Efficient Data Propagation in Traffic-Monitoring Vehicular Networks

Antonios Skordylis and Niki Trigoni

Abstract—**Road congestion and traffic-related pollution have a large negative social and economic impact on several economies worldwide. We believe that investment in the monitoring, distribution, and processing of traffic information should enable better strategic planning and encourage better use of public transport, both of which would help cut pollution and congestion. This paper investigates the problem of efficiently collecting and disseminating traffic information in an urban setting. We formulate the traffic data acquisition problem and explore solutions in the mobile sensor network domain while considering realistic application requirements. By leveraging existing infrastructure such as traveling vehicles in the city, we propose traffic data dissemination schemes that operate on both the routing and the application layer; our schemes are frugal in the use of the wireless medium, rendering our system interoperable with the proliferation of competing applications. We introduce the following two routing algorithms for vehicular networks that aim at minimizing communication and, at the same time, adhering to a delay threshold set by the application: 1) delay-bounded greedy forwarding and 2) delay-bounded minimum-cost forwarding. We propose a framework that jointly optimizes the two key processes associated with monitoring traffic, i.e., data acquisition and data delivery, and provide a thorough experimental evaluation based on realistic vehicular traces on a real city map.**

Index Terms—**Ad hoc network, data muling (DM), delay-tolerant networks, intervehicle communication, multihop (MH) communication, routing, sensor participation, traffic monitoring, vehicular ad hoc networks (VANETs), vehicular networks.**

I. Introduction

RESEARCHERS and automotive industries are envisioning the deployment of *ambient traffic-monitoring applications*, wherein vehicles equipped with the Global Positioning System (GPS) detect local traffic and periodically report it to one of the stationary roadside units dispersed throughout the city. These units are referred to as access points (APs) and act as gateways to the city's traffic-monitoring center (TMC) and the outside world.

One of the most important attributes of traffic data is *freshness*, i.e., the interval between the time that the data are generated by a vehicle on a particular road and the time that the data are made available to the user as a query response. Informally, data freshness indicates how stale the data are and to what extent they can be used to estimate trip times or to select the fastest route to a destination in a reliable manner. Depending on the expected rate of change in traffic conditions, users may have different freshness requirements for different parts of the city or for different times of the day. It is crucial that the ambient traffic-monitoring application provides deterministic guarantees that the available traffic data satisfy the specified freshness requirements.

At the same time, the ambient traffic-monitoring application will share bandwidth resources with various applications that run on the same vehicular ad hoc network (VANET), e.g., applications that provide Internet access to passengers, commercial applications that flood advertisements about nearby stores, and safety applications that provide drivers with emergency braking services.

Thus, our high-level goal is to design an ambient traffic-monitoring system that minimizes bandwidth utilization while adhering to user-defined data freshness requirements. To achieve this goal, we investigate the following two intertwined aspects of traffic monitoring, both of which significantly impact both data freshness and bandwidth utilization: 1) data acquisition and 2) data delivery. Our contributions are listed as follows.

1) We formulate a novel problem in the context of ambient traffic monitoring, i.e., minimizing the communication cost required to monitor traffic while providing deterministic guarantees of data freshness.
2) We propose two novel delay-tolerant routing algorithms for vehicular networks, i.e., delay-bounded greedy forwarding (D-Greedy) and delay-bounded minimum-cost forwarding (D-MinCost), which leverage locally available information about traffic and global traffic statistics to reach forwarding strategy decisions that minimize communication.
3) We propose a framework for vehicular networks that jointly optimizes the two key processes associated with monitoring traffic, i.e., data acquisition and data delivery.
4) We evaluate the benefits of our approach using realistic traffic traces on a real city map.

The remainder of this paper is organized in six sections. In Sections II and III, we present our assumptions and objective. In Section IV, we discuss our traffic data acquisition algorithm, and in Section V, we present and evaluate two novel data delivery algorithms for vehicular networks that are suitable for traffic information propagation. In Section VI, we present an

Manuscript received October 10, 2010; accepted March 27, 2011. Date of publication July 14, 2011; date of current version September 6, 2011. This work was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/C547640/1. The Associate Editor for this paper was Y. Wang.

A. Skordylis was with the Department of Computer Science, University of Oxford, OX1 3QD Oxford, U.K. He is now with Thomson Reuters, New York, NY 10036 USA (e-mail: antonios.skordylis@comlab.ox.ac.uk).

N. Trigoni is with the Department of Computer Science, University of Oxford, OX1 3QD Oxford, U.K. (e-mail: niki.trigoni@comlab.ox.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2159857

in-network data reduction methodology by jointly considering the *data acquisition* and *data delivery* phases for traffic data. Section VII offers the concluding remarks.

II. MODEL

We assume that the network consists of vehicles that travel in an urban environment and several stationary APs spread across the city, which do not provide full-city coverage. APs act as collection points for sensor readings and feed data through a backbone connection to the TMC, where user applications consume traffic data. We assume that they can only be installed on road intersections. Using short- to mid-range transceivers, vehicles can communicate with neighboring vehicles or stationary APs within a range of 250 m.

We assume that vehicles can obtain their absolute position using a positioning service (e.g., GPS). Vehicles are also equipped with a digital street 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 that connects a and b and vehicles can travel from a toward b on that segment. We also assume that the map is preloaded with traffic statistics about the street network, i.e., the average speed $\overline{u}$ and average vehicle density $\overline{d}$ at each road segment. The map also contains the locations of the AP nodes. In addition, we assume that, using onboard sensors (e.g., GPS and laser), vehicles can estimate the current average speed $\overline{u}$ and average vehicle density $\overline{d}$ on the road segment that they are traversing.

Depending on the location where a message is generated, it may need to be relayed multiple times through several vehicles before it reaches one of the APs. When traffic density is low or 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 these messages. Vehicles can either choose to continue carrying buffered messages as they move closer to one of the APs or to forward these messages to other vehicles in their vicinity.

III. GOALS

We envisage that a variety of applications could benefit from traffic data that are acquired and collected by the vehicular network. Applications could widely vary in their requirements for data freshness; for example, an emergency response application, e.g., an ambulance coordination service, has stringent constraints on data freshness on the order of a few minutes. On the other hand, a road maintenance company that works overnight could tolerate data staleness of tens of minutes to decide how to plan road repair work.

We aim at minimizing the bandwidth utilization of a traffic-monitoring system while adhering to user-defined data freshness requirements. To achieve this goal, we investigate two system aspects that significantly impact both data freshness and bandwidth utilization: 1) *data acquisition* and 2) *data delivery*.

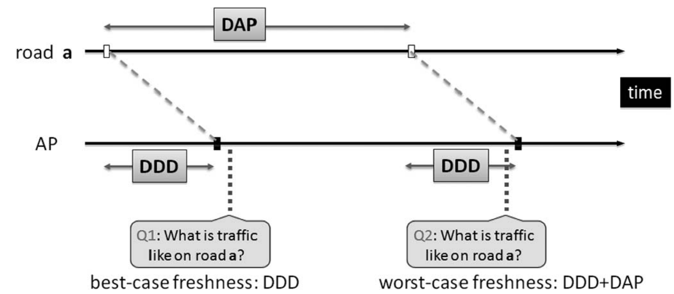


Fig. 1. Depending on when a user asks the TMC about traffic on a particular road, he/she will receive results of varying freshness. The worst-case freshness is DDD+DAP. A data acquisition period DAP can only be achieved if at least one vehicle passes from road a every DAP time units. Similarly, a DDD can be achieved only if there are enough vehicles in the map to enable a packet to travel from road a to the AP in DDD time units. Details about the data acquisition and data delivery algorithms are provided in Sections IV and V, respectively.

Data acquisition refers to the sampling of road traffic information by passing vehicles. High sampling rates can be achieved by having vehicles participate in the sampling process and generate traffic information messages with high frequency. The lower the data acquisition period (DAP) is, the fresher the traffic data that become available for each road, but the larger the number of traffic messages propagated through the network.

Data delivery refers to the propagation of traffic messages from the originating vehicle to one of the APs dispersed in the city. Traffic messages can be delivered either by wireless multihop forwarding (MF) or by physically carrying messages at the vehicle's speed toward an AP. We propose hybrid algorithms that carefully combine MF and data muling (DM) to achieve a desirable delivery delay. Clearly, the lower the data delivery delay (DDD), the fresher the traffic data available at the APs, but the higher the use of MF, and thus, the higher the communication cost.

Fig. 1 shows that the freshness of traffic data is directly dependent on the DAP and DDD. Consider the example where users wish to query the speed of vehicles on a particular road. Let traffic messages concerning this road be generated every DAP time units and let these messages take DDD time units to be delivered from the source vehicles to the AP. As shown in Fig. 1, users who query traffic information immediately after the arrival of a traffic message get the freshest data, whereas users who pose their queries just before the arrival of a traffic message get the stalest data. The best case freshness equals the DDD, whereas the worst case freshness equals the sum of the DDD and the DAP (DDD+DAP). Data freshness is of particular interest to users in simple yet common traffic-monitoring systems, in which the most recent reading about a road segment serves as an indication of the current traffic conditions on that road segment. This condition is a key assumption underlying this paper. In applications where time-series analysis and prediction techniques are employed to estimate future traffic conditions, we should also consider other factors, e.g., the temporal granularity of recently acquired data. In such applications, users are interested not only in getting fresh data but in achieving a desired accuracy in the predicted traffic values as well. In this paper, we limit our study to the first class of applications, where users are interested in using the latest reports on traffic, as long as they remain sufficiently fresh.

