

# Persistent Content-based Information Dissemination in Hybrid Vehicular Networks

Ilias Leontiadis<sup>a</sup>, Paolo Costa<sup>b</sup>, Cecilia Mascolo<sup>c</sup>

<sup>a</sup>*University College London (UCL), United Kingdom*

<sup>b</sup>*Microsoft Research Cambridge, United Kingdom*

<sup>c</sup>*University of Cambridge, United Kingdom*

---

## Abstract

Content-based information dissemination has a potential number of applications in vehicular networking, including advertising, traffic and parking notifications and emergency announcements. In this paper we describe a protocol for content based information dissemination in hybrid (i.e., partially structureless) vehicular networks. The protocol allows content to “stick” to areas where vehicles need to receive it. The vehicle’s subscriptions indicate the driver’s interests about types of content and are used to filter and route information to affected vehicles. The publications, generated by other vehicles or by central servers, are first routed into the area, then continuously propagated for a specified time interval. The protocol takes advantage of both the infrastructure (i.e., wireless base stations), if this exists, and the decentralized vehicle-to-vehicle communication technologies.

To show the feasibility and assess the performance of our approach, we implemented and tested our framework using a small number of vehicles. Furthermore, we run a large scale simulation over a number of realistic vehicular traces based scenarios. Results show that our protocol achieves high message delivery while introducing low overhead, even in scenarios where no infrastructure is available.

---

## 1. Introduction

Vehicular networks are a peculiar class of mobile networks in which vehicles are equipped with radio interfaces and are therefore able to communicate with an infrastructure (if existing) or other vehicles in an opportunistic way. Content-based information dissemination enjoys wide applicability in these types of networks, ranging from traffic information and warnings, to parking availability, fuel prices, road conditions, and advertisements. While 3G networks can be

---

*Email addresses:* [i.leontiadis@cs.ucl.ac.uk](mailto:i.leontiadis@cs.ucl.ac.uk) (Ilias Leontiadis),  
[paolo.costa@microsoft.com](mailto:paolo.costa@microsoft.com) (Paolo Costa), [cecilia.mascolo@cl.cam.ac.uk](mailto:cecilia.mascolo@cl.cam.ac.uk) (Cecilia Mascolo)

used to offer these services, these come with a cost, both at network and hardware level<sup>1</sup>, and with limitations in granularity and coverage. Our approach aims at integrating infrastructure with ad hoc approaches to decrease costs and increase coverage.

Unfortunately, while several protocols geared to content dissemination for mobile networks have been developed [1, 2, 3, 4], we have seen very few approaches specifically targeting vehicular networks[5, 6, 7, 8, 9], and, to the best of our knowledge, none which allows *persistent* dissemination in hybrid or completely infrastructureless scenarios.

In this work, we address the current research gap by presenting a protocol for *persistent content based dissemination in vehicular networks*. The protocol enables applications to:

- *publish* messages to geographical locations by first sending them to the relevant areas;
- *store* the messages in the relevant area (generally roads adjacent to where the specific event is happening) using a combinations of infostations (if any) and vehicles. Replicas of messages are stored in appropriate numbers to allow delivery to subscribers and, if stored on vehicles, they are hopped from one to another to allow them to stay in specific locations (homeZones);
- *deliver* the messages to *subscribers* (i.e., interested vehicles) inside the area, when these are met by any of the replica holders.

We take advantage of the information from the navigation system (NS), available on more and more vehicles. NSs provide valuable information on the *suggested route*. This information makes the mobility patterns of the vehicles more predictable and can be used to efficiently select the best carriers to forward messages to the affected areas and vehicles.

The suggested routes can also be used to extract interests in order to automatically filter only information relevant to the driver. For example, the NS can automatically subscribe to receive traffic warnings that affect the suggested route, to receive fuel prices from nearby fuel stations when the vehicle is running out of petrol, or to receive free parking notifications concerning the vehicle's destination (*automatic subscriptions*). However, a user is also allowed to insert specific subscription interests, which are not automatically calculated, e.g., information on nearby restaurants or hotels (*custom subscriptions*). The subscriptions and the navigation system will then be used to filter incoming messages that concern the vehicle and its driver, to geographically route messages to/from infostations and to efficiently and persistently disseminate information in specific geographic areas. This enables our protocol to disseminate information mainly to subscribers without affecting non-interested vehicles.

---

<sup>1</sup>The cost of a cellular chipset is currently 5-10 times higher than Bluetooth and WiFi chips.

We evaluate the protocol both by testing a real implementation in a small vehicular testbed, and by simulation in various scenarios, using realistic traffic traces generated by various traffic simulators [10, 11], under different conditions. The results exhibit good performance in various settings in terms of overhead and message delivery, largely outperforming existing epidemic dissemination protocols [12].

## 2. Scenario

We assume a network of vehicles, equipped with navigation systems that contain information about the geographical location of local infostations (i.e., access points to the backbone) and the planned route to reach the desired destination. Each vehicle is provided with an omnidirectional antenna and is able to wirelessly communicate with neighbouring vehicles. In addition, we assume that power consumption is not critical and that storage space is virtually unlimited: this is reasonable for vehicular networks.

Information is generated by *publishers* in the network: these can be either central servers on the backbone or vehicles themselves. The publisher defines the area where the information should be disseminated (*Persistence Area*). Publications can be the result of information collected beforehand in the network or fresh information only distributed locally by vehicles directly. We assume information is collected by a centralized system: it is not the scope of this paper to discuss how it is collected, we can even assume that the very same vehicles are mobile sensors that collect information about traffic conditions, accidents, etc (e.g., like in CarTel [13]).

A vehicle, may act as a subscriber, by expressing its interests in a certain set of messages. In vehicular applications, relevant information ranges from traffic news (e.g., road works or congestion) to gas stations and hotel advertisements. Figure 1 describes one of our scenarios: a car needs to travel on a certain road, indicated with the red arrow following the main yellow road. At a certain point, road works are scheduled or an accident happens on that road (indicated in the right side of the picture): we will call this *Point of Interest* (POI) from now on. The cars heading in the directions of the POI need to be informed so that they can potentially follow another route.

As illustrated in Figure 1, our approach aims at letting information *stick* to an area (generally the area leading to the POI) for a certain time period. This is done by creating *homeZones*, which are locations where message replicas are stored for distribution to subscribers. The homeZones will be placed on the road segments where subscribers drive to reach the POI. As we will see later, the homeZones represent either the location of road-side infostations, if present, or the location where the replicas should be kept by vehicles traveling nearby.

In order to keep disseminating information, we use content based dissemination which allows us not to inform all vehicles in the area but just those interested. Content-based routing (CBR) differs from classical routing paradigms as messages (i.e., the published information) are routed based on their *content*

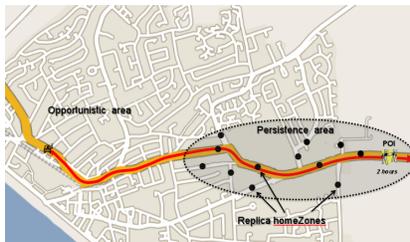


Figure 1: Black dots represent homeZones where messages should be stored to inform approaching vehicles about the road works (POI).

rather than their destination address. This form of implicit, multi-point communication fosters a high degree of decoupling, since the communicating parties are not necessarily aware of each other, and can therefore change dynamically without affecting the rest of the system.

### 3. Protocol Description

Given the heterogeneity of the scenarios we target, a widespread presence of infostations cannot be guaranteed at any time and any place. Therefore, we developed a novel approach, striking a balance between the efficiency of infrastructure and the flexibility of opportunistic communication. Indeed, as we detailed in the following, our protocol seamlessly exploits both infostations (where available) and vehicle-to-vehicle communication to *deliver* and *store* messages in the intended locations.

Hereafter, for sake of clarity, we illustrate our approach in different steps by first describing the basic version of the protocol, assuming pervasive infrastructure availability, and then we show how we can relax this assumption, incorporating opportunistic and ad hoc communication to i) extend the infostation dissemination range and ii) maintain the message persistence even in areas without infostations. Nevertheless, these three variants are not independent but co-exist in our protocol to provide a single solution addressing content-based dissemination in heterogeneous environments.

