Pervasive and Mobile Computing 5 (2009) 1–18

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Review

Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification

James Bernsen, D. Manivannan *

Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA

ARTICLE INFO

Article history:
Received 18 February 2008
Received in revised form 26 August 2008
Accepted 1 September 2008
Available online 7 September 2008

Keywords:
Vehicular ad hoc networks
VANET
Routing
Mobile ad hoc network
Inter-vehicle communication

ABSTRACT

Vehicular Ad hoc NETworks (VANETs) allow vehicles to form a self-organized network without the need for a permanent infrastructure. As a prerequisite to communication, an efficient route between network nodes must be established, and it must adapt to the rapidly changing topology of vehicles in motion. This is the aim of VANET routing protocols. In this paper, we discuss the design factors of unicast routing protocols for VANETs, and present a timeline of the development of the existing unicast routing protocols. Moreover, we classify and characterize the existing unicast routing protocols for VANETs, and also provide a qualitative comparison of them. This classification and characterization gives a clear picture of the strengths and weaknesses of existing protocols in this area and also throws light on open issues that remain to be addressed. Multicast routing protocols are also very important in VANETs; however, they are outside the scope of this paper.

Contents

* Corresponding author. Tel.: +1 859 257 9234; fax: +1 849 323 3740.
E-mail addresses: james@bernsen.us (J. Bernsen), mani@cs.uky.edu (D. Manivannan).
URL: http://www.cs.uky.edu/~manivann (D. Manivannan).

1574-1192/$ – see front matter
doi:10.1016/j.pmcj.2008.09.001

1. Introduction

Less than a century since the automobile was made affordable enough for the general public, hundreds of millions of vehicles now travel along highways and streets around the world. Innovations in safety, comfort, and convenience have made vast improvements in automobiles during that time, and now new technologies promise to change the face of vehicular travel once again. One such new technology is the vehicular ad hoc network or VANET, which provides the ability for vehicles to spontaneously and wirelessly network with other vehicles nearby for the purposes of providing travelers with new features and applications that have never previously been possible.

1.1. What is a VANET?

A VANET is a wireless network that is formed between vehicles on an as-needed basis. To participate in a VANET, vehicles must be equipped with wireless transceivers and computerized control modules that allow them to act as network nodes. Each vehicle's wireless network range may be limited to a few hundred meters, so providing end-to-end communication across a larger distance requires messages to hop through several nodes. Network infrastructure is not required for a VANET to be created, although permanent network nodes may be used in the form of roadside units. These roadside units open up a wide variety of services for vehicular networks, such as acting as a drop point for messages on sparsely populated roads, serving up geographically-relevant data, or serving as a gateway to the Internet.

VANETs are a special class of mobile ad hoc networks (MANETs) with their own unique characteristics. Most nodes in a VANET are mobile, but because vehicles are generally constrained to roadways, they have a distinct controlled mobility pattern that is subject to vehicular traffic regulations. In urban areas, gaps between roads are often occupied by buildings and other obstacles to radio communication, so communication along roads is sometimes necessary. Vehicles generally move at higher rates of speed than nodes in other kinds of MANETs. Speeds of vehicles moving in the same direction are generally similar, and they can therefore remain in radio contact with one another for much longer periods of time than with vehicles moving in the opposite direction.

Even though several unicast routing protocols have been developed for MANETs [4,5,32,6,9–12,16,18,20,28–31,33,40,41,46,52–54,57,60,61,66,70] because of the unique characteristics of VANETs, these protocols cannot be directly used in VANETs efficiently. Hence, because of the expected potential impact of VANETs, several researchers have developed unicast routing protocols that are suitable for VANETs. The effect of traffic patterns and congestion on VANETs are studied in [25,55,67,68]. Security issues in VANETs are discussed in [13,22,24,27,56]. Simulation and testing of unicast routing protocols for VANETs presents its own challenges in reproducing realistic vehicular motion on a simulated environment that is representative of real cities and roadways. These simulation issues unique to VANETs are discussed in [7,19,48,58]. Fairness of bandwidth sharing is discussed in [64].

1.2. VANET applications

Numerous applications await users of vehicular ad hoc networks, serving the interests of consumers, businesses, governments, law enforcement agencies, and emergency services. Many of these applications center on the idea of improving the safety of motor vehicles. Accident avoidance warnings could quickly notify drivers of numerous conditions that could cause a collision. When vehicles ahead brake quickly, an audio warning could alert the driver or an onboard computer might automatically apply the brakes. If vehicles on the road ahead swerve to avoid a road obstruction, the driver may be advised to change lanes. In a scenario in which a driver fails to observe a traffic signal, putting it on a collision course with a cross-street vehicle, both drivers may be alerted or corrective action may be taken by the vehicles' onboard computers. In case an accident does occur, trajectory and velocity information exchanged between vehicles prior to collision may allow the accident to be reconstructed more easily. Rescue vehicles could instantly receive exact coordinates of the location of an accident, which can help them to reach the scene of the emergency faster.

Several features that enhance the convenience of drivers could be offered through the use of VANETs. Today, GPS navigation systems can provide detour information based on reported accidents downloaded through cellular carriers for a monthly fee. It may not be possible to obtain such information in real-time through cellular carriers. When VANETs are in widespread use, information about traffic and road hazards could be acquired in real-time and fed into vehicle navigation systems to provide alternate driving routes. In such situations, reliability and authenticity of the information disseminated need to be ensured. Existing research results on ensuring reliability and authenticity in the context of MANETS need to be explored and carefully adopted to VANETs. Current electronic toll road payments are typically made via RFID tags installed in vehicles, but there are incompatibilities in the hardware used in different regions. Toll roads could be wirelessly paid without the installation of additional hardware to a vehicle, and protocol standardization [51] could make the technology ubiquitous among various regions. By utilizing the positioning information provided by GPS, geographically-oriented retrieval of local shopping and service information could be provided in real-time, just as this information is preloaded in GPS systems today. Traffic signals equipped with communication equipment could more accurately control intersection traffic, and by feeding their data into centralized systems, similar traffic control efficiencies could be applied at a more macroscopic level, such as an entire district or downtown area. Further into the future, VANETs will provide the communication network required by cooperative driving applications, which would allow vehicles to navigate without driver intervention by communicating with other vehicles about velocity, proximity, and other factors.

From radio to television and video players, the possibilities for in-vehicle entertainment have been constantly evolving, and several new applications in this area could be made possible with VANETs. One does not yet know what the killer application would be in this area. Internet access for passengers, communication with other vehicles, multimedia entertainment, and cooperative games are just a few potential entertainment services that VANETs may provide.

1.3. Popular MANET routing protocols adapted to VANETS

The Dynamic Source Routing Protocol (DSR) [33] and the Optimized Link State Routing Protocol (OLSR) [62] are well-known unicast routing protocols for MANETs and have been successfully adapted to VANETs as well. DSR is an on-demand routing protocol which searches for a route only when needed. Each node maintains the known routes in its cache. A route consists of the full source route, containing all the intermediate nodes in the route. New routes are discovered by a source by flooding the network with route request messages. When the destination receives a route request, it sends a route reply. The route reply sent by the destination accumulates all the nodes through which the route reply propagates. When the route reply reaches the source, it gets the source route to the destination from the reply. On the other hand, OLSR is a proactive routing protocol, which maintains routes between any two nodes in the network. HELLO messages are used for maintaining the routes. The main advantage of this is that each node always has a route to every other node in the network. This advantage comes as a result of large message overhead for maintaining the routes. DSR has low overhead and is suitable for networks in which not all nodes need a route to every other node in the network and the user traffic is low.

1.4. Organization of the paper

The rest of the paper is organized as follows. We discuss design factors of VANETs in Section 2. A timeline, classification and qualitative comparison of the existing unicast routing protocols for VANETs are presented in Section 3. An overview of each protocol is given in Section 4. We conclude with a discussion of open issues in Section 5.

2. Design factors

There are many factors to consider when designing a VANET. Will the network be vehicle-to-vehicle only, or could roadside units be used for communication? What forms of communication will be available? Which vehicular systems will be employed in the network? These and many other aspects will require analysis when determining the features and capabilities of a VANET.

2.1. Vehicle-to-roadside communication

New technologies that will allow vehicles to communicate with roadside units are in progress. The IEEE 802.11 working group continues to actively develop draft amendment 802.11p [2] in order to provide support for Intelligent Transportation System (ITS) applications. In the existing infrastructure and ad-hoc modes of the IEEE 802.11 wireless standard, the time required to authenticate and associate with a basic service set (BSS) is too long to be employed by VANETs. The 802.11p standard will provide wireless devices with the ability to perform the short-duration exchanges necessary to communicate between a high-velocity vehicle and a stationary roadside unit. This mode of operation, called WAVE (wireless access in vehicle environments) will operate in a 5.9 GHz band and support the Dedicated Short Range Communications (DSRC) standard [65] sponsored by the US Department of Transportation. These standards will support systems that communicate from vehicle-to-roadside, vehicle-to-vehicle, or both.

2.2. Vehicle-to-vehicle communication

Installing fixed infrastructure on roads incurs great expense, so vehicle-to-vehicle (V2V) communication will be necessary to extend the effective range of networked vehicles. VANETs require features not provided by cellular network-based systems, such as low data transport times for emergency warnings and robustness due to the network's decentralized structure [42]. In an emergency situation, cellular base stations are often overwhelmed with calls, but distributed communications have the potential for load balancing traffic to avoid network congestion.

2.3. Communication paradigms

Like other kinds of networks, different communication paradigms can be supported in VANETs. *Unicast* communication provides the ability for one node to communicate with a target node in the network. The target node may be in a precise known location or an approximate location within a specified range [69]. While unicast is a useful mode of communication in VANETs, many applications will require dissemination of messages to many different nodes in the network.