The question that arises is: Given a user-defined freshness threshold F that bounds the sum of DAP and DDD, how should we split it into DAP and DDD to minimize the total message transmissions in the network? Should we frequently sample traffic (select low DAP) and allow the routing algorithm to incur a high DDD (select high DDD) or is it preferable to infrequently generate traffic messages (select high DAP) and deliver them as fast as possible (select low DDD)?

Our objective is to strike a good balance between the delay budgets allocated to data acquisition and data delivery while keeping their sum below the freshness threshold. Our approach for achieving our objective is twofold.

1) We propose and evaluate delay-tolerant data delivery algorithms that trade message delay for communication, and we define a distributed data acquisition algorithm that compensates for variable traffic conditions and node movement.
2) We propose a framework that jointly optimizes the data acquisition and data delivery; we investigate how their combined operation trades data freshness for communication cost and propose a mechanism that fine-tunes their parameters to minimize the communication incurred by the traffic-monitoring system.

IV. Data Acquisition

A. Background

A large part of the literature on sensor participation schemes for field coverage refers to stationary sensor networks [1]–[5]. Previous work that concentrates on mobile sensors operates on the assumption that sensor mobility can be controlled, and therefore, sensors can be moved on demand to ensure the coverage of the sensing field [6]. Other recent works discuss selection schemes, where the problem is to decide which sensor to move to compensate for node failures [7], [8]. To optimize node selection for a particular task, most of the aforementioned approaches present distributed algorithms that require message exchange between the mobile nodes.

B. Algorithm

For a traffic-monitoring scheme to be successful in an urban environment, it must ensure complete coverage of the sensing field. In the scenario that we consider, this case translates into providing regular traffic information updates for every road segment in the network. If a stationary sensor network will be used, it would suffice to position one or more traffic sensors on each road, uniformly distributed across the road's length, and task them to generate traffic update messages with DAP. In our case, however, sensor nodes are mobile, and we have no control over their mobility. We would like to task the mobile nodes in such a way so that at least one traffic message per road is generated every DAP time units.

To optimize node selection for a particular task, most of the approaches in the literature present distributed algorithms that require message exchange between the mobile nodes, incurring undesirable communication overhead. Because our main goal is to reduce the communication cost associated with traffic monitoring, we have opted to use a probabilistic sensor participation scheme, wherein each node independently and probabilistically decides on whether to participate in the sensing task. Each node participates in sensing, i.e., generates a traffic information message, with probability P_g. The value of P_g is computed based solely on locally available data.

We would like our mobile sensor network to provide an output similar to a stationary sensor network: one traffic message per road every DAP time units. Node mobility introduces the following two issues that need to be addressed: 1) variable node position and 2) variable traffic conditions. We address the first issue by only allowing vehicles to generate messages at a predefined fixed point on each road segment, e.g., the segment midpoint, effectively simulating a stationary sensor mounted on that point. To compensate for variable traffic conditions, we carefully tune the message generation probability P_g.

Adhering to a constant DAP requires messages to be generated with frequency $f_g = 1/DAP$. The vehicle can locally derive its average speed $\overline{u}$ and the average vehicle density $\overline{d}$ for the road that it traverses using onboard sensor information. We assume that vehicles that run the traffic-monitoring application broadcast short beacons at regular intervals (e.g., every $T =$ 5 s) and use these beacons to discover their neighbors. A vehicle can estimate the vehicle density around it by counting the number of beacons received by distinct vehicles within the communication range. We clarify that we are interested in the density of vehicles that run the traffic-monitoring application and thus take part in the data acquisition process and not in the density of all vehicles on the road. Let b be the number of beacons received from distinct vehicles in the last $1s$. We assume that the vehicle has moved very little ($<$ 11 m if the speed is $<$ 40 km/h) in the last $1s$, and it can hear $1/T$ of the cars within its communication range R. Thus, an estimate of the current density is $d = (T \times b)/(2 \times R)$. The average density $\bar{d}$ is locally computed by each vehicle by averaging density measurements over several consecutive 1-s sliding windows. When a vehicle comes within the communication range of an AP, it offloads the density estimates made since the last encounter with an AP. These data are further forwarded into a centralized data base, which maintains historical information about road segment densities at different times of the day. Assuming uninterrupted flow conditions, we can derive the average flow q of vehicles on each road as follows: $\overline{q} = \overline{u} \cdot \overline{d}$. The desired probability is given as follows: $P_g = (f_g/q) \Rightarrow P_g = 1/(DAP \cdot \overline{u} \cdot \overline{d})$. Intuitively, the higher the flow of vehicles over the road midpoint where sensing is performed, the lower the value of P_g necessary to maintain a constant sensing period DAP.

V. Data Delivery

Once the traffic information message has been generated, the underlying routing protocol will forward it to the AP. The routing protocol is responsible not only for the message delivery delay but for the number of transmissions until successful delivery occurs as well.

A. Background

Previous studies on network capacity [9], [10] consider ad hoc networks with multiple pairs of users who want to communicate with each other. In contrast, we consider mobile nodes (vehicles) that all want to upload their sensor readings to fixed APs. Thus, our model is similar to the infostation model described in [11]. The capacity of the wireless network is constrained by the limited number of APs and the interference of concurrent transmissions when uploading data to the same AP.[1]

If, without interference, a vehicle within the range of an AP can upload data with a bit rate of $\Theta(B)$ b/time unit, then the capacity of the network is $\Theta(B|AP|)$, where $|AP|$ is the number of APs in the monitored area. Let us assume that vehicles can share this capacity in a fair manner by generating data at similar rates, encountering APs with a similar frequency, and equally sharing the wireless medium with other vehicles collocated near the same AP. If the user has no stringent constraints on the delivery of their sensor readings, the available throughput per node is $\Theta(B|AP|/n)$, where n is the number of vehicles. This throughput can be achieved if vehicles carry their data at the vehicle's speed until they come within communication range of an AP, at which point they upload their stored data. Thus, the transfer capacity of the vehicular network is $\Theta(B|AP|)$.

Driver behavior, high speeds, and constraints on mobility imposed by the road infrastructure have important implications for the design of routing protocols in VANETs. 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 [12]. Unlike traditional routing protocols for mobile ad hoc networks (MANETs) [13], epidemic routing achieves message delivery, even in the case where a connected path from the source to the destination is rarely available.

Several protocols have been proposed in the literature with regard to vehicular networks. Chen *et al.* studied the efficiency of carry-and-forward algorithms for data dissemination among vehicles in the context of highways [14]. Briesemeister *et al.* [15] 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 [16] for resource discovery among vehicles. When vehicles are within the communication range, they evaluate the relevance of their resources using a spatiotemporal function and exchange only the most relevant resources; the least relevant resources that do not fit in the memory are purged. The authors in [17] have developed protocols that disseminate information to a set of target zones rather than specific destination nodes. Unlike this paper, these studies do not exploit the statistics and patterns of vehicle mobility to carefully design their data dissemination protocols. Unlike [18], we assume no control over vehicle movement, i.e., we cannot proactively modify vehicle trajectories for communication.

[1]Note that APs are assumed to be far away from each other and, thus, do not interfere with each other.

MOVE [19] considers the scenario where location-aware mobile nodes attempt to deliver information to a stationary destination whose position is globally known, unlike our model's APs. Unlike our work, the approach 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.

MDDV [20] aims at routing 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, which depends on the type of road that it represents. The forwarding trajectory of a message is predecided, whereas in this paper, we allow intermediate nodes to modify and improve the message trajectory.

Zhao and Cao [21] make very similar assumptions to ours, because 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.

Due to space limitations, we have mainly focused on routing techniques for vehicular networks. This paper is clearly related to a large body of research on routing for sensor networks, an overview of which is provided in [22].

B. Algorithms

We propose the following two novel routing algorithms for VANETs: 1) *D-Greedy* and 2) *D-MinCost*. The goal of our algorithms is to exploit a user-specified delay threshold to save communication while delivering messages from vehicles to an AP. Both algorithms assume that vehicles are equipped with a digital street map of the area, which is abstracted as a direct graph, as discussed in detail in Section II. This map is loaded once when a vehicle is within the range of an AP, and it very infrequently changes thereafter. Hence, the communication cost of updating the map is considered negligible. The first proposed algorithm (D-Greedy) exploits *local traffic conditions*, i.e., information about the speed and density of cars at the road segment that it currently traverses. This algorithm is suitable for scenarios where vehicles are not aware of traffic conditions on every road of the city but can nevertheless sense traffic conditions in their vicinity. The second algorithm (D-MinCost) assumes knowledge of *global traffic conditions*, i.e., statistical information about the speed and density of cars on every road segment of the city. Note that the proposed algorithms do not assume any knowledge of the vehicles' planned trajectories.

