Delay-bounded Routing in Vehicular Ad-hoc Networks

Antonios Skordylis
antonios.skordylis@comlab.ox.ac.uk

Niki Trigoni
niki.trigoni@comlab.ox.ac.uk

Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford, OX1 3QD, United Kingdom

ABSTRACT

Ad hoc networks formed by traveling vehicles are envisaged to become a common platform that will support a wide variety of applications, ranging from road safety to advertising and entertainment. The multitude of vehicular applications calls for routing schemes that satisfy user-defined delay requirements while at the same time maintaining a low level of channel utilization to allow their coexistence. This paper focuses on the development of carry-and-forward schemes that attempt to deliver data from vehicles to fixed infrastructure nodes in an urban setting. The proposed algorithms leverage local or global knowledge of traffic statistics to carefully alternate between the Data Muling and Multihop Forwarding strategies, in order to minimize communication overhead while adhering to delay constraints imposed by the application. We provide an extensive evaluation of our schemes using realistic vehicular traces on a real city map.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer-Communication Networks; C.2.2 [**Network Protocols**]: [Routing protocols]

General Terms

Algorithms, Performance

Keywords

VANET,inter-vehicle communication, vehicular network, ad hoc network, routing, minimum cost forwarding, delay.

1. INTRODUCTION

Recent trends in Intelligent Transportation Systems show that an increasing number of vehicles will be equipped with wireless transceivers that will enable them to communicate with each other and form a special class of wireless networks, known as *vehicular ad hoc networks or VANETs.* Researchers and automotive industries are envisioning the deployment of a large spectrum of applications running on VANETs, ranging from cooperative driver assistance systems for collision avoidance, to notification services for traffic congestion and parking availability. We classify VANET-based applications into two categories: 1) those that require broadcasting of information from one vehicle to many nearby vehicles, e.g. for collision avoidance; and 2) those that require the propagation of information hop-by-hop to a single destination point or area, e.g. sending an advertisement from an attraction site to a busy intersection, or sending an emergency message from an accident site to the closest roadside unit that is connected to a fixed network.

The focus of this paper is the second class of applications; our motivating example is the *ambient traffic sensor application* wherein vehicles are equipped with sensors that detect accidents, road faults and traffic congestion. On detection of an interesting event, vehicles attempt to notify the city's traffic monitoring center, by sending the information to one of the stationary roadside units dispersed in the city. These are referred to as *access points (APs)* and act as gateways to stream traffic information through a fixed network to the outside world.

We note that messages may have very different priorities, and thus delay thresholds until they are delivered to one of the APs. For example, information about a serious accident has higher priority than information about a road fault. The former must be delivered to one of the APs much faster than the latter, since it calls for immediate assistance from fire, hospital or police departments. It is therefore vital that packet forwarding algorithms are designed to prioritize packets based on their urgency and deliver them within user-defined delays.

At the same time, the proliferation of applications that may be running on a VANET network in the near future suggests that they will be competing for the use of the wireless medium. Network resources will be shared by applications that provide internet access to passengers, propagate advertisements about nearby places of interest, provide the driver with safety information (e.g. emergency braking) and so on. This matter is further aggravated if we take into account inter-system interference. This type of interference is of particular significance in wireless systems that operate in similar frequency bands, where it appears in the form of co-channel interference both among multiple users of the same technology and among different co-existing technologies. For instance, Bluetooth and WLAN which have a strong presence in urban settings,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.
Copyright 2008 ACM 978-1-60558-073-9/08/05 ...$5.00.

are shown to heavily interfere with one another in [8]. Thus, our goal is to design algorithms that try to optimize bandwidth utilization, by being frugal in wireless packet transmissions. To do so, we plan to leverage knowledge of traffic information on different parts of the city; our proposed algorithms are traffic-informed and they adapt their behavior depending on the traffic density and the average vehicle speed on different road segments.

We can therefore argue that in order to bring vehicular networks to their full potential, we must try to satisfy application requirements for bounded delays in packet delivery, whilst trying to minimize the utilization of the wireless medium. The key to achieve this goal is to take into consideration statistics of vehicle density and speed in various parts of the city. The contributions of this work are summarized as follows:

- We define the problem of timely and bandwidth-efficient data dissemination from vehicles acting as data sources to one of several access points dispersed in the city, given statistical information about road traffic. We carefully study the tradeoff between the competing requirements for timely data delivery and low bandwidth utilization.

- We propose two novel algorithms, D-Greedy and D-MinCost, that exploit traffic information to forward packets to the most convenient access point. D-Greedy exploits only local traffic information, whereas D-MinCost leverages traffic information about the entire city. Unlike existing vehicular-assisted forwarding algorithms [16], D-Greedy and D-MinCost do not try to minimize delay of packet delivery. Their goal is to minimize the number of packet transmissions required to satisfy packet-specific delay thresholds.

- In our extensive simulation study, we evaluate the performance of the proposed algorithms in terms of packet delivery ratio and bandwidth utilization, and compare them with the epidemic protocol proposed in [13] and the MinDelay protocol inspired by the VADD protocols [16]. Our experiments are conducted using realistic vehicle traces on a real city map.

The remainder of this paper is organized as follows: Section 2 outlines the assumptions, model and objectives of our work. Section 3 describes our two novel algorithms, D-Greedy and D-MinCost, and shows how they leverage traffic statistics to deliver packets in a timely and efficient manner. Section 4 evaluates the performance of the proposed algorithms using realistic vehicle traces. Section 5 discusses related work and we conclude in Section 6.

2. MODEL

2.1 Assumptions

We assume location-aware vehicles that obtain their geographical position from a GPS receiver or other location service. Vehicles also have access to a digital map of the area. The street map is abstracted as a directed graph $G(V, E)$. For any two intersections a and b, $(a, b) \in G$ if and only if there is a road segment connecting a and b and vehicles can travel from a towards b on that segment.

