REIHE INFORMATIK
TR-2007-012

Unicast Ad-Hoc Routing in Vehicular City Scenarios

Sascha Schnaufer, Holger Füßler, Matthias Transier, Wolfgang Effelsberg
University of Mannheim
– Fakultät für Mathematik und Informatik –
Praktische Informatik IV
A5, 6
D-68159 Mannheim, Germany

Unicast Ad-Hoc Routing in Vehicular City Scenarios

Sascha Schnaufer* Holger Füßler Matthias Transier
Wolfgang Effelsberg

Within Vehicular Ad-Hoc Networking (VANET), i.e., networking between radio-equipped vehicles, unicast packet forwarding can be separated into the one-dimensional highway case and the two-dimensional city case. In this report, we survey the routing methods developed in the FleetNet [1] and Network-on-Wheels [2] projects plus a novel combination of two well-known methods called PBR-DV or *Position-Based Routing with Distance-Vector recovery*. On the quest for a city-capable candidate routing algorithm as a possible standard, we discuss the usability and performance of the protocols in city scenarios. Finally, we conclude proposing PBR-DV as a candidate protocol for small-hop-count unicast VANET scenarios.

1 Introduction

Vehicular Ad-Hoc Networks (VANETs) are a special kind of Mobile Ad-Hoc Networks (MANETs), where wireless-equipped (road) vehicles form a network without any additional infrastructure. While many communication scenarios exist for these networks, government-sponsored research activities like the German Network-on-Wheels project [2] mainly focus on the application of VANETs to increase vehicular safety, but also considering applications increasing driver convenience. For these convenience applications, unicast data communication is believed to be a potentially important function provided by a VANET communication system, on one hand to support IP-style vehicle-to-vehicle applications on the other hand as a necessity for vehicle-to-Internet communication.

---

*Sascha Schnaufer acknowledges the support of the German Ministry for Education and Research (BMB+F) for the "Network-on-Wheels" project under contract no. 01AK064F and Matthias Transier the support of the DFG for Priority Program 1140.

1.1 Characteristics of City Scenarios

In this report we focus on methods providing multi-hop unicast connectivity between road-bound vehicles. While algorithms for vehicles traveling on the same highway are well-researched and understood [12, 37, 13], a more challenging scenario is communication between cars driving in a city environment. The most fundamental difference between these highway and city scenarios is that highway scenarios are largely one-dimensional which is very beneficial for greedy position-based routing approaches. Cities, however, face the following challenges to routing protocols:

Geometric Two-Dimensionality In a city scenario, vehicles change their movement direction all the time. Moreover, cars can move with any relative angle allowed by the street geometry. In contrast to the highway, this weakens the correlation of the destination position to a suitable next-hop.

Obstacles A city is usually characterized by the presence of radio obstacles, considering 2.4 or 5 GHz radio transmissions with power settings similar to WLANs on non-elevated radio antennas. Also, this creates problems with position-based next-hop selection [19]. For all of the modeling in these scenarios usually the simple assumption is made that whenever the line-of-sight between two nodes touches an obstacles, the nodes are not able to communicate. However, multipath propagation and complex obstacle surfaces create a much more complicated situation in reality.

Node Density In cities of industrialized countries, the node density can be expected to be rather high with respect to the radio range, especially at “density hot spots” like junctions. On one hand, node density creates a better ad-hoc connectivity, but on the other hand, it poses a challenge to flooding mechanisms that need to be very efficient [31].

Low Mobility Compared to highway scenarios, the nodes move at rather low speeds, influenced by node density (the more nodes the slower the movement) and constrained by speed limits. Also, the mobility is location-dependent, e.g., it is lower at junctions.

1.2 Characteristics of Expected Data Traffic

When designing a communication system—esp. for ad-hoc networks—a very important issue are the communication needs the system has to support. Of course, application developers ask the opposite question: What will your communication system offer to us? Obviously, this is a chicken-egg problem since both goals can not easily be optimized globally.

In the Internet, this problem is solved by transport protocols like TCP [38] trying to use the complete available bandwidth and at the same time to be fair to other data streams. With this preposition, a communication system only has to support the capability to send data packets to arbitrary nodes, regardless if they are direct neighbors or only reachable across multiple hops.

However, the decisive question in ad-hoc networks is not really the bandwidth but the hop-distance between communicating nodes. [16, 15] state that the bandwidth in an ad-hoc network is, even under optimal circumstances, $\Theta\left(\frac{W}{\sqrt{n}}\right)$ where W is the link bandwidth and n is the number of nodes. While this is only an asymptotic statement, [25] shows by simulation how network performance diminishes with a growing number of hops. In a vehicular context, [17] shows the same for cars running TCP.

While the capacity constraint will force ad-hoc communication to be local, TCP problems (e.g., [10, 6]) and mobility [29, 30] still add more weight to this necessity. Obviously, high-bandwidth communication can also hamper low-bandwidth communication for other VANET purposes. From all this, we derive the following design goals for our unicast routing protocol:

Hop Limit Unicast Communication will be mostly local like at most ≈ 5 hops.

Semi-Mobility The most important application case will include communication to a (multi-hop) hot spot, making one end of the traffic static.

The remainder of this report is organized as follows: The next section outlines candidate protocols for application in the scenarios characterized above. Section 3 analyzes their suitability for our routing problem. Finally, Section 4 will give conclusions including protocol recommendations.