Multicast communication allows messages to be sent to multiple destinations using the most efficient route possible. For instance, when a traffic jam occurs at a particular location on a roadway, it would be valuable to send messages to vehicles that are approaching that point so that they can take alternate routes. Sending a multicast message that reaches vehicles not affected by the traffic jam would waste valuable network bandwidth. Instead, it is desirable to only target affected vehicles, whose positions can be determined by analyzing a road map. If each vehicle is equipped with knowledge of its own global coordinates, then a specialized form of multicast communication called *geocast* is possible. In geocasting, a message is sent to all of the nodes in a particular geographic position, usually relative to the source of the message. A similar form of communication is *anycast*, where a node sends a message to any destination node in a group of nodes. Anycast provides a data acquisition feature that is the intuitive inverse of geocasting, where a node sends a message to a certain geographic area to request data from any node found there, called geographical anycast. Geographic anycasting has the ability to provide "distributed floating car data" [23]. Another communication paradigm called *scan* operates like a sonar echo, sending a message that traverses a certain region once [69]. Much research [3,8,34,36,37,69] has focused on providing these various multicast services in VANETs. This paper primarily focuses on routing protocols that deliver unicast services.

2.4. Environmental constraints

VANETs operate in a very different environment than most computing applications. The high velocities at which vehicles move will sometimes reduce the amount of time available for message exchanges. Protocols will need to take advantage of vehicles moving in the same direction at relatively similar speeds to maintain connections for longer periods of time [50]. Protocols must operate well in both city roads and highways. City streets pose a unique set of geographic constraints as buildings between streets often form obstacles for radio signals that must be routed around. Road characteristics, such as traffic signals and stop signs will affect the flow of traffic in urban areas, breaking any reliable streams of similar-velocity vehicles that may be found on highways. On highways, low vehicle density must be considered. Traffic density, often measured in the number of vehicles per unit distance, has a large influence on road capacity and vehicle velocity. In low traffic densities, vehicles tend to move at faster rates, but as traffic density increases, vehicles slow down. Very high traffic density (in the case of a serious road block, for example) also causes both relative speed and distance between vehicles to become stable [50].

3. Timeline and classification

In this section, we present a timeline of unicast routing protocols for VANETs and a qualitative comparison of them. For the reader's reference, publication dates for each protocol surveyed were used to build the timeline shown in Fig. 1. In addition, where a paper cited another protocol as a reference point for its own improvements, Fig. 1 captures this dependency with an arrow from the predecessor protocol to the potentially influenced protocol.

In Fig. 2, we give a qualitative comparison of the existing VANET unicast routing protocols. We have classified VANET unicast routing protocols based on three sets of criteria: objectives, characteristics, and assumptions. In the set of objectives criteria, we categorize based on the design goals of each protocol. Some protocols are aimed at providing *vehicle-to-vehicle* services, while others focus on vehicle-to-roadside communication. Other protocols intend to provide communication in *delay tolerant/sparse* networks. In contrast, there are *QoS* (Quality of Service) oriented protocols, some of which provide *Internet connectivity* to vehicles.

In the set of characteristics criteria, we categorize based on the various strategies used by each protocol to achieve its objectives. Nearly all of the protocols are *position-based*, using knowledge of vehicles' positions and velocities to route messages. Many also utilize the *greedy forwarding* strategy for sending messages to the farthest neighbor in the intended direction. We also observe several *predictive* approaches, where some speculation is made about characteristics of the nodes involved in a route. Some algorithms make predictions on the current locations of nodes based on the last known position, and velocity of the node. Other algorithms use this same information to make predictions about the stability or estimated

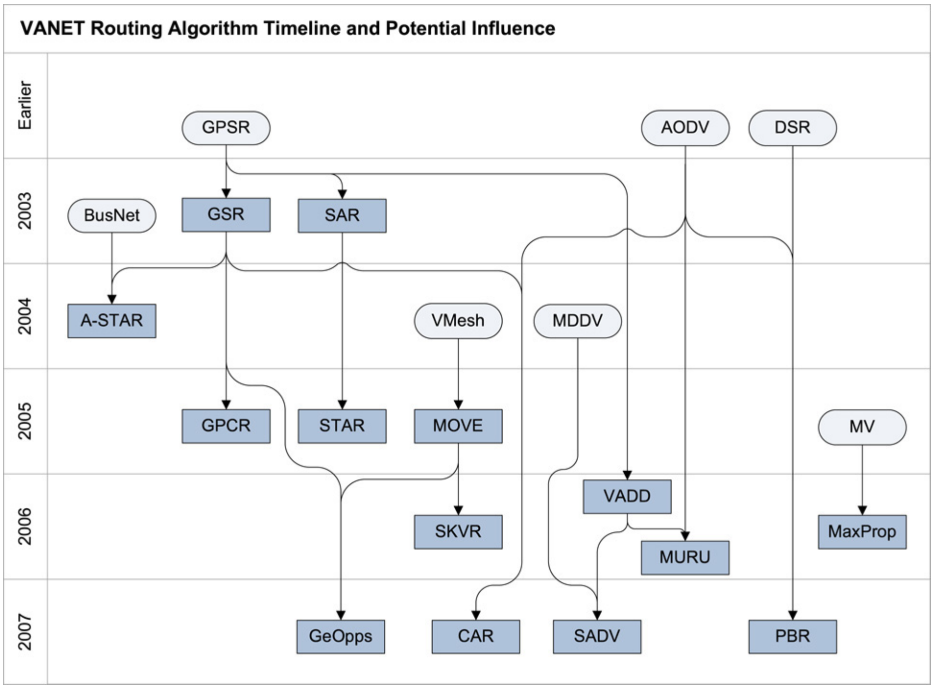

Fig. 1. VANET unicast routing protocols timeline and potential influence.

		Objectives				Characteristics												Assumptions						
		vehicle to vehicle	delay tolerant / sparse	QoS	Internet connectivity	position-based / geographic	greedy forwarding	predictive	buffering (carry-and-forward)	street-aware	traffic-aware (probabilistic)	traffic-aware (real-time)	node-anchored routes	position-anchored routes	route-repair or recovery	route caching	geographic marker messages	map required	positioning system required	location service required	transport routes required	transport schedules required	traffic data required	mobile gateways required
Position-Based Greedy V2V	GSR	✓				✓	✓			✓				✓	✓			✓	✓	✓				
	SAR	✓				✓	✓		✓	✓				✓	✓			✓	✓	✓				
	A-STAR	✓				✓	✓			✓	✓			✓	✓		✓	✓	✓	✓	✓			
	STAR	✓				✓	✓			✓		✓		✓	✓			✓	✓	✓				
	GPCR	✓				✓	✓			✓				✓	✓				✓	✓				
	CAR	✓				✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓					
Delay Tolerant	MOVE		✓			✓		✓	✓										✓					
	SKVR	✓	✓						✓												✓	✓		
	VADD		✓			✓	✓	✓	✓	✓	✓							✓	✓	✓			✓	
	SADV		✓			✓	✓		✓	✓		✓		✓				✓	✓	✓				
	GeOpps		✓			✓	✓	✓	✓	✓								✓	✓	✓				
	MaxProp	✓	✓			✓		✓	✓										✓					
QoS	MURU	✓		✓		✓		✓	✓	✓					✓	✓		✓	✓	✓				
	PBR	✓		✓	✓	✓		✓					✓		✓	✓			✓					✓

Fig. 2. Qualitative comparison of VANET unicast routing protocols.

lifetime of a route. To provide higher rates of delivery in sparse networks, a *buffering (carry-and-forward)* strategy is often used. In this strategy, a node may hold a packet in a local buffer until a forwarding opportunity is available, instead of simply dropping the packet.

We adopt the term *street-aware* to imply that the protocol builds route paths along streets, whether an external map is used to perform this function or not (this is noted independently in the assumptions criteria). We use a similar term, traffic-aware, to refer to a protocol's ability to utilize traffic information to select an efficient route. We divide the traffic-aware algorithms into two groups: (1) *traffic-aware (probabilistic)*, which includes those protocols that make probabilistic assumptions about traffic density by using static knowledge such as bus routes and lane information, and (2) *traffic-aware (real-time)*, which includes those that determine traffic density by real-time measurement. In establishing routes, we can categorize the unicast routing protocols based on whether hops on the route are *node-anchored* (by remembering the exact vehicles used in a route) or *position-anchored* (remembering approximate geographic positions used in the route).

The criterion *route-repair or recovery* refers to protocols which either use a strategy to recover from a greedy local optimum in a position-based route or have a mechanism for repairing broken routes. We also denote *route caching* protocols which remember routes that have been previously established. We observe that a small number of protocols include the use of what we call *geographic marker messages*. These are messages that are passed from one vehicle to another to proliferate previously discovered information about a geographic location. They may indicate a location that is disconnected [59] or they may provide forwarding information for a relocated node like the "guards" in [49].

In the final set of criteria, assumptions, we identify the external sources of information that each protocol depends upon for its operation. These sources of information include external maps, positioning systems (such as GPS), and location services. Some protocols take advantage of public transport routes and schedules to approximate the positions of those vehicles. Traffic data from external sources may be utilized, and one protocol [47] requires the presence of "external gateways", which are vehicles that have a wireless WAN connection to the Internet.

4. Overview of protocols

4.1. Position-based greedy V2V protocols

A significant number of unicast routing protocols use a position-based, greedy approach to provide vehicle-to-vehicle communication. Position-based approaches use information about the geographic coordinates or relative positions of nodes to generate an efficient route through the network. Like most VANET unicast routing approaches, position-based routing allows for unicast communication, but it also allows for delivery of messages to all nodes in a geographic area, called geocasting.

When feasible, position based routing is beneficial, in that no global route from a source node to a destination node need be created and maintained: each node decides where to forward each packet it receives. It has been shown that routing approaches using position information can adapt to the high mobility of nodes found in highway situations [25].

In the greedy forwarding strategy, an intermediate node in a route forwards messages to the farthest neighbor in the direction of the next anchor or the destination. This approach requires the intermediate node to have three important data points: the position of itself, the position of its neighbors, and the position of the destination. Typically, a node's own position is acquired through GPS. (All of the protocols in this category assume a positioning system of some kind.) The node's neighbors' positions are obtained through message exchanges, and the position of the destination node is usually found through the use of a location service [43]. Location servers may be periodically placed external to the system, but this offers no guarantee that such a server will be within range. To alleviate this problem, quorum-based location services may be built into nodes, or fully-distributed location services similar to AODV route request may be utilized [44].

4.1.1. Geographic source routing

The geographic source routing (GSR) algorithm [42] tries to overcome the disadvantages of position-based routing approaches designed for MANETs when applied to VANETs in urban scenarios. For example, consider the position-based MANET algorithm GPSR [35]. GPSR utilizes the strategy of greedy forwarding of messages toward a known destination. If, at any hop, there are no nodes in the direction of the destination, then GPSR has a recovery strategy called perimeter mode that routes around this void. The perimeter mode has two components. The first component is a distributed planarization algorithm that makes a local conversion of the connectivity graph into a planar graph by eliminating redundant edges. The second component is an online routing algorithm that operates on planar graphs.