Both algorithms attempt to reduce the total number of message transmissions needed to forward a message to an AP within the message-specific delay threshold. To do so, they proactively alternate between the following two forwarding strategies.

1) MF refers to the aggressive forwarding of messages to vehicles that are better positioned to deliver them to an AP.

2) DM refers to buffering messages in the local memory and carrying these messages at the vehicle's speed.

The novelty of our proposed algorithms lies in their careful alternation between the MF and DM strategies to achieve a good tradeoff between delay and communication cost. This approach is in stark contrast with the previously proposed VADD protocols [21], which aim at minimizing message delay and thus always prefer MF to DM when the former approach 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 MF strategy for one message and DM for another message.

1) D-Greedy: The D-Greedy algorithm defines a forwarding strategy that assumes no knowledge of traffic information beyond node speed, which can locally be derived from the available location information. Based only on local knowledge, D-Greedy assumes that the best path to an AP is the shortest path, i.e., the path that minimizes the sum of the lengths of the edges on the directed graph G that abstracts the street map. We use Dijkstra's algorithm to compute the shortest path between the current vehicle position and the geographically closest AP. When multiple APs exist, the algorithm selects the closest path, i.e., the path on the shortest path beginning at the vehicle's position.

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 AP $(distToAP)$. $distToAP$ 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 tolerated delay value (TTL) and is considered useful only if delivered before TTL has elapsed.

a) 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 greedy decision depends on the remaining delay (TTL) until the expiration time of a message and on its distance to the closest AP $(distToAP)$. Because global traffic information is not available, D-Greedy assumes that the remaining message delay budget can uniformly be 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. If, for a certain message, the delay allocated to the current edge exceeds the delay of the vehicle that travels along that edge, the DM strategy is selected for that particular message. Otherwise, the algorithm assigns the MF 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 travels. $distToAP$ denotes the current distance from the closest AP on the shortest path, and u is the average speed of the vehicle, calculated during a constant-size 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 DM strategy will be used to carry the message to the next intersection as

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

If $Del_{DM} \leq Del$, then the algorithm opts for the DM strategy, i.e., it refrains from transmitting the message to save bandwidth while adhering to the delay budget. Otherwise, the MF strategy is chosen. In this case, the message is forwarded to the neighboring vehicle in range that is closest to the AP (see Fig. 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 MF strategy switch to DM. Similarly, if the carrying vehicle moves away from the closest AP, messages that were originally assigned to use the DM strategy switch to the MF strategy.

Fig. 3 shows the strategy selection of D-Greedy in action. Observe that, when the message is carried by a vehicle with high speed, it is propagated with the DM strategy, whereas when a vehicle with low speed carries the message, it is propagated with the MF strategy. DM is allowed at lower speeds during the early lifetime of a message, because the algorithm overestimates the delay allocated at each edge, because it assumes that the message will follow the shortest path to the

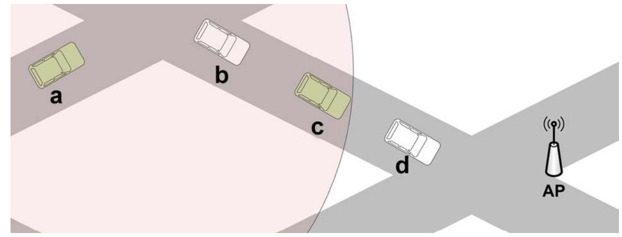


Fig. 2. Node a will choose to forward the message to node c, which is the closest node to the AP among the nodes in range.

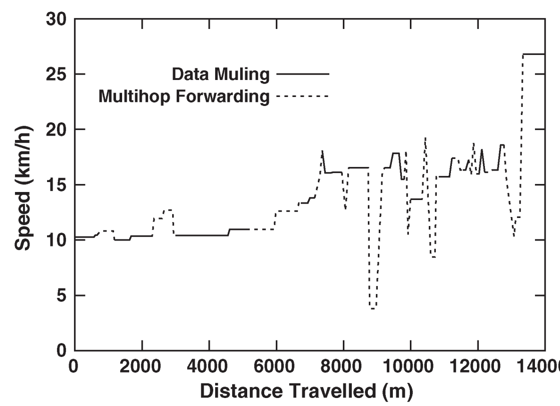


Fig. 3. Correlation between node speed and forwarding strategy.

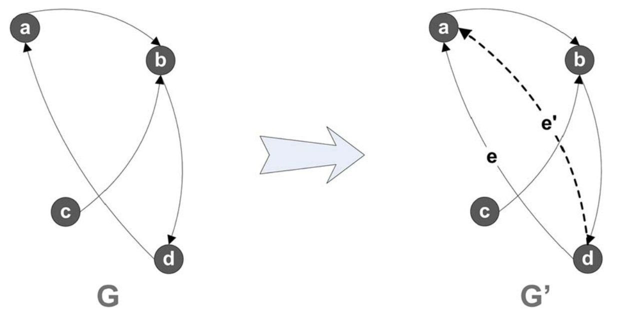


Fig. 4. Replacing edge (d, a) in G by two sibling edges, one edge per strategy.

AP. As the message progresses through the network, the delay budget tightens, and only high-speed carriers are allowed to perform DM.

To summarize, D-Greedy is an algorithm that locally runs on each vehicle and periodically decides the fate of each message in the vehicle's buffer, i.e., whether to wirelessly forward it to another vehicle or to continue to locally carry it at the vehicle's speed. The decision for each message is based on a simple calculation (multiplication by a constant and comparison with a constant). Thus, D-Greedy takes linear time and space in the number of messages.

2) 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.

a) 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 the following two metrics: 1) cost C, which represents the number of message transmissions along the edge, and 2) delay Del, which denotes 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 use the DM or the MF strategy. To solve this case, 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 $e' \in G$ that connects the same two vertices. The original edge e corresponds to a road segment when the DM strategy is utilized, whereas edge e' corresponds to the same road segment when the MF strategy is used. Consider, for example, the graph in Fig. 4, where 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.

Let us now consider how we can annotate the edges of the extended graph G' with the following two metrics: 1) cost C and 2) delay Del.

For edges associated with the DM strategy, we have

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

where ℓ denotes the length of the edge, and $\overline{u}$ is the average vehicle speed along that edge. We fix the communication cost of the DM strategy to one 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.

For edges associated with the MF strategy, we must first check whether multihop (MH) is feasible on the road segment. A necessary condition is that $\ell > R$ and $\overline{d} \geq (1/R)$, where $\overline{d}$ is the average vehicle density for the edge in question. However, this condition is not sufficient, and the higher the average vehicle density and the communication range, the higher the probability of MH connectivity. For simplicity, we only check the necessary condition ($\ell > R$ and $\overline{d} \geq (1/R)$), and if true, we create a MH edge and label it with the following cost and delay[2]:

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

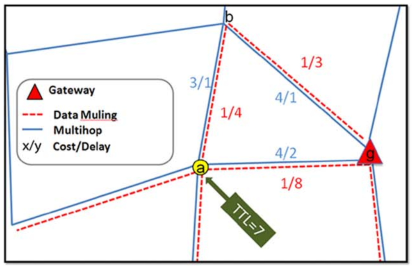
where q denotes the time required for the node to check its neighbor list and identify the best next hop.

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 selected not only 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 [23], and various heuristics have been proposed in the literature. D-MinCost utilizes one such heuristic, i.e., the *delay-scaling algorithm* (*DSA*) [24], to efficiently compute delay-constrained least cost paths from the vehicle's location to all APs in the network. By computing these least cost paths, we can identify the following factors:

- the AP that can be reached with the least cost;
- the exact minimum-cost path to that AP;
- the strategy that should be followed at each edge of the path to adhere to the message's remaining delay budget.

D-MinCost maintains a neighbor list at each node through periodic beacon broadcasts, similar 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. Based on the paths returned by $DSA(I, TTL)$, D-MinCost selects the minimum-cost path that leads to an AP and encodes this path in the message header. If the first edge of the path suggests the use of DM, 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

[2]In future work, we will consider a probabilistic model for measuring the cost and delay of a MH edge, taking into account the vehicle density and communication range.



Path	Cost	Delay	Feasible Path
a MF g	4	2	yes
a DM g	1	8	no
a DM b DM g	2	7	yes
a DM b MF g	5	5	yes
a MF b DM g	4	4	yes
a MF b MF g	7	2	yes

Fig. 5. D-MinCost considers all outgoing edges and selects the feasible path (path delay <= TTL) with the minimum communication cost.

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 case can happen if, for example, there are no available vehicles on the recommended edge.[3]

In this case, the edge is removed from graph G', and the DSA heuristic is reinvoked on the resulting graph to compute an alternative minimum-cost path.