2 Overview of Candidate Protocols

In the following we will coarsely describe candidate protocols for VANET routing in a city scenario. Since the utilization of position-information is easily possible in VANETs and obviously beneficial for routing, all protocols considered (except AODV in 2.5) make extensive use of the positions of the communicating nodes. While the sender's position is obtained from the local GPS/Navigation system, a so-called location service like RLS [22] , i.e. a distributed network protocol, provides the position of the destination. Table 1 shows an overview over the protocols described in the this section. The table consist of the following rows:

Map Usage Indicates if the protocol needs a map or can use one optionally.

Forwarding Based on There are two basic forwarding strategies: One is based on the position of the destination and the other one is based on the discovered network topology.

Greedy Recovery Position-based routing approaches typically use a so-called greedy routing mode. In this greedy mode the neighbor which is the nearest to the position of the destination is selected as the next hop. If there is no suitable neighbor,

	STBR	**GSR**	**GPSR**	**GPCR**	**AODV**	**PBR-DV**
Map Usage:	mandatory	mandatory	—	optional	—	—
Forwarding based on:	Position	Position	Position	Position	Topology	Position
Greedy Recovery:	—	—	Planar Graph Routing	Planar Graph Routing	—	Distance Vector
Recovery requires Flooding:	—–	no	no	no	—–	yes
Variable Header Size:	—–	junction source route	—	—	—	—
Proactive Overhead:	Junction-to-Junction Beacons	Neighbor-hood Beacons	Neighbor-hood Beacons	Neighbor-hood Beacons	—	Neighbor-hood Beacons
CBF compatibility:	yes	yes	no	no	yes (CBDV)	yes (CBDV)

Table 1: Protocol Comparison

a recovery mode is started to find a non-greedy route. In this row you can see the used recovery strategy.

Recovery Requires Flooding This row indicates whether the recovery mode of a protocol starts a flooding process.

Variable Header Size Routing protocols add routing information to the header of every packet. This row shows if this additional information has a variable length.

Proactive Overhead Some routing protocols continuously send control packets to build neighbor tables. This row indicates if a protocol sends such control packets.

CBF Compatibility This row shows if a protocol can be extended with CBF (see section 2.7).

2.1 PBR and Greedy Perimeter Stateless Routing

The term Position-Based Routing (PBR) summarizes a class of routing algorithms. They share the property of using geographic positioning information in order to select the next forwarding hops. If a node wants to send a packet, a location service is used to determine the position of the destination. The packet is sent without any map knowledge to the one-hop neighbor which is the closest to the position of the destination. To make this possible, every node continuously sends beacon packets with their own position and node id. This is necessary to build one-hop neighbor tables.

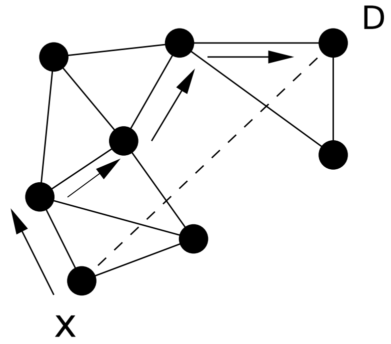


Figure 1: GPSR's Perimeter Strategy (from [20])

If no neighbor is closer to the position of the destination as the receiving node, the packet has reached a local optimum. Various algorithms have been proposed to overcome such situations. One of them is Greedy Perimeter Stateless Routing (GPSR) [20]. GPSR use a so-called perimeter mode as recovery strategy to leave the local optimum. This perimeter mode is comparable to the face-2 algorithm proposed in [8] which is a distributed algorithm to find a path in a graph with the local knowledge of a node. The idea is to use the so-called right hand rule [5] to find a path. However, the right hand rule only works if the graph is planar, i.e., it can be drawn so that no links intersect. Therefore GPSR uses a distributed algorithm to build a planar one-hop neighbor table at every node. A benefit of this planar graph routing strategy is that no flooding is necessary because it only uses existing neighborhood information. In Figure 1 we see an example for the path finding process of the right hand rule.

The node which starts the perimeter mode stores its own position in the packet header. If a node receives such a packet and has a neighbor that is closer to the destination than the position in the header, it changes back to the greedy routing mode.

2.2 Street Topology Based Routing

The idea of Street Topology Based Routing (STBR) [11] is to interpret a given street map as a planar graph. Every junction is interpreted as a graph node and ever street between two junctions as a link. On these links, small junction-to-junction beacons are sent checking for the usability of a street to transport data packets. Additionally, there is link-state information kept at the nodes (junctions) about the connectivity to the neighboring junctions.

In principal, the protocol works as follows: On a junction one node is selected as master. All other nodes on the junction operate as slaves and nodes on streets between junctions are used as forwarders. So there are three valid states for a node: master, slave and street (forwarder). The job of the master node is to check if the links to the next junctions is up or down. Therefore every master broadcasts beacon packets. These beacons are forwarded by the street nodes to the masters of the neighboring junctions (which are one hop away in the planar graph). The beacons are forwarded using CBF

