Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication

Marios D. Dikaiakos, Andreas Florides, Tamer Nadeem, and Liviu Iftode

Abstract— **Recent advances in wireless inter-vehicle communication systems enable the establishment of Vehicular Ad-hoc Networks (VANET) and create significant opportunities for the deployment of a wide variety of applications and services to vehicles.**

In this work, we investigate the problem of developing services that can provide car drivers with time-sensitive information about traffic conditions and roadside facilities. We introduce the Vehicular Information Transfer Protocol (VITP), a location-aware, application-layer, communication protocol designed to support a distributed service infrastructure over Vehicular Ad-hoc Networks. We describe the key design concepts of the VITP protocol and infrastructure. We provide an extensive simulation study of VITP performance on large-scale vehicular networks under realistic highway and city traffic conditions. Our results demonstrate the viability and effectiveness of VITP in providing location-aware services over VANETs.

Index Terms— **Protocols, services, road vehicles, simulation, vehicular ad-hoc networks, location-aware services.**

I. INTRODUCTION

IN THE last couple of years, Inter-Vehicle Communication (IVC) has emerged as a promising field of research, where advances in Wireless and Mobile Ad-Hoc Networks can be applied to real-life problems and lead to a great market potential [1]. Already, several major automobile manufacturers and research centers are investigating the development of IVC protocols and systems, and the use of inter-vehicle communication for the establishment of *Vehicular Ad-Hoc NETworks (VANETs)* [2], [3], [4], [5], [6]. Despite the similarities that VANETs share with general, mobile ad-hoc networks, such as short-radio transmission range, low bandwidth, omni-directional broadcast (at most times) and low storage capacity, several new challenges arise because of the unique characteristics of the vehicular context: (i) rapid changes in link topology because of the relative fast movement of vehicles; (ii) frequent network disconnections, especially in the case of low vehicle density, where the gap between two vehicles might be several miles; (iii) data compression/aggregation required to accommodate for the limited bandwidth of the wireless medium; (iv) the feasibility of partially predicting vehicular position, since vehicles normally run along pre-built roads, which remain unchanged over the years; (v) the fact that energy is not an important issue since the vehicle itself can be used as a source of electric power. Last, but not least, an important challenge is the exploitation of vehicular ad-hoc networks for the provision of higher-level services to vehicles and drivers.

In this paper, we focus on the problem of providing *location-aware* services to moving vehicles by taking advantage of short-range, inter-vehicle wireless communication and vehicular ad-hoc networks. We concentrate on services that distribute on-demand information describing road conditions and available facilities in some geographic area; in particular: *traffic conditions* (e.g., congestion, traffic flow), *traffic alerts* that result from on-road emergencies (e.g., a traffic accident or a broken vehicle obstructing traffic on a road), and *roadside service directories* (e.g., location and price-lists of gas stations, location and menus of restaurants). We are particularly interested in providing traffic-related, time-sensitive information at a medium space and time boundaries. This information can serve drivers who are interested in local and short-term adjustments of their routes or in time-sensitive, roadside service-related information. Moreover, this information can be fused with GPS navigation information, extending the functionality of state-of-the-art, on-board navigation systems.

For the deployment and provision of vehicular services, we propose the development and deployment of an ad-hoc service infrastructure on top of emerging vehicular ad-hoc networks. This infrastructure is based on the *Vehicular Information Transfer Protocol (VITP)*, an application-layer communication protocol we introduced in [7]. VITP specifies the syntax and the semantics of messages exchanged between the software components of our proposed service infrastructure, which we call *VITP peers*. A VITP peer runs on the computing device of a vehicle, uses its IVC capabilities, and accesses the vehicle's sensors to retrieve useful information. VITP peers establish on-demand dynamic, ad-hoc groups, which collect, communicate, and combine information from the on-board sensors of different vehicles in order to resolve incoming requests.

The remaining of this paper is organized as follows. In Section II, we describe a motivating scenario for providing location-aware services using car-to-car communication, introduce the VITP service model, and present the key design concepts of VITP. In Section III, we give a short overview of the VITP message specification. Section IV presents the evaluation of our approach with a simulation study that uses two different vehicular traffic generators for investigating key

---

Manuscript received February 23, 2007; revised June 12, 2007.

Marios D. Dikaiakos and Andreas Florides are with the Department of Computer Science, University of Cyprus, P.O. Box 20537, 1678 Nicosia, Cyprus (e-mail: mdd@cs.ucy.ac.cy, florides@cs.ucy.ac.cy).