Consider the example in Fig. 5. A node has arrived at intersection a, carrying a message whose remaining TTL value is equal to 7. The table lists all possible paths from a to the AP g with all strategy selection combinations: either DM or MF. Assuming that a vehicle is available at every outgoing edge of intersection a, our previous algorithm, which uses greedy forwarding, will only consider propagating over edge (a, g), because this edge is on the shortest path. It would then detect that DM would incur a higher delay than TTL allows and would have opted for MF over (a, g), spending four transmissions. D-MinCost, on the other hand, will try to find the cheapest path that satisfies the delay requirement. It will consider all outgoing edges and eventually choose to propagate over (a, b) using DM and, subsequently, over (b, g) using DM as well. This case will incur a communication cost of two transmissions, which is less than D-Greedy, and a delay of 7, which is equal to the TTL value.

C. Evaluation

1) Node Mobility: It is widely accepted in the literature that the results of ad hoc network protocol studies are heavily influenced by the mobility model utilized [25]. The random-waypoint mobility model is among the most commonly used approaches, which, however, fails to capture the dynamics of the urban vehicular scenarios for which our protocols are destined. In this paper, we base our evaluation on realistic vehicular traces from the city of Zurich, Switzerland. The traces have been produced by a multiagent traffic simulator that simulates public and private traffic over a real map based on actual travel plans of individuals [26]. The size of the area is 250 km × 260 km, with 260 000 vehicles involved.

2) Experimental Setup: For our evaluation, we have extracted a rectangular street area of size 20 km × 10 km, which covers the center of the city and surrounding areas and contains around 30 000 distinct vehicle trajectories during a 30-min interval in the morning rush hour. We analyzed the trajectories to identify the four busiest intersections and placed one stationary AP on each. We evaluate our protocols using a discrete event simulation environment developed with vehicular networks in mind in Java. Our simulator supports *openstreetmap* [27] geographic data; however, we have opted to extract the area map from the vehicular traces in an attempt to eliminate unused streets and alleys from the resulting graph and render our simulations more tractable. We simulate 30 min of traffic and set the neighbor discovery beacon period at 5 s. We have selected the simulation interval to coincide with the morning rush hour in the traces. Fig. 6 shows a simulation snapshot where vehicle density and speed on different road segments of the map can be observed.

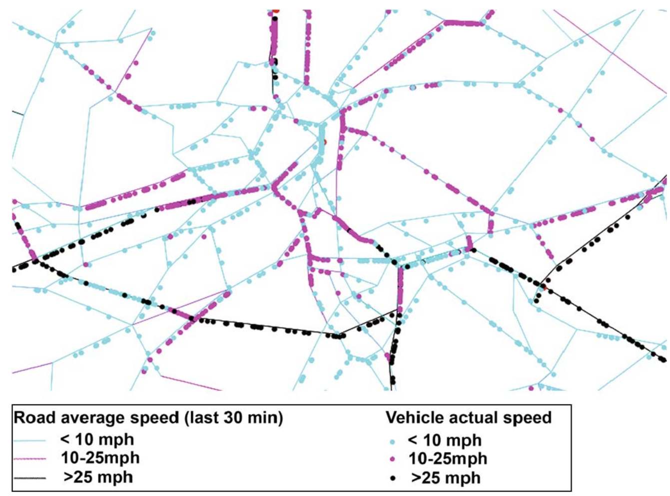


Fig. 6. Snapshot of the map during the simulation. Road segments have been classified based on the average vehicle speed, and vehicles have been classified according to their actual speed.

All simulations run during the same 30-min 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-min interval, ending at t_0. One hundred messages are generated during the first 50 s of the simulation and are randomly distributed among the participating vehicles. Our results are averaged over 30 iterations. Table I lists the parameters of our experiments.

3) Performance Metrics: We compare D-Greedy and D-MinCost with the Epidemic protocol, as defined in [12], and the *MinDelay* protocol, which is inspired by the VADD protocols [21]. By exploiting all possible vehicle contacts, Epidemic provides an upper bound for message delivery ratio

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

TABLE I
SIMULATION PARAMETERS

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

and a lower bound for delivery delay under our infinite buffer assumption. We cannot do better than Epidemic in terms of delivery ratio and delay for our scenario. MinDelay tries to identify the minimum-delay path on the extended graph G' described in Section V-B2. It makes aggressive use of the MF strategy, because its goal is to minimize delay. However, unlike the Epidemic protocol, it typically forwards messages through a single minimum-delay path and thus incurs lower communication cost.

For each algorithm, we measure the following metrics.

1) *Message delivery ratio*: The percentage of messages that have reached an AP and do not exceed the delay requirement λ.
2) *Average message delivery delay*: The time between message generation and delivery at an AP, averaged over all delivered messages. Again, only messages that do not exceed λ are considered.
3) *Bytes transmitted*: The total number of bytes transmitted by the algorithm. This value is used as an indication of bandwidth utilization.

D. Simulation Results

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

Fig. 7 shows the message delivery ratio for different car densities. λ is set at 1200 s. 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, because more contacts between vehicles are exploited.

Fig. 8 shows the message delivery ratio for different values of the delay threshold. For low delay thresholds, only packets that are close enough to an AP 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. Fig. 8 confirms that the behavior that we observed in Fig. 7 is consistent for different values of λ.

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 and acknowledgments) and protocol-specific headers.

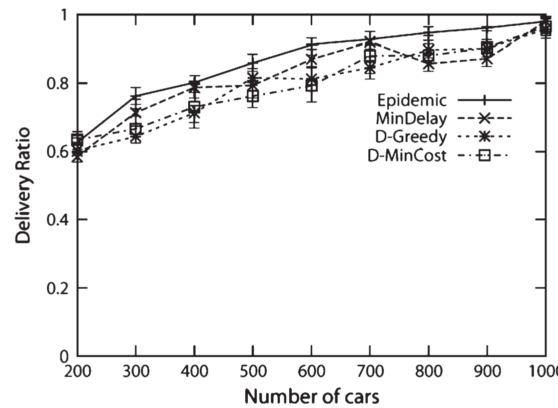


Fig. 7. Message delivery ratio varying the number of cars ($\lambda = 1200$ s).

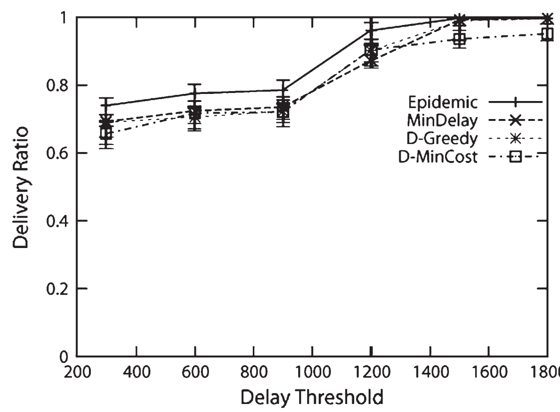


Fig. 8. Message delivery ratio varying λ (number of cars $= 900$).

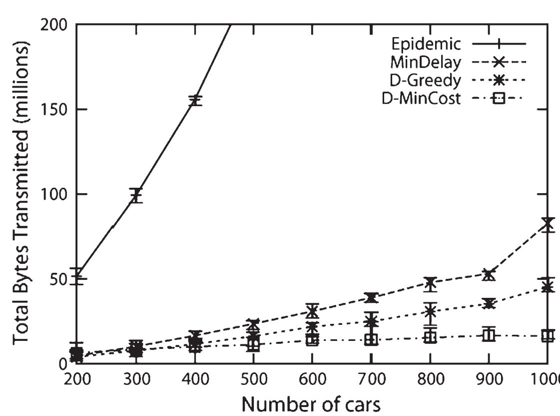


Fig. 9. Total number of bytes sent for different car densities ($\lambda = 1200$ s).

Figs. 9 and 10 show that our algorithms outperform MinDelay in terms of bandwidth usage. A multiple-copy scheme such as 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; therefore, we have focused on the lower portion of the graphs to better distinguish between MinDelay, D-Greedy, and D-MinCost.

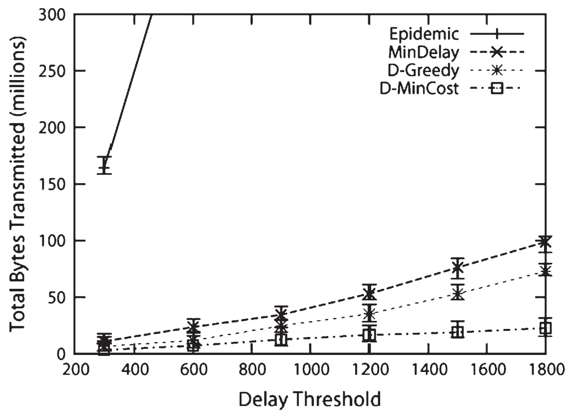


Fig. 10. Total number of bytes sent for different values of λ (number of cars = 900).