In urban areas, buildings and other radio obstructions restrict effective network routes to run along streets, so in VANET scenarios, the perimeter mode of GPSR would be frequently required. In GPSR, if a radio obstruction or void causes the algorithm to enter perimeter mode, then the planar graph routing algorithm begins operating. While attempting to route messages around the obstruction, the planarized connectivity graph causes messages to be sent to immediate neighbors, instead of sending messages to the farthest node on the street in the perimeter route. Because this routing method uses immediate neighbors instead of the farthest reachable nodes, more nodes will carry the message, causing increased delays and greater hop counts. In addition, with the rapid movement of vehicles inherent to VANETs, routing loops can be introduced while in the perimeter mode of GPSR, causing further inefficiencies. Even worse, these routing loops can cause messages to be transmitted in the wrong direction by persistently following the right hand rule. For example, consider five nodes a, b, c, d, and e arranged clockwise in a semicircular pattern. Node a attempts to send a message to e, but an obstruction blocks the path. Perimeter mode is entered, following the right-hand rule, and the message travels $a \to b \to c \to d$, but during this sequence, d moves within range of b. When link (d, b) forms, b becomes the next destination by the right-hand rule, and a loop is formed.

Geographic source routing [42] uses a map of the urban area to avoid these problems. Using a static street map and location information about each node, GSR computes a route to a destination by forwarding messages along streets. The sender of a message computes a sequence of intersections that must be traversed in order to reach the destination. This sequence of intersections can be placed in the packet header or they can be decided by each forwarding node. The path between the source and destination is computed using Dijkstra's shortest-path algorithm.

GSR is a seminal algorithm for position-based VANET unicast routing. However, the vehicle density on a chosen street is not considered, and the authors acknowledge that there is potential to improve the algorithm by considering this information. The algorithm also requires externally-provided static street maps for operation.

4.1.2. Spatially aware packet routing

Similar to GSR, the Spatially-Aware Packet Routing (SAR) protocol [63], attempts to overcome some of the weaknesses of the recovery strategy used by GPSR. In particular, when GPSR reaches a local optimum at a network void, its stateless recovery strategy will cause each packet reaching the void to engage in its recovery strategy as well. If the network void is a permanent one, then the frequent use of the recovery strategy will degrade the algorithm's performance. SAR is a position-based unicast routing protocol that predicts and avoids route recovery caused by permanent network voids. SAR relies upon the extraction of a static street map from an external service such as GIS (Geographic Information Systems) to construct a "spatial model" for unicast routing.

In SAR, a node determines its location on the "spatial model" and uses the street information to calculate a shortest path to a packet's destination. When this path is determined, the set of geographic locations to be traveled is embedded into the header of the packet. Note that no intermediate nodes are included in the source route; it is position-anchored: a route based on immobile physical locations. When a node needs to forward a packet, it inspects the packet header for the next geographic location in the route. Rather than utilizing a strictly greedy strategy toward the destination, a neighbor that is located along the route listed in the packet is chosen.

When a forwarding node cannot find a node along the predetermined routing path in SAR [63] it is suggested that one of the following strategies could be employed. First, the node can choose to place the packet in a "suspension buffer" where it will remain until a suitable node is located along the routing path. Second, the node can attempt to greedily forward a packet towards its destination. Finally, the node may recompute the unicast routing path stored in the packet header. However, none of the suggested recovery strategies is recommended, nor is it specified how the node might select one of these strategies. The authors compared SAR with no recovery strategy and SAR using the suspension buffer approach (SARB) in their simulations and noted that SARB provides a higher packet delivery ratio at the expense of message delay in the presence of sparse networks.

4.1.3. Anchor-based street and traffic aware routing

Unlike other greedy position-based unicast routing protocols, Anchor-Based Street and Traffic Aware Routing (A-STAR) [59] utilizes city bus routes as a strategy to find routes with a high probability for delivery. A-STAR also aims to remedy the problem where the perimeter mode of GPSR utilizes next-neighbor hops along a street instead of selecting the farthest neighbor along a street for a next hop. In addition, the authors of A-STAR note that the use of the right-hand rule at a local optimum is inefficiently biased toward one direction, while going in the opposite direction (following a left-hand rule instead) will sometimes result in a shorter route to a destination.

The A-STAR algorithm uses "anchor-based" unicast routing, which involves inserting a sequence of geographic forwarding points into a packet, through which the packet must travel on its route to the destination. Like GSR, A-STAR also utilizes a static street map to route messages around potential radio obstacles, such as city buildings. In order to take advantage of the fact that some streets contain denser traffic than others, the A-STAR algorithm also utilizes bus route information. The assumption here is that buses will typically follow major streets, and major streets are more likely to contain vehicle traffic in greater densities. A weight inversely proportional to the number of bus routes that serve that street is assigned to each street. This information is used to compute an anchor path using Dijkstra's least-weight path algorithm.

When reaching a local optimum, the recovery mode of A-STAR works differently than that of its predecessors. In recovery mode, A-STAR computes a new route along an anchor path from the node at which the local optimum was reached and rewrites the path stored in the packet header. To prevent other packets from reaching the same local optimum, A-STAR uses geographic marker messages. The street that led to the local optimum is temporarily marked as "out of service" by piggy-backing this information along with the message that reached the local optimum. Any node that receives a message that contains the "out of service" information will update its local map so that "out of service" routes are not used for computing anchor paths. These routes remain inoperative until a timeout duration threshold is met.

In their simulations of A-STAR, the authors assume that buses are equipped to act as communicating nodes; a noteworthy assumption, since this would insure that a city would have additional vehicular nodes traveling along regular paths. The addition of bus schedules into the algorithm is noted as potential future work. The authors observe that estimating the traffic density of a street based on bus route information is less optimal than more dynamic approaches that utilize latest traffic condition information, and they also leave the details of this for future work.

4.1.4. Spatial and traffic aware routing

In designing the Spatial and Traffic Aware Routing (STAR) algorithm [26], the drawbacks of the SAR algorithm were observed. The SAR algorithm had the advantage of its underlying spatial model, allowing it to forward packets along streets. However, SAR did not have knowledge of whether any vehicles were actually positioned along the streets it selected. The STAR algorithm is designed to overcome this problem by only forwarding packets along streets that are occupied by vehicles. STAR also computes its route more lazily than SAR by providing a partial route in the packet header and then relying on intermediate nodes to provide additional segments of the route. This provides the advantages of a fixed packet header length and a route that can be adjusted dynamically to accommodate accurate local traffic information and the movement of the destination.

The traffic monitoring component of STAR is interested in determining two primary conditions: streets that contain dense traffic and streets that contain sparse or non-existent traffic. Beacon messages are utilized to observe ''node neighborhoods'' and this information is maintained in a neighbors-table stored at each node. The neighbors-table contains the position of each neighboring node. The neighbors-table and two dependent data structures called the ''presence vector'' and the ''persistence vector'' are used to determine sparse and dense traffic conditions relative to each node. The presence vector counts the number of neighboring nodes in each of the cardinal directions, and when a value in this vector reaches a low or high threshold, this event causes an update to the persistence vector. The persistence vector is used to capture sparse and dense traffic situations that have persisted for a substantial amount of time while ignoring temporary abnormal conditions. When the persistence vector shows that traffic is significantly sparse or dense in any direction for a considerable period of time, this observation is recorded in the node's traffic-table. Each node periodically sends a beacon message that includes its ID, its position, and information from its traffic-table.

The routing and forwarding component of STAR computes routes on-demand, utilizing information from the node's traffic-table. Each node dynamically maintains a weighted graph of street map and traffic information. Streets with dense traffic are represented by edges with lesser weights, making them preferred for routing. When a source node s wishes to send a message to a destination node d, it computes part of the route toward d using its weighted graph. When the end of that route is reached, the node where the route ends computes another route toward d, stores this in the packet header, and continues forwarding the packet. If a route fails (due to sparse traffic or other reasons), it is handled as if the end of the route was reached, and the node at which the route failed computes a new route to the destination from its weighted graph.

The STAR algorithm's reliance on beacon messages may introduce scalability and wasted bandwidth problems since there appear to be no heuristics for adapting the beacon to conditions such as high node density or network congestion. Furthermore, the authors show that the size of beacon messages in STAR are approximately six times the size of beacons in GPSR and SAR. It is also unclear how STAR maintains its presence vector when moving at high speed in one direction for an extended period of time, such as on a freeway. For example, if a node is moving east at a high velocity, it appears that over time the presence vector may become skewed such that it appears that traffic is sparse to the east and dense to the west. It is true that a node's traffic table is updated via beacon message exchanges in STAR, but a node moving at high velocity has less time to exchange large messages, thus exacerbating the possibility of a skewed presence vector.

4.1.5. Greedy perimeter coordinator routing

Like GSR, the Greedy Perimeter Coordinator Routing (GPCR) algorithm [43] takes advantage of the fact that city streets form a ''natural planar graph''. This algorithm improves upon GSR by eliminating the requirement of an external static street map for its operation. GPCR consists of two components: a restricted greedy forwarding procedure and a repair strategy for routing along streets towards a destination when the greedy strategy reaches a local optimum. Both algorithms avoid graph planarization by utilizing the innately planar graph of city streets.

To avoid potential radio obstacles such as buildings, in GPCR, the typical destination-based greedy forwarding strategy is modified such that it only routes messages along streets. In this way, routing decisions are only made at street intersections. As such, the goal is to forward messages to nodes at an intersection, rather than forwarding them to a node that is already past the intersection. (Contrast this to the traditional greedy strategy that always forwards to the farthest node in the direction of the destination.) However, since this algorithm does not require an external static street map, there is a problem in determining which nodes are located at intersections. GPCR solves this problem by defining two heuristic methods for determining whether nodes are in an intersection, and it designates those nodes as ''coordinators''. A coordinator has the responsibility of making routing decisions, and it periodically broadcasts its role along with its position information. Nodes can determine whether they are coordinators in one of the following two ways.

