as the next hop of the packet. As long as there are no qualified neighbors which are coordinators the node with the largest distance to the forwarding node is chosen. If coordinators are qualified then one coordinator is randomly chosen as next hop. With this approach packets will not be forwarded across junctions. Figure 1(a) shows an example of how the next hop is selected on a street. Node a receives a packet from node b. Because a is located on a street and not on a junction it should forward the packet along this street. First the qualified neighbors of a are determined. Then it is checked whether at least one of them is a coordinator. As in this example there are three coordinator nodes that qualify as a next hop one of these coordinator nodes is chosen randomly and the packet will be forwarded to this coordinator.

If the forwarding node is located on a junction (i.e., it is a coordinator) then the node needs to determine which street the packet should follow next. To this end the topology of the city is regarded as a planar graph and the well known right-hand rule [1, 4] is applied.

We illustrate the use of the right hand rule in figure 1(b). A packet with destination D reaches a local optimum at node S. The forwarding of the packet is then switched to the repair strategy and it is routed along the the street until it hits the first coordinator node. Node C_1 receives the packet and has to decide on the street the packet should follow. Using the right-hand rule it chooses the street that is the next one counter-clock wise from the street the packet has arrived on. Therefore node I will be chosen to forward the packet. The packet will then be forwarded along the street until the next junction is reached. When the packet arrives at the coordinator C_2 this node has to decide again on the next street that is to be taken and decides to forward the packet to node L. At this point the distance to the destination is less than at the beginning of the repair strategy at node S. Hence the mode is switched back to the greedy



(a) Avoid missing a junction by detecting coordinator nodes.

(b) Repair strategy.

Figure 1: Graphical representation of finding a forwarding node.

strategy.

4 Detecting junctions

One key challenge of GPCR is to detect whether a node is located on a junction without using external information. In the following we present two alternative approaches. In the first approach each node regularly transmits beacon messages including the position of the node sending the beacon as well as the position of all of its neighbors. By observing the beacon messages a node has the following information for each neighbor: its position and the position and presence of the neighbor's neighbors. A node x is then considered to be located in a junction if it has two neighbors y and z that are in transmission range to each other but do not list each other as neighbors. This indicates that those neighbors are separated by an obstacle and that x is able to forward messages around the obstacle.

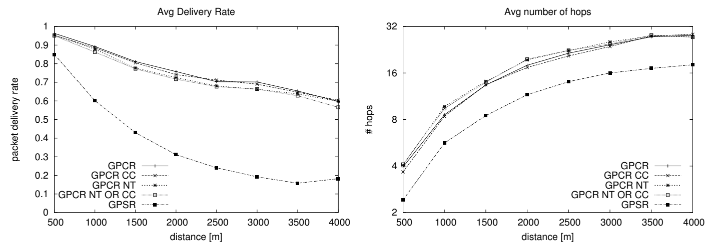
The second approach does not require special beacon messages. Each node calculates the correlation coefficient with respect to the position of its neighbors. The correlation coefficient is defined as:

$$\rho_{xy} = \left|\frac{\sigma_{xy}}{\sigma_x \sigma_y}\right| = \left|\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)\left(\sum_{i=1}^{n}(y_i - \bar{y})^2\right)}}\right| \tag{1}$$

with $\rho_{xy} \in [0, 1]$. A correlation coefficient of 1 indicates a linear coherence as it is found when the node is located in the middle of a street. A correlation coefficient of 0 shows that there are neighbors located on more than one straight line (i.e., the node is located on a junction). By adjusting a threshold ϵ a node can evaluate the correlation coefficient and assume with $\rho_{xy} \geq \epsilon$ that it is located on a street and with $\rho_{xy} < \epsilon$ that it is located within the area of a junction.

5 Simulation Results

We simulated the performance of GPCR with the `ns-2` simulator version ns-2.1b9a. To obtain realistic measures we used a real city topology which is a part of Berlin, Germany. The scenario consists of 955 cars (nodes) on 33 streets in an area of 6.25 km × 3.45 km. The movement of the nodes was generated with a dedicated vehicular traffic simulator [5]. IEEE 802.11 was used as MAC with a transmission rate of 2 Mbps. The transmission range was set to 500 m, real world tests with cars have shown this to be a reasonable value when using external antennas. For each simulation run we randomly selected ten sender-receiver pairs. Each pair exchanges 20 packets over 5 seconds. We measured the achieved packet delivery rate (Fig. 2(a)) versus the distance between the two communication partners (at the beginning of the communication) and the number of hops (Fig. 2(b)). Each point in the above mentioned graphs is based on 10 independent simulation runs.



(a) Delivery rate.

(b) Avg. no. of hops.

Figure 2: Simulation of GPCR vs. GPSR.

Several simulations on the different approaches to detect a junction were done. As one can see in Fig. 2(a) detecting junctions by calculating only the correlation (CC) coefficient performs slightly better than relying solely on the comparison of the neighbortables of the neighbors (NT). We also analyzed a compound decision consisting of the neighbortable comparison and correlation coefficient, concatenated by OR as well as by AND. The latter one outperforms the other approaches significantly. As a result we defined this procedure as the GPCR-approach. In general the study on achievable packet delivery rate (Fig. 2(a)) shows good results for our approach compared GPSR which does not use a repair strategy suitable to city topologies. This improvement in performance comes at the expense of a higher average number of hops, increasing the latency slightly.

6 Future work

In GPCR the next street to be taken is determined without considering account whether there are sufficient nodes on the street to allow packet forwarding to the next junction. We plan to augment GPCR with a very low overhead proactive probing scheme to predict whether the next junction in a given direction can be reached or not.

References

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[2] T. Camp, J. Boleng, and L. Wilcox. Location Information Services in Mobile Ad Hoc Networks. In *Proc. of IEEE ICC '02*, pages 3318–3324, New York City, New York, April 2002.

[3] B. N. Karp. Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning. Talk at DIMACS Workshop on Pervasive Networking, May 2001.

[4] B. N. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. of ACM MobiCom '00*, pages 243–254, Boston, Massachusetts, August 2000.

[5] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve. A Routing Strategy for Vehicular Ad-Hoc Networks in City Environments. In *Proc. of IEEE IV'03*, pages 156–161, Columbus, OH, 2003.