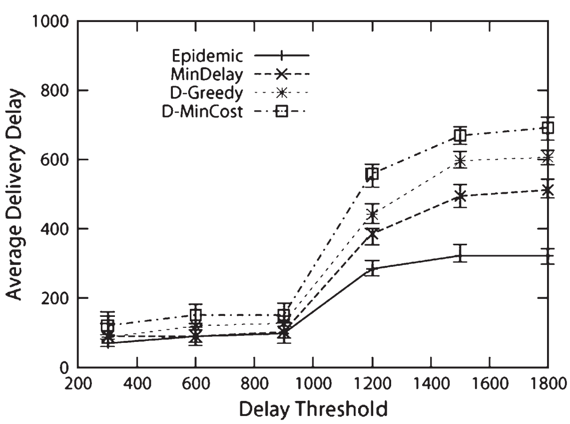


Fig. 11. Average delivery delay (DDD) for different values of λ (number of cars = 900).

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

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

3) Message Delay: Fig. 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, because it takes advantage of every contact opportunity and forwards the message over all possible paths.

We observe that D-Greedy and D-MinCost, on the average, deliver messages later than MinDelay, particularly for large values of λ. This case 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 shown in Fig. 8) but delivers messages later than the other schemes. This case is because D-MinCost will always follow the minimum-cost path to the AP that involves more DM, whereas D-Greedy will follow the shortest path, ignoring possibly cheaper (and more time-consuming) alternatives. In general, D-MinCost's paths are more likely to utilize the DM Strategy than D-Greedy's paths. MinDelay does not proactively utilize DM but merely when there is no other alternative.

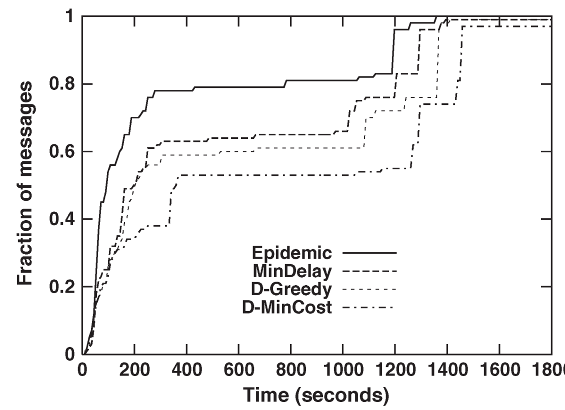


Fig. 12. CDF of DDD (900 cars, $\lambda = 1500$).

For each of the simulated routing schemes, we have plotted the cumulative density function (cdf) of the message delivery delay in Fig. 12. The y-axis represents the fraction of delivered messages over all generated messages, and λ is set at 1500 s. This figure confirms that D-MinCost better exploits the delay threshold than any other algorithm: it delivers almost half the messages in the interval [1200, 1500] s. D-Greedy delivers 29% of the messages during the same interval—a 9% improvement over MinDelay.

4) Effect of λ: D-Greedy and D-MinCost do not aggressively use the MF strategy similar to MinDelay; instead, they gracefully alternate between the MF and DM strategies, aiming at exhausting the message delay threshold and minimizing the communication cost, effectively trading allowable delay for bandwidth. To show the effect of the delay threshold λ on our algorithms, we run two simulations with different λ values, i.e., 600 and 1800, where we generate ten messages and examine the strategy followed by each message throughout its journey toward the AP.

Figs. 13 and 14 show, for D-Greedy, the strategy chosen per message during the simulation as a function of the distance covered by the message. In Fig. 13, where the delay threshold is set at 600 s, we observe that messages that need to travel long distances to an AP make aggressive use of the MF mode, whereas messages closer to an AP alternate between the two modes. A similar trend is observed in Fig. 14, where the delay threshold is set to 1800 s. Comparing the two figures verifies that the DM strategy is much more frequently used when messages have a high delay threshold.

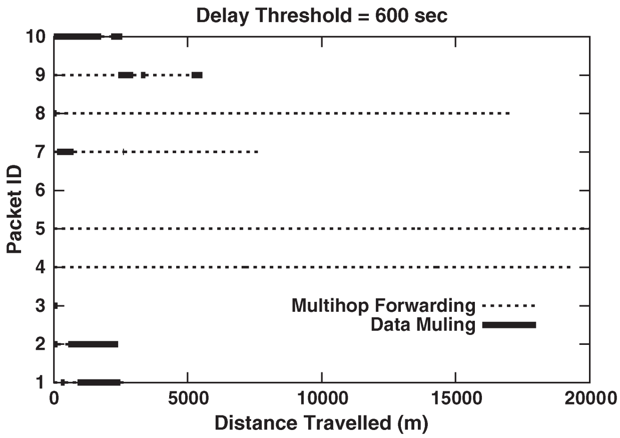


Fig. 13. Strategy chosen for low $\lambda = 600$ s.

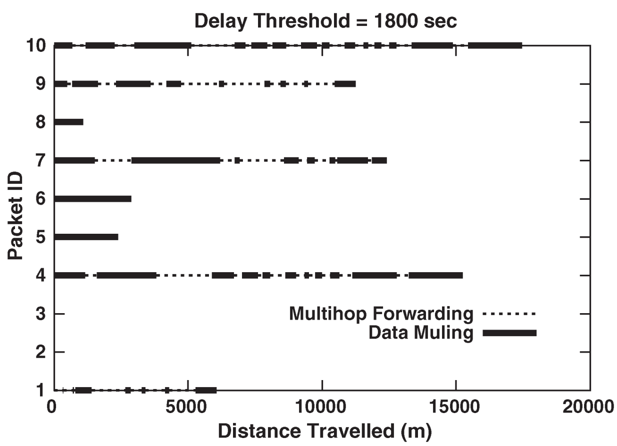


Fig. 14. Strategy chosen for high $\lambda = 1800$ s.

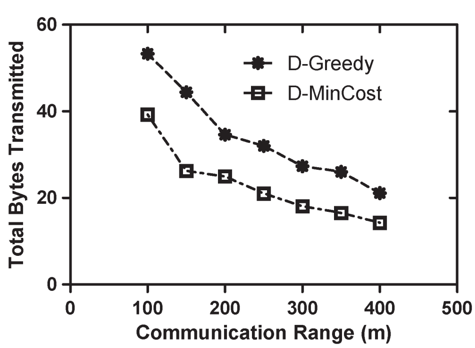


Fig. 15. Total number of bytes sent for different communication ranges ($\lambda = 1200$ s, number of cars $= 900$).

5) Effect of Communication Range: Figs. 15 and 16 show the effect of varying the communication range on the achieved delivery ratio and communication cost of the proposed algorithms. We observe that both algorithms achieve very similar delivery ratios under different communication range scenarios and exhibit similar sensitivity to these approaches, with the delivery ratios dropping when the range is reduced. This case is expected, because the list of discovered neighbors will significantly be shorter for small communication ranges. Although the delivery ratios do not drop to unacceptable levels, this condition comes at a very high communication cost. D-MinCost maintains the lead in communication cost performance throughout the different range scenarios.

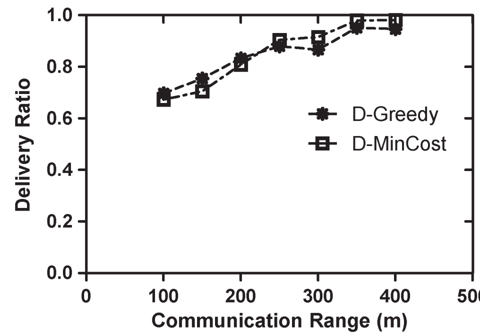


Fig. 16. Delivery ratio for different communication ranges ($\lambda = 1200$ s, number of cars $= 900$).

TABLE II
SIMULATION PARAMETERS

Simulation area	$3.5km \times 8km$
Number of sensors	112
Simulation duration	336 hours
Antenna type	omnidirectional
Propagation model	free space
Transmission range	250 m
Packet size	30 bytes (constant)
Packet payload	18 bytes
Report period	24 hours
FC-Temporal window size	1 week

E. DDD Analysis

We have chosen to utilize D-Greedy as the data delivery algorithm for the remainder of our analysis to render our simulations more tractable. For our subsequent analysis, we have doubled the duration of our simulations, increasing it to a 60-min interval during the morning rush hour. We have also uniformly distributed 150 stationary APs on road intersections in the area. All other simulation parameters remain the same (see Table II).

Recall that the delay budget that is initially available to a message is an algorithm parameter that the user can vary, called the delay threshold (λ). D-Greedy attempts to deliver the message to the closest AP within the user-defined delay threshold λ. In fact, it endeavors to deliver as close to λ as possible by aggressively utilizing MF for low values of λ and using DM when λ is high.

Whether D-Greedy can achieve the λ delay target inevitably depends on the underlying network topology; it may be impossible for messages generated far from an AP to be delivered within certain low λ thresholds, whereas messages that originate near an AP might be delivered much sooner than λ, even if DM is used for the duration of the routing phase. In our scenario, we would like to know the actual delay DDD incurred by the routing algorithm. Knowing DDD allows us to allocate the remaining data freshness budget to the DAP.