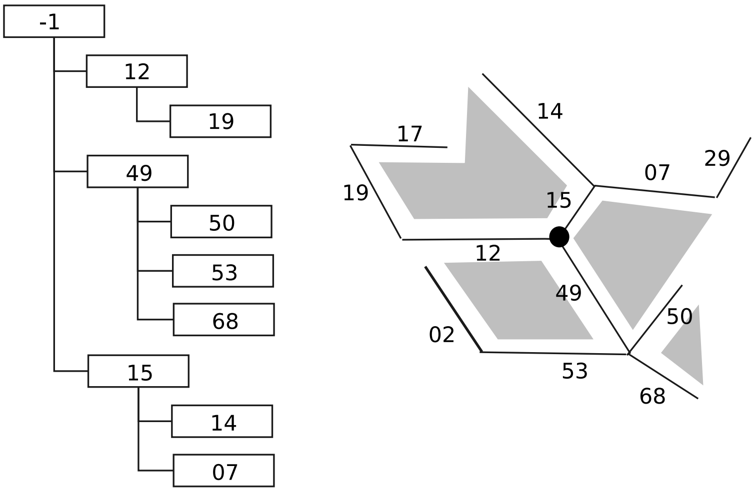


Figure 2: Example of an STBR Neighbor Table (as in [11])

(see section 2.7). The status of the links is stored in a so-called junction neighbor table. The first level, i.e., the list of directly neighboring junctions, is sent out by the master nodes with every beacon. This information is used by the receiving masters of the neighboring junctions to build a two-hop neighbor table and by all other nodes on the streets to build a one-hop neighbor table. In Figure 2 we see an example for a two-hop neighbor table.

If the position of the destination node is unknown, the source node uses a location service to acquire the position. After this a complete routing process from source to destination consists of three parts:

1. Routing from the source node to the first junction.
2. Routing from junction to junction.
3. Routing from the last junction to the destination node.

If the sender is a street node and the destination is not on the same street, the packet is forwarded to the junction which is the closest to the position of the destination. The master node from this junction uses the two-hop neighbor table to find the junction which is reachable and have the most progress to the destination. Then the master starts the forwarding of the packet to the chosen junction. This second part is repeated until the destination is located on a street which is directly connected to the current junction. Then the junction master uses the street nodes to forward the packet to the destination node.

There are many special cases the protocol has to take care of, e.g., if there is no junction with progress in the neighbor table. For more details please have a look at [11]. Also, [32] describes a similar protocol.

2.3 Geographic Source Routing

Like STBR Geographic Source Routing (GSR) [7, 28] uses a map and a position-based address scheme to send packets to the destination. As before, the source node uses a location service to acquire the position of the destination node. Now the source node evaluates the shortest path between itself and the destination (Dijkstra [9] or Breadth-First-Search [3]). All junctions on this shortest path are added to the header of the packet like in DSR [18]. The Packet is forwarded from street to junction, from junction to junction and from junction to street in a position-based routing (PBR) fashion. Therefore every node continuously sends beacons with its own position and its node id. With the position information of the beacon every node can build a one-hop neighbor table. So a receiving node can select the neighbor with the highest progress to the position of the next junction as the next hop. After reaching the junction, the junction is deleted from the packet header and the position of the next one is used as new destination. After the last junction the position of the destination node is chosen. When the packet is forwarded in junction-to-junction mode and there is no node closer to the next junction than the current node, a global position-based routing is started. In this case the position of the destination node can directly be used. This is equivalent to the greedy mode in GPSR.

2.4 Greedy Perimeter Coordinator Routing

The basic idea of Greedy Perimeter Coordinator Routing (GPCR) [26, 27] is to use position-based routing without map knowledge and with a better recovery strategy than the perimeter mode of GPSR. Like in GPSR, every node continuously sends beacon packets with their own position and their node id. Additionally, the beacon includes information about whether the sender is located on a junction or on a street. When a node wants to send a packet a location service is used to estimate the position of the destination. Similar to GPSR, packets are sent to the one-hop neighbor which is closest to the position of the destination. The major difference is that neighbors on junction are preferred even if their progress to the position of the destination is smaller. This selection strategy is called restricted greedy routing. Figure 3 depicts a small example.

Node u wants to send a packet to D. The normal greedy mode would select 1a because it is the neighbor that is located closest to D. Node 1a would select 1b and then a local optimum is reached. In restricted greedy mode u knows that 2a is located on a junction and would prefer this one. 2a has, compared to 1a, the advantage that it has no obstacles between itself and 2b. We can see in this small example that junction nodes have the benefit that they can reach neighbors located on all connected streets.

The right hand rule only finds a path if the graph is planar. For this reason GPSR uses a distributed algorithm to build a planar one-hop neighbor table by virtually deleting links. GPCR's use the right hand rule slightly different: In the recovery mode a junction node decides based on the right hand rule to which junction the packet should be forwarded. Between two junctions restricted greedy routing is used. So the recovery mode uses a graph which is identical to the real word street map and thus planar. Iden-

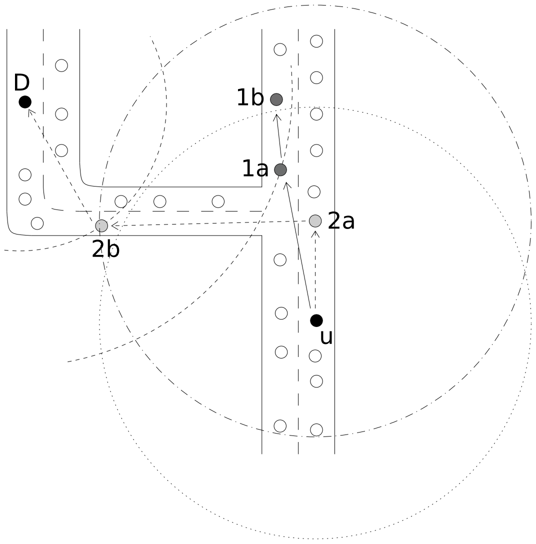


Figure 3: Example for GPCR's "Restricted Greedy" (as described in [26])

tical to the recovery mode in GPSR no flooding is necessary because GPCR also only uses existing neighborhood information.