#### 3.1. Infrastructure-based Persistence

As detailed in Section 2, the aim of our protocol is to ensure that all drivers are promptly informed about relevant events, affecting their route (e.g., traffic jams or gas stations). To this end, information about these events must be stored at specific locations, called *homeZones* (indicated by black dots in Figure 1) such that all approaching vehicles can be notified.

The identification of the exact position and the number of these homeZones can be done automatically or in an application specific way and will vary according to the type of information and the road topology. For example, we can use a simple algorithm to make sure that there are replicas in every path leading to the POI from a certain distance. Or, a highway agency can strategically define

the areas where the information should be persistent: for instance, in case of the traffic jam on a highway in Figure 1, the homeZones sit on the main roads to access the highway so that vehicles can avoid entering the highway and choose alternative paths.

For each homeZone a *replica* of the original message is created and then routed and stored at the corresponding geographic location. Under the assumption of widespread infrastructure, messages can be sent to the nearby infostations to be then disseminated. A simple yet inefficient solution would be to have each infostation to periodically re-broadcast the message to all nearby vehicles. This, however, would incur a significant network overhead because i) messages are transmitted even if no subscribers are around and ii) subscribers are very likely to receive the same message multiple times by encountering several infostations along their paths.

To circumvent this issue and to remove unnecessary transmissions, we devised a two-phase scheme where each vehicle periodically advertises its planned route and its additional interests, e.g., fuel or parking slot, if any. Through this information, the infostations can derive the actual subscriptions (both automatic and custom) and compare them against the stored messages.

We assume that subscriptions and messages are matched through a **match** function. Messages are defined as a list of attributes values (for example, `<gas company="abc", fuel type="unleaded", price="1.25">`) whereas subscriptions are expressed as queries over these attributes (e.g., `<*, diesel, less than 2>`) but more sophisticated patterns including regular expressions can also be used.

If a match occurs, the message is then transmitted and received by the subscribers. To avoid duplicate receipts, the subscriber also piggybacks the IDs of the last  $\lambda$  messages received. In this way, before forwarding the message, the infostation could check whether that message has already been delivered. Only subscribers that did not previously receive the message can trigger a broadcast, which, however, can be heard by more than one subscribers in the area.

### 3.2. Opportunistic Dissemination

Even in presence of infostations, opportunistic vehicle-to-vehicle communication can greatly enhance the performance of the above protocol by enabling the dissemination of messages in a broader area at a very small additional cost. To this end, we let the vehicles (subscribers and non) which have heard a message store it and retransmit it when detecting a subscriber in the vicinity. This allows for opportunistic exploitation of vehicles which in any case have heard the information and can act as additional carriers for the information in the persistence area and beyond it when vehicles exit it. Interestingly, if no subscriber is encountered, no additional traffic is generated, thus effectively implementing an *interest-driven* routing scheme in which messages propagate only in areas populated by subscribers, possibly extending the persistence area defined by the application.

<p><b>Variables</b></p> <ul style="list-style-type: none"> <li>• self: node's own id</li> <li>• <math>\mathcal{R}</math>: node's route, expressed as an ordered list <math>\langle (I_1, t_1), (I_2, t_2), \dots, (I_r, t_r) \rangle</math> of intersection points <math>I_i</math> and (expected) arrival time <math>t_i</math></li> <li>• <math>S</math>: node's subscription set</li> <li>• <math>\mathcal{I}</math>: set of the identifiers of the last <math>\lambda</math> messages received</li> <li>• <math>\mathcal{M}</math>: the message buffer</li> </ul> <p><b>Messages</b></p> <ul style="list-style-type: none"> <li>• DATA<math>\langle ID, POI, kind, HomeZone, TTL \rangle</math>: a data message, uniquely identified by <math>ID</math>. <math>POI</math> represents the point-of-interest of the message (e.g., "junction A15") while <math>kind</math> indicates whether the message is a <i>replica</i> or a <i>notification</i>. <math>HomeZone</math>, expressed as a pair of coordinates <math>(x, y)</math>, specifies the geographic location where the replica should be delivered. Finally, <math>TTL</math> represents the expiration time. Message payload is not indicated here for simplicity.</li> <li>• CONTROL<math>\langle R, S, I, n \rangle</math>: control message disseminated periodically by each node. It contains the planned route (<math>R</math>), the subscriptions (<math>S</math>) and the identifiers of the last <math>\lambda</math> messages received (<math>I</math>) of the sender node <math>n</math>.</li> </ul> <p><b>Functions</b></p> <ul style="list-style-type: none"> <li>• <b>send</b> (<math>m, n</math>): send an unicast message <math>m</math> to neighbor <math>n</math></li> <li>• <b>broadcast</b> (<math>m</math>): broadcast message <math>m</math> to all 1-hop neighbours</li> <li>• <b>deliver</b> (<math>m</math>): deliver message <math>m</math> to the application</li> <li>• <b>matches</b>(<math>m, S, R</math>): returns TRUE if the message <math>m</math> matches a custom subscription <math>s \in S</math> or is relevant for route <math>R</math> (i.e., <math>m.POI \in R</math>)</li> <li>• <b>matches</b>(<math>c, R</math>): returns TRUE if the content <math>c</math> is relevant for the route <math>R</math> (automatic subscription)</li> <li>• <b>expired</b>(<math>m</math>): returns TRUE if the message expiration time has passed</li> <li>• <b>computeUtility</b>(<math>m, R</math>): calculate the utility for the message replica <math>m</math> given the route <math>R</math>, using formula (1)</li> <li>• <b>assignHomeZone</b>(<math>m</math>): returns the pair of coordinate <math>(x, y)</math> representing the assigned <i>homeZone</i> for the replica <math>r</math></li> </ul>
---

Figure 2: Pseudo-code definitions of the ad hoc protocol.

### 3.3. Ad-hoc Persistence

In some scenarios the existence of infostations is not sufficient to allow the dissemination to all interested vehicles. This could be for various reasons: i) the infrastructure is partially collapsed due to accidents or attacks ii) the infrastructure is not covering the whole interested area as this is vast iii) the information has a very fine granularity with respect to the infostation coverage (e.g., parking slots positions may be of interest only to be persistently maintained only in a couple of streets away from the parking area).

With respect to the approach presented in Section 3.1, if infostations are not available, we then need to find ways to i) *route* message from the publisher to the homeZones and to ii) *keep a replica* persistently close to its homeZone.

While the just described opportunistic dissemination helps to partially solve these issues in a hybrid scenario, it does not guarantee that a replica is persistently maintained in the homeZone as most of the vehicles which have heard

the message might leave. Alternative approaches based on 3G technology like UMTS are not viable as the available bandwidth per user in a crowded area would be severely limited and the price the users should pay to network companies would discourage them from adopting the service.

Therefore, a more proactive mechanism is needed to enforce persistence in a semi or totally decentralized scenario, relying on *scalable* and *inexpensive* ad hoc communication among vehicles and exploiting the route information provided by navigation systems.

While power consumption is not critical in vehicular networks, bandwidth is a critical factor which may drastically reduce the capacity of wireless communication [14] and limiting its usage it is of utmost importance to enable ad hoc communication. It is therefore unfeasible to store messages on all vehicles and let them flood the network because this would rapidly saturate all the available bandwidth. Hence, it is key to bound the number of transmissions by properly identifying which are the nodes interested in a given messages and which are the best carriers to keep the replica close to its homeZone. Ideally a carrier would be a vehicle which is always near the replica's homeZone. This is clearly unrealistic as in general vehicles continuously move from one location to another and, hence, new carriers must be selected. In particular, not only the current location of a node is important but also its future one as carriers moving *towards* a homeZone are much better than those departing *from* it.

Notably, however, while in traditional mobile networks predicting user movements is hard (if not impossible) in vehicular networks information about future movements can be derived by looking at the planned route provided by navigation systems. This is particularly beneficial as a host can detect whether one of its neighbors is a better carrier for some of the buffered messages. Note that information about future route is broadcasted even in the infrastructure-based version of our protocol to derive automatic subscription so no additional traffic is required.