The first approach for coordinator determination is known as the neighbor table approach. In this approach, nodes periodically transmit beacon messages which contain their position information and the last-known position information of all of their neighbors. By listening to these beacon messages, a node has information about its own position, the position of its neighbors, and the positions of its neighbors' neighbors. Using this information, a node x considers itself to be within an intersection if it has two neighbors y and z that are within transmission range to each other but neither of them lists the other as its neighbor. Such a situation implies that y and z are separated by an obstacle and that node x can forward messages around this obstacle.

The second approach (the correlation coefficient approach) for coordinator determination does not use beacon messages. Instead, each node uses its position information and the position information of its immediate neighbors to calculate the correlation coefficient, ρ_{xy}, of its position with respect to its neighbors. A correlation coefficient indicates the strength and direction of a linear relationship between two independent variables. A strong linear correlation between the positions of the node's neighbors (i.e., ρ_{xy} is close to 1) indicates that the node is on a street. If there is no linear correlation among the positions of the node's neighbors (i.e., ρ_{xy} is close to 0), this indicates that the node is within an intersection. Through adjustment of a threshold ε, a node can evaluate the correlation coefficient and assume that $\rho_{xy} \geq \varepsilon$ indicates the node is on a street, and $\rho_{xy} < \varepsilon$ indicates the node is at an intersection. Performance studies show that this correlation coefficient approach performs better than the neighbor table approach, therefore GPCR only utilizes this strategy.

When a forwarding node that is not a coordinator forwards a message from its predecessor, it considers only those neighbors whose paths are approximate extensions of the line between the forwarding node's predecessor and the forwarding node itself. If none of the considered nodes are coordinators, then the farthest of these nodes is chosen as the next hop. If however, some of the considered nodes are coordinators, then one of the coordinators is chosen at random to be the next hop. Since GPCR avoids radio obstacles by only sending messages along streets, once a packet reaches a coordinator, a routing decision can be made. The decision about which street should be followed is performed using a greedy strategy once again. The street occupied by the neighboring node that is closest to the destination is selected.

By using the correlation coefficient approach for coordinator determination, the algorithm can avoid any dependency on an external street map. However, despite the improvements of using calculated street information for the greedy strategy, scenarios will still occur where the greedy strategy reaches a dead-end in a local optimum. This is where the repair strategy is exercised. Like the forwarding procedure, the repair strategy makes routing decisions at coordinator nodes in intersections, but it continues to use greedy forwarding between coordinators (to eliminate the extra hops noted in GPSR above). To determine which street to follow at an intersection, the repair strategy uses the well-known right-hand rule. Using this rule, the coordinator node chooses the street that is the next one counter-clockwise from the street that the packet arrived on. Since streets do not move, the problems that plagued GPSR in VANET scenarios (creating routing loops and sending packets in the wrong direction) are avoided. When a node is reached whose distance to the destination is less than the distance from the destination to the node where the repair strategy began, the greedy strategy is resumed.

The authors acknowledge that, like GPSR, GPCR does not account for low node density on selected streets, and they plan to augment the algorithm with "a very low overhead proactive probing scheme" for determining whether a selected street will allow another intersection to be reached.

4.1.6. *Connectivity-aware routing protocol*

Like other position-based vehicular routing protocols, the Connectivity-Aware Routing (CAR) protocol [49] finds a route to a destination, but a unique characteristic of CAR is its ability to maintain a cache of successful routes between various source and destination pairs. This characteristic was prompted by observations of other position-based routing protocols, and their inability to utilize information gathered about disconnected paths (due to unoccupied streets, for instance) after those disconnections are detected. CAR also predicts positions of destination vehicles, repairs routes as those positions change, and employs geographic marker messages.

Nodes using the CAR protocol send periodic HELLO beacons that contain their "velocity vector" information (heading and speed). Upon receiving a HELLO beacon, a node will record the sender in its neighbor table and calculate its own velocity vector and the velocity vectors of its neighbors. Entries expire from the neighbor table when distance between nodes exceeds a threshold or after two HELLO intervals. The beaconing interval adapts to traffic density by increasing in frequency when there are few neighbors (to maintain contact in sparse traffic) and decreasing in frequency when neighbors are abundant (to reduce unnecessary message overhead). Beacons can also be piggybacked on forwarded data packets. These measures help to reduce wasted bandwidth and network congestion.

The CAR protocol establishes the notion of a "guard", a geographic marker message that is buffered and passed from one vehicle to another to proliferate forwarding information about a node that has moved to a new location. A guard is a temporary message that has an ID, a TTL (time-to-live) counter, a radius, and some state information (usually about the velocity vector of a node, but other types of information could be stored). CAR provides two different forms of guards: the "standing guard" and the "traveling guard". A standing guard is tied to specific geographic coordinates, while a traveling guard has initial coordinates, initial time, and a velocity vector. A guard, which is kept alive by nodes in its geographic area, is transmitted as part of the nodes' HELLO beacons, and is stored in a Guards-Table by any node that receives such a beacon. To maintain traveling guards, nodes use the guard's initial coordinates, time, and velocity vector to compute the guard's current geographic area. A node that is aware of a guard can use the information to ensure delivery of messages to destination nodes that have moved.

Velocity vectors are considered parallel if the smallest angle between them is less than a configurable value, and non-parallel otherwise. Since CAR does not use external maps, this information is useful to determine when a message travels around a curve or changes course in an intersection. When an intermediate node's velocity vector is not parallel with the previous message-bearer's velocity vector, an "anchor" is added to the PD message to indicate that the message changes direction in this location. The anchor contains the coordinates and velocity vectors of the current and previous nodes. When the PD message reaches the destination, it contains a series of anchors to mark the path from the source.

After receiving one or more PD messages, the destination chooses a path with high connectivity and low delay. It then sends a RREP back to the source node, containing its coordinates, its velocity vector, and information from the original PD message. Advanced Greedy Forwarding (AGF) is used to send the RREP back to the source. When it reaches the source, the route is recorded by the source node, and communication begins. Communication over the established path also utilizes a slightly modified AGF strategy. Instead of always choosing the farthest neighbor in the direction of the destination, the farthest neighbor in the direction of the next anchor point is chosen.

To maintain the established routing paths as the vehicles in the network change positions, guards are utilized to avoid the repetition of the costly discovery phase. If a node at a route endpoint changes its direction, then the node activates a guard that contains its old and new velocity vectors. Depending on the values of the old and new velocity vectors, the guard may be a standing guard that remains at the point of direction change, or a traveling guard that moves in the direction of the old velocity vector. If a node that is aware of the guard receives a message addressed to the relocated node, it will add the guard coordinates as an anchor point to the message, calculate a new estimated position for the destination, and forward the message. The endpoint also sends a notification message to the other endpoint of the route, to inform that node of the new anchor points along the route. Notification messages may also be sent if either endpoint passes an anchor point, crosses the established routing path, or if the endpoint's change in velocity may cause a substantial position shift between its estimated and actual positions.

Routing errors may occur due to communication gaps between anchor points or due to guards that were not maintained due to low traffic density. When routing errors occur, CAR has two recovery strategies to cope with the problem. The first recovery strategy is called "timeout algorithm with active waiting cycle". In this mode, the forwarding node places the packet in a suspension buffer and periodically checks to see if a potential next-hop neighbor has entered into range. To do this, it sends a "non-propagating next-hop request" to inform other nodes that it has detected a disconnection and tries to find a next-hop node. Upon receiving this request, which contains the forwarder's coordinates and the next anchor point in the route, a node may identify itself as being between those locations and reply with a HELLO beacon. The second recovery strategy is called "walk around error recovery". In this mode, the forwarding node informs the source node of the disconnection and initiates a local path discovery process. The node also initiates a guard to declare that it is a buffering node, and it receives and buffers all messages on this route. If the local path discovery fails, this information is sent back to the source node, the buffered messages are either dropped or returned to the source, and the source starts a new path discovery. However, if the buffering node's path discovery was successful, then the new path is sent to the source node and the buffered messages are delivered.

The CAR protocol's ability to generate a virtual infrastructure in the form of guards give it a distinct advantage over other protocols. It provides street awareness, performs basic traffic awareness during path discovery, maintains routes, adapts well in low traffic densities, and it does not require map or location services.

4.2. Delay tolerant protocols

In urban daytime scenarios, where vehicles are densely packed, locating a node to carry a message is not typically a problem. But in rural highway situations or even in cities at night, fewer vehicles are running, and establishing end-to-end routes may not be possible. Even in densely-populated urban scenarios, sparse sub-networks can be prevalent. Law enforcement, military, and financial armored vehicles may each wish to exchange data privately within their own vehicular networks, due to the sensitivity of the information exchanged. One existing application is observed in the VMesh Demand-Response project [17] where utility pricing information is exchanged from roaming utility vehicles for utility usage information from consumer homes.

In such cases, some consideration needs to be given to routing in sparse networks. In particular, in the early stages of vehicular networks, when few vehicles are equipped with wireless transceivers, networks will frequently be sparse. In these situations, delay-tolerant routing algorithms are needed. Delay-tolerant networks are sometimes also known as disruption-tolerant networks.

4.2.1. Motion vector routing algorithm

In routing a message from a vehicle to a roadside unit, the Motion Vector (MOVE) routing algorithm [38] uses knowledge of neighboring vehicles' velocities and trajectories to predict which vehicle will physically travel closest to a fixed message destination. This algorithm assumes a sparser network where rare opportunistic routing decisions must be made predictively.

The MOVE algorithm is an algorithm for specialized sparse VANET scenarios. In these scenarios, vehicles act as mobile routers that will have intermittent connectivity with other vehicles or stations in their network. The network is so sparsely populated that there is seldom, if ever, a completely connected path from source to the static destination (a fixed data collection point or roadside unit). A carry-and-forward approach is used for vehicles to store data for long periods between connections. Connection opportunities must be scrutinized carefully since they occur infrequently and the global topology is unknown and rapidly changing. At each opportunity, the algorithm must predict whether forwarding a message at that instant will provide progress toward the intended destination.

Under the MOVE algorithm, it is assumed that each node has knowledge of its own position and heading, and it is assumed that the message destination D is a fixed location that is known globally. From that information, the current vehicle node C can calculate d_C, the closest distance between the vehicle and the message destination, D, along its current trajectory. The current vehicle node periodically sends a HELLO beacon. If a neighbor vehicle N in the network receives the beacon, it replies with a RESPONSE message to make itself known. Given the direction that the neighbor vehicle N is heading, the current vehicle C determines the shortest distance d_N to the destination D along the neighbor vehicle's trajectory. The current vehicle then makes a forwarding decision based on this information, each vehicle's current distance from the destination, and rules about each vehicle's trajectory relative to each other and to the destination. (One example rule: if the current vehicle is moving away from the destination and the neighbor vehicle is moving toward it, forward the message.)