To detect if a node is on a junction, [26] proposes two strategies. First all nodes add the positions of their neighbors to the continuously sent beacon packets. A node is located on a junction if two neighbors are in transmission range but do not list each other in their neighbor tables. It is assumed that an obstacle is between these two nodes and thus they are not on the same street. In the second approach no additional information has to be sent. The node is on a street if all one-hop neighbors are coarsely located on a line in the driving direction of the car. Otherwise the node is on a junction. This criterion is checked with a correlation coefficient [4].

2.5 Ad-Hoc On-Demand Distance Vector Routing

Ad-Hoc On-Demand Distance Vector Routing (AODV) [34] is a reactive routing protocol. It only builds a route from source to destination if a node wants to send a packet. If the source node does not know a path to the destination, it broadcasts a route request packet. If the receiving node knows a path to the destination or is the destination, it sends a route reply packet to the neighbor who sent the route request. In case the receiving node does not know a path, it rebroadcasts the route request packet. Furthermore the node stores which neighbor has sent the request, which node has initiated the route discovery and the hop count to/from this node. The tuple {destination node / next hop / distance in hops} is called a *distance vector*, the data structure in which it is stored is called *distance vector table*. If a node receives a route request packet it has already sent, it discards the packet. It is assumed that the first route request packet traveled on a short path. When the destination node receives the route request, it looks into its distance vector table and sends the route reply back to the request originator along the path in the table which is identified by the next hop. This next hop continues to do so until the original requester is reached. This route reply phase is also called backward path setup, since the distance vector tables are now updated with information about how to reach the destination. That way the source node, the destination node and all nodes in between can use the discovered route.

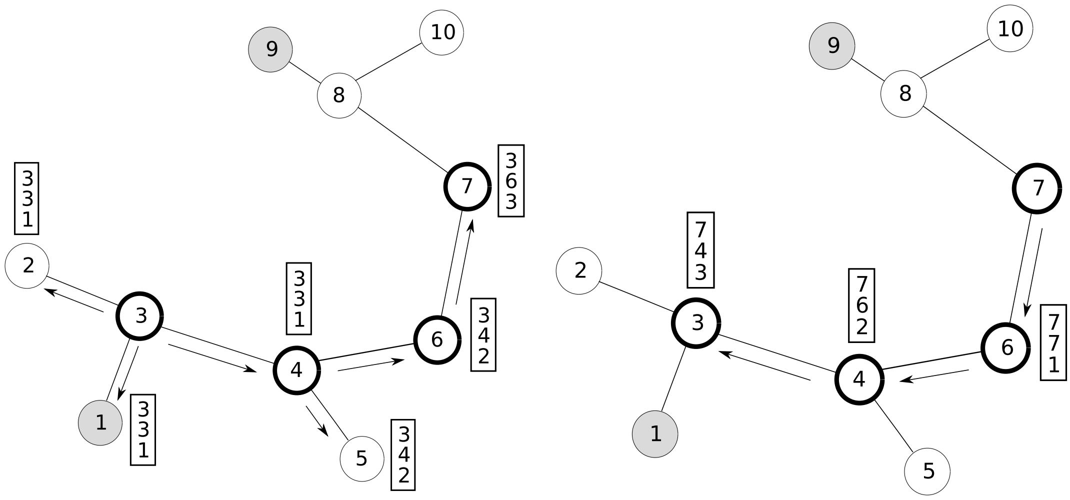
AODV comes with a complete set of methods for route and neighbor maintenance making it a complete and well-evaluated protocol [35, 33]. From the protocols listed here, it surely is the most mature.

2.6 Position Based Routing with Distance Vector Recovery

In this section, we present Position Based Routing with Distance Vector Recovery (PBR-DV), our proposal of combining position-based greedy routing with AODV-style recovery. This approach uses the well-known position-based greedy routing scheme which is also used in GPSR (see above). Thus every node periodically sends beacon packets with their position and node id. If the position of the destination is unknown, the source node uses a location service to acquire it.

While greedy routing is the default behavior, the novelty starts when a node reaches a local optimum. In that case, PBR-DV changes to a distance vector mode. The task of this recovery mode is to forward the packet to a node which is closer to the position of the destination as the node starting the recovery. Of course, this also includes the destination node itself. The route discovery is similar to AODV. Thus the node at the local optimum broadcasts a route request packet. The route request packets of PBR-DV additionally include the position of the node which starts the recovery and the position of the destination node. If the receiver of the route request is not located closer to the destination than the node which starts the recovery, it rebroadcast the packet and stores the id of the sender in the routing table. Otherwise, i.e., when it provides distance progress, it sends a route reply packet with its own position to the neighbor which broadcasts the request. The receiver of the reply packet uses the entry in the routing table to forward the packet to the previous node which broadcast the request and so on. During the packet travels back to the node which starts the recovery, every forwarding node stores the previous hop in the routing table. This way the node which starts the recovery, can forward the packet to a node which is closer to the destination. On this closer node the mode is changed back to PBR.