Tamer Nadeem is with Siemens Corporate Research, Inc., 755 College Road East, Princeton, NJ 08540, USA (e-mail: tamer.nadeem@siemens.com).

Liviu Iftode is with the Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854-8019, USA (e-mail: iftode@cs.rutgers.edu).

Digital Object Identifier 10.1109/JSAC.2007.071008.

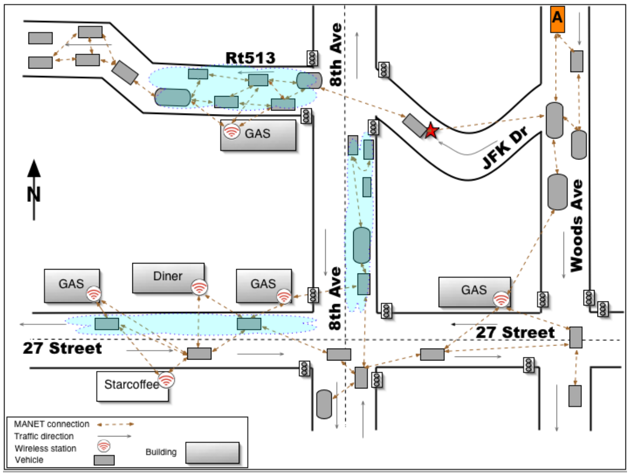


Fig. 1. A vehicular service provision scenario.

performance metrics of VITP. Section V presents an overview of related work. We conclude in Section VI.

II. A Service Model for VANETs

A. System Context

To describe the infrastructure required for providing vehicular services and to present the VITP design, we introduce a simple motivating scenario taking place in the city setting of Figure 1. This figure represents the plan of a small city-district, which is traversed by five streets; the direction of traffic is depicted with arrows placed near the street names. A snapshot of traffic conditions is superimposed on the plan. This snapshot depicts a number of vehicles (shown as grey boxes) of various sizes located on the district streets.

We assume that most vehicles are equipped with an embedded computer with a display interface, a GPS receiver, a wireless network interface for inter-vehicle communication (compliant to standards like 802.11x or DSRC) [8], and an on-board diagnostics (OBD) interface. Some vehicles may have alternative wireless network connectivity support based on an on-board cellular communication device. The OBD can be used to acquire a small set of data values from mechanical and electronic sensors mounted on the vehicle. We assume that all subsystems (GPS, OBD, wireless networking) are connected and provide data to the embedded computer. We also assume that a navigation software system is installed on the computer and enables the association of the vehicle's geographic position to an internal data-structure representing the road networks of a large geographic area around the vehicle. Such a data structure can be easily constructed from publicly available geographic referencing systems [9], [10].

Vehicles establish a vehicular ad-hoc network infrastructure through their wireless connections; Figure 1 depicts VANET connections as dashed, double-headed arrows connecting vehicles. A number of roadside service facilities (gas stations, coffee shops, restaurants) are also equipped with short-range wireless interfaces and participate in the VANET infrastructure. Vehicles and roadside stations use this infrastructure to exchange messages. Multi-hop message delivery can be supported by geographic routing protocols, which push messages toward their geographic destination [11], [12], [13], [14], [15]. In the absence of a VANET infrastructure, messages can be transported to their destination area through alternative wireless/cellular or wireline networks (Internet), and then passed onto moving vehicles via roadside wireless gateway stations.

B. Motivation and Problem Statement

The types of services we wish to support are illustrated with the following example: We assume that vehicle A, located at the top right of Figure 1 and moving southward on Woods Ave, is heading to a destination on the West side of the city. The driver wants to go either through Rt513 or through 27th Str. She is also interested in getting gas along the way, but is not willing to pay over 1.8 dollars per gallon for gasoline. The driver asks the on-board navigation system for the traffic conditions on alternative routes that lead to Rt513 West and 27th Str West, and for the location of drive-in coffee shops and gas stations along those routes. Notably, a possible way to Rt513 goes through JFK Dr, which is only a few meters down the road from the present location of A. Therefore, the service infrastructure should try to come up with a reply to the driver's requests, before the driver decides whether or not to take the JFK Dr exit. The interaction between the driver and the on-board navigation system can be performed either with a voice or with a simple touch-screen interface.

The information requested by the driver of vehicle A can be *computed* out of data available on vehicles and roadside facilities located in the road segments specified by A's inquiries. For instance, the traffic-flow on the segment of Rt513 shown in Figure 1 can be derived by estimating the average speed of vehicles moving on that segment for a short period of time; a congestion in that road segment can be established from a consistently low average speed and/or a high density of vehicles on that road. Similarly, the operation of a gas station on 27th Str can be deduced from information dispatched by the gas-station's wireless access point, which specifies the type of service offered (selling gas), the business address, and gas prices.