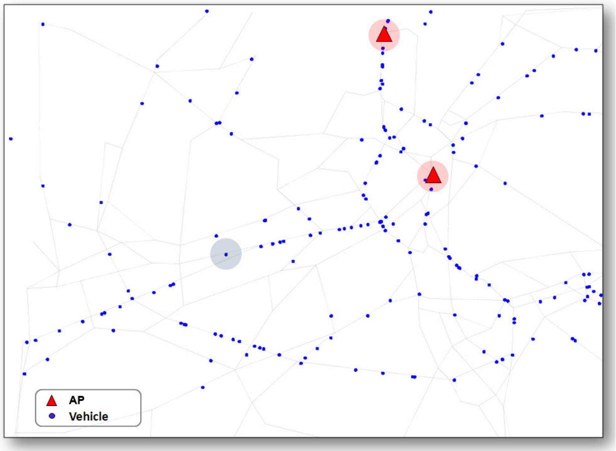

Figure 1: Our Model. Shaded circles indicate the communication range.

For the needs of the second algorithm described in Section 3 (D-MinCost), we assume that the map is preloaded with historical traffic statistics about the street network. These contain the average speed $\overline{u}$ and average vehicle density $\overline{d}$ per road segment. We do not assume preloaded statistical information to be highly accurate since traffic patterns vary throughout the day. However, traffic statistics can be updated once a vehicle comes into contact with an access point.

Using short to mid range transceivers, vehicles can communicate with neighboring vehicles or stationary access points within a range of 250 meters. Access points are attached to a fixed backbone and can communicate with each other. APs are infrastructure nodes whose absolute location in known to all vehicles and can only be installed on road intersections.

In the urban scenario we are considering, the network consists of mobile nodes (vehicles) and a few stationary access points (APs) that do not provide full city coverage (Figure 1). When a message is generated at a mobile node, depending on its location, it may need to be relayed multiple times thru several vehicles before reaching an AP. Upon sensing an interesting event, the vehicle produces a message containing the event description and all event-specific information, the message generation time t_g and a time-to-live value λ. The message is considered to be successfully delivered, if it arrives at one of the access points before time $t_g + \lambda$. We will refer to λ as the message *delay threshold* in the rest of this paper.

2.2 Objective

Wireless communication is unreliable in vehicle-to-vehicle communications due to multi-path fading, shadowing and Doppler shifts caused by the high mobility of vehicles. The quality of wireless links decreases with the relative and average velocities of vehicles using 802.11b [12]. Our focus is not on designing physical and MAC protocols for reliable packet transmission; we assume that such protocols are already in place, and our goal is to design routing protocols for propagating messages along multiple hops to a convenient AP. We are considering a densely populated urban area where typically the wireless medium is shared by a large number of vehicles, potentially running a variety of applications competing for network resources. We therefore assume that the network condition does not exhibit large variability throughout the area but rather operates at, or near, capacity. Under such conditions it is crucial to be frugal in the use of the wireless channel, refraining from transmitting whenever this is possible. In this work we aim to minimize bandwidth utilization by minimizing the number of transmitted messages, whilst adhering to the message requirements for bounded delivery delay.

When traffic density is low or when only few vehicles carry a wireless transceiver, the vehicular network often becomes disconnected. Hence, carry-and-forward protocols are required for the reliable delivery of messages between vehicles in dynamically changing network partitions. We assume that vehicles have very large buffers to store messages before forwarding them, since storage is no longer a limited resource in most mobile devices. Vehicles can choose to either continue carrying buffered messages as they move closer to an AP, or to forward messages to vehicles in range.

Our objective is thus to devise carry-and-forward algorithms that leverage knowledge of traffic statistics in an urban setting to enable timely delivery of messages from vehicles to stationary access points, whilst minimizing wireless transmissions and optimizing bandwidth utilization.

3. PROPOSED ALGORITHMS

In this section we present two novel routing algorithms for VANETs, *Delay-bounded Greedy Forwarding (D-Greedy)* and *Delay-bounded Min-Cost Forwarding (D-MinCost)*. The goal of our algorithms is to deliver messages originating in vehicles to an access point with bounded delay while minimizing the number of wireless transmissions. D-MinCost requires knowledge of *global traffic conditions*, i.e. statistical information about the speed and density of cars on every road segment of the city. In this work we do not study the precise process of maintaining a fairly accurate set of urban traffic statistics but rather assume that, when in the vicinity of access point, vehicles can update the preloaded street map with the latest statistical information. D-Greedy, on the other hand, requires no such knowledge. It only relies on local information, i.e. vehicle speed, to make forwarding decisions.

Our algorithms intend to minimize the number of transmissions while forwarding a message to an access point within the message-specific delay threshold. To do so, they proactively alternate between two forwarding strategies: *a*) *Multihop Forwarding*, which refers to the aggressive forwarding of messages to vehicles that are better positioned to deliver them to an access point and *b*) *Data Muling*, which refers to buffering messages in local memory and carrying them at the vehicle's speed. For the Multihop Forwarding strategy to be a feasible option, traffic needs to be dense enough so that better positioned vehicles exist within communication range. The Data Muling strategy is a feasible option as long as the current vehicle is traveling on the path selected by the routing algorithm.

The novelty of our proposed algorithms lies in their careful alternation between the Multihop Forwarding and Data Muling strategies to achieve a good tradeoff between delay and communication cost. This is in stark contrast with the previously proposed VADD protocols [16], which aim at minimizing message delay, and thus always prefer Multihop Forwarding to Data Muling when the former is possible. An additional difference from existing work is that our algorithms treat each buffered message in a different way depending on its remaining delay budget; the same vehicle may decide to adopt the Multihop Forwarding strategy for one message and Data Muling for another.