On the left side of Figure 4 we see a small route request example. Node 1 wants to send a packet to node 9. Node 1 forwards the packet to node 3 because this is the node with the most progress to the position of node 9. However, node 3 has no neighbor which is closer to the destination and switches the packet to distance vector mode. Node 3 broadcasts a route request packet to node 1, 2 and 4. The receivers store in the routing table that they can reach node 3 in one hop and rebroadcast the request packet because no node is closer to node 9 than the recovery starting node. Node 6 receives the forwarded request packet and rebroadcasts it. Finally, the packet reaches node 7 which is closer to node 9 than node 3. In its neighbor table, node 7 stores the information that



(a) Request Packet (b) Reply Packet

Figure 4: PBR-DV: Path Setup in Recovery Mode

it can reach node 3 over node 6 in tree hops and it sends a route reply packet to node 3. On the right side of Figure 4 we see the path of the route reply packet. After node 3 receives the route reply it forward the packet along the acquired path to node 7 which changes the packet back to PBR mode and forwards the packet to node 8. Finally, node 8 forwards the packet to the destination node.

The recovery strategy of PBR-DV has—compared to the ones of GPSR and GPCR—the disadvantage that an additional flooding is necessary to discover the non-greedy part of the route. This effect can be minimized by flooding the request with a small hop limit.

2.7 Contention-Based Forwarding

Contention-Based Forwarding [14] is a novel position-based packet forwarding method for both general ad-hoc networks and VANETs. The fundamental difference to all methods listed above is that the neighbor selection is not done by the node currently in possession of the packet but by the potential forwarders. The main advantages of CBF are (a) almost perfect mobility resilience and (b) higher performance with realistic radio channel models [37]. In the first proposal, CBF was only usable for greedy routes based on position CBDV [24, 23], or Contention-Based Distance-Vector Routing is the contention-based cousin[1] of PBR-DV and can be applied to 1D and 2D routing.

While CBF simulation results indicate its supremacy to explicit next-hop selection algorithms, the basic problem for practical CBF application is that its more difficult to

---

[1] Standard DV routing has to know about a node's potential forwarders/neighbors to select them explicitly. CBDV operates truly contention-based without explicit forwarder selection, by greedily forwarding on the hop-count property.

operate with standard 802.11 MAC hardware and operating system user space implementations due to its sensitivity to timer granularity. Thus, we will stick to explicit-select protocols for the remainder of this report.

However, most protocols we discuss could be modified to support CBF operation. In detail, while the distance-vector can be modified to be contention-capable as described above, GSR could incorporate CBF as a junction-to-junction forwarding method as in STBR. Oppositely, this is quite different with GPSR and GPSR, due to their founding on distributed graph planarization. This planarization works on the resp. neighbor tables of the involved nodes by removing virtual links such that the resulting network graph is planar. CBF, however, does not have neighbor information due to the lack of beaconing. Therefore, the planarization is not applicable making GPSR and GPCR less compatible for a forwarding method based on contention.

3 Suitability Analysis

In this section we analyze the suitability of the presented routing approaches in a city scenario. Therefore we classify the approaches into the ones using maps (Section 3.1), methods based on distance vectors (Section 3.2) and finally algorithms that only use node positions (Section 3.3).

3.1 Map-Based Approaches

The first group consists of the routing approaches GSR and STBR. Both approaches assume that the communication system has access to the digital representation of a street map, e.g., provided by a navigation system. In theory, the additional map knowledge can improve the routing capabilities, since the node positions and therefore the network structure is highly correlated with the streets. Nevertheless, following GSR's and STBR's description, we have to keep in mind that this improvement is bought dearly with a more complex architecture of the communication system. Furthermore the system is required to have up-to-date information of the street situation, moreover, in the algorithms discussed, the used maps have to be the same.

The path calculation of GSR assumes that all links (streets) are up. A link is called up if there are enough cars to forward the packets from the start junction to end junction of the street. If this is not the case, GSR switches to the global position-based routing mode and forwards the packet directly to the position of the destination node. In this mode the receiving node selects the neighbor with the highest distance progress to the position of the destination as next hop. If there is no neighbor node closer to the destination than the current node, the packet is in a local optimum and will be discarded. For this reason GSR is most suitable for scenarios with many nodes resulting in many "up links", i.e., streets with a continuous node distribution being able to forward packets from one junction to the next. However, especially in sparse scenarios it is very unlikely that most links are up. Altogether, GSR is a comparatively simple routing approach which does not exploit all the potentials of the map usage.

STBR uses the map in a more complex manner and considers whether a link (street) is up or down. Therefore STBR sends beacons within the first level of the junction neighbor table from junction to junction. Every street node is used as possible forwarder for the beacons. If an application needs the local neighborhood information, PBR can be used instead of CBF for forwarding the beacons from junction to junction. In this case there are junction beacons and beacons used for PBR forwarding. These packets create continuous traffic to build all necessary neighbor tables even if there are no data packets. In the scenario we have in mind unicast traffic is only one communication type. Its importance is minor compared to active safety applications, and even if CBF is used, the continuous multi-hop control traffic might conflict with active safety data which is consequently a disadvantage of this approach. Another challenge is that without modification, STBR would try to send junction beacons along a highway because it is not suitable for mixed scenarios. Furthermore, the high amount of special cases the protocol has to handle, e.g., the selection of the junction master or the transfer of the two-hop neighbor table to the new master when the old one leaves the junction and so on, increases the complexity of the protocol state machine. Altogether, STBR is a heavy-weight protocol most likely providing an advantage in city scenarios with much long-distance (in terms of hops) unicast communication, at least spanning multiple junction traversals.