We introduce the notion of *utility* for the selection of message carriers. The utility  $U_v(m)$  of a vehicle  $v$  with respect to a message  $m$  represents how good of a carrier  $v$  is to deliver  $m$  to its homeZone. This depends on how close that node will be w.r.t. the homeZone, i.e., is the nearest point (NP), and how fast the node will get to there according to its travel speed. Then, we apply the Dijkstra algorithm to estimate the minimum time occurring to reach the homeZone under the assumption that a new carrier is found.

More formally, we have:

$$U_v(m) = T_{NP} + \widehat{T}_{homeZone} \quad (1)$$

being  $T_{NP}$  the time needed to reach NP given the current route and speed of the vehicle and  $\widehat{T}_{homeZone}$  the estimated time to go from  $NP$  to the homeZone based on the Dijkstra algorithm<sup>2</sup>.

---

<sup>2</sup>Note that this calculation can be easily executed by the on-board computer by leveraging off the map and the planned route of the neighbor  $v$ .

For instance, in the example depicted in Figure 4, a message is published at the infostation  $I$  and needs to be route to the homeZone denoted by the red dot. Both the vehicle  $V_A$  and vehicle  $V_B$  are potentially eligible as carrier but the latter is preferable because its route will get closer to the final destination, i.e., the message’s homeZone. Nevertheless, while approaching  $NP_B$  the node encounters another vehicle,  $V_C$ , whose route happens to cross the homeZone and hence it takes over the message and delivers it to the destination. Note that this process never ceases because as soon as  $V_c$  will pass the homeZone, it will need to find another carrier going in the opposite direction back to the homeZone.

Thus far, we focused our attention only on carriers but the goal of the protocol is to deliver the message to the subscribers. To this end, when a node receives the planned route from its neighbors, beside checking whether there is any potentially better carrier, it also verifies whether its messages are of interest for any of its neighbors, adopting the same approach as in the infrastructure-based version (e.g. replica routing and information dissemination occur at the same time). Therefore, we distinguish between two kind of messages: *replicas* and *notifications*. The former are the messages that need to be located as close as possible to the corresponding homeZone and can never be deleted. Notifications, instead, are those messages that have been delivered to subscribers or overheard.

### 3.4. Protocol Details and Pseudocode

The pseudocode for the protocol is reported in Figure 3 along with the necessary definitions listed in Figure 2.

Periodically each node broadcasts a *CONTROL* message containing the description of its planned route to its 1-hop neighbors along with the list of its custom subscriptions as indicated in Figure 3 (*Subscription Dissemination*). Also the identifiers of the last  $\lambda$  message received are piggybacked in this message. The information contained in the message is key for the neighbours to determine message forwarding decision.

Indeed, when a *CONTROL* message  $c$  is received (see second block in Figure 3), the buffer is re-evaluated to assess whether the originator of  $c$  may be interested in some messages either as a new carrier or a subscriber. First, the expiration time of each buffered message is evaluated (lines 2-3). If the message is still valid and the node is a carrier for that message, i.e., the message is a replica, the route  $R$  contained in the control message is analyzed and the utility  $U'$  of the neighbour for the given message is compared with the one of the current node (line 6-9). If it is lower, i.e., the neighbor will travel closer (or faster) to the destination than the current node, the message is transferred to the neighbor by means of a unicast transmission and then is removed from the buffer (lines 10-12). Otherwise, the content of the message is evaluated against neighbour’s subscriptions, both automatic (i.e., depending on the route  $R$ ) and custom. If a match occurs and the message has not already been received by the node (i.e.,  $m.ID \notin I$ ) a copy of the message is sent (lines 14-17). Naturally, a number of optimizations are possible (e.g., packing multiple matching messages

```

Subscriptions Dissemination
1: create new message  $c$ : CONTROL<  $\mathcal{R}, \mathcal{S}, \mathcal{M}$ , self>
2: broadcast ( $c$ )

Invoked on receipt of a CONTROL message from neighbour  $n$ .
receive CONTROL<  $R, S, I, n$  >
1: for all  $m \in \mathcal{M}$  do
2:   if expired( $m$ ) then
3:      $\mathcal{M} \leftarrow \mathcal{M} \setminus \{m\}$ 
4:   else
5:      $sent \leftarrow \text{FALSE}$ 
6:     if  $m.kind = replica$  then
7:        $U \leftarrow \text{computeUtility}(m, \mathcal{R})$ 
8:        $U' \leftarrow \text{computeUtility}(m, R)$ 
9:       if  $U' < U$  then
10:        broadcast( $m$ )
11:         $sent \leftarrow \text{TRUE}$ 
12:         $\mathcal{M} \leftarrow \mathcal{M} \setminus \{m\}$ 
13:       if  $\neg sent$  then
14:         if matches( $m, S, R$ )  $\wedge m.ID \notin I$  then
15:           create a copy  $m'$  of message  $m$ 
16:            $m'.kind \leftarrow notification$ 
17:           send( $m'$ )  $\rightarrow n$ 

Invoked on receipt of a DATA message from neighbour  $n$ .
receive DATA<  $ID, POI, kind, HomeZone, TTL$  >
1: if matches( $m, S, \mathcal{R}$ )  $\wedge m.ID \notin \mathcal{I}$  then
2:   deliver ( $m$ )
3:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$ 

Message Publishing.
1: create a set of replicas  $\Gamma = \{m_1, m_2, \dots, m_r\}$  of the published message DATA  $m$ 
2: for all  $m_i \in \Gamma$  do
3:    $m_i.kind \leftarrow replica$ 
4:    $m_i.homeZone \leftarrow \text{assignHomeZone}(m_i)$ 
5:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{m_i\}$ 

```

Figure 3: Pseudo-code of the ad hoc protocol.

in a single transmissions) but they are not discussed here to avoid complicating the pseudocode.

Dually, when a *DATA* message is received, if it is of interest for the application, i.e., it matches either an automatic or custom subscription, it is delivered to the applicative layer (lines 1-2 of **receive DATA**). Then, regardless the outcome of this check, the message is inserted in the buffer. Indeed, if the node has been designated to be a carrier ( $m.replica = \text{TRUE}$ ), it stores the message until a better one is found. Otherwise ( $m.replica = \text{FALSE}$ ), the message is stored too but it will be delivered only to the subscribers *opportunistically* encountered along the path.

Finally, *Message Publishing* consists simply of inserting the  $r$  replica of the published message into the local buffer. The message will then be taken care and forwarded to the interested subscribers as well as “moved” to its homeZone,



Figure 4: Routing a message replica to its homeZone

possibly through a better carrier, if and when encountered, according to the routing protocol we described thus far. In other words, our protocol works based on whatever the content of the buffer is, regardless of how such content got inserted. Each replica is routed independently, i.e., whenever a better carrier is encountered only relevant replicas are removed from the local buffer and sent to the new carrier. The publisher is the only node that duplicates messages, and does so only at publish time. Therefore, at any time the network contains at most  $r$  replicas of the message.

### 3.5. Discussion

Our approach largely relies on routes available in the navigation systems. This implicitly assumes that users are cooperative and willing to insert their destination. One might argue, however, that this assumption is only partly verified in practice as users tend to avoid using navigation systems for known routes (e.g., when going to the work places). Nevertheless, we expect the drivers will have an incentive to insert their destination in the navigation system as this action will automatically subscribe them to any type of notification about it, thus being able of receiving critical information about traffic congestion, accidents, etc which are of utmost importance even (if not more) for daily routes.

Similarly, at a first glance, privacy concerns seem to hamper the adoption of our protocol since drivers may be reluctant to publicly advertise their route. While we do not feel this being an issue for most, it can be easily solved by employing specific techniques to anonymise this information. Although a throughout discussion of these aspects is out of the scope of the paper, a simple yet efficient mechanism would be to use encrypted transmissions and to frequently change the ID associated to a given vehicle<sup>3</sup>.