3.1 Delay-bounded Greedy Forwarding (D-Greedy)

The D-Greedy algorithm defines a forwarding strategy that assumes no knowledge of traffic information beyond node speed, which can be derived locally from the available location information. Based only on local knowledge, D-Greedy assumes that the best path to an access point is the shortest one, i.e. the path that minimizes the sum of the lengths of the edges on the directed graph G that abstracts the street map. When multiple APs exist, the algorithm selects the closest one, i.e. the one on the shortest path beginning at the vehicle's location.

Each vehicle maintains a neighbor list by periodically broadcasting beacons. A beacon message contains the unique vehicle identifier (id) and the length of the shortest path between the vehicle's current location and the location of the closest access point ($distToAP$). $distToAP$ is computed by running a single invocation of Dijkstra on G just before broadcasting a beacon. As soon as a vehicle senses an event and generates a new message, the message is assigned a delay threshold value (TTL) and is considered to be useful only if delivered before TTL has elapsed.

3.1.1 Greedy Strategy Selection

Vehicles periodically iterate through their buffers and make greedy decisions about the strategy that will be used for forwarding each message to the closest AP. The decision depends on the remaining delay budget (TTL) until the message expires as well as on the distance to the closest AP ($distToAP$). Since global traffic information is not available, D-Greedy assumes that the remaining message delay budget can be uniformly distributed among the edges that compose the shortest path to the AP. As a result, *each edge on the path is allocated a delay budget that is proportional to its length.* The algorithm periodically monitors the forwarding progress of each message; as long as the actual time spent by the carrying vehicle that travels along an edge does not exceed the delay allocated to that edge, the Data Muling strategy is selected for the message. Otherwise, the algorithm assigns the Multihop Forwarding strategy to the message.

More formally, let $distToInt$ be the remaining length, until the next intersection, of the current street segment e on which the vehicle is traveling. $distToAP$ denotes the current shortest-path distance from the closest AP and u the average speed of the vehicle calculated during a k-second historical window. D-Greedy computes the available delay budget Del for forwarding the message along the current edge up to the next intersection as follows:

$$Del = TTL \times \frac{distToInt}{distToAP}$$

It subsequently calculates the expected delay if the Data Muling strategy were to be used to carry the message to the next intersection:

$$Del_{DM} = \frac{distToInt}{u}$$

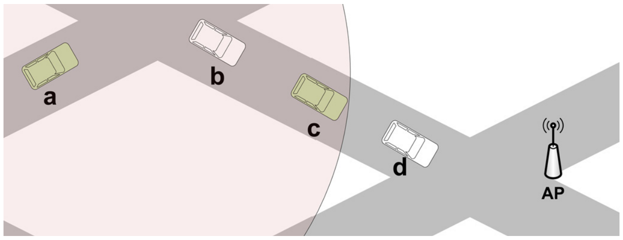

Figure 2: Node a will choose to forward the message to node c, the closest node to the AP among those in range.

If $Del_{DM} \leq Del$ then the algorithm opts for the Data Muling strategy, i.e. it refrains from transmitting the message to save bandwidth whilst adhering to the delay budget. Otherwise, the Multihop Forwarding strategy is chosen. In this case, the message is forwarded to the neighboring vehicle in range that is closest to the AP (Figure 2) and it is deleted from the node's buffer.

There are two extreme cases in which a vehicle does not apply the selected forwarding strategy for the message. When there is no better-positioned neighbor node to forward the message than the current node, messages that were originally assigned to use the Multihop Forwarding strategy switch to Data Muling. Similarly, if the carrying vehicle is moving away from the closest AP, messages that were originally assigned to use the Data Muling strategy switch to the Multihop Forwarding strategy until a vehicle traveling towards the AP is found.

Figure 3 shows the strategy selection of D-Greedy in action. Observe that when the message is being carried by a vehicle with high speed, it is propagated with the Data Muling strategy, whereas when a vehicle with low speed carries the message, it is propagated with the Multihop Forwarding strategy. Data Muling is allowed at lower speeds during the early lifetime of a message because the algorithm overestimates the delay allocated at each edge, since it assumes the message will follow the shortest path to the AP. As the message progresses through the network, the delay budget tightens and only high-speed carriers are allowed to perform Data Muling.

3.2 Delay-Bounded Minimum Cost Forwarding (D-MinCost)

Our second proposed algorithm leverages the knowledge of global traffic statistics, i.e. estimated values of average vehicle speed $\overline{u}$ and density $\overline{d}$ for all edges of the street graph G. Based on this information, D-MinCost computes bandwidth-efficient delay-constrained paths for every message in the node's buffer.

3.2.1 Graph extension

Recall that in the graph that abstracts the street map, edges represent road segments and vertices represent road intersections. We would like to annotate each edge with two metrics: 1) cost (C), representing the number of message transmissions along the edge, and 2) delay (Del), denoting the time required to forward a message along the edge.

However, the cost and delay of forwarding a message along an edge depends on whether we are using the Data Muling strategy or the Multihop Forwarding strategy. To solve this, we convert the original directed graph $G(V, E)$ that represents the street map to a new graph $G'(V, E')$, which contains the same set of vertices, and twice as many edges. For each directed edge $e \in G$ that connects two vertices, we create a new *sibling* edge $e' \in G$ connecting the same two vertices. The original edge e corresponds to a road segment when the Data Muling strategy is utilized, whereas edge e' corresponds to the same road segment when the Multihop Forwarding strategy is used. Consider for example the graph of Figure 4: the directed edge (d, a) in the original graph G is replaced by two sibling edges in the extended graph G', one for each strategy. Edges (c, b), (a, b) and (b, d) will each be replaced by two sibling edges in the same manner.