To retrieve such information, the on-board system of vehicle A has to translate end-user inquiries into a sequence of location-sensitive queries. Each of these queries should be routed toward its designated location of interest, either via the vehicular ad-hoc network or through some other available network. Upon arrival to its destination area, the query must be picked up by the local vehicular service infrastructure. Nodes of that infrastructure (vehicles and/or roadside services) collaborate on-the-fly to compute a reply, which is dispatched back to the location where the query came from.

The goal of our work is to take advantage of VANETs established among vehicles equipped with the capabilities described in section II-A, in order to design a vehicular service infrastructure that is capable of carrying out transactions like the ones described above, and allow vehicular services such as EZCab [16] to be built on top of it.. As key building blocks of this infrastructure, we introduce:

- The **Vehicular Information Transfer Protocol**. VITP is an application-layer, stateless communication protocol

that specifies the syntax and the semantics of messages carrying location-sensitive queries and replies between the nodes of a vehicular ad-hoc network. VITP is independent of underlying VANET protocols that undertake the transmission and routing of VITP messages.
- The **VITP peer**, which is a lightweight software component to be deployed on the embedded computer of modern vehicles. VITP peers implement the VITP protocol and operate as clients, intermediaries, or servers in a VITP-protocol interaction.
- A **location encoding scheme**, which organizes and represents symbolically road segments and directions. This scheme is used by VITP for the specification of location-aware queries and for supporting underlying geographic routing protocols, which make use of on-board navigation services to transform symbolic locations into GPS coordinates [17].
- A number of protocol features designed to support **performance optimizations** (message caching, VITP traffic reduction), **quality assurance** for VITP results (termination conditions of VITP queries), and the **protection of privacy** of vehicle drivers.

In contrast to recent traffic monitoring systems, which are based on the continuous dissemination of traffic conditions through vehicular ad-hoc networks [17], [10], VITP proposes the *pull-based* retrieval of traffic information, which can be triggered on-demand by *location-sensitive queries* issued from VITP-enabled vehicles. The pull mechanism of VITP can help drivers make adjustments to their path while driving to some destination on a given route; these adjustments can be based on information collected by VITP for shorter-term traffic conditions of nearby road segments. The pull-based mechanism of VITP can also provide time-sensitive information about services, such as the current value of gas in a nearby gas-station, or the current number of free parking spaces in a nearby parking and so on.

VITP supports also the *push-based* propagation of messages, as a mechanism for disseminating various alerts that carry information about emergencies or serious deviations from normal traffic conditions. Finally, VITP-collected information can be used by higher-level software to derive traffic estimates across larger time-scales.

C. Key Design Concepts

We anticipate that service provisioning over vehicular ad-hoc networks will be based on an extended client-server computing model. In this model, a driver inquires information about traffic conditions or available facilities on some road segment. This inquiry is translated into query messages sent toward that road segment, via the underlying VANET. Vehicles in the destination area collaborate to establish a server, to resolve the incoming queries and to send back messages carrying the requested information. The *Vehicular Information Transfer Protocol* specifies the format and the semantics of query and reply messages exchanged between vehicular clients and servers. The main design concepts behind the VITP architecture and message specification are described below:

1) Location-aware requests: In a vehicular-service provisioning model, service requesters are interested primarily in attributes describing traffic conditions and service facilities available to drivers in some particular geographic area. Therefore, vehicular-service queries must be *location-sensitive*, specifying explicitly the *target location* of their inquiry. Given that the motion of vehicles is constrained within the road system, we can assume that the geographic areas of interest are restricted to roads, road segments, directions of traffic, and adjacent roadside areas.

Accordingly, locations are represented in VITP as two-value tuples `[road_id,segment_id]`, where `road_id` is a unique key representing a road and `segment_id` is a number representing a segment of that road; opposite traffic directions on the same part of a road are represented as different road-segments. The feasibility of such schemes has been demonstrated in recent literature [18], [10].

2) Virtual Ad-Hoc Servers (VAHS): The server that computes the reply to a VITP query is essentially a dynamic collection of VITP peers, each of which:
- runs on a vehicle that moves inside the query's target-location area, and
- is willing and able to participate in the query's resolution by contributing information from its on-board diagnostics sensors or local cache.

The establishment of this collection of peers is done in an ad-hoc manner, and relies on the vehicular ad-hoc network established by vehicles moving inside the target-location area.