## 4. System Architecture

In this section we illustrate our framework. As you can see in Figure 5, the architecture is composed by a number of components that interact. We will now examine these components in detail.

<sup>3</sup>Given the high mobility, ID collisions are unlikely to happen and, hence, IDs can change often without harnessing the correct behavior of the protocol.

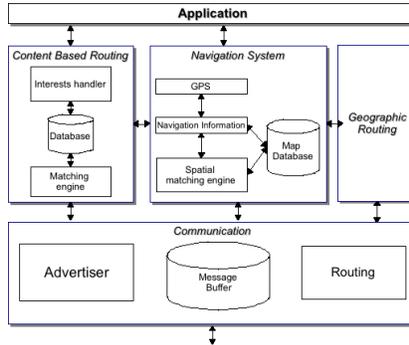


Figure 5: System architecture.

**Application:** The application invokes the appropriate primitives (i.e. `notify()`) in order to create a publication. Furthermore It can use the primitives provided by the Content-Based Routing component to express interests (i.e. `subscribe()`). Finally, this component receives a call-back whenever a matching notification is received.

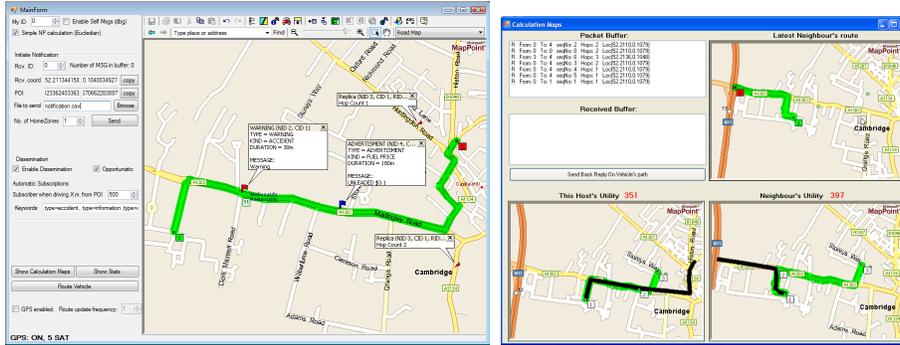
**Navigation System.** The navigation system is a core part of our architecture. First of all, it holds the navigation information of the vehicle (suggested route, estimated time, map database etc). Furthermore, it contains *spatial matching engine* to match whether a notification is relevant at a specific location. Finally, this component provides support to perform the utility value calculations to route the replicas near the homeZones.

**Geographic Routing.** The geographic routing component is invoked every time the vehicle holds a replica in the *message buffer*. When an advertisement is received by one of the neighbours this component, with the aid of the navigation system, determines whether this vehicle should keep or forward the replica.

**Content-Based Routing:** . This component supplies the application with the calls to handle the node’s interests. Additionally, it determines whether a notification is relevant by evaluating the vehicle’s route and the current interests. Here we implement the CBR matching engine with the help of the navigation’s system spatial-matching component. When a new matching notification is received it handles the message to the application component.

**Communication :** This component is responsible for disseminating the known notifications to the neighbours. When a new advertisement is received, it is invoking the CBR module to check whether any of the notifications in the message buffer are relevant. Furthermore, apart from disseminating information this component is responsible to route replicas to the homeZones. Finally, this component is responsible of advertising the vehicle’s interests, routes, and known notifications.

**Network:** . The network can be any type of wireless network that supports broadcasts to the neighbours (e.g., 802.11b). It is forwarding the received in-



(a) Navigation/Application window.

(b) Utility calculation window.

Figure 6: CarView prototype implementation.

formation to the communication component to further handle it.

## 5. Implementation

We implemented our framework and evaluated it using a small number of vehicles as well as simulations based on realistic traces. This section outlines the main features of our implementation while the next two sections report the results of our experiments.

We developed a prototype of our dissemination framework in C# 3.0 using the Microsoft MapPoint 2006 platform. As for network communication, we relied on an unmodified TCP/IP stack and therefore any 802.11a/b/g card can be used. While more efficient and tailored solutions (e.g., the upcoming 802.11p MAC layer) could be put in place, our aim was to show the feasibility of the approach, rather than providing a fully-fledged implementation. Yet, as discussed in the next section, the results we obtained on our prototype are good enough to make IEEE 802.11g with off-the-shelf network cards as a viable option even for a market release.

To advertise the route of the vehicle and its interests, periodically every node transmits a UDP packet to a predefined port. Nearby vehicles that receive the advertisement transfer the matching replicas using the same mechanism.

An application screenshot of our prototype is available in Figure 6. The main window (Figure 6(a)) displays the vehicle’s navigation system enhanced with the notifications received.

The left side window provides the interface to two basic functions: i) Send a notification about a specific POI. Together with the POI the user has to define the persistent area and the number of homeZones to create ii) Define interests. More specifically our implementation allows automatic subscriptions near the vehicles route and custom subscriptions based on keywords (content-based).

Finally, the lower buttons are used to show debugging windows which we used throughout our development and evaluation phases. In particular, the cal-

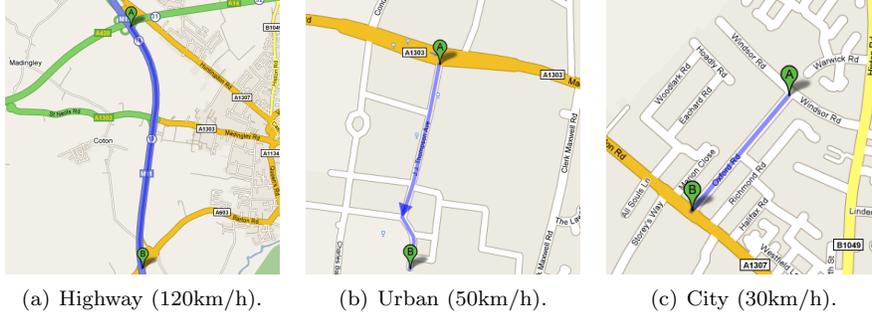
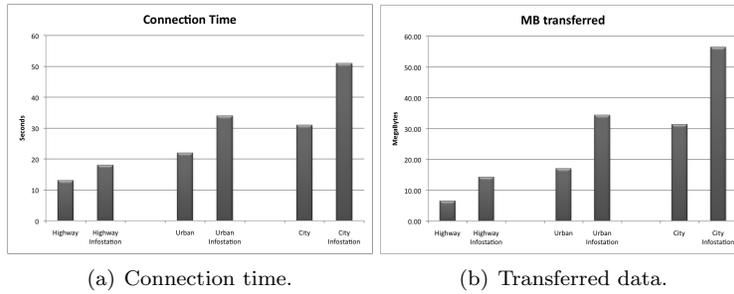


Figure 7: Testbed locations.



culcation map window (depicted in Figure 6(b)) shows a graphical representation of the *matching* process and, furthermore, the *utility calculation*.

Upon receiving an advertisement of interests, our prototype evaluates whether any message in the buffer is matching. At the same time, if the buffer contains *replicas*, the system computes the utility function and forwards the appropriate messages marking them as replicas if appropriate.

In the example reported in Figure 6(b), both the host and the neighbour are located in the position 1 while the replica should be delivered to position 3. In this case, the current host has a better utility (i.e., a lower Minimum Estimated Time of Delivery) than the neighbour because its route exhibits a larger overlap with the messages's one. This is shown in the two bottom windows where the dark line represents the route of the host (resp. the neighbour) whereas the light line indicates the ideal route of the message. Hence, the host should keep the message until a better neighbour, i.e., one with a lower utility, is encountered. Similarly, a notification will be delivered if the neighbour is travelling near the POI.

Finally our implementation places incoming matching notifications on the map of the navigation system.

## 6. Experimental Evaluation

In order to assess the actual feasibility and efficiency of our approach, we evaluated the implementation described in the previous section in different road scenarios. Unfortunately, due to the intrinsic difficulties of setting up a large scale vehicular test bed, we restrict our analysis to measure the exchange performance when two vehicles come across. The correctness and efficiency of the protocol in large-scale settings are instead thoroughly evaluated on simulation basis in the next section.