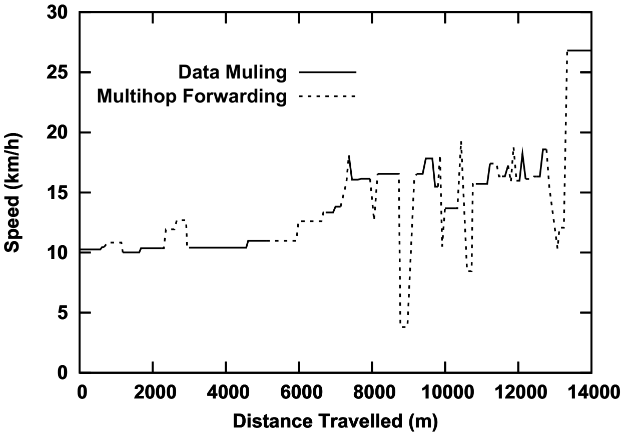

Figure 3: Correlation between node speed and forwarding strategy

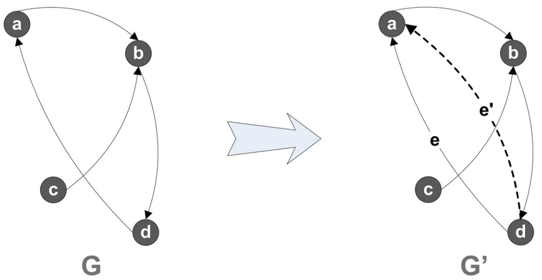

Figure 4: Replacing edge (d, a) in G by two sibling edges, one per strategy.

Let us now consider how we can annotate the edges of the extended graph G' with the two metrics: cost (C) and delay (Del).

For edges associated with the Data Muling strategy, we have:

$$Del_{DM} = \frac{\ell}{\overline{u}},\ C_{DM} = 1$$

where ℓ denotes the length of the edge and $\overline{u}$ the average vehicle speed along that edge. We fix the communication cost of the Data Muling strategy to 1 message transmission regardless of the segment length ℓ. The reason is simple: the vehicle carries the message along the entire road segment, and in the worst case, transmits it only once upon reaching the intersection.

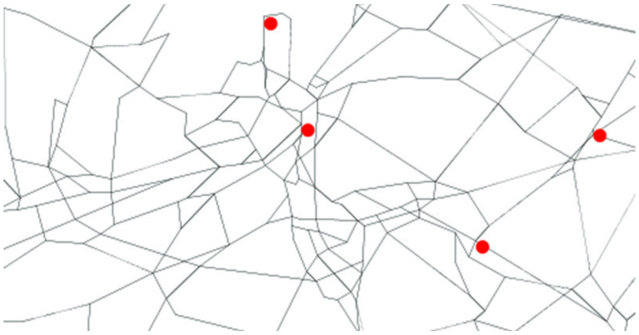

Figure 5: The section of the map of Zurich used in our experiments. Circles represent access points.

For edges associated with the Multihop Forwarding strategy, we must first check whether multihop is feasible on the road segment. For wireless communication range R, Multihop Forwarding is an available option if $\ell > R$ and $\overline{d} \geq \frac{\ell}{R}$, where $\overline{d}$ is the average vehicle density for the edge in question. If this is true, then

$$C_{MH} = \tfrac{\ell}{R},\ Del_{MH} = C_{MH} \times q$$

where q denotes the time required for the node to check its neighbor list and identify the best next hop.

3.2.2 *Path selection*

After annotating the edges of the extended graph G' with their corresponding delays and costs, the next step is to choose the minimum cost path, such that the total delay of the path does not exceed the message delay budget. By doing so, we will have not only selected the sequence of edges through which the message should be forwarded, but also the strategy that vehicles must adopt at each edge for the particular message. The *delay-constrained least-cost routing problem* is known to be NP-complete [6] and various heuristics have been proposed in the literature. D-MinCost utilizes one such heuristic, the *Delay Scaling Algorithm (DSA)* [7], in order to efficiently compute delay-constrained least cost paths from the vehicle's location to all access points on the network. By computing these least cost paths we are able to identify:

- The access point that can be reached with the least cost.
- The exact min-cost path to that access point.
- The strategy that should be followed at each edge of the path in order to adhere to the message's remaining delay budget.

D-MinCost maintains a neighbor list at each node through periodic beacon broadcasts, similarly to D-Greedy. When a message p is generated at the node, the algorithm applies the DSA heuristic on the extended graph G' for message p with delay budget TTL. The next intersection I is used as the location of the message. From the paths returned by $DSA(I, TTL)$, D-MinCost selects the minimum cost path that leads to an access point and encodes it in the message header. If the first edge of the path suggests the use of Data Muling, the vehicle carries the message until the next intersection I. Otherwise, the message is forwarded to the neighboring vehicle in range that is closest to I. Upon successful message reception, the neighbor returns an acknowledgment so that the sending node can remove the message from its buffer. Subsequently, the new message carrier will obey the strategy encoded in the message header together with the suggested path. The message path will be recomputed at the next intersection by its carrier only if it is not feasible to follow the suggested edge and its associated strategy. This can happen if, for instance, there are no available vehicles on the recommended edge[1]. In this case the edge is removed from graph G' and the DSA heuristic is reinvoked on the resulting graph in order to compute an alternative min-cost path.

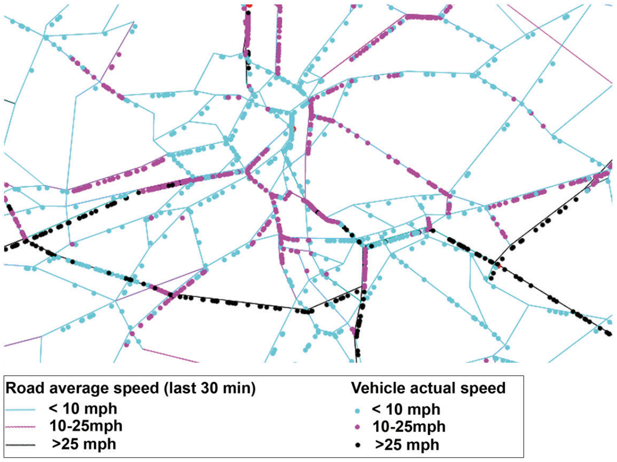

Figure 6: A snapshot of the map during the simulation. Road segments have been classified based on average vehicle speed and vehicles according to their actual speed.