In sparse networks, data delivery rates were higher for the MOVE algorithm than a greedy, position-based routing algorithm. The MOVE algorithm also used less system-wide buffer space. However, the authors noted that, in a preliminary performance evaluation where vehicle routes were consistent and uniform (bus routes), a greedy, position-based routing algorithm performed better than MOVE. Further work is planned in this area to take advantage of knowledge of the bus routes and schedules to optimize the algorithm's performance. The MOVE algorithm is designed for sparse networks and for vehicles that transfer data from sensor networks to a base station.

4.2.2. Scalable knowledge-based routing

Observing that knowledge-based schemes, such as MOVE, can improve the effectiveness of a vehicular network, the Scalable Knowledge-Based Routing (SKVR) algorithm [1] utilizes the relatively predictable nature of public transport routes and schedules. In particular, bus routes are used. Position knowledge of each bus is not required. SKVR assumes that bus routes are not looped routes; they have a start and an end, between which they travel forward and then backward.

SKVR divides the network into domains such that each bus route is a separate domain. The algorithm works on two hierarchical levels: the top level is inter-domain routing, where a source and destination are on different bus routes, while the bottom level consists of intra-domain routing within a bus route. Even the most dependable public transport system is not completely predictable, as vehicles are prone to delays due to traffic, mechanical failures, driver habits, road conditions, or other factors that may cause them to deviate from their routes and schedules. Because of this, SKVR makes use of two types of knowledge: static knowledge that does not change over time and dynamic knowledge that changes often. For inter-domain routing (among bus routes), SKVR relies on static knowledge only, namely the bus routes themselves and their intersections, which are fixed and do not change often. For intra-domain routing (when the source and destination are in the same bus route), SKVR uses both static and dynamic knowledge. The dynamic knowledge used in intra-domain routing is the schedule of the bus time tables, which may often be prone to variations, due to the aforementioned circumstances.

For intra-domain routing, when a source node wants to send a message to a destination node that is within the same physical route, there are only two directions in which to send the message, forward or backward. To determine the destination node's position without location information, a simple method is used: each vehicle maintains a list of other vehicles it has encountered that are also traveling along the same route, clearing the list when it reaches the end of the route. Each vehicle also transmits its direction (a single Boolean value) along the route as it travels. If the destination is in this previous-contact list, then it must be in the backward direction, and so the message's direction flag is marked accordingly. When vehicles along the same route encounter one another, a node carrying a message must decide whether to continue buffering the message, or to forward it, based on the direction information of the other vehicle. In this way, messages are passed to vehicles traveling along the route in the direction of their direction flag until the destination is reached. Eventually, each vehicle along the current route will reach the end of the route, at which time it continues back down the route in the reverse direction. This ensures that even when nodes are temporarily disconnected, their buffered messages will eventually be delivered when they cross paths with another vehicle in their route.

For inter-domain routing, SKVR forwards a message to a vehicle traveling in the destination domain, or to a vehicle that will eventually make contact with that domain. When the destination's domain is reached, the intra-domain behavior can complete the delivery. If the sending vehicle's contact list does not contain any vehicle in the destination's domain, then copies of the message are sent to other vehicles in the contact list.

4.2.3. Vehicle-assisted data delivery

The Vehicle-Assisted Data Delivery (VADD) protocol [71] uses a "carry-and-forward" strategy to allow packets to be carried by vehicles in sparse networks for eventual forwarding when another appropriate node enters the broadcast range, thereby allowing packets to be forwarded by relay in case of sparse networks. VADD requires each vehicle to know its own position and also requires an external static street map that includes traffic statistics.

With the VADD protocol, when wireless channels are not available, the carry-and-forward strategy is used by transferring packets along vehicles on the fastest roads available. Since vehicles may deviate from predicted paths, the routing path should be recomputed continuously during the forwarding process. To aid in this process, VADD uses a street graph weighted with expected packet delivery delays.

Each packet has three modes: Intersection, StraightWay, and Destination, where each mode is based on the location of the node carrying the packet. Appropriately, the *Intersection* mode is used when the packet has reached an intersection, at which routing decisions can be made for the packet to be forwarded to a vehicle along any of the available directions of the intersection. In *StraightWay* mode, the current node is on a road, and there are only two possible directions for the packet to travel: in the direction of the current node or in the opposite direction. Both directions are considered using a simplified version of the algorithm found in Intersection mode, and when the optimal direction is chosen, packet forwarding occurs in a greedy manner. *Destination* mode is briefly entered when the packet is close to its final destination.

The most complex mode is the Intersection mode because of the many decisions that are made while in this mode. Within an intersection, the forwarding node prioritizes the available streets (based on distance to destination, expected packet delivery delay, and probability of packet delivery) and chooses the next intersection in the highest prioritized direction as the "target intersection". It then attempts to select a candidate node for that intersection. If no candidate node can be found for the current target intersection, the forwarding node chooses the next best direction, chooses a target intersection in that direction, and resumes the strategy of finding a candidate node for that intersection. If no nodes are found in any suitable direction, the forwarding node carries the packet.

When choosing a candidate node to which a message will be forwarded, there are several variations of VADD with different selection criteria. Location First Probe VADD (L-VADD) selects nodes that are located closely to the target intersection, regardless of the direction the node is traveling. In this way, L-VADD is similar to the position-based greedy strategy: choose the farthest node in the routing direction. Because the selection of the next hop node is based solely on proximity to the target intersection, as nodes approach an intersection, L-VADD has the potential for causing routing loops. For instance, consider node *A* that selects north as the highest prioritized direction for its packet. There are no nodes to the north of *A*, and east is chosen as the next best direction, so as *A* approaches an intersection to the east, it may forward its packet to another node *B* that is southeast of *A* and closer to the eastern intersection. However, when *B* receives the packet, it determines that north is the best direction, and *A* is further north than *B*, so it sends the packet back to *A*. These simple loops can be avoided by recording the previous hop and not returning a packet to that node, but this method is not scalable for larger loops, and will result in making many valid forwarding paths not usable in order to prevent loops. Another variation of the protocol, Direction First Probe VADD (D-VADD), prefers nodes that are moving in the same direction as the highest priority route direction. This strategy eliminates the possibility of routing loops, because all nodes agree that a carrying node should always be a vehicle traveling in the direction with highest priority. However, due to the possibility that a node may not encounter the most optimal transfer node immediately upon entering an intersection, the authors introduced Multi-Path Direction First Probe VADD (MD-VADD), which is a multi-path variant of D-VADD. MD-VADD is more likely to find the most optimal routes, but it also wastes bandwidth by duplicating packets among multiple receivers.

Hybrid Probe VADD (H-VADD) is a hybrid protocol that harnesses the strengths of L-VADD, D-VADD and MD-VADD. Upon entering an intersection, H-VADD operates in the greedy manner of L-VADD. However, if a routing loop is detected, D-VADD or MD-VADD is used in the current intersection instead. The source paper is not clear on how D-VADD or MD-VADD is selected in the presence of a routing loop. In simulations, H-VADD shows the best performance in terms of delivery ratio, message delay, and packet loss.

VADD makes an interesting assumption that vehicles with GPS capability are generally programmed with long-range trajectory information (for example, to map driving directions to a destination for the driver). VADD proposes that this trajectory be sent along with a packet, so that when a return packet is sent, the intermediate nodes can continually update the predicted location of the packet's destination. The authors leave the details for this idea as future work.

4.2.4. Static-node assisted adaptive routing protocol in vehicular networks

The Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks (SADV) [21] aims at reducing message delivery delay in sparse networks. In experiments using VADD, the authors observed that the network became unstable as vehicle density decreased, because optimal paths were not available and because VADD relies upon probabilistic traffic density information. To remedy this, the authors proposed the idea that static nodes could be placed at intersections to assist with packet delivery. Each static node has the capability to store a message until it can forward the message to a node traveling on the optimal path. SADV also dynamically adapts to varying traffic densities by allowing each node to measure the amount of time for message delivery. SADV assumes that each vehicle knows its own position through GPS and that each vehicle has access to an external static street map.

SADV has three different modules: "Static Node Assisted Routing" (SNAR), "Link Delay Update" (LDU), and "Multi-Path Data Dissemination" (MPDD). SNAR attempts to deliver messages on a path with the shortest expected delay to the destination, using both vehicular nodes and static nodes. It operates in two modes: the "in-road mode" and the "intersection mode" (similar to VADD's StraightWay and Intersection modes). The in-road mode operates while a vehicular node carries a message in a road. In this mode, greedy geographic forwarding is used to transport a message toward a static node at an intersection. When the static node is reached, the intersection mode assumes control. In intersection mode, the static node calculates the optimal next intersection for the message based on its "delay matrix", explained below. The packet is stored at the static node until it can be forwarded to a vehicle traveling toward the optimal next intersection. If multiple vehicles traveling in that direction are immediately available, the static node forwards the message to the farthest of those vehicles.

As with any store-and-forward communication, buffer management is an important issue in SADV. When the buffer at a static node becomes full, packets will be eliminated from the buffer by forwarding them along the best available path. The strategy used for determining which messages will be eliminated from the buffer is called "least delay increase", which attempts to send packets along paths which will not significantly increase their delivery delay. Under this strategy, the static node checks which paths are currently available, and eliminates packets that will be least affected by traveling along the available paths.

As described above, SNAR makes extensive use of optimal paths. Optimal paths are determined based on a graph abstracted from a static road map and weighted with "expected path forwarding delays" from a delay matrix. SADV generates these expected path forwarding delays based on real-time traffic density information. The LDU module maintains the delay matrix dynamically by measuring the delay of message delivery between static nodes. When a static node s_i receives a message, it places a timestamp in the message header and proceeds to forward the message. When that message reaches the next static node s_j, the elapsed time is calculated, and the timestamp in the message header is updated again. The elapsed time is retained by s_j, and in this way, a static node s_j can obtain the delay for packets arriving from all incoming directions. Each static node calculates the mean delay for each incoming direction over a specified time interval and propagates this information to neighboring static nodes in a periodic "delay update message" distributed by vehicles.