In our experiments we leveraged off two cars equipped with Dual Core Intel based laptops running windows XP with Atheros 802.11 cards and external GPS receivers. The laptops, operating in ad hoc mode, had pre-defined IPs (no DHCP) and network settings. We selected three different locations in order to represent three recurring scenarios in vehicular networks: highway, urban and city (see Figure 7). For each location we designed two different experiments: one with the two vehicles going in opposite directions and the other with a static laptop, representing an infostation, and a mobile vehicle. Each experiment was executed multiple times and median results are presented here.

This represents the worst case for our protocol as the connection between the two vehicles lasts only a handful of seconds. Conversely, in the most favorable case, the two vehicles are proceeding in the same direction and, hence, the connection can last for several minutes, thus allowing for much larger exchange.

For each of the scenarios we created a number notifications about various locations. Furthermore, we created replicas that have to be routed to one pre-selected homeZone. We pre-selected these locations so that when the vehicles meet a large number of messages should be evaluated and forwarded (i.e. there are always enough data to forward).

### 6.1. Highway

Our first experiment was carried out on a main highway with the two vehicles traveling at speeds up to 120km/h which yielded an average relative speed of 229 km/h and an average connection time of 13.1s in the vehicle-to-vehicle (V2V) experiment and 18s in the vehicle-to-infostation (V2I) one (see Figure 8(a)). This represents a very challenging scenario.

Nonetheless, as reported in Figure 8(b), the amount of data exchange is still significant. We could transfer 6.5MB per pass in the V2V experiment and 14MB in the V2I configuration.

Note that these results also account for the time required to perform the initial content matching and evaluate the utility functions. The setup time (the time required between receiving the first advertisement and starting to transmit the first packet) was 0.83s. Apart from the setup time, no further delay due to processing was noticeable because these calculations were performed in parallel to the transmission.

### 6.2. Urban

For the urban scenario we selected an area with very few buildings that are far from the road, thus minimizing the impact of interference and fading effects. The average relative speed between vehicles in the V2V experiment was 85 km/h while 46 km/h was the average speed in the V2I experiment. Connection times are significantly higher than the highway scenario reaching up to 31s for the V2I. This favorably increases the volumes of data exchanged, allowing to transfer 17MB between vehicles and 34MB to the infostation.

### 6.3. City

Finally, we evaluated our approach in a busy and densely populated road with two storey terraced houses on each side of the road. The average relative speed during our experiments was 51km/h between the two vehicles and 26km/h with the infostation resulting in connection times of 31 and 51 seconds respectively. The very low speeds and the reflections from the houses led to impressive volume of transferred data. Vehicles exchanged 31MB and managed to transfer to the infostation 56MB.

## 7. Simulation evaluation

To prove the validity of our approach and evaluate its performance in large scale settings, we report on the protocol performance over several simulated scenarios, generated from realistic vehicular traces. We analyzed our protocol under a synthetic load of both automatic and custom subscriptions. In particular, for automatic subscriptions, all vehicles with planned route intersecting the POI are considered subscribers. This is the typical situation with traffic warnings, which are of interest to any vehicle enroute towards the affected destination. Conversely, custom subscriptions (e.g., hotel or restaurants) are not relevant for everybody but will involve only a fraction of vehicles travelling towards the POI.

To put our work in the context of related efforts and to capture the tradeoffs involved, we compared our solution with an epidemic approach, reminiscent of [12], in which all nodes store each message received and re-broadcast it to all neighbours, which have not heard that message yet.

### 7.1. Simulation Settings

As simulation platform, we used *OMNet++* [15], an open-source discrete event simulator and the *mobility framework plug-in* [16]. We use 802.11b wireless radio interface (max range is 250m). In the default configuration we have 450 vehicles, advertise interval equal to 10 s and 10 replicas. Each simulation lasts for 2 hours of simulated time and results are averaged over multiple runs.

To accurately assess the performance of our protocol in the context of vehicular networking, we exploited traffic traces generated by a multi-agent microscopic traffic simulator at ETH, Zurich [10]. These traces contain mobility patterns of 260,000 vehicles over real road maps in the canton of Zurich within

a period of 24 hours. However, we extracted smaller areas (50x50 km) to decrease the duration of the simulation (by implicitly reducing the number of vehicles involved). Additionally, we used the GMSF generator [11] to produce GIS traffic-light traces for the rural, urban and city scenarios, which have finer granularity (3x3km).

The characteristics of the scenarios considered are:

- *City scenario*: High vehicle and street density scenario where up to 880 vehicles are concurrently present (default 700). We place the POI on an intersection of two secondary roads: subscribers can be up to 40 of the 700 vehicles (automatic subscriptions). Average speed near POI is 20km/h and maximum is 60km/h. An example of this is provided in Figure 12(a) where POI and replicas are also shown.
- *Urban scenario*: Medium street and vehicle density. 420 Vehicles are present at the same time and up to 30 of them can be subscribers. Maximum speed is 60km/h but average speed is 25km/h (Figure 12(e)).
- *Rural scenario*: This is a low density scenario where only 100 vehicles concurrently present. In the scenario of Figure 12(i). Subscribers can be up to 40 of the 100 vehicles (automatic subscriptions). Average speed is 28km/h (max is 60km/h).
- *Highway scenario*: This is a much larger 50x50km area as illustrated in Figure 12(m). The simulation includes a maximum of 830 concurrent vehicles and up to 350 can be subscribers (most of the vehicles will drive near the POI). The average speed is higher than the previous scenarios (93km/h) and the highest 120km/h.

## 7.2. Simulation Results

Hereafter, we will first present results achieved in the city-base scenario, with and without infostations, as this represents the more challenging case for our protocol, given the complex road topology. Then, we will show the performance obtained in the urban, rural, and highway scenarios to demonstrate the suitability of our approach to different environments.

In all our experiments, we measured the *delivery ratio*, expressed as the fraction of subscriber that successfully received the messages; and the *network overhead*, defined as the number of transmissions received per minute by each vehicle.

**Infostations.** As a first experiment, we focus on a fully infrastructure-based scenario in which the persistence area is instrumented with several infostations. Our goal is twofold: on one hand we want to demonstrate the correctness of our protocol and on the other hand we want to assess the impact of the additional opportunistic dissemination in such scenario. To this end, in Figure 8(c) we measured the delivery of our protocol under two different configurations, i.e., with and without opportunistic dissemination.

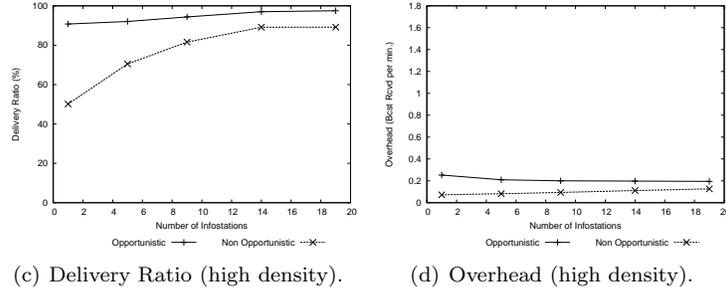


Figure 8: Number of Infostations.

Remarkably, through the opportunistic dissemination introduced in Section 3.2, delivery is above the 90% even with just one infostation. On the other hand, if opportunistic dissemination is not used, at least 14 infostations are needed to achieve similar performance. This is a prominent result as it proves that even in a fully infrastructured environment, opportunistic dissemination represents an asset to our approach. Indeed, although the network overhead does not change with the number of infostations (see Figure 8(d)), still resorting to opportunistic dissemination enables the reduction of the number of infostations, thus simplifying their deployment.

**Ad-hoc.** Nonetheless, despite the above results, assuming a widespread availability of infostations is unrealistic in many scenarios. Hence, to ensure efficient content-based dissemination in hybrid scenarios, as those targetted in this paper, it is fundamental to support infrastructure-less communication. In our work, this is achieved by means of the *ad hoc* persistence solution, described in Section 3.3. To avoid any bias and to isolate the contribution, in the rest of this section we will assume that no infostation is present and that all communication relies on vehicle-to-vehicle technology. Clearly, in case of semi-infrastructure environments, we can have an interplay of the two approaches.