4. EVALUATION

4.1 Node Mobility

The results of ad hoc network protocol studies are heavily influenced by the mobility model utilized [4]. The random-waypoint mobility model is amongst the most commonly used, which however fails to capture the dynamics of the urban vehicular scenario our protocols are designed for. In this study we are basing our evaluation on realistic vehicular traces from the city of Zurich. The traces have been produced by a multi-agent traffic simulator that simulates public and private traffic over a real map, based on actual travel plans of individuals [10]. The size of the simulation area is $250km \times 260km$ with 260.000 vehicles involved.

4.2 Simulation Setup

We have extracted a rectangular street area of size $20km \times 10km$, which covers the centre of the city and surrounding areas, and contains around 30000 distinct vehicle trajectories during a 30-minute interval in morning rush hour. We analyzed the trajectories to identify the four busiest intersections, and placed one stationary access point on each(Figure 5). We evaluate our protocols using

[1] Note that edges are directed; when a vehicle advertises its edge in the beacon message, it also implies its direction.

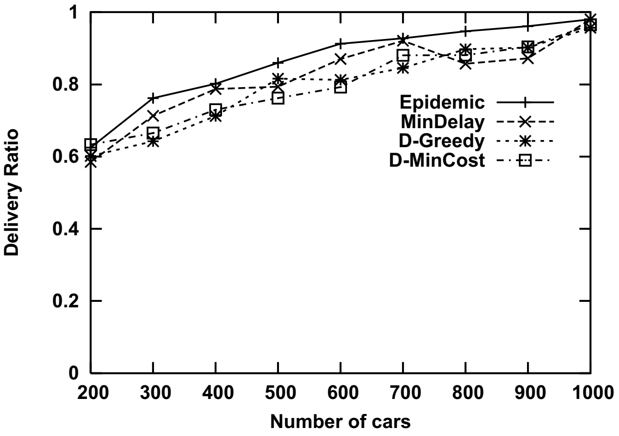

Figure 7: Message Delivery Ratio varying the number of cars (λ=1200sec)

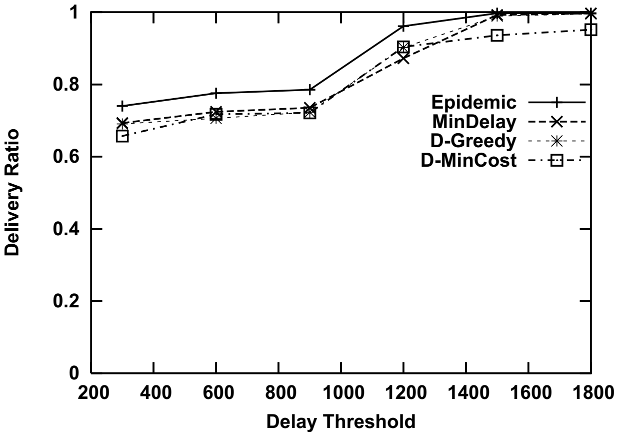

Figure 8: Message Delivery Ratio varying λ (number of cars=900)

a discrete event simulation environment developed with vehicular networks in mind in Java. Our simulator supports *openstreetmap* [1] geographic data, however, we have opted to extract the area map from the vehicular traces instead, in an attempt to eliminate unused streets and alleys from the resulting graph and render our simulations more tractable. We simulate 30 minutes of traffic and set the neighbor discovery beacon period at 5 seconds. We have selected the simulation interval to coincide with morning rush hour in the traces; Figure 6 shows a simulation snapshot where vehicle density and speed on different road segments of the map can be observed.

All simulations run during the same 30 minute interval that starts at t_0. For the evaluation of the D-MinCost algorithm we preload the street graph with traffic statistics computed during the 30 minute interval ending at t_0. 100 messages are generated during the first 50 seconds of the simulation and are randomly distributed among the participating vehicles. Our results are averages over 30 iterations. Table 1 lists the parameters of our experiments.

Number of Iterations	30
Iteration Duration	1800 sec
Beacon Period	5 sec
Number of Vehicles	[200-1000]
Delay Threshold (λ)	[300-1800] sec
Number of Messages Generated	100
Message Generation Interval	First 50 sec
Message Size	1500 KBytes
Communication Range	250m
Bitrate	500 Kbps

Table 1: Simulation Parameters

4.3 Results

We are comparing D-Greedy and D-MinCost with Epidemic as defined in [13] and *MinDelay*, inspired by the VADD protocols [16]. By exploiting all possible vehicle contacts, epidemic provides an upper bound for packet delivery ratio and a lower bound for delivery delay under our unlimited buffer assumption; one cannot outperform epidemic in terms of delivery ratio and delay for our scenario. MinDelay attempts to identify the minimum delay path on the extended graph G' described in Section 3.2. It aggressively utilizes the Multihop Forwarding strategy to achieve low delivery delays; however, unlike Epidemic, it is a single-copy scheme that forwards the message over a single min-delay path, significantly reducing the number of transmissions.

4.3.1 Delivery Ratio

In this section we compare the delivery ratio of D-Greedy and D-MinCost with Epidemic and MinDelay. We are measuring the fraction of messages that have reached an access point without exceeding the delay threshold λ. Suppose a message was generated at timestamp t_g and delivered at t_d. We consider the message successfully delivered only when $t_d - t_g < \lambda$.