3.2 Distance Vector-Based Approaches

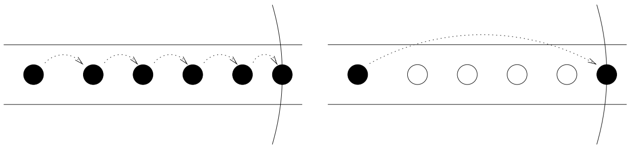
AODV is the only distance vector-based approach in this report. In [27, 26, 28, 11, 36, 37], AODV is compared to other routing approaches in vehicular scenarios. In most of these simulations AODV performs well for communication over few hops in the city case. However, a disadvantage of AODV is that it uses explicit routes. If a route is broken, a new route request will be flooded. PBR-based approaches decide the next hop only with the knowledge of the local neighborhood. Therefore, they can better respond to changes in the environment like moving cars but only if the next hop is greedy. Obviously, this effects gain gravity with increasing mobility causing any topology-based approach to behave poorly on the highway.

An important VANET use case is likely to be (multi-hop) communication to a static hot spot. In such a scenario the position of the hot spot is known or it is only necessary to acquire it once. If the position is known, position-based approaches can use this position directly and do not even need a location service. AODV is not position-based and has to establish an explicit route every time. Thus, AODV is only suitable for few-hop communication and non static nodes.

3.3 PBR-Based Approaches

The next group consists of GPSR, GPCR and PBR-DV. All of them are based on PBR and use a special recovery strategy. GPCR also slightly modifies greedy forwarding by preferring cars having radio access to multiple streets. However, GPCR only reaches low junction detection rate (see below) lessening the effect of this modification. Consequently, we stick to the differences in recovery strategies to analyze the projected protocol performance.

In the recovery mode **GPSR** use the right hand rule to find a path. However, this rule only works if the graph is planar. Therefore GPSR uses a distributed algorithm to build a planar one-hop neighbor table at every node. This algorithm deletes intersecting links. A drawback of this method is that the deleted links are mostly the longer ones. On the left side of Figure 5 we see a typical routing process in perimeter mode and on the right side we see the same situation in greedy mode. So the average progress per hop is much lower than in greedy mode. This leads to increased delays and to hop counts that may exceed the threshold.



(a) Perimeter Mode Routing (b) Greedy Routing

Figure 5: GPSR: Progress Per Hop (from [28])

There are two common planarization methods, Gabriel Graph (GG) and Relative Neighborhood Graph (RNG). Both assume that the connectivity between nodes only depends on the node distances. If there are obstacles like buildings which avoid the communication between nodes, it is possible that the planarization algorithm deletes non intersecting links, virtually creating an unconnected graph. Figure 6 depicts such a situation.

Following the planarization rules, Node u deletes the link to node v from its neighbor table because there is another node in the gray-shaded region (node W). This is due to the assumption that all nodes inside this area are able to communicate with each other. However, removing $\overline{uv}$ disconnects the graph, because v is no longer reachable since an obstacle blocks the assumed link $\overline{wv}$.

Another problem of GPSR is that mobility can induce routing loops for packets being routed in perimeter mode. For that reason we can summarize that the perimeter mode is not suitable for city scenarios.

The recovery mode of **GPCR** requires a reliable junction detection method. [26] has evaluated the combination of both junction detection approaches described in Section 2.4. It is only assumed that the node is on a junction if both checks agree on this result. The authors make a simulation study in a city scenario to analyze how well this combination works. One problem of GPCR is that only 55 percent of all junction checks correctly detect an existing junction. The remaining checks lead to incorrect results, where the study differentiates two kinds of errors. a) A node does not recognize that it is on a junction. b) A node is on a street but considers itself on a junction. 33 percent of the incorrect results are of the first error type and 12 percent of the second. Without a better junction detection the recovery mode and the used right hand rule do not work

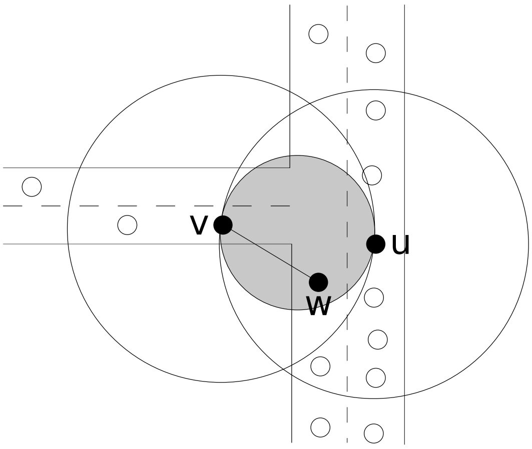


Figure 6: Planarization Problems (as in [28])

properly. Both parts of the junction detection use neighborhood information. These methods work well if there are many neighbors and fail in sparse scenarios. Obviously, this problem could be alleviated by using a map for junction detection. However, even with a perfect detection the right hand rule performs badly in many situations. In Figure 7 we see a bidirectional routing flow. Node `u` sends a packet to node `v` which sends a packet back to node `u`. The complete route to node `v` is greedy and works without a recovery mode. However, the route back to node `u` is only greedy until node `w`, where GPCR will switch to the recovery mode. The right hand rule prefers the right hand direction to the destination and is a good choice if a short route lies to the right. The figure, however, depicts the path a packet travels if the right hand direction is not available. Obviously, this creates a very long route with the potential consequence of a packet drop induced by the hop limit.

[12, 21] indicate that PBR performs very well for routing on highways, whereas AODV suffers from link-breaks due to the high mobility. Furthermore, PBR works well for communication over few hops in city scenarios, since recovery is not needed very often. Thus, PBR is a good choice as a basic routing scheme[2]. In recovery mode, both GPSR and GPCR use a variant of the right hand rule to route packets. Due to the reasoning outlined above, these methods are not suitable for city scenarios. The basic idea of PBR-DV is to combine a position-based greedy heuristic with a general-purpose recovery mode based on distance vectors. This method is very suitable because it finds non-greedy routes and performs well for small-hop-count communication in a city scenario (as shown with the example of AODV in [28]). Moreover, it is practically indifferent to radio obstacles, and distance vector-based approaches are well-researched and understood.