**Number of Replicas.** The first parameter we explore is the number of replicas created to guarantee the persistence of the message within the specified area. Also, as we did in the infostation scenario, to assess the impact of the opportunistic dissemination, we run two different versions of our protocol: the former relying only on replicas to disseminate messages and the latter exploiting also the opportunistic routing. Since results strongly depend on the density of vehicles, we tested it both in a low and high density scenario (200 and 700 vehicles).

Results in Figure 9(a) confirm our claims. When the density is high, even a small number of replicas is sufficient to achieve a high delivery. Interestingly, however, this remarkable result is due to the combination of two different strategies: the *ad hoc* persistence and the opportunistic dissemination. Indeed, when the opportunistic dissemination is not used, the delivery drops to 50%, unless many more replicas are introduced. This however, as shown in Figure and 9(b),

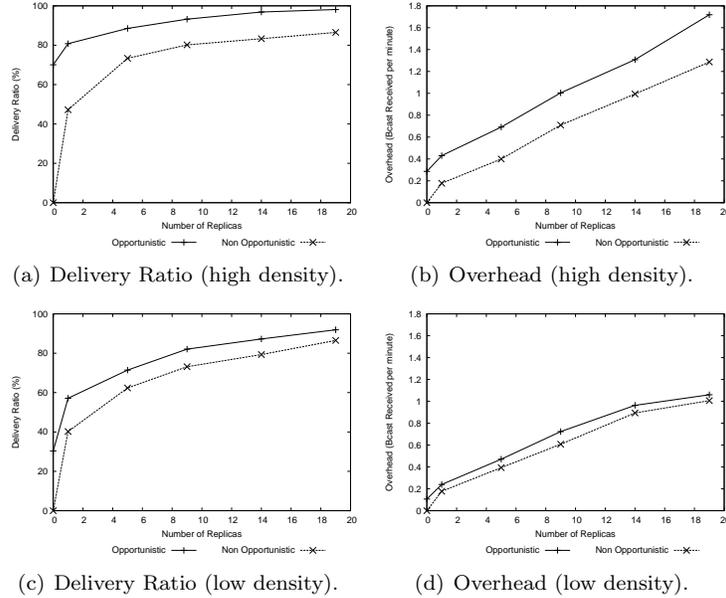


Figure 9: Number of Replicas.

generates a significant overhead. Indeed, to achieve the same delivery of 80%, 9 replicas are needed without opportunistic dissemination (instead of just 1) with almost doubled overhead (0.7 against 0.4 broadcasts per minute). Notably, the opportunistic dissemination only slightly affects the overhead because the most of it is due to keep replicas in the persistence area. Furthermore, if opportunistic dissemination is not used, even a high number of replicas does not bring significant improvements to the delivery.

In case of low density, as expected, the overall improvement provided by the opportunistic dissemination decreases as there are fewer vehicles around. Hence, the main transmissions will occur from replica carriers and this explains why the delivery is mainly impacted by the number of replicas. Nevertheless, the opportunistic dissemination is still useful because it yields an improvement of about 10% in terms of delivery regardless how many replicas are used.

Looking at these results, one might argue that the main contribution to the message delivery comes from the opportunistic dissemination while the ad hoc persistence plays only a marginal role. This, however, is strongly contradicted by performance achieved with zero replicas, both in the high density and, especially, in the low density scenario. Indeed, in the former, opportunistic dissemination alone delivers the message only to the 70% of subscribers while in the low density scenarios only the 30% of subscribers are notified. This is consistent with the conclusions drawn above: opportunistic dissemination provides a valuable contribution only in dense scenarios while in sparse scenarios it becomes less useful. Nevertheless, even in dense networks, to get reasonable results, it must

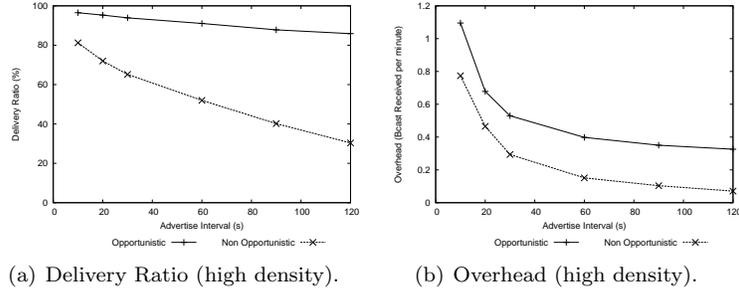


Figure 10: Advertise Interval.

be coupled with persistence strategy since, otherwise, if the message disappears from the area, by no means later subscribers can be notified.

The results in the high density scenario (Figure 9(a)) closely resemble the ones with infostations in Figure 8(c). Not surprisingly, however, overall performance is slightly worse: This behavior stems from the fact that now replicas are hosted on vehicles, as opposed to infostations. Hence, even non-subscribers play a key role to ensure proper persistence, by continuously passing replicas from one vehicle to another. Delivery of subscribers is also affected because in some cases, replicas may abandon the homeZones (e.g., because no alternative carriers were found). Consequently, incoming subscribers may miss the notification, thus demanding for more replicas to be in place.

**Advertise Interval.** Advertise interval is a complementary parameter w.r.t. the number of replicas. If we keep the number of replicas fixed, we can reduce the advertise interval to improve the message delivery. In this way, the probability for a subscriber to miss a replica is lower because subscribers advertise their interests more frequently.

This property is charted in Figure 10 in which we studied the protocol behavior over different advertise intervals. As described above, decreasing the advertise interval is beneficial to the delivery which increases to almost 100% (here we used 10 replicas). Interestingly, the improvement in terms of delivery is more evident when opportunistic dissemination is not used: without opportunistic dissemination, missing a replica is far more critical because the chances to encounter another one are few. Conversely, opportunistic dissemination alleviates this issue since messages can be obtained also from other vehicles and not exclusively from replica carries.

Note, however, that reducing the advertise interval comes at a cost. Beside incrementing the advertisements per minute, it increments the overall number of broadcasts received. Indeed, given that information about nearby vehicles is more accurate, replicas will hop more frequently from one vehicle to another because better carriers are found. This explains why the number of broadcast exhibits a steep trend as soon as the advertise interval gets small.

**Custom Subscriptions.** Thus far, we concentrated our attention only on

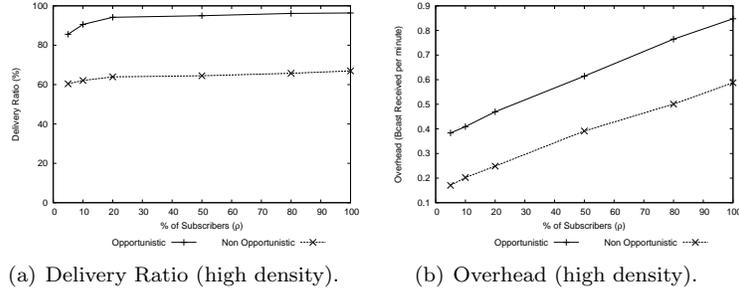


Figure 11: Custom Subscriptions.

automatic subscriptions. Nevertheless, a prominent feature of our approach is the ability to incorporate also driver’s interests, which are not necessarily shared by all other drivers. To model this scenario, we assume that only a fraction  $\rho$  of vehicles going towards the POI are actually interested in the message and we analyze our protocol under different values of  $\rho$  (see Figure 11).

Remarkably, as reported in Figure 11(a) our protocol shows high event delivery even for small values of  $\rho$ . This means that regardless of the fraction of subscribers, our protocol ensures that the vast majority (e.g., 90% for  $\rho=10\%$ ) of them receives the message. Furthermore, we also observe that when there are more subscribers, the message overhead increases. This verifies that low interest messages are spread less than more popular ones (i.e., the spread/overhead depends on the interest about an event).

These charts demonstrate the high flexibility of our protocol, which is able to tune to network conditions and to selectively contact almost only intended subscribers.