Figure 7 shows the message delivery ratio for different car densities. λ is set at 1200 seconds. D-Greedy, D-MinCost and MinDelay exhibit very similar behavior, never falling behind Epidemic's optimal values by more than 10%. Naturally we expect the delivery ratio to increase for all algorithms as we increase the vehicle density, since more contacts between vehicles are exploited.

Figure 8 shows the message delivery ratio for different values of the delay threshold. For low delay thresholds, only packets close enough to an access point will be delivered leading to lower delivery ratio values. Our schemes are shown to perform very well, within 9% of Epidemic, across the different delay thresholds. MinDelay behaves similarly. Figure 8 confirms that the behavior we observed in Figure 7 is consistent for different values of λ.

4.3.2 Transmitted Bytes

In this section we measure the total number of bytes transmitted by each algorithm. This metric reflects the bandwidth utilization of each scheme. The total number of bytes is inclusive of any overhead incurred by control messages (e.g. beacons, acknowledgments) and protocol-specific headers.

Figures 9,10 show that our algorithms outperform MinDelay in terms of bandwidth usage. A multiple-copy scheme like

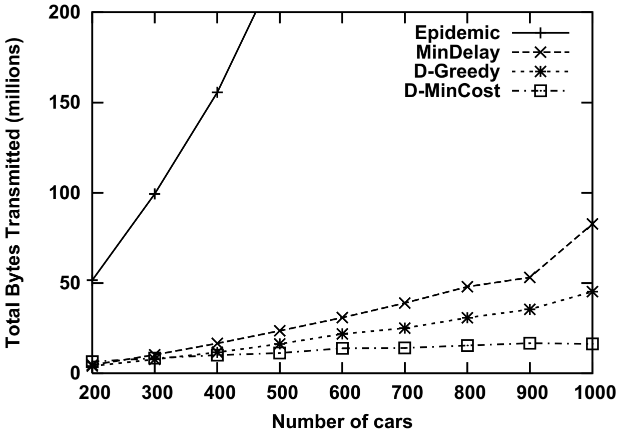

Figure 9: The total number of bytes sent for different car densities (λ=1200sec)

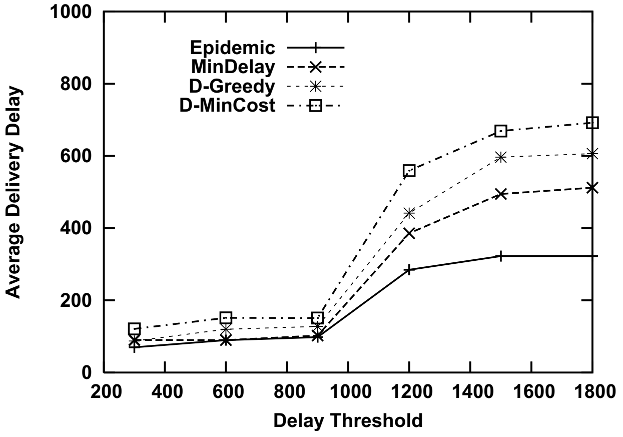

Figure 11: Average delivery delay for different values of λ (number of cars=900)

Epidemic is not expected to perform well in this case; in fact, it transmits at least an order of magnitude more bytes than the rest of the schemes, so we have focused on the lower portion of the graphs to better distinguish between MinDelay, D-Greedy and D-MinCost.

As we increase the number of cars in Figure 9, we observe that the rate of increase of bandwidth utilization for both D-Greedy and D-Mincost is significantly lower than that of MinDelay. D-Greedy transmits up to 45% less bytes than MinDelay while D-MinCost is even more conservative in its transmissions, outperforming MinDelay by up to 75%. This behavior is consistent across the different delay thresholds (Figure 10). D-MinCost is the top performer among all algorithms, which is not a surprise since we expect it to utilize Data Muling more frequently than D-Greedy, resulting in fewer transmissions.

By carefully alternating between the Multihop Forwarding and Data Muling strategies, our algorithms introduce very significant communication savings over the MinDelay scheme that scale gracefully with car density, while at the same time maintaining the delivery ratio close to optimal levels.

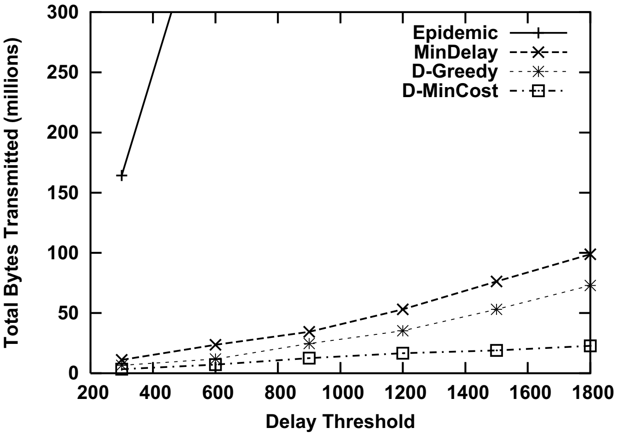

Figure 10: The total number of bytes sent for different values of λ (number of cars=900)

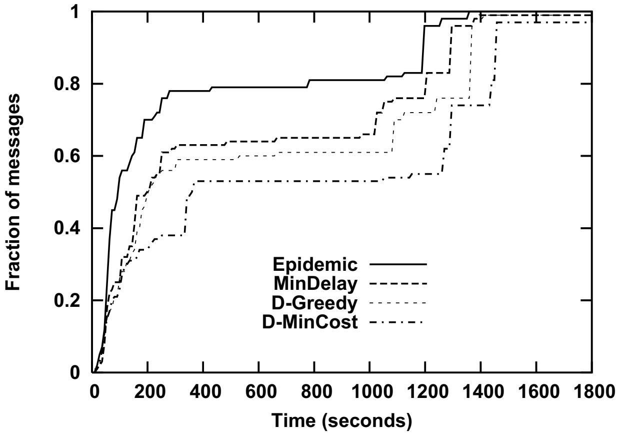

Figure 12: CDF of message delivery delay (900 cars, λ=1500)

4.3.3 Message Delay

Figure 11 shows the effect of different delay thresholds on the average message delivery delay. It is computed as the average of the delivery delays of all successfully delivered messages within the delay threshold λ. Epidemic always finds the minimum delay path as it takes advantage of every contact opportunity and forwards the message over all possible paths.