When traffic is low, the MPDD module of SADV takes advantage of the situation by utilizing multi-path routing. To reduce the impact on the protocol stack of vehicles, only static nodes may deliver messages on multiple paths. In the MPDD module, each static node chooses the best two paths for each message and attempts to forward the message along both of those paths. To avoid excessive routing overhead caused by loops, each static node maintains a record of the messages it has sent during a time interval and ignores any duplicate packets received during that time.

The simulations used in the performance analysis did not include any one-way streets. Since one-way streets are not effective for sending data in the opposite direction of traffic flow, a scenario that includes one-way traffic may have an impact on the performance results of SADV.

4.2.5. Geographical opportunistic routing

Like other routing algorithms designed for delay-tolerant networks, the Geographical Opportunistic (GeOpps) routing algorithm [39] uses opportunistic routing and a carry-and-forward approach to route messages in sparse networks. Its uniqueness derives from its utilization of navigation information to route packets efficiently. GeOpps assumes that each vehicle has knowledge of its position through GPS and a navigation system that provides a suggested route to a traveling destination. It is also assumed that each vehicle's navigation system can provide the location of static roadside units.

When a message is being sent from a source node to a destination using GeOpps, intermediate nodes use the following method to select the next hop. Each neighbor vehicle that is following a navigation-suggested route calculates its future nearest point to the message destination. It also uses a "utility function" built into its navigation system to calculate the amount of time required to reach that point. The vehicle that can deliver the packet fastest or closest to the destination will be chosen as the next hop for the message.

GeOpps also considers some special cases. If drivers cease to follow their navigation-suggested routes, then the algorithm will forward the message to any neighbor. If a vehicle remains stopped for a certain amount of time, or if the engine is switched off, then all messages carried by the vehicle will be forwarded to the next available vehicle. The algorithm does not require all vehicles to be equipped with navigation systems. In these situations, a greedy algorithm is used to route messages instead.

GeOpps performs well in sparse networks and with delay-tolerant applications. The differences in navigation systems that are available to consumers may create challenges for GeOpps. For instance, there may be some difficulties in the differences between "utility functions" for time calculations, depending on the factors (speed limit, traffic conditions, road types, average driver speed) that each navigation system considers.

4.2.6. MaxProp

Another algorithm in the delay-tolerant network category, MaxProp [14] utilizes carry-and-forward and packet prioritization techniques to maximize message delivery in a network with limited transfer opportunities between nodes. MaxProp is implemented in a real network called UMass DieselNet, where it is deployed on buses, allowing each bus to communicate its location and performance information to wireless access points or other buses as they are encountered. The algorithm is an extension of a previous routing protocol called MV [15], and it assumes that each node in the network has knowledge of its position through GPS.

MaxProp operates in three basic stages: the *neighbor discovery* stage, in which nodes discover each other before packet transfers can begin, the *data transfer* stage in which a limited amount of data can be transferred between nodes, and the *storage management* stage, in which each node manages its local storage buffer by selecting packets to delete according to a prioritization algorithm.

By default, a node will carry a message until the next neighbor discovery phase occurs. The node will continue to forward this message at each node meeting until the message's timeout is met, the delivery of the message is confirmed by an ACK in a data transfer stage, or until the message is dropped in the storage management stage due to a full buffer.

In the data transfer stage, messages are transferred based on a highest-priority-first scheme. In the storage management stage, MaxProp deletes messages based on a lowest-priority-first scheme. Each message stored at a node is prioritized based on the estimated cost to deliver that message to its destination. To determine the estimated delivery cost, each node in MaxProp maintains a probability of meeting every other node in the network. MaxProp uses incremental averaging to modify these probabilities at each meeting between nodes, and at each meeting, these probabilities are exchanged between the nodes. With these values, each node calculates the estimated cost to deliver a message for each possible path to that message's destination. This is done by summing the probabilities that each connection on the path will not occur. The estimated cost for that message to reach its destination is the least costly path among all of those calculated.

When the data transfer stage begins, information exchange occurs in a predetermined order. First, messages destined for the neighbor node are transferred. Secondly, the aforementioned meeting probability information is exchanged between the nodes. Third, delivery acknowledgements are exchanged. This allows delivered messages remaining in the node's buffer to be safely deleted. Fourth, messages with low hop count are transferred, regardless of estimated delivery cost. Finally, the remaining messages that have not yet been transmitted are sent according to their priority, based on estimated delivery cost.

Prioritizing based on two different factors (delivery cost and hop count) is done in the following manner. The priority-sorted buffer at each node is divided into two groups according to whether messages in the buffer have a hop count of less than a threshold t hops. Messages with a hop count below this dynamically set threshold t are sorted by their hop count (newer messages have higher priority), while messages with larger hop counts are prioritized by the aforementioned estimated delivery cost criterion.

The storage management stage handles the maintenance of each node's finite buffer. Messages are deleted when the buffer is full. First, messages that have been acknowledged as delivered are deleted. These acknowledgements are distributed across the entire network (not just to the source) to clean up message copies that might otherwise remain in each node's buffers for an extended period of time. Secondly, messages that have a hop count of greater than t hops are deleted in the order of the highest estimated delivery cost first. Finally, if the buffer is still full, messages with hop counts below t are deleted in the order of the largest hop counts first.

MaxProp offers novel ideas about message prioritization, buffer management, and routing messages based on probability of node meetings. System-wide acknowledgements and tracking the probability of meeting every other node in the network are practical techniques in a network of 30 buses, but these approaches may not scale well to larger populations.

4.3. Quality of service protocols

Using the term Quality of Service (QoS) to describe current protocols in this category is a trifle misleading. In the strictest sense, a QoS protocol should provide guarantees about the level of performance provided. This is often achieved through resource reservation and sufficient infrastructure. However, in an ad-hoc wireless network, this is a difficult task. With the exception of the potential for roadside units, there is no infrastructure to be relied upon for guaranteed bandwidth. The dynamic and cooperative nature of an ad-hoc network does not lend itself to resource reservation. What variables can we adjust in order to provide guarantees about service?

These QoS routing strategies attempt to provide a robust route among vehicles. Factors such as link delay, node velocity and trajectory, node position, distance between nodes, and reliability of links all contribute to the stability of a particular route. Some performance guarantees can be made in vehicular routing, and we survey the algorithms that can estimate the duration for which a route will remain connected, and minimize the amount of time required to rebuild the connection if it is broken. With those exceptions, the current suite of QoS VANET routing protocols is most aptly described as a set of "best effort" protocols.

4.3.1. Multi-hop routing protocol for urban VANETs

Merging position-based and QoS factors, the Multi-Hop Routing Protocol for Urban VANETs (MURU) [45] balances hop minimization with the ability to provide a robust route connection. In doing so, a new metric called the "expected disconnection degree" (EDD), is introduced to estimate the quality of a route based on factors such as vehicle position, speed, and trajectory. MURU requires each vehicle to know its own position and to have an external static street map available. The presence of an efficient location service is also assumed.

EDD, the expected disconnection degree, is an estimation of the probability that a given route might break during a given time period. Using this measure, a low EDD is preferred. Given knowledge of vehicle positions, speeds, and trajectories, one can make some guesses about the stability of a route along a sequence of nodes. Intuitively, nodes moving in similar directions at similar speeds are more likely to maintain a stable route. The authors also show that, given certain assumptions about vehicle traffic, routes with very small and very large routes have higher packet error rates. The formula for calculating EDD takes these factors into account to make a prediction about the breakability of a route. In MURU, each RREQ message stores the cumulative EDD for the path that message traveled.

To find a route to a destination, a source node calculates the shortest trajectory to the destination, based on their locations and the static street map. It then initiates a RREQ message, broadcasting it in a rectangular "broadcast area" that encloses the shortest trajectory and is bounded by the positions of the source and destination. The shortest trajectory is stored in the packet and is used as a directional guideline for the RREQ message. Nodes outside of the "broadcast area" will drop the packet.

A node n_i, upon receiving a RREQ message from node n_{i-1}, calculates the EDD of the link from node n_{i-1} to node n_i using the formula

$$\mathrm{EDD}_{i-1,i} = \alpha * |D(i-1,i) - D_0|^l + \beta * f(L(i), T(0,d)) + \gamma * g(M(i-1), M(i), T(0,d)).$$

Here, α, β and γ are predefined tuning parameters, l is the path loss exponent which is determined by the signal propagation model used in the urban areas. $D_{i-1,i}$ is the geographic distance from node n_{i-1} to node n_i, D_0 being a correction factor. $L(i)$ is the current location of node n_i, $M(i)$ is n_i's predicted movement information including expected velocity during the time period T. $T_{0,d}$ is the shortest trajectory from source n_0 to destination n_d. $g(M(i-1), M(i), T(0,d))$ returns 0 if n_i and n_{i-1} are expected to be within the transmission range of each other for a period longer than T, and 1 otherwise. $f(L(i), T(0,d))$ returns 0 if n_i is on the shortest trajectory towards the destination for a time period longer than T, and 1 otherwise. RREQ message has a field to carry $\mathrm{EDD}_{\mathrm{path}}(0,i)$, which is the EDD of the path from source n_0 to the current node n_i through which the RREQ has propagated. When RREQ is received by node n_i from node n_{i-1}, it calculates $\mathrm{EDD}_{\mathrm{path}}(0,i)$ using the following formula and includes it in the RREQ before rebroadcasting it.

$$\mathrm{EDD}_{\mathrm{path}}(0,i) = \begin{cases} 0 & \text{if } i = 0 \\ \mathrm{EDD}_{\mathrm{path}}(0,i-1) + \mathrm{EDD}_{\mathrm{path}}(i-1,i) & \text{otherwise.} \end{cases}$$

If every node receiving the RREQ message immediately re-broadcasts it, the message overhead would quickly become not scalable. To avoid this, the MURU protocol provides a pruning mechanism for route requests which allows each node to delay forwarding a RREQ message. A node receiving the RREQ will wait for a calculated backoff delay (milliseconds) that is directly proportional to the EDD between the previous forwarder of the RREQ and the current node. (If the EDD is large, then the current node's forwarding delay is also large, to allow for the possibility that a path with a smaller EDD may be found.) During this backoff delay, the node listens for RREQ messages at other nodes. If during that window of time, the node overhears a counterpart to this RREQ (from the same source with the same sequence number) whose EDD is smaller than its own EDD to the source, then it will drop the RREQ. (This causes RREQ messages received along the path to be dropped by nodes whose links are more likely to be broken.) If the node has not dropped the RREQ during this time, then it will rebroadcast the RREQ into a rectangular area defined by its position and the position of the destination. Thus, each time the RREQ is re-broadcast, the broadcast area becomes an iteratively smaller rectangular area. When the destination finally receives some RREQ messages from different routes, it selects the route with the smallest EDD.