In the situation shown in Figure 7 node `v` did not find a usable—since reasonably short—route to node `u` by the application of the right hand rule. The packet is greedily

---

[2] Obviously, plain PBR would benefit from the "restricted greedy" mode proposed in [28].

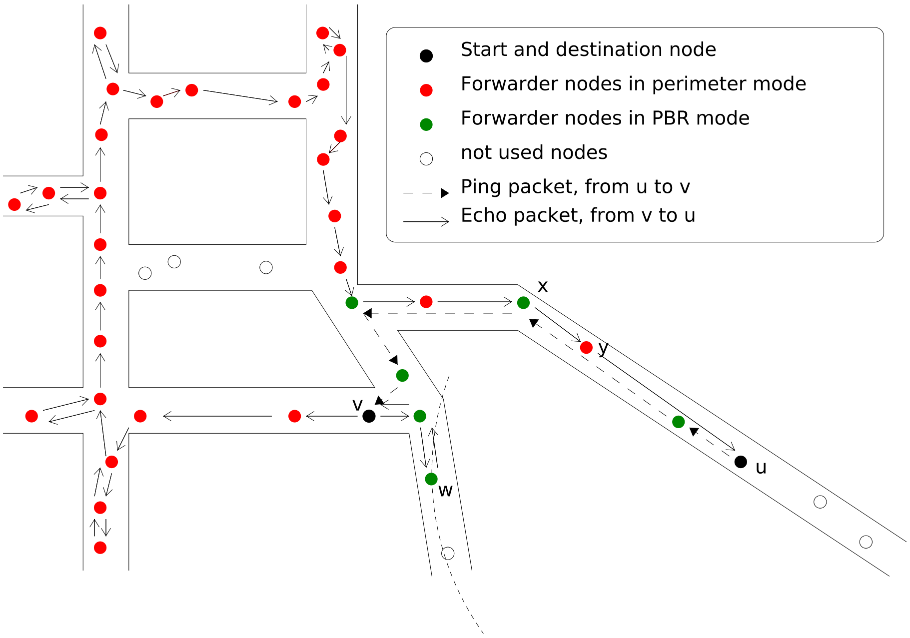


Figure 7: Problem of the Right Hand Rule (from [26])

forwarded to node w, stalling in a local optimum. The distance vector based recovery mode of PBR-DV would then start a route discovery process resulting in a topology-based route to node x. At node x, PBR-DV changes back to greedy forwarding because node x is closer to the position of node u than node w.

Altogether, PBR-DV is the combination of two well-researched and understood routing approaches. Since both methods are hybridly applied in the scenarios they perform best, i.e., greedy when possible and DV when not, we are convinced that it is highly suitable for the given task. With this, it combines the strengths of a general-purpose routing scheme like AODV with the power of geographic forwarding. Geographic unicast routing to a node or a hot spot located on the same street comes at minimal cost.

4 Conclusions: Protocol Recommendation

In this report, we have conducted a collection of ad-hoc routing methods applicable to vehicular city scenarios. Furthermore, we have analyzed them w.r.t. analytical and simulative performance properties and deduced their suitability for these scenarios. Keeping in mind the scenario restriction of a small number of hops (as explained in Section 1) plus the restrictions of routing devices to be very cheap and easily maintainable, we recommend PBR-DV, a combination of position-based (greedy) routing and AODV-like distance vector routing in case greedy routing fails. The reasons can be summarized as follows:

1. PBR is the protocol-of-choice of the FleetNet [1] project and was shown to work very well for small-hop-count scenarios, since recovery is not needed very often. However, there can be—even static—cases, where PBR will not find any route at all, making it unacceptable as a stand-alone solution.
2. PBR is performing almost perfectly on highways, only excelled by contention-based forwarding.
3. AODV shows reasonable performance in city scenarios with their low mobility. It works practically indifferently to radio obstacles and does not require any map information.
4. DV routing is well-researched and understood for mobile ad-hoc networks. Also, it can be used in a purely ad-hoc fashion.
5. While DV routing does not exploit the geometric situation in the city, it does also not require a map and is of fairly low complexity as opposed to protocols like STBR, which we expect to take over performance lead for a growing number of hops.
6. While GPCR does not need extra flooding in the recovery case as opposed to DV, it has the problem of finding the right direction to bypass the void.

While we do not have an implementation of PBR-DV, we expect the building and optimizing of that protocol to be straight-forward but time-consuming engineering work. In the context of this procedure, the following design decisions could be answered under precise assumptions the scenario dictates:

- Should recovery only go round the void or should it go from source to destination (or from void to destination)?
- Should the restricted greedy variant of GPCR go into PBR?
- What are reasonable values for soft-state parameters like time outs or beacon intervals?

Another significant advantage of PBR-DV is that it is replaceable with CBDV [24], creating the possibility to use the performance-promising contention-based forwarding methods.

References

[1] The FleetNet Project Homepage. `http://www.et2.tu-harburg.de/fleetnet/`, 2006. [Online: Accessed at 2006-04-07].

[2] The Network on Wheels Project. `http://www.network-on-wheels.de`, 2006. [Online: Accessed at 2006-04-07].