We have examined the effect of the algorithm parameter λ (delay threshold) on the actual delivery delay (DDD) incurred for different roads. Fig. 17 shows the effect of λ on the maximum delivery delay incurred for a road. We show 95%

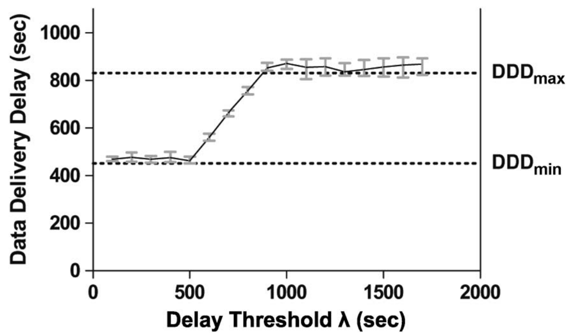


Fig. 17. Effect of the D-Greedy parameter λ on the actual DDD.

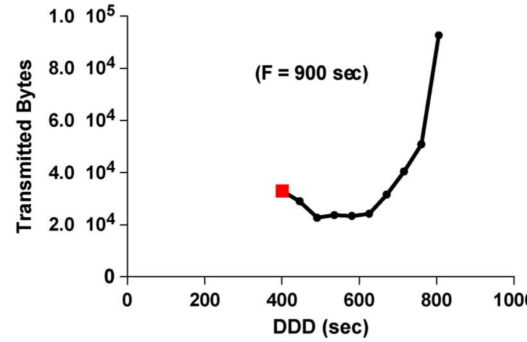


Fig. 18. Communication cost across the valid DDD range for a single road.

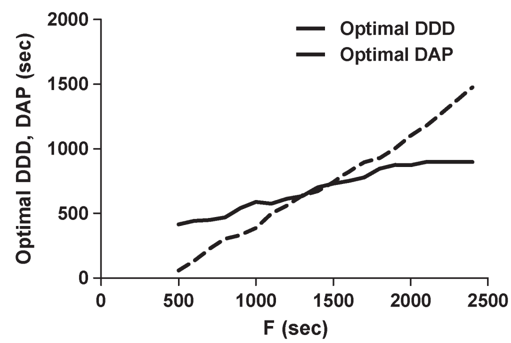


Fig. 19. (DDD, DAP) pairs for optimal communication savings and a single road.

confidence intervals as a result of 30 iterations, each with a different set of participating vehicles that are randomly chosen from our traces. We observed that, for every road, there is a lower bound $DDD_{\min}$ on how fast the data can be propagated and an upper bound $DDD_{\max}$ above which the routing algorithm cannot further delay messages to save extra bandwidth. We also observed that, for $DDD_{\min} \leq DDD \leq DDD_{\max}$, D-Greedy always achieves the λ target, resulting in a linear relationship between λ, i.e., the algorithm parameter, and DDD, i.e., the resulting delay. By storing $DDD_{\min}$ and $DDD_{\max}$, as well as the slope a and intercept b of the least squares fit between the two points, we now can predict not only the range of allowable DDD values per road but the corresponding λ parameter of the D-Greedy algorithm that results in the desired DDD as well. To aid our optimizations in the next section, we preload the street map with the values $DDD_{\min}$ and $DDD_{\max}$, as well as the slope a and intercept b for each road.

VI. Joint Optimization

A user query with a data freshness requirement of F provides an upper bound for the worst case freshness allowed by the system. Based on Fig. 1, recall that the following condition needs to be satisfied:

$$DDD + DAP \leq F. \quad (1)$$

In Sections IV-B and V-E, we have discussed how increasing the value of either DDD or DAP will result in less message transmissions in the network. Thus, to keep the number of message transmissions to a minimum, we need to maintain the sum $DDD + DAP$ as close to F as possible to exhaust the available freshness budget. The naive approach for splitting the budget between DDD and DAP would be to select $DDD_{\min}$ for the DDD, i.e., route data as fast as possible, and utilize the full remaining budget $(F - DDD_{\min})$ to slow down data acquisition. We refer to this basic approach as *rapid delivery*. In other words, *rapid delivery* aims at reducing the rate of traffic information generation as much as possible.

This basic approach does not necessarily yield optimal communication savings. We investigate whether we can outperform *rapid delivery* by jointly optimizing the data acquisition and data delivery tasks as follows. In Section VI-A, we examine how we can divide the freshness budget into DDD and DAP in search for the optimal balance that minimizes communication. We measure how this balance is affected by different freshness budgets and by road proximity to the AP. In Section VI-B, we compare the communication savings of *rapid delivery* to our joint optimization approach.

A. Algorithm Tuning

In Section V-E, we have noticed that the actual DDD DDD incurred by the routing algorithm lies within a certain interval for each road $[DDD_{\min}, DDD_{\max}]$. We measure the communication cost incurred, in the form of transmitted bytes, for DDD values within this interval and their corresponding DAP values, where $DAP = F - DDD$. Each DDD value corresponds to a λ value used to set up the routing algorithm (Section V-E), whereas DAP values control the data acquisition rate for each road (see Section IV-B).

Fig. 18 shows the bytes transmitted for different values of DDD for a single road when the freshness requirement F is set to 900 s. We observe that, for $DDD \cong 500$, the bandwidth utilization is minimized for this road. This case essentially means that, for a specific freshness budget, it is worth allocating part of the budget to slow down data delivery rather than to use it all to slow down data acquisition. Observe the square point on the graph that corresponds to $DDD_{\min}$ and, thus, to the *rapid delivery* algorithm: By jointly optimizing, we achieved a 30% reduction in communication cost compared to *rapid delivery* for this road.

For the same road, Fig. 19 shows the optimal DDD value as we vary the freshness budget. The corresponding optimal DAP value that results from the choice of DDD is also shown. A comparison of the DDD and DAP slopes reveals that, as the

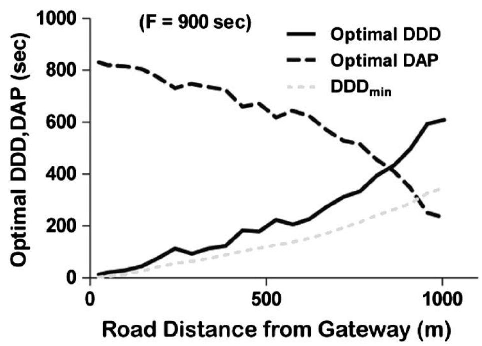


Fig. 20. Optimal (DDD, DAP) pairs as a function of road distance from the AP.

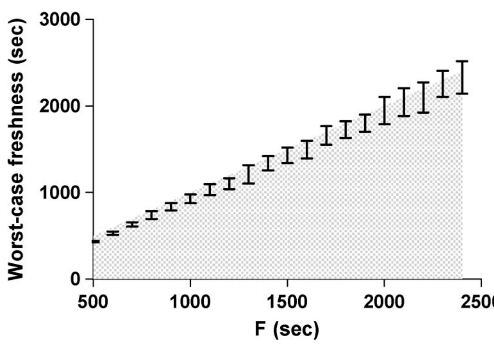


Fig. 21. Worst case freshness of traffic data for a road 840 m away from the AP.

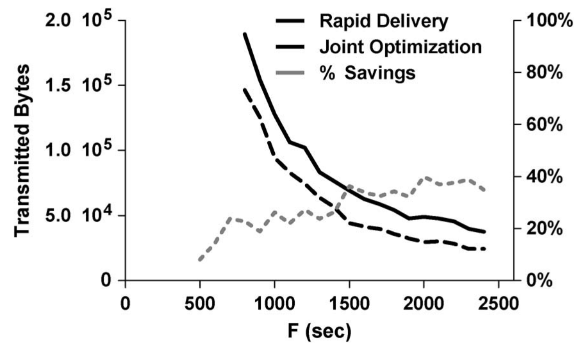


Fig. 22. Transmitted bytes (single road) as a function of F.

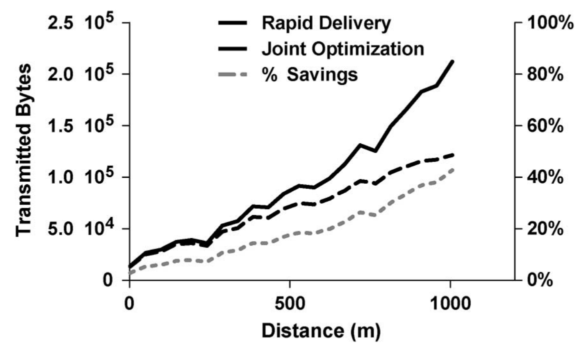


Fig. 23. Transmitted bytes for roads at various distances from the AP.

freshness budget increases, we should allocate proportionally more delay to data acquisition than data delivery for optimal behavior. Observe that, after the 2000-s mark, DDD ceases to increase, because $DDD_{\max}$ has been reached. From that point onward, the extra freshness budget is exclusively absorbed by DAP.