In the paper, the authors show that MURU is loop-free and that MURU always chooses a path from source to destination with the smallest EDD, both initially and when repairing a route. Due to the rectangular broadcast areas used in MURU, it may be susceptible to local optimums. If the rectangular broadcast area is strictly bounded by the forwarding node and the destination node, this problem could occur in cases where the only available next hop node is located outside of the broadcast rectangle. MURU aims to provide a quality route that delivers a high percentage of packets while controlling overhead and delay.

4.3.2. Prediction-based routing

While many algorithms have concentrated on vehicle-to-vehicle communication in VANETs, the Prediction-Based Routing (PBR) algorithm [47] is focussed on providing Internet connectivity to vehicles. Building an infrastructure of roadside static gateways is costly, especially outside of urban areas. Instead, PBR explores the possibility of mobile gateways with wireless WAN connections that can act as Internet gateways for other vehicles, focusing specifically on highway scenarios. The PBR algorithm assumes that each vehicle has knowledge of its own position through GPS or other means. The algorithm takes advantage of the less erratic vehicle movement patterns on highways to predict the duration and expiration of a route from a client vehicle to a mobile gateway vehicle. Just before a route failure is predicted, PBR preemptively seeks a new route to avoid loss of service.

To communicate to a location on the Internet, a node checks its routing table for an existing route. Like many reactive MANET protocols, if the node finds no existing route, the node broadcasts a RREQ message with a limited number of hops. When the RREQ reaches a mobile gateway, a RREP message is returned to the source node through the sequence of nodes stored in the RREQ. (Note that due to the low lifetime of vehicular routes, intermediate nodes with cached gateway routes do not send RREP messages.) If multiple gateways are found, the source node chooses the route with the shortest number of hops, where the most hops are moving in the same direction as the source node. If multiple routes to the same gateway are found, then the route with the longest predicted lifetime is chosen. Once the route to the gateway is established, communication begins. If this route remains active over time, then just before the route is predicted to fail, PBR attempts to establish a new route to a gateway.

The RREP message is used to predict the lifetime of the route. When the gateway receives the RREQ message, it writes its position and velocity information and a maximum lifetime value in the RREP message that it returns to the source node. As each intermediate node handles the RREP message it estimates the lifetime of the link between its predecessor and itself, based on wireless communication range, direction of travel, velocities of nodes, and distance between the nodes. If that lifetime is less than the lifetime stored in the RREP message, then it updates the message with the lesser lifetime. When the source node receives the RREP message, it contains the estimated lifetime of the route.

Regardless of the effectiveness of the PBR algorithm in the mobile gateway situation, it is unknown how realistic the situation itself is. It is unclear how a vehicle would be motivated to share its wireless Internet connection with others, when that connection is likely to be costly, although the integration of micropayments may be a possible area for exploration. The prospect of Internet providers charging for the use of their roaming WAN-connected vehicles may be economically feasible, but further analysis would be required to determine this. In addition, the bandwidths of the mobile gateway's wireless WAN connections would need to be significant enough to support the bandwidth demand of numerous client vehicles.

5. Conclusion and open issues

In this paper, we have presented a survey of VANET routing protocols. We have discussed potential applications for the technology and addressed many of the design factors involved. We provided a timeline of VANET routing protocol development and a qualitative comparison of their objectives, characteristics, and assumptions. We summarized the protocols and categorized them into three groups: position-based greedy V2V protocols, delay-tolerant protocols, and QoS protocols.

The potential applications of vehicular ad hoc networks (VANETs) offer vast opportunity, but much work remains to be done in order to make this technology practical. Each of the routing protocols surveyed in this paper has contributed its own unique ideas to the advancement of this area of research. GeOpps [39] considers long-range trajectory information provided by navigation systems, and VADD [71] also mentions the potential of this. Utilization of this information could be integrated into other protocols. STAR [26], CAR [49], and SADV [21] all explore slightly different methods for real-time traffic density detection. A careful study of these methods would be prudent, as areas for improvement may be found. Traffic density detection and the propagation of this information could be modularized. Like positioning and location services, routing protocols could be decoupled from traffic density mechanisms, allowing them to be designed separately. Propagation of data about network and traffic anomalies may also yield opportunities for improvement and possible modularization. The concept of geographic marker messages such as guards [49] have been explored, but the use of roadside units may be a further area for expansion. Additional areas for improvement include the integration of privacy and security mechanisms into routing protocols and the establishment of priority routes for emergency and safety messages.

A foundation of VANET routing protocols has been established, and this body of research begs future work. While each protocol has generally provided a performance evaluation against a few other protocols, a comprehensive performance evaluation of each category of protocols would offer significant value. In such an evaluation, variables such as traffic density, map layouts, route distance, and radio range could be made consistent for each protocol. Urban and highway environments could be simulated using real maps or by incrementally varying factors such as intersection density. Criteria such as routing overhead, packet delivery ratio, average packet delay, delay variance, link reliability, average number of hops, and average buffer sizes could be compared uniformly.

We have seen that position-based, greedy, vehicle-to-vehicle approaches utilize the absolute or relative locations of each node to greedily route messages toward a next anchor or a destination vehicle. These approaches tend to focus on highly dense traffic scenarios. Delay-tolerant routing protocols aim at a different objective. They provide the ability to communicate slowly when there are few vehicles on the road by analyzing each forwarding opportunity carefully. In reality, both low and high traffic density will be encountered, and a successful routing protocol must adapt to both of these scenarios. A hybridization of these approaches still remains as an area for future work. QoS features might also be incorporated into such a hybrid algorithm.

This survey focused on unicast VANET routing protocols, but many of the features required to make VANETs a viable technology rely on the use of multicast communication paradigms. Much research has been focused on multicast, geocast, and anycast communication in VANETs. A survey of these routing protocols is an open topic for future work.

With the massive number of vehicles on roads today, the potential for vehicular ad hoc networks is vast. Consumer and corporate interests promise bright prospects for this technology. Routing protocols for VANETs have separated themselves from other mobile routing protocols, due to the inherent characteristics of communication on roadways. Many distinguishing qualities of this environment are not yet explored, leaving ample opportunities for further research in the area.

Acknowledgements

We thank the reviewers and the editor for their excellent comments which helped in improving the content and presentation of the paper greatly. This material is based in part upon work supported by the US National science Foundation under Grant No. IIS-0414791. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Ahmed, S.S. Kanere, SKVR: Scalable knowledge-based routing architecture for public transport networks, in: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET'06, ACM, New York, NY, USA, 2006.
[2] L. Armstrong, W. Fisher, Status of project IEEE 802.11 task group p: Wireless Access in Vehicular Environments (WAVE). http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm, meeting Update, Nov 2007.
[3] A. Bachir, A. Benslimane, A multicast protocol in ad hoc networks: Inter-vehicle geocast, in: Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, 2003, vol. 4, IEEE, 2003.
[4] R. Bai, M. Singhal, DOA: DSR over AODV routing for mobile ad hoc networks, IEEE Transactions on Mobile Computing 5 (10) (2006) 1403–1416.
[5] R. Bai, M. Singhal, BRD: Bilateral route discovery in mobile ad hoc networks, in: Proceedings of the 6th International IFIP-TC6 Networking Conference, in: Lecture Notes in Computer Science, vol. 4479, 2007.
[6] R. Bai, M. Singhal, Y. Wang, On supporting high-throughput routing metrics in on-demand routing protocols for multi-hop wireless networks, Journal of Parallel and Distributed Computing 67 (10) (2007) 1108–1118.
[7] R. Baumann, Vehicular ad hoc networks (VANET): Engineering and simulation of mobile ad hoc routing protocols for VANET on highways and in cities, Master's Thesis, Swiss Federal Institute of Technology Zurich, 2004.
[8] A. Benslimane, Optimized dissemination of alarm messages in vehicular ad-hoc networks (VANET), in: Proceedings of High Speed Networks and Multimedia Communications, in: Lecture Notes in Computer Science, vol. 3079/2004, Springer, Berlin, Heidelberg, 2004, pp. 655–666.
[9] J. Boleng, T. Camp, Adaptive location aided mobile ad hoc network routing, in: Proceedings of the 2004 IEEE International Performance, Computing, and Communications Conference, 2004.
[10] P. Bose, P. Morin, I. Stojmenovi, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networks 7 (6) (2001) 609–616.
[11] A. Boukerche, El-Khatib, L. Xu, An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks, Computer Communications 28 (10) (2005) 1193–1203.
[12] A. Boukerche, H. Owens, Energy aware routing protocol for mobile and wireless ad hoc networks, in: Proceedings 28th Annual IEEE International Conference on Local Computer Networks, 2003.
[13] I. Broustis, M. Faloutsos, Routing in vehicular networks: feasibility, security, and modeling issues, Tech. Rep. UCR-CS-2006-05219, Department of Computer Science and Engineering, University of California, Riverside, Oct 2006.
[14] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: Routing for vehicle-based disruption-tolerant networks, in: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, IEEE, 2006.