To better reflect the dynamic and ad-hoc establishment of VITP servers, we refer to the dynamic collection of VITP peers that are inside the target-location of a VITP query and take part in the query's resolution, as a *Virtual Ad-Hoc Server (VAHS)*. Through its participation to a VAHS, a VITP peer has the opportunity to improve the accuracy of the information collectively acquired and distributed over the VANET, without having to pay any significant cost. It would be interesting, however, to adopt mechanisms that would provide VITP peers with incentives to join Virtual Ad-Hoc Servers and to participate in VITP-query resolutions. The study of such mechanisms is beyond the scope of this paper and will be part of future work.

The collection of peers that constitute a VAHS, and the VITP peers that manage the VITP communication, follow a *best-effort approach* in their operation. A VITP peer that has joined a Virtual Ad-Hoc Server, does not have information about other members of the group. It is also possible that a VITP peer joins a VAHS, participates in its computation, and leaves the target-location area before the completion of the query's resolution. The Virtual Ad-Hoc Server, on the other hand, does not maintain explicit knowledge of its members. The VAHS is established on-the-fly; its constituents can be derived only by the choice that VITP peers make individually about serving or simply forwarding VITP requests, and by the semantics of the VITP messages they exchange. In other words, the Virtual Ad-Hoc Server is identified with a query and its target-location area, rather than with the VITP peers that participate in it. We have taken this approach in order to make the VITP protocol stateless and lightweight, and to keep the VITP state-machines as simple as possible. This approach

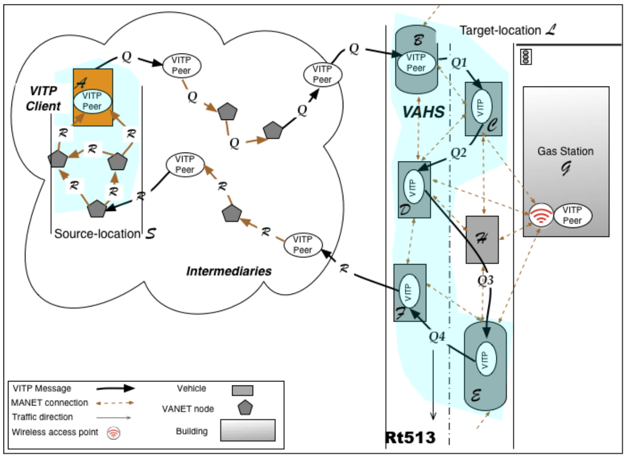


Fig. 2. Clients and Servers in a VITP transaction.

should allow for a simple, web-server-like implementation of VITP peers that could easily fit even on embedded processors.

3) VITP transactions: Figure 2 depicts a typical VITP-transaction that takes place in the context of the service-provision scenario presented earlier in Figure 1. This transaction is initiated by vehicle $\mathcal{A}$, which is located in road segment $\mathcal{S}$ (Woods Ave in Fig. 1) and inquires information about the average speed of at least four vehicles inside road segment $\mathcal{L}$ (first segment of Rt513), as an estimate of traffic-flow conditions in $\mathcal{L}$. To this end, the VITP peer of $\mathcal{A}$ submits a VITP request $\mathcal{Q}$ with $\mathcal{A}$'s inquiry.

The VITP transaction consists of four phases. In the initial *dispatch-query phase*, a request $\mathcal{Q}$ is transported through the underlying VANET toward its target area $\mathcal{L}$. $\mathcal{Q}$ goes through a number of intermediary VANET nodes, which push the message toward its destination using geographic routing. Intermediary nodes may not be VITP-enabled (these are depicted as grey pentagons in Figure 2); these nodes simply pass the message on toward $\mathcal{L}$.

When $\mathcal{Q}$ is received by a peer $\mathcal{B}$ that is inside target-area $\mathcal{L}$ and is willing to join a Virtual Ad-Hoc Server to resolve $\mathcal{Q}$, the VITP transaction enters its second phase, the *VAHS-computation phase*. During this phase, the VITP request is routed between the VITP peers of the VAHS. These peers modify the VITP request in order to: (i) indicate that the request is part of an ongoing VAHS computation (this modification takes place only at the first peer that joins the VAHS), and (ii) piggybVTack partial query results to the VITP message's payload. Referring, for example, to Figure 2, when peer $\mathcal{B}$ receives the VITP request $\mathcal{Q}$, it parses the request, extracts the requested information from its on-board diagnostics system, *rewrites* the query in order to store the partial result into the query's body and to indicate that the query is now part of a VAHS computation, and passes the message on to its neighbor. The semantics of the query indicate how the underlying network protocol will treat the rewritten VITP query (unicasting or broadcasting it to neighboring peers).