We observe that D-Greedy and D-MinCost are, on average, delivering messages later than MinDelay, especially for large values of λ. This is attributed to the fact that our algorithms try to exhaust the available delay threshold, by delivering messages as late as possible. By exploiting traffic statistics, D-MinCost is more effective than D-Greedy in doing so: it maintains a high delivery ratio (as seen in Figure 8) but delivers messages later than the other schemes. This is because D-MinCost will always follow the minimum cost path to the gateway that involves more Data Muling, whereas D-Greedy will follow the shortest one, ignoring possibly cheaper - and more time consuming - alternatives. In general, D-MinCost's paths are more likely to utilize the Data Muling Strategy that D-Greedy's paths. MinDelay

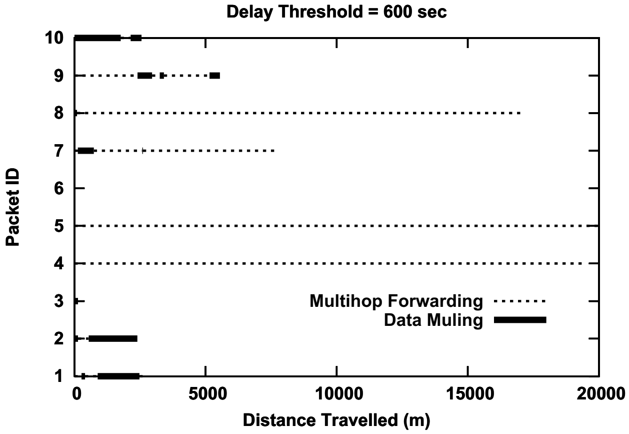

Figure 13: Strategy chosen for low $\lambda = 600s$

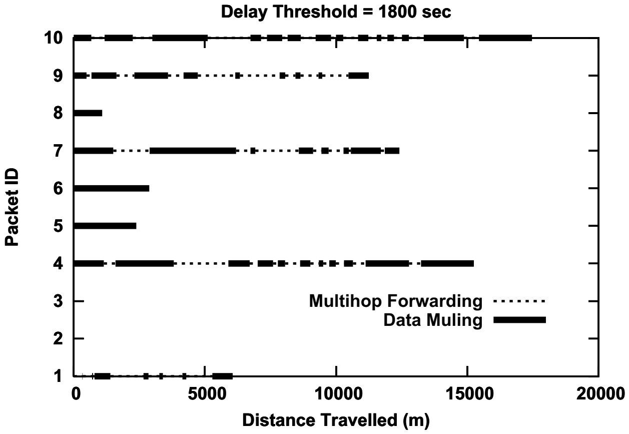

Figure 14: Strategy chosen for high $\lambda = 1800s$

does not utilize data muling proactively but merely when there is no other alternative.

For each of the simulated routing schemes we have plotted the CDF (Cumulative Density Function) of the message delivery delay in Figure 12. The y axis represents the fraction of delivered messages over all generated messages; λ is set at 1500 seconds. This figure confirms that D-MinCost better exploits the delay threshold than any other algorithm: it delivers almost half of the messages in the interval $[1200, 1500]sec$. D-Greedy delivers 29% of the messages during the same interval, a 9% improvement over MinDelay.

4.4 Effect of λ

D-Greedy and D-MinCost do not aggressively use the Multihop Forwarding strategy like MinDelay; instead, they gracefully alternate between the Multihop Forwarding and Data Muling strategies aiming to exhaust the message delay threshold and minimize the communication cost, effectively trading allowable delay for bandwidth. In an attempt to show the effect of the delay threshold λ on our algorithms, we run two simulations with different λ values, 600 and 1800, where we generate 10 messages and examine the strategy followed by each message throughout its journey towards the gateway.

Figures 13 and 14 show, for D-Greedy, the strategy chosen per message during the course of the simulation as a function of the distance covered by the message. In Figure 13, where the delay threshold is set at 600 secs, we observe that messages that need to travel long distances to an access point make aggressive use of the Multihop Forwarding mode, while messages closer to an access point alternate between the two modes. A similar trend is observed in Figure 14 where the delay threshold is set to 1800 secs. Comparing the two figures verifies that the Data Muling strategy is much more frequently used when messages have a high delay threshold. The results we obtained for D-MinCost are very similar and thus have been omitted.

5. RELATED WORK

Driver behavior, high speeds and constraints on mobility imposed by the road infrastructure have important implications for the design of routing protocols in automotive ad hoc networks. Rapid changes in link connectivity cause many paths to disconnect before they can be utilized. Routing protocols must therefore be designed to handle frequent topology changes and network disconnections [2], rendering carry-and-forward protocols designed with Delay Tolerant Networks in mind a reasonable choice.

Epidemic routing addresses the challenge of sparsely and intermittently connected mobile networks, by allowing nodes to carry their content and opportunistically forward it to other nodes that they encounter [13]. Unlike traditional routing protocols for MANETs [11], epidemic routing achieves message delivery even in the case where a connected path from source to destination is rarely available. When two nodes meet they exchanges the messages they don't carry in common until their buffers are identical. This flooding-like behavior allows epidemic to achieve very high delivery rates and very low latency at the cost of high bandwidth utilization.