[15] B. Burns, O. Brock, B.N. Levine, MV routing and capacity building in disruption tolerant networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 1, IEEE, 2005.
[16] T. Camp, Y. Liu, An adaptive mesh-based protocol for geocast routing, Journal of Parallel and Distributed Computing 63 (2) (2003) 196–213.
[17] H.C. Chang, H. Du, J. Anda, C.-N. Chuah, D. Ghosal, H.M. Zhang, Enabling energy demand response with vehicular mesh networks, in: Mobile and Wireless Communication Networks, in: IFIP International Federation for Information Processing, vol. 162/2005, Springer, Boston, 2005, pp. 371–382.
[18] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, Wireless Networks 8 (2002) 481–494.
[19] D.R. Choffnes, F.E. Bustamante, An integrated mobility and traffic model for vehicular wireless networks, in: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks, VANET'05, ACM, New York, NY, USA, 2005.
[20] D. De, S. Das, Dynamic multipath routing (DMPR): An approach to improve resource utilization in networks for real-time traffic, in: Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001.
[21] Y. Ding, C. Wang, L. Xiao, A static-node assisted adaptive routing protocol in vehicular networks, in: Proceedings of The Fourth ACM International Workshop On Vehicular Ad Hoc Networks, VANET'07, ACM, New York, NY, USA, 2007.
[22] S. Eichler, F. Dötzer, C. Schwingenschlögl, F.J.F. Caro, J. Eberspächer, Secure routing in a vehicular ad hoc network, in: Proceedings of the IEEE 60th Vehicular Technology Conference, VTC2004-Fall, IEEE, 2004.
[23] A. Festag, H. Füßler, H. Hartenstein, A. Sarma, R. Schmitz, FleetNet: Bringing car-to-car communication into the real world, in: Proceedings of the 11th World Congress on ITS, 2004.
[24] E. Fonseca, A. Festag, A survey of existing approaches for secure ad hoc routing and their applicability to VANETS, Tech. Rep. NLE-PR-2006-19, NEC Network Laboratories, Mar 2006.
[25] H. Füßler, M. Mauve, H. Hartenstein, D. Vollmer, M. Käsemann, A comparison of routing strategies in vehicular ad-hoc networks, Tech. Rep. 3/2002, Universität Mannheim, Mar 2002.
[26] F. Giudici, E. Pagani, Spatial and traffic-aware routing (STAR) for vehicular systems, in: Proceedings of High Performance Computing and Communications, in: of Lecture Notes in Computer Science, vol. 3726/2005, Springer, Berlin, Heidelberg, 2005, pp. 77–86.
[27] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in VANETs, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET'04, ACM, New York, NY, USA, 2004.
[28] Z. Haas, A new routing protocol for the reconfigurable wireless networks, in: Proceedings of the IEEE International Conference on Universal Personal Communications, 1997.
[29] Z. Haas, M. Pearlman, The performance of query control schemes for the zone routing protocol, in: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, WA, United States, 1999.
[30] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, Directed diffusion for wireless sensor networking, IEEE/ACM Transactions on Networking 11 (2003) 2–16.
[31] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE 2001 International Multi Topic Conference, IEEE INMIC 2001, IEEE, 2001.
[32] Q. Jiang, R.A. Finkel, D. Manivannan, M. Singhal, RPSF: A routing protocol with selective forwarding for mobile ad-hoc networks, Wireless Personal Communications 43 (2) (2007) 411–436.
[33] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Imielinski Korth (Ed.), Mobile Computing, vol. 353, Kluwer Academic Publishers, 1996.
[34] H.P. Joshi, Distributed robust geocast: A multicast protocol for inter-vehicle communication, Master's Thesis, North Carolina State University, 2006.
[35] B. Karp, H.T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom'00, ACM, New York, NY, USA, 2000.
[36] M. Kihl, M. Sichitiu, T. Ekeroth, M. Rozenberg, Reliable geographical multicast routing in vehicular ad-hoc networks, in: Proceedings of International Conference on Wired/Wireless Internet Communications, in: Lecture Notes in Computer Science, vol. 4517/2007, Springer, Berlin, Heidelberg, 2007, pp. 315–325.
[37] G. Korkmaz, E. Ekici, F. Özgüner, Ümit Özgüner, Urban Multi-hop broadcast protocol for inter-vehicle communication systems, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET'04, ACM, New York, NY, USA, 2004.
[38] J. LeBrun, C.-N. Chuah, D. Ghosal, M. Zhang, Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks, in: Proceedings of the 61st IEEE Vehicular Technology Conference, 2005, in: VTC 2005-Spring, vol. 4, IEEE, 2005.
[39] I. Leontiadis, C. Mascolo, GeOpps: Geographical opportunistic routing for vehicular networks, in: Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, 2007, IEEE, 2007.
[40] H. Li, M. Singhal, A Scalable routing protocol for ad hoc networks, in: Proceedings of IEEE 61st Vehicular Technology Conference, VTC2005, 2005.
[41] H. Li, M. Singhal, An anchor-based routing protocol with cell ID management system for ad hoc networks, in: Proceedings of 14th International Conference on Computer Communications and Networks, 2005.
[42] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, M. Mauve, A routing strategy for vehicular ad hoc networks in city environments, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2003.
[43] C. Lochert, M. Mauve, H. Füßler, H. Hartenstein, Geographic routing in city scenarios, ACM SIGMOBILE Mobile Computing and Communications Review 9 (1) (2005) 69–72.
[44] M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad hoc networks, IEEE Network 15 (6) (2001) 30–39.
[45] Z. Mo, H. Zhu, K. Makki, N. Pissinou, MURU: A multi-hop routing protocol for urban vehicular ad hoc networks, in: Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems — Workshops, IEEE, 2006.
[46] M. Singhal, V.C. Giruka, A self-healing on-demand geographic path routing protocol for mobile ad-hoc networks, Ad Hoc Networks 5 (7) (2007) 1113–1128.
[47] V. Namboodiri, L. Gao, Prediction-based routing for vehicular ad hoc networks, IEEE Transactions on Vehicular Technology 56 (4) (2007) 2.
[48] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'06, ACM, New York, NY, USA, 2006.
[49] V. Naumov, T. Gross, Connectivity-aware routing (CAR) in vehicular ad-hoc networks, in: Proceedings of the 26th IEEE International Conference on Computer Communications, INFOCOM 2007, IEEE, 2007.
[50] Z. Niu, W. Yao, Q. Ni, Y. Song, DeReQ: A QoS routing algorithm for multimedia communications in vehicular ad hoc networks, in: Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, IWCMC'07, ACM, New York, NY, USA, 2007.
[51] OmniAir Consortium, DSRC for the public sector: Transportation facility operators. http://www.omniair.org/benefits/forpublicsector.html, 2008.
[52] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of the INFOCOM, 1997.
[53] C. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of the Conference on Communications Architectures, Protocols and Applications, London, United Kingdom, 1994.
[54] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, 1999.
[55] H. Pucha, S.M. Das, Y.C. Hu, The performance impact of traffic patterns on routing protocols in mobile ad hoc networks, in: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'04, ACM, New York, NY, USA, 2004.
[56] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN'05, ACM, New York, NY, USA, 2005.
[57] A. Roy, S.K. Das, QM2RP: A QoS-based mobile multicast routing protocol using multiobjective genetic algorithm, Wireless Networks 10 (3) (2004) 271–286.

[58] A.K. Saha, D.B. Johnson, Modeling mobility for vehicular ad-hoc networks, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET'04, ACM, New York, NY, USA, 2004.
[59] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, K.-K. Lee, A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications, in: Proceedings of the Third International IFIP-TC6 Networking Conference, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications, NETWORKING 2004, in: Lecture Notes in Computer Science, vol. 3042/2004, Springer, Berlin, Heidelberg, 2004, pp. 989–999.
[60] P. Sinha, R. Sivakumar, V. Bharghavan, CEDAR: A Core-extraction distributed ad hoc routing algorithm, in: Proceedings of the INFOCOM, 1999.
[61] A. Srinivas, E. Modiano, Minimum energy disjoint path routing in wireless ad hoc networks, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, San Diego, CA, USA, 2003.
[62] E.T. Clausen, E.P. Jacquet, Optimized link state routing protocol, in: Request for Comments# 3626, 2003.
[63] J. Tian, L. Han, K. Rothermel, C. Cseh, Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks, in: Proceeding of the IEEE Intelligent Transportation Systems, 2003, vol. 2, IEEE, 2003.
[64] M. Torrent-Moreno, P. Santi, H. Hartenstein, Fair sharing of bandwidth in VANETs, in: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks, VANET'05, ACM, New York, NY, USA, 2005.
[65] USDOT, Intelligent transportation systems standards fact sheet: ASTM E2213-03 — standard specification for telecommunications and information exchange between roadside and vehicle systems — 5 GHz band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) specifications. http://www.standards.its.dot.gov/fact_sheet.asp?f=66, fact Sheet ASTM E2213-03, Jan 2006.
[66] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, Mobile Network Applications 9 (2) (2004) 141–149.
[67] H.F. Wedde, S. Lehnhoff, B. van Bonn, Highly dynamic and scalable VANET routing for avoiding traffic congestions, in: Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, VANET'07, ACM, New York, NY, USA, 2007.
[68] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, O. Tonguz, Routing in sparse vehicular ad hoc wireless networks, IEEE Journal on Selected Areas in Communications 25 (8) (2007) 1538–1556.
[69] H. Wu, R. Fujimoto, R. Guensler, M. Hunter, MDDV: A mobility-centric data dissemination algorithm for vehicular networks, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET'04, ACM, New York, NY, USA, 2004.
[70] J. Wu, F. Dai, M. Gao, I. Stojmenovic, On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks, Journal of Communications and Networks 5 (2) (2002) 169–178.
[71] J. Zhao, G. Cao, VADD: Vehicle-assisted data delivery in vehicular ad hoc networks, in: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, 2006, pp. 1–12.

James Bernsen is a graduate student of the computer science program at the University of Kentucky. His academic interests include distributed computing and mobile networking. He also works as a software engineer and system analyst in the field of product lifecycle management (PLM) for Lexmark International, Inc.

D. Manivannan is currently an associate professor in the Department of Computer Science at the University of Kentucky, Lexington, USA. Dr. Manivannan received the Ph.D. degree in computer and information science from The Ohio State University, Columbus, Ohio, in 1997. His areas of research interests include fault-tolerance and synchronization in distributed systems, routing in wormhole networks, channel allocation in cellular networks and wireless networks. He served as program committee member for several International Conferences and also served as reviewer for several International Journals published by ACM, IEEE, Elsevier, Springer, Oxford University Press and others. He is on the Editorial Board of Information Sciences journal, published by Elsevier and Wireless Personal Communications journal, published by Springer. He is also an Associate Technical Editor for IEEE Communications Magazine. Information about his publications and research can be found at his lab site http://dmcslab.netlab.uky.edu/. Dr. Manivannan is a recipient of the CAREER Award from the US National Science Foundation. He is a senior member of IEEE, senior member of the ACM. He is also a member of ACM SIGMOBILE and IEEE Computer Society and IEEE Communications Society.