**Distribution.** In all previous charts, we focused our analysis on the city scenario, since this represented the most challenging test for us. Nevertheless, to carefully evaluate our protocol, we experimented also with other traces, available at [11], representative, of an urban, a highway, and a rural scenario and compare it with results obtained in the city scenario.

We first plot the distribution of informed vehicles to get a visual intuition of the performance of our protocol in the three scenarios, as depicted in Figure 12. Looking at the Figure 12(a), 12(e), 12(i), and 12(m), the different topologies of the four scenarios emerge. In the city scenario, much more roads and potential routes are present while in the latter three, the topology is simpler. Figure 12(b), 12(f), 12(j), and 12(n) depict the distribution of subscribers across the whole simulation area. Note that these include all nodes travelling towards the POI depicted in the left most charts. Because of the more complex topology, in the city scenario, only nodes close to the POI are actually subscribers while in the other scenarios, since there are fewer roads, all nodes travelling on the main road are subscribers, i.e., all nodes are going towards the POI.

Regardless the underlying topology, the main contribution from the deliv-

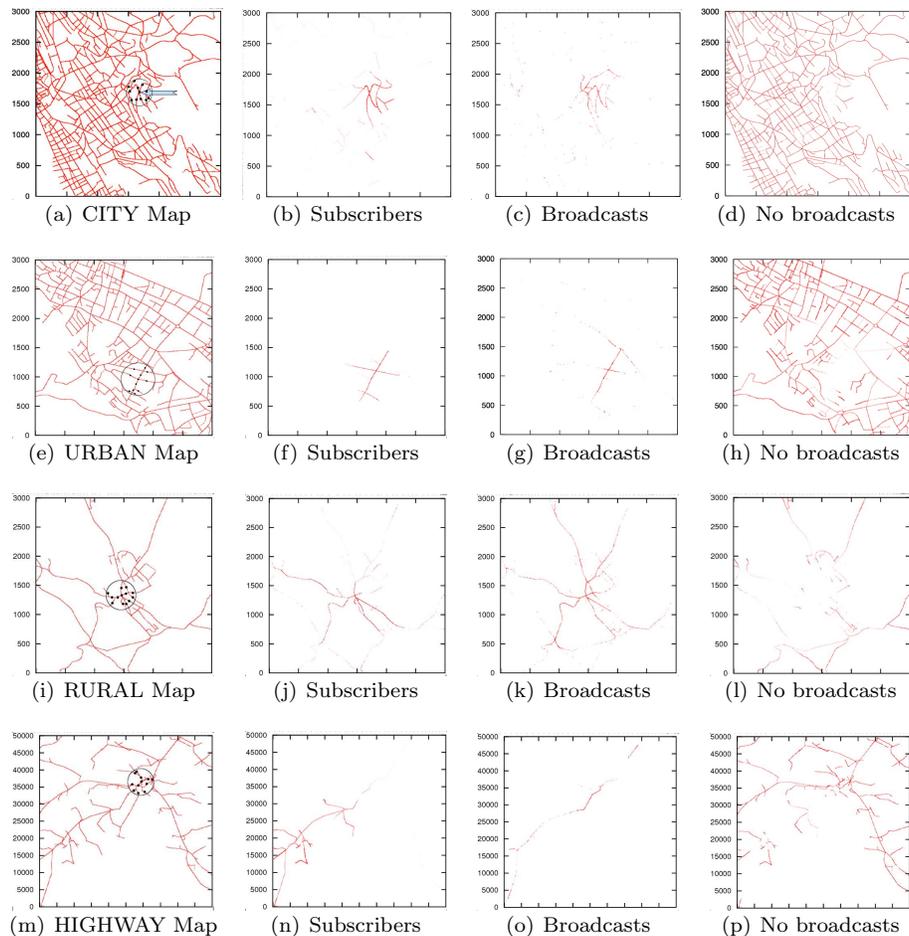


Figure 12: City (a-d). urban (e-h), rural (i-l), and highway (m-p) scenarios. First column illustrates the Map, POI, replicas (black dots), and persistence zone (circle). Second contains road segments with high percentage of subscribers. Third depicts the broadcast distribution while the fourth demonstrate road segments with no broadcasts.

ery, as already outlined, comes from the replicas in the persistence area. Indeed, the distribution of broadcasts (see Figure 12(c), 12(g), 12(k), and 12(o)) is higher in the persistence area than in the rest of the chart, as plot in Figure 12(d), 12(h), 12(l), and 12(p).

Note that message propagation extends also beyond the persistence area but almost only subscribers are reached by the message. This behavior is due to the opportunistic dissemination which keeps on informing new subscribers, exploiting vehicles which overheard the message in the persistence area. In this way, subscribers are informed, at virtually no cost, much earlier than the time they would enter the persistence area, thus enabling them to take the proper

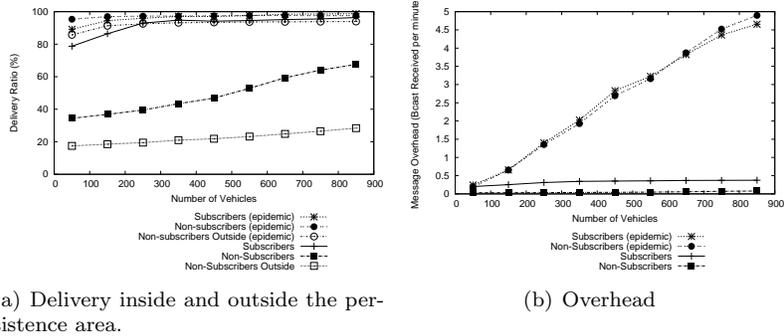


Figure 13: Density of vehicles (City)

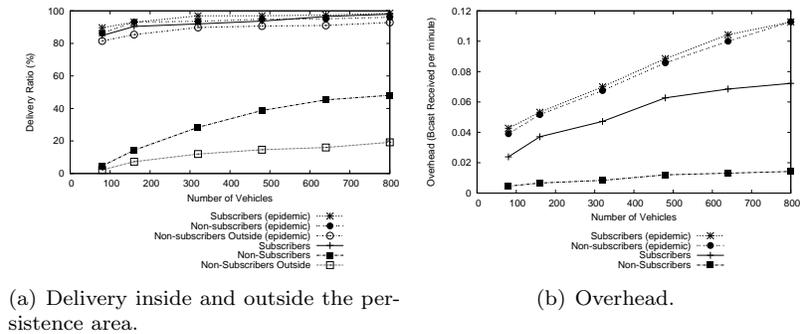


Figure 14: Density of vehicles (Highway)

actions, e.g., in case of a traffic congestions or emergency, in advance. This is even more evident in the highway and rural scenario because, given the scarcity of roads, subscribers leaving the persistence area are much likely to travel on the same road, but in the opposite direction, of subscriber going towards that area, thus increasing the probability of opportunistically exchange messages.

Finally, if vehicle density is low, e.g., in the rural scenario, replicas can leave the persistence area because the current carrier might not find any suitable vehicle to forward the replica and, hence, the replica is kept until a better carrier is encountered. This explains why in Figure 12(k) we have some broadcasts also in areas where there are no subscribers.

**Density of Vehicles.** To get further insights on the efficiency of our protocol, we compared it against an epidemic version, inspired to [12]. In this protocol, all nodes gossip to all neighbours which have not previously received the message. In this way, the epidemic infection is kept alive and eventually all vehicles gets informed. This protocol can be seen as an extension of our opportunistic dissemination in which all vehicles, not just subscribers and vehicles which overheard it, receive the message. We already shown in Figure 9 and 10 that opportunistic dissemination is not sufficient unless coupled with persistence

(either infrastructure-based or ad hoc). Here we make a further step in this direction and show that epidemics provide good performance in terms of delivery but the overhead is order of magnitude higher than ours. This is observable in Figure 13(a) and 13(b): although delivery is quite high, the overhead increases enormously. Furthermore, while the overhead of our protocol increases sub-linearly density of vehicles, the epidemic overhead increases linearly. The difference is due to the selectivity of our protocol which delivers message only to proper subscribers and hence it is less impacted by the density of vehicles. On the other hand, the epidemic protocol infects all vehicles, not just subscribers, as illustrated by the much higher delivery ratio of non-subscribers. This becomes even more critical if we extend our analysis of non-subscribers outside the persistence area. Indeed, while, outside the persistence area, our protocol affects around the 20% of vehicles non-subscribers, the epidemic protocol has to contact *all* vehicles, which is unacceptable in real situations.