[3] Breadth-first search — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Breadth-first_search&oldid=54584556`, 2006. [Online: Accessed at 2006-05-24].

[4] Correlation — Wikipedia, the free encyclopedia. `"http://en.wikipedia.org/w/index.php?title=Correlation&oldid=53253922`, 2006. [Online: Accessed at 2006-05-24].

[5] Right-hand rule — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Right-hand_rule&oldid=46812021`, 2006. [Online: Accessed at 2006-04-21].

[6] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking (TON)*, 5(6):756–769, December 1997.

[7] L. Blažević, S. Giordano, and J.-Y. LeBoudec. Self-Organizing Wide-Area Routing. In *Proc. of SCI 2000/ ISAS 2000*, Orlando, July 2000.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc Wireless Networks. In *Proc. of ACM DIAL-M '99*, pages 48–55, Seattle, WS, August 1999.

[9] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, (1):269–271, 1959.

[10] Z. Fu, X. Meng, and S. Lu. How Bad TCP Can Perform In Mobile Ad Hoc Networks. In *Proc. of IEEE ISCC '02*, Taormina/Giardini Naxos, Italy, July 2002.

[11] D. Förderer. Street-Topology Based Routing. Master's thesis, University of Mannheim, May 2005.

[12] H. Füßler, M. Mauve, H. Hartenstein, M. Käsemann, and D. Vollmer. A Comparison of Routing Strategies for Vehicular Ad Hoc Networks. Technical Report TR-02-003, Department of Computer Science, University of Mannheim, July 2002.

[13] H. Füßler, M. Torrent-Moreno, R. Krüger, M. Transier, H. Hartenstein, and W. Effelsberg. Studying Vehicle Movements on Highways and their Impact on Ad-Hoc Connectivity. Technical Report TR-2005-003, Department of Mathematics and Computer Science, University of Mannheim, 2005.

[14] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier's Ad Hoc Networks*, 1(4):351–369, 2003.

[15] P. Gupta, R. Gray, and P. R. Kumar. An Experimental Scaling Law for Ad Hoc Networks. Technical report, University of Illinois at Urbana-Champaign, 2001.

[16] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[17] H. Hartenstein, H. Füßler, M. Mauve, and W. Franz. Simulation Results and Proof-of-Concept Implementation of the FleetNet Position-Based Router. In *Proc. of PWC '03*, pages 192–197, Venice, Italy, September 2003.

[18] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[19] B. N. Karp. Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning, May 2001.

[20] 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.

[21] T. King, H. Füßler, M. Transier, and W. Effelsberg. On the Application of Dead-Reckoning to Position-Based Routing for Vehicular Highway Scenarios. In *Proc. of CoNEXT '05*, pages 258–259, Toulouse, France, October 2005.

[22] M. Käsemann, H. Füßler, H. Hartenstein, and M. Mauve. A Reactive Location Service for Mobile Ad Hoc Networks. Technical Report TR-02-014, Department of Computer Science, University of Mannheim, November 2002.

[23] A. Leibscher. Recovery-Strategies for Beaconless, Position-Based Routing in Mobile Ad-hoc Networks. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, March 2004.

[24] A. Leibscher, J. Widmer, M. Transier, and W. Effelsberg. Contention-Based Distance-Vector Routing (CBDV) for Mobile Ad-Hoc Networks. In *Proc. of IEEE ICNP '04*, Berlin, Germany, October 2004.

[25] J. Li, C. Blake, D. S. J. DeCouto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *Proc. of ACM MobiCom '01*, pages 61–69, Rome, Italy, July 2001.

[26] C. Lochert. Ad Hoc Routing für die Kommunikation zwischen Fahrzeugen in Stadtszenarien. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, 2003.

[27] C. Lochert, H. Füßler, M. Mauve, and H. Hartenstein. Geographic Routing in City Scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 9(1):69–72, January 2005.

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

[29] D. A. Maltz, J. Broch, J. G. Jetcheva, and D. B. Johnson. The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications (special issue on mobile and wireless networks)*, 17(8):1439–1453, August 1999.

[30] D. A. Maltz, J. Broch, and D. B. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed. Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, 1999.

[31] S.-Y. Ni, T. Yu-Chee, C. Yuh-Shyan, and S. Jang-Ping. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proc. of ACM MobiCom '99*, pages 151–162, Seattle, Washington, August 1999.

[32] A. Oberhoeffken. MIRP - Map Information Routing Protocol for Mobile Ad-Hoc Networks. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, 2003.

[33] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, `http://www.ietf.org/rfc3561.txt`, 2003. Status: EXPERIMENTAL.

[34] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proc. of IEEE WMCSA '99*, pages 90–100, New Orleans, LA, February 1999.

[35] E. M. Royer and C. E. Perkins. An Implementation Study of the AODV Routing Protocol. In *Proc. of IEEE WCNC '00*, Chicago, IL, September 2000.

[36] F. Schmidt-Eisenlohr, M. Torrent-Moreno, H. Füßler, and H. Hartenstein. Packet Forwarding in VANETs, the Complete Set of Results. Technical Report TR-2006-02, University of Karlsruhe, January 2006.

[37] F. Schmidt-Eisenlohr, M. Torrent-Moreno, H. Füßler, and H. Hartenstein. Effects of a Realistic Channel Model on Packet Forwarding in Vehicular Ad Hoc Networks. In *Proc. of IEEE WCNC '06*, Las Vegas, NV, April 2006, to appear.

[38] W. R. Stevens. *TCP/IP Illustrated*, volume 1. Addison Wesley Longman, 1994.