One important variable that affects the behavior of the routing algorithm is the road distance from the closest AP. For roads that are farther away, messages need to travel longer distances and over more hops to reach the AP. Fig. 20 shows the optimal (DDD, DAP) pairs for roads at different distances from the AP. $DDD_{\min}$ is also shown here, which corresponds to the delivery delay that *rapid delivery* incurs. There are several conclusions that can be derived from Fig. 20. The optimal (DDD, DAP) pairs are almost linearly dependent on distance, which provides us with a mechanism for assigning (DDD, DAP) pairs to any road based solely on its distance from the AP. One interesting observation is that, for roads closer to the AP, the freshness budget should mostly be allocated to data acquisition. For roads farther from the AP, the freshness budget should increasingly be allocated to data delivery. Note that, for roads farther away from the AP, the optimal delay for data delivery (DDD) is significantly larger than the minimum possible delay for data delivery $(DDD_{\min})$.

Fig. 21 shows the worst case freshness achieved for a specific road when using our optimization scheme. For different values of F, we measured the worst case freshness, i.e., the message that a user would receive if he/she issued a query just before the arrival of a new message at the AP. The shaded area represents the freshness budget F. Our scheme performs as desired, because it comes very close to exhausting the available freshness budget.

B. Benefits

Figs. 22 and 23 depict the benefits of the joint optimization. In Fig. 22, we observe increasing benefits of our approach over the approach that utilizes *rapid delivery* as we relax the freshness requirement, which reaches up to 38%. Fig. 23 outlines the benefits as a function of road distance from the AP. As anticipated, following our observations in Fig. 20, joint optimization saves more communication cost compared to *rapid delivery* for roads that are farther away from the AP, reaching up to 42% for the farthest roads. Our approach is not very effective for roads close to the AP. We could thus omit optimization for roads that surround the AP without significantly impacting the number of bytes transmitted.

C. Effect of Traffic Volume

To perform our analysis and joint optimization, we have selected the simulation interval to coincide with the morning rush hour in the traces. Our results are based on the early morning traffic patterns that occur between 7 A.M. and 8 A.M. Rush hour generally results in high car densities on the urban street map.

The traffic traces that we have available have been produced from a traffic simulator that generates routes based on actual travel plans of individuals. The simulator seems to mostly consider commuter travel plans, leaving a gap between 10 A.M. and 2 P.M. Fig. 24 shows how the average traffic density varies during the day for a single road from our traces. To examine

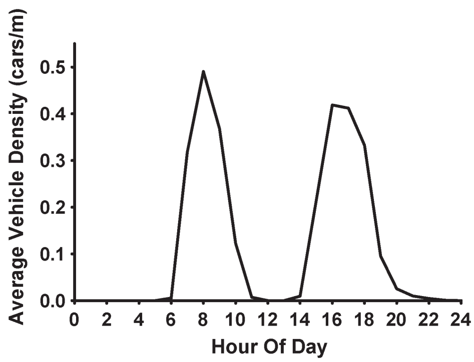


Fig. 24. Average vehicle density for a single road during a 24-h period.

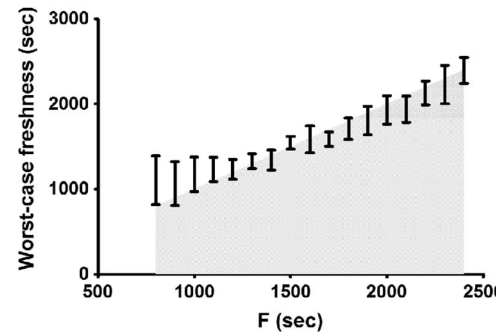


Fig. 26. Worst-case freshness of traffic data for a road 840 m away from the AP in the sparse network. The shaded area represents the freshness budget F.

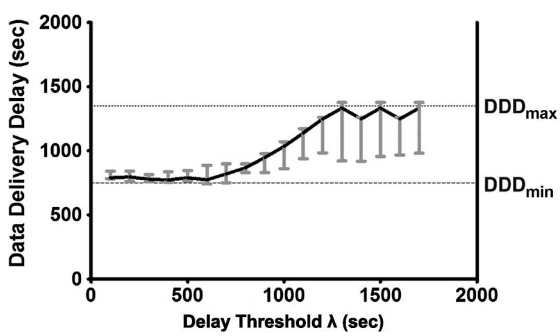


Fig. 25. Effect of the D-Greedy parameter λ on the actual DDD in the sparse network.

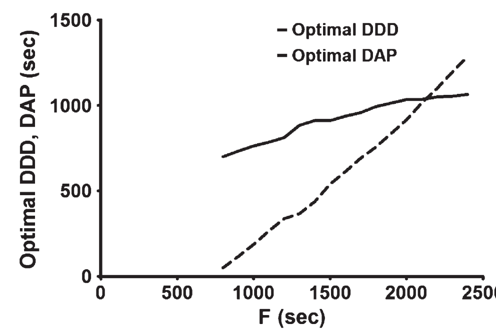


Fig. 27. Optimal (DDD, DAP) pairs as a function of road distance from the AP.

how our optimization is affected by lower traffic densities, we utilize the interval 14:30–15:30 for our simulations. During this interval, the average traffic density observed is 40% lower than in our previous simulations.

1) Effect on the Routing Algorithm: It is reasonable to expect that lower traffic volumes will introduce extra delay during the routing phase. Fig. 25 shows that sparse traffic has a significant impact on the routing delay. Compared with Fig. 17, based on our rush-hour results, we observe that, for the same road, $DDD_{\min}$ has increased from 450 s to about 750 s. This case is for a road that is roughly 800 m away from the AP. As a result, we cannot expect better freshness results than 750 s for that particular road. λ has a similar effect on the behavior of the routing algorithm, but it is now more erratic. Our results show higher variation, which we attribute to the fact that there are not enough cars for the routing algorithm to route over similar paths at every iteration. $DDD_{\max}$ can be as low as during the rush hour, but it can also reach much higher values in this case. We utilize the maximum value of $DDD_{\max}$ for mapping λ to DDD. For example, for this particular road, we set $DDD_{\max}$ to 1350 s compared with about 900 s in the rush-hour scenario.

2) Effect on the Data Acquisition Algorithm: Recall that, in the data acquisition phase, we would like our mobile sensor network to provide an output similar to a stationary sensor network, i.e., one traffic message per road every DAP time units. To achieve this goal, we task each vehicle to transmit a traffic message with a certain probability, i.e., $P_g = (f_g/q) \Rightarrow P_g = 1/(DAP \cdot \overline{u} \cdot \overline{d})$, at the road segment midpoint, accounting for the different speed and density conditions that are directly observable by the vehicle's sensors. However, the fact that we consider low vehicle densities in this data set, i.e., low values of $\overline{d}$, could mean that, to maintain the DAP requirement, P_g takes values higher than 1 for some roads. A P_g value greater than 1 effectively means that each vehicle would need to transmit more than one message to satisfy the DAP requirement, or in other words, there are not enough vehicles on the road to produce traffic messages as often as required.

As shown in Fig. 26, this fact impacts the received message freshness when the freshness budget is low, because this case is when $DDD = DDD_{\min}$ and short DAP intervals are required to achieve the freshness target. As the freshness budget relaxes and part of it is allocated to the DAP, freshness values again begin to fall within range.

3) Effect on the Optimization Results: Fig. 27 shows the optimal DDD value as we vary the freshness budget. The corresponding optimal DAP that results from the choice of DDD is also shown. Our conclusion based on Fig. 19 that, as the freshness budget increases, we should proportionally allocate more delay to data acquisition than data delivery, still holds and is even more pronounced. Data delivery has less effect on the number of transmissions when the network is sparse. This condition can be justified by the fact that, in a sparse network, the data delivery algorithm does not have as several opportunities to perform MF, and thus, relaxing the delay requirement has a less significant impact on the number of transmissions. To verify our hypothesis, we measure the average number of hops traveled by messages generated at a specific road until they reach the AP for both the sparse and dense networks (see Fig. 28).

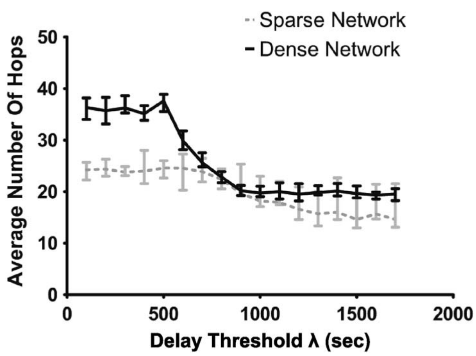


Fig. 28. Average number of hops traveled by messages generated at a specific road until they reach a gateway for both the sparse and dense networks.

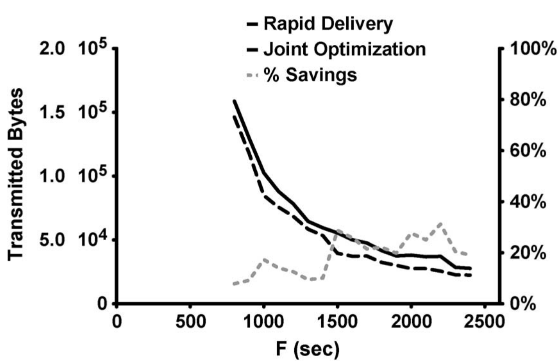


Fig. 29. Transmitted bytes (single road) as a function of F in the sparse network.