The same trends are observed in the highway scenario in Figure 14(a), although here most nodes are subscribers and hence the fraction of non-subscribers informed is much lower with our approach. The overhead in Figure 14(b) follows a behaviour akin to the one observed in the city scenario, although the absolute values are lower. On the highway the set of neighbors changes less frequently and, hence, broadcasts are less triggered. Similar tradeoffs also emerged in the urban and rural scenarios (not shown for space reasons).

These results fully confirm that our protocol deals effectively with the characteristics of hybrid vehicular networks, ensuring high event delivery with reasonable overhead in a heterogeneous set of realistic scenarios.

## 8. Related work

To the best of our knowledge, Abiding Geocast [3] is the most related work in terms of delivering time-stable message in a geographical area. In order to disseminate the message in the area it employs periodic flooding or epidemic dissemination. With respect to this work, our protocol considers subscriptions, topics and navigation systems in order to filter and route the message to the interested vehicles (it is a content-based dissemination protocol and not a Geocast protocol). In addition, through the use of ad hoc persistence, we drastically reduce the overhead as opposed to epidemic approaches.

LPS [1], L-ToPSS [2], and STEAM [4] are location-based publish-subscribe systems in which location is defined as a range from the publisher or subscribers. Conversely, in our approach we have detached the persistence area from the subscriber or publisher location, thus improving the flexibility. Also, none of these approaches enable persistent dissemination. They provides delivery only to nodes which are inside a destination region exactly at publishing time while instead we allow messages to persistently remain in a given area.

Chen et al [17] proposed two policies to deliver notifications to subscribers in specific areas. In their first method, they used a server to monitor the location of the subscribers. When subscribers enter the areas defined by the publisher, the notification is routed to them. In the second method, the server sends

the publication zone to the subscriber and when the latter enters the zone, it contacts the server to receive the notification. Despite placing emphasis on location, none of these techniques were designed to work in a fully distributed and highly mobile environment.

In the field of vehicular networks, some works exploited contextual information to steer message propagation. In [18] a system for dissemination of information in an area is presented: the framework, however, does not take advantage of topological constraints of the road map to drive the dissemination and only concentrates on notifying a percentage of nodes in a circular area. Works targeting multicast communication in vehicular networks recently appeared in literature [9, 19, 20, 21, 5]. They used different versions of scoped epidemic protocols to constrain the propagation of a message within the given area specified by the publisher. Other works [6, 8, 7], instead, define a notion of relevance to enable the routing layer to self-identify the areas in which the messages should be delivered. In contrast to these approaches, our work offers a much richer semantics in which publishers and subscribers are completely decoupled as the former define the persistent area while the latter express their interests and these are matched against their route to filter out unnecessary information.

Nonetheless, beside the aforementioned differences, the most prominent novelty of our approach is that none of the above approaches targeting vehicular networks support persistence. This is a major flaw because the vast majority of information disseminated (e.g. warnings, gas station advertisements) are usually meant to last several hours. Hence, the only way to ensure this in the cited works is either to periodically re-broadcast the message (thus largely degrading the overall performance of the system) or exploiting an epidemic-like approach in the targeted areas, which, however, as we discussed in Section 7, would introduce unacceptable overhead.

## 9. Conclusions

We have presented a protocol for persistent content dissemination in hybrid vehicular networks. Messages specify a point of interest and a persistence area (i.e., a set of roads) in which the information needs to remain in order to notify subscribers. Subscribers are vehicles interested in the information. Our mechanism exploits the navigation system information to generate and match subscriptions, to route and store the message in the relevant area. Furthermore, we designed a simple framework to accommodate such kind of communication and we implemented it. Our implementation is using Microsoft MapPoint as a navigation system, GPS, and 802.11 wireless network.

Firstly we evaluated our approach using a small number of real vehicles. Secondly, we run an extensive large-scale simulation using realistic traffic traces. The results show good performance in various settings in terms of overhead and message delivery, also with respect to epidemic dissemination.

**Acknowledgments:** We would like to acknowledge the support of the EPSRC through the Project CREAM.

## References

- [1] P. T. Eugster, B. Garbinato, A. Holzer, Location-based publish/subscribe, in: In Proc. of 4<sup>th</sup> Symp. on Network Computing and Applications, 2005.
- [2] I. Burcea, H.-A. Jacobsen, L-ToPSS - Push-Oriented Location-Based Services, in: TES, 2003.
- [3] C. Maihofer, T. Leinmuller, E. Schoch, Abiding geocast: time-stable geocast for ad hoc networks, in: Proc. of VANET '05, 2005.
- [4] R. Meier, V. Cahill, STEAM: Event-Based Middleware for Wireless Ad-Hoc Networks, in: Proc. of DEBS '02, 2002.  
URL [citeseer.ist.psu.edu/meier02steam.html](http://citeseer.ist.psu.edu/meier02steam.html)
- [5] S. Eichler, C. Schroth, T. Kosch, M. Strassberger, Strategies for context-adaptive message dissemination in vehicular ad hoc networks, in: Proc. of V2VCOM'06, 2006.
- [6] C. Adler, R. Eigner, C. Schroth, M. Strassberger, Context-Adaptive Information Dissemination in VANETs Maximizing the Global Benefit, in: CSN, 2006.
- [7] M. Caliskan, D. Graupner, M. Mauve, Decentralized discovery of free parking places, in: Proc. of VANET'06, 2006.
- [8] T. Kosch, C. Schwingenschlgl, L. Ai, Information Dissemination in Multihop Inter-Vehicle Networks - Adapting the Ad-hoc On-demand Distance Vector Routing Protocol (AODV), in: The 5<sup>th</sup> Int. Conf. on Intelligent Transportation Systems, 2002.
- [9] D. Sormani, G. Turconi, P. Costa, D. Frey, M. Migliavacca, L. Mottola, Towards Lightweight Information Dissemination in InterVehicular Networks, in: Proc. of VANET 2006, 2006.
- [10] ETH Vehicular Traces, <http://lst.inf.ethz.ch/ad-hoc/car-traces>.
- [11] GMSF generator, <http://gmsf.hypert.net>.
- [12] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad Hoc Networks, Tech. rep., Duke University (2000).
- [13] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, S. Madden, The CarTel mobile sensor computing system, in: SenSys '06, 2006. doi:<http://doi.acm.org/10.1145/1182807.1182866>.
- [14] P. Gupta, P. R. Kumar, The capacity of wireless networks, IEEE Trans. on Information Theory.
- [15] OMNET++ Website, <http://www.omnetpp.org/>.
- [16] M. F. for OMNeT++, <http://mobility-fw.sourceforge.net/>.
- [17] X. Chen, Y. Chen, F. Rao, An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services, in: DEBS '03, ACM Press, New York, NY, USA, 2003, pp. 1-6. doi:<http://doi.acm.org/10.1145/966618.966625>.
- [18] I. Leontiadis, C. Mascolo, Opportunistic Spatio-Temporal Dissemination System for Vehicular Networks, in: Proc. of MobiOpp'07, 2007.
- [19] G. Korkmaz, E. Ekici, F. Ozguner, U. Ozguner, Urban multi-hop broadcast protocol for inter-vehicle communication systems, in: Proc. of VANET'04, 2004. doi:<http://doi.acm.org/10.1145/1023875.1023887>.
- [20] B. Xu, A. Ouksel, O. Wolfson, Opportunistic resource exchange in inter-vehicle ad-hoc networks, in: Proceedings of the IEEE International Conference on Mobile Data Management (MDM 2004), 2004.
- [21] S. Dornbush, A. Joshi, StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's, in: Proceedings of the 65<sup>th</sup> Vehicular Technology Conference, Dublin, Ireland, 2007.