Several protocols have been proposed in the literature with vehicular networks in mind. Chen et al. studied the efficiency of carry-and-forward algorithms for data dissemination among vehicles in the context of highways. Briesemeister et al. [3] proposed an epidemic-style protocol to multicast messages about an accident to cars with a specific role (e.g. geographic location, speed and direction), limiting message propagation to a certain number of hops. Opportunistic exchange of messages is also explored in [15] for resource discovery among vehicles. When vehicles are within communication range, they evaluate the relevance of their resources using a spatio-temporal function, and exchange only the most relevant resources; the least relevant resources that do not fit in memory are purged. The authors of [5] have developed protocols that disseminate information to a set of target zones, rather than specific destination nodes. They utilize a propagation function whose value is minimized over the target zones. Unlike our work, these studies do not exploit statistics and patterns of vehicle mobility to carefully design their data dissemination protocols. In our work, we exploit knowledge of average speed and car density on nearby road segments, but no knowledge of individual vehicle trajectories. Unlike [17], we assume no control over vehicle movement, i.e. we are not able to modify vehicle trajectories proactively for communication purposes.

MOVE [9] considers the scenario where location-aware mobile nodes attempt to deliver information to a stationary destination whose position is globally known, not unlike our model's access points. It relies on the relative velocity of a node and its neighbors to make forwarding decisions and assumes that a node will maintain its heading until it reaches the destination. This is not realistic for our urban vehicular scenario where a node may change its heading multiple times while turning at intersections before reaching an access point.

MDDV [14] aims to route information to receivers that have expressed an interest for it. The road network is abstracted as a directed graph and weights are assigned to each edge of the graph that depend on the type of road it represents. The forwarding trajectory of a message is predecided and set as the trajectory that minimizes the sum of weights on that graph between the source and a vertex in the destination region. The main weakness of MDDV is the restriction it imposes on messages to move on the predefined forwarding trajectory. We provide a smarter forwarding algorithm, where intermediate nodes are allowed to modify and improve the message trajectory.

Zhao and Cao [16] make very similar assumptions to ours, in that they assume knowledge of traffic statistics on different road segments, and they design vehicle assisted data delivery (VADD) protocols taking into account traffic patterns over a predefined road layout. However, their goal is to identify lowest-delay delivery paths, whereas our goal is to deliver packets within a user-specified delay threshold over minimum cost paths.

6. CONCLUSIONS AND FUTURE WORK

We have proposed two novel packet forwarding schemes for urban vehicular networks, D-Greedy and D-MinCost, that aim to route packets towards fixed infrastructure nodes. Our algorithms leverage local knowledge as well as global traffic statistics to reach forwarding strategy decisions that minimize communication cost and at the same time adhere to a per-packet application defined delay threshold. We have conducted a thorough experimental evaluation of our schemes utilizing realistic vehicular traces on a real city map. We have compared with Epidemic routing that achieves optimal delay and delivery ratio under our scenario, as well as with MinDelay, a greedy delay-minimizing scheme inspired by [16]. We have shown that our schemes significantly outperform the competing algorithms in terms of communication cost, while achieving a similar delivery ratio to Epidemic for a wide range of car densities and delay thresholds. The cost savings of D-Greedy and D-MinCost are derived from carefully alternating between the Multihop Forwarding and Data Muling strategies while maintaining delivery delay below the required threshold. For high delay budgets, the cost-saving Data Muling strategy is heavily utilized.

We are planning future extensions for our schemes in an attempt to further enhance their behavior. D-MinCost currently considers global traffic statistical information in order to provide predictions for strategy costs. The accuracy of the algorithm's predictions depends on this information, which can easily become out of date if the vehicle has not met an infrastructure node for a while, or if the infrastructure has not recently received information for a certain part of the network. It would be interesting to extend D-MinCost to allow for dynamic updates to a node's traffic statistics by utilizing the knowledge of neighboring nodes; a node could then obtain very accurate traffic information for the surrounding area, allowing for more efficient path computation and strategy selection.

In this work we have considered a busy urban scenario, where the wireless medium is expected to be congested throughout. In future work we plan to extend our algorithms for operation in scenarios with high variability of network conditions. In areas where the network bandwidth is underutilized, multihop forwarding can be utilized aggressively, reserving delay budget for resorting to the data muling strategy in more congested parts of the network.

7. REFERENCES

[1] `http://www.openstreetmap.org`.

[2] J. Blum, A. Eskandarian, and L. Hoffman. Challenges of inter-vehicle ad hoc networks. In *IEEE Transactions on Intelligent Transportation Systems, 5(4)*, pages 347–351, 2004.

[3] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Annual Workshop on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2000.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[5] P. Costa, D. Frey, M. Migliavacca, and L. Mottola. Towards lightweight information dissemination in inter-vehicular networks. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 20–29, New York, NY, USA, 2006. ACM Press.

[6] M. Garey and D. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W.H. Freeman, January 1979.

[7] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis. Efficient computation of delay-sensitive routes from one source to all destinations. In *Conf. on Computer Communications (INFOCOM)*, pages 854–858, 2001.

[8] N. Golmie, R. E. V. Dyck, and A. Soltanian. Interference of bluetooth and ieee 802.11: simulation modeling and performance evaluation. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 11–18, New York, NY, USA, 2001. ACM Press.

[9] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 4, pages 2289–2293, May/June 2005.

[10] V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *ACM International*

Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2006.

[11] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. In *IEEE Personal Communications*, 1999.

[12] J. Singh, N. Bambos, B. Srinivasan, and D. Clawin. Wireless lan performance under varied stress conditions in vehicular traffic scenarios. In *IEEE VTC Fall 2002, volume 2*, pages 743–747, 2002.

[13] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks.

[14] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. Mddv: a mobility-centric data dissemination algorithm for vehicular networks. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 47–56, New York, NY, USA, 2004. ACM Press.

[15] B. Xu, A. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *IEEE International Conference on Mobile Data Management (MDM)*, 2004.

[16] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. In *Conf. on Computer Communications (INFOCOM)*, 2006.

[17] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 187–198, 2004.