The following interesting observations can be made after examining Fig. 28.

- In the sparse network, the number of hops starts at a lower value, because there are not as several opportunities for MH.
- The confidence intervals are much tighter in the dense network, where the routing algorithm is likely to use similar paths, because there is a higher probability that a vehicle will be available as the next hop on the preferred path.
- In the sparse network, the effect of relaxing the delay threshold λ is not as pronounced, which verifies our hypothesis.
- In the sparse network, we observe that the number of hops drops lower than in the dense-network case for large λ values. It would appear that this case represents energy savings that the algorithm should also achieve in the dense network. However, D-Greedy uses shortest path routing to reach the AP. In a sparse network, a message might have to veer off the shortest path during DM, because next-hop vehicles on the shortest path are less likely to exist, causing it travel on edges outside the preferred path for larger periods of time.

Fig. 29 shows the benefits of our optimization for the sparse-network scenario. As shown in Fig. 22, we observe increasing benefits of our approach as we increase the freshness budget. However, in this case, our benefits reach up to 27% compared with 42% during the rush hour. Because the DDD does not play such a significant role in this case, our optimization is closer to the rapid-delivery approach results, where the entire freshness budget is allocated to the DAP.

VII. Conclusion

In this paper, we have defined the problem of minimizing the communication incurred by traffic-monitoring systems while providing deterministic guarantees of information freshness. We have proposed algorithms that will be utilized in both key processes associated with monitoring traffic, data acquisition, and data delivery.

For data delivery, we have proposed the following two novel packet-forwarding schemes for vehicular network scenarios, which route messages toward fixed infrastructure nodes: 1) D-Greedy and 2) D-MinCost. Our algorithms leverage locally available information about traffic and global traffic statistics to reach forwarding strategy decisions that minimize communication and, at the same time, adhere to a delay threshold set by the application for each packet. We have conducted a thorough experimental evaluation of our schemes, utilizing realistic vehicular traces on a real city map. We have compared them with the Epidemic scheme, which achieves optimal delay and delivery ratio under our scenario, and with MinDelay, a greedy delay-minimizing scheme. It has been shown that our schemes significantly outperform the competing algorithms in terms of communication cost while maintaining a reasonably high packet delivery ratio and low delivery delay and are thus very well suited for our scenario.

We have subsequently proposed a framework that jointly optimizes the data acquisition and data delivery stages in the traffic-monitoring system. Our results have shown that the optimal allocation of freshness budget to these processes depends on the freshness budget itself and the distance of the monitored road from the closest gateway. Roads farther away from the gateway are the roads that benefit the most from our optimization. By striking an optimal balance between data acquisition and DDDs, we obtain communication savings of up to 42% compared to the basic approach. We have shown that our optimizations yield very good results in both sparse and dense vehicular traffic, with the benefits of our approach being more pronounced in dense networks. Joint optimization relies on the computation of the optimal (DDD, DAP) pairs, which can be calculated during an initial system configuration phase and used thereafter during normal operations. If an anomaly, e.g., a traffic accident, disrupts joint optimization, this condition can be detected at the AP side, and a message broadcast can switch the protocol used to *rapid delivery*. As our future work, we aim at investigating the real-time adaptive calculation of (DDD, DAP) pairs so that the system remains optimized during nonrecurring traffic events.

In this paper, we have considered a busy urban scenario, where the wireless medium is expected to be congested throughout. In our 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, MF can aggressively be utilized, reserving delay budget for resorting to the DM strategy in more congested parts of the network. We also plan to embed more sophisticated propagation models in our simulations and study how the behavior of our algorithms is effected.

This paper has proposed algorithms in the context of traffic-monitoring sensor networks; however, their applicability is not limited to traffic-monitoring systems. The routing techniques proposed could be used to deliver other types of content over a vehicular network, e.g., atmospheric pollution or noise level data. However, a careful study of these techniques must be performed within the context of each new application to investigate their suitability and tune them to meet specific application needs.

REFERENCES

[1] M. A. Perillo and W. B. Heinzelman, "Optimal sensor management under energy and reliability constraints," in *Proc. IEEE WCNC*, 2003, pp. 1621–1626.

[2] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *ACM Wireless Netw.*, vol. 11, no. 3, pp. 333–340, May 2005.

[3] K.-P. Shih, Y.-D. Chen, C.-W. Chiang, and B.-J. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in *Proc. IEEE ISCC*, Jun. 2006, pp. 923–928.

[4] J. Lu, L. Bao, and T. Suda, "Coverage-aware sensor engagement in dense sensor networks," in *Proc. EUC*, 2005, pp. 639–650.

[5] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proc. SenSys*, 2003, pp. 51–62.

[6] K. S. Kwok, B. J. Driessen, C. A. Phillips, and C. A. Tovey, "Analyzing the multiple-target–multiple-agent scenario using optimal assignment algorithms," *J. Intell. Robot. Syst.*, vol. 35, no. 1, pp. 111–122, Sep. 2002.

[7] A. Sekhar, B. S. Manoj, and C. S. R. Murthy, "Dynamic coverage maintenance algorithms for sensor networks with limited mobility," in *Proc. PERCOM*, 2005, pp. 51–60.

[8] G. Wang, G. Cao, and T. L. Porta, "A bidding protocol for deploying mobile sensors," in *Proc. ICNP*, 2003, pp. 315–324.

[9] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[10] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[11] R. Frenkiel, B. Badrinath, J. Borres, and R. Yates, "The infostations challenge: Balancing cost and ubiquity in delivering wireless data," *IEEE Pers. Commun.*, vol. 7, no. 2, pp. 66–71, Apr. 2000.

[12] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, Tech. Rep. CS-200006, Apr. 2000.

[13] E. Royer and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Pers. Commun.*, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[14] Z. Chen, H. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proc. Annu. Workshop MobiHoc*, 2001, pp. 247–250.

[15] L. Briesemeister and G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *Proc. Annu. Workshop MobiHoc*, 2000, pp. 45–50.

[16] B. Xu, A. Ouksel, and O. Wolfson, "Opportunistic resource exchange in intervehicle ad hoc networks," in *Proc. IEEE Int. Conf. MDM*, 2004, pp. 4–12.

[17] P. Costa, D. Frey, M. Migliavacca, and L. Mottola, "Towards lightweight information dissemination in intervehicular networks," in *Proc. 3rd Int. Workshop VANET*, 2006, pp. 20–29.

[18] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM Int. Symp. MobiHoc*, 2004, pp. 187–198.

[19] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE 61st VTC—Spring*, May/Jun. 2005, vol. 4, pp. 2289–2293.

[20] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A mobility-centric data dissemination algorithm for vehicular networks," in *Proc. 1st ACM Int. Workshop VANET*, 2004, pp. 47–56.

[21] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. INFOCOM*, 2006, pp. 1–12.

[22] Y. Yu, V. Prasanna, B. Krishnamachari, and V. Kumar, *Information Processing and Routing in Wireless Sensor Networks*. Singapore: World Scientific, 2006.

[23] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. San Francisco, CA: Freeman, Jan. 1979.

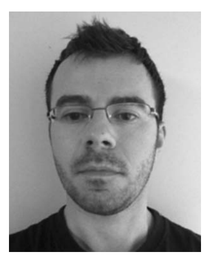
[24] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis, "Efficient computation of delay-sensitive routes from one source to all destinations," in *Proc. INFOCOM*, 2001, pp. 854–858.

[25] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," in *Wireless Commun. Mobile Comput. (WCMC)—Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Aug. 2002, vol. 2, no. 5, pp. 483–502.

[26] V. Naumov, R. Baumann, and T. Gross, "An evaluation of intervehicle ad hoc networks based on realistic vehicular traces," in *Proc. ACM Int. Symp. MobiHoc*, 2006, pp. 108–119.

[27] Open Street Map. [Online]. Available: http://www.openstreetmap.org

[28] S. A. Hosseini Tabatabaei, M. Fleury, N. N. Qadri, and M. Ghanbari, "Improving propagation modeling in urban environments for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 705–716, Sep. 2011.



Antonios Skordylis received the D.Phil. degree in computer science from the University of Oxford, Oxford, U.K., in 2010.

He is currently with Thomson Reuters, NY. His research interests include distributed data management and routing for energy-constrained devices. His recent work has focused on routing and data compression techniques for vehicular sensor networks.

Niki Trigoni received the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 2001.

For two years, she was a Postdoctoral Fellow with the Department of Computer Science, Cornell University, Ithaca, NY. Then, she was a University Lecturer with Birkbeck College, University of London, London, U.K. In 2007, she joined the University of Oxford, Oxford, U.K., where she is currently a University Lecturer with the Department of Computer Science and the Leader of the Sensor Networks Group. Her research interests include distributed algorithms for wireless ad hoc networks. She has worked on several aspects of fixed and mobile sensor networks, including medium access control, routing, in-network storage, in-network processing, and distributed localization techniques.