A VITP request is transported between VAHS peers until some *Return Condition* is satisfied. The VAHS peer that detects the upholding of the Return Condition creates the VITP reply and posts it toward source-region $\mathcal{S}$ through the VANET. The transportation of the VITP reply toward $\mathcal{S}$ corresponds to the third phase of the VITP transaction, the *dispatch-reply phase*. When the VITP reply reaches area $\mathcal{S}$, the VITP transaction enters its final phase, the *reply-delivery phase*. During this phase, the underlying network protocol broadcasts the VITP reply to the VANET nodes of $\mathcal{S}$, so that the reply can be received by the VITP peer that originated the transaction.

In the scenario described above, we make the assumption that the geographic routing protocol can discover a route between the source and the target location areas, $\mathcal{S}$ and $\mathcal{L}$, using the underlying VANET connections. If such a route cannot be found, however, due to insufficient VANET connectivity, the VITP messages can be dropped leading to a VITP transaction failure.

4) Return Conditions: An important issue arising in the context of the VAHS-computation phase is how to define the Return Condition for a VITP request. A Return Condition determines at which point the resolution of a VITP request can be considered accomplished. In other words, the Return Condition indicates if a VITP reply can be created and dispatched back to the originator of the request. The decision on what constitutes success in the resolution of a VITP query, however, depends on the semantics of the query itself. Therefore, the Return Condition must be defined explicitly as part of the query's specification.

For instance, referring to our example of Figure 2, suppose that vehicle $\mathcal{A}$ is looking for a gas station on road segment $\mathcal{L}$; when the corresponding VITP request reaches the VITP peer of gas station $\mathcal{G}$, the peer switches to the VAHS-computation phase, parses the incoming query, detects that the query requests information about *at least one* gas station in $\mathcal{L}$, and decides that it can fully resolve the query and that the Return Condition is satisfied. Consequently, it creates a VITP-reply message with $\mathcal{G}$'s coordinates and prices, and sends the reply to $\mathcal{S}$. In contrast, if $\mathcal{A}$ is looking for the prices of *more than one* gas stations in road segment $\mathcal{L}$, $\mathcal{G}$'s peer will start the VAHS-computation phase, re-write the incoming query, and try to pass the query on through the VANET, in search of other gas stations: clearly, the query's Return Condition is not satisfied yet. In the absence of other gas stations in $\mathcal{L}$, however, this Return Condition will never be met, the original VITP query will not be resolved, and $\mathcal{A}$'s peer will not receive any VITP reply. To address such cases, VITP supports an alternative Return Condition, which is constrained on the total time a query can spend in the VAHS-computation phase.

5) VITP Protocol layering: The operation of the VITP protocol presumes an underlying networking infrastructure that undertakes the transport and routing of VITP messages between peers installed in vehicles and roadside services. Typically, networking support will be provided by vehicular ad-hoc networks, although VITP messages could also be transported via other networks, such as cellular. It is important to observe that the way VANET nodes handle VITP messages is influenced by VITP semantics. In particular, a VITP message that is part of a VITP transaction in either dispatch-query or dispatch-reply phase, must be routed *geographically* toward its target location. In contrast, a VITP reply that is part of a VITP transaction in the reply-delivery phase, should be *broadcast* inside its target-location area (and/or nearby areas), so that it

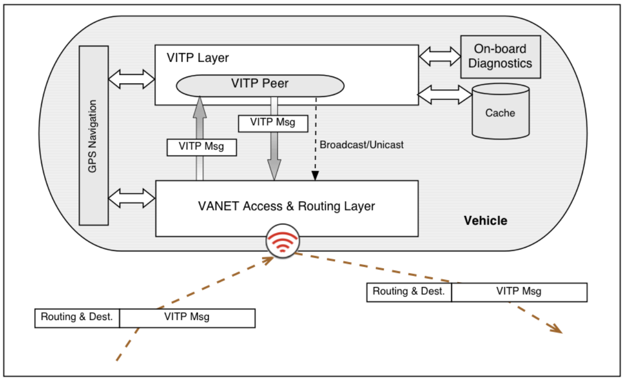


Fig. 3. VITP Protocol layering.

reaches the VITP peer that originated the VITP transaction. Finally, the routing of a VITP message that is part of a VAHS-computation phase, depends on the semantics of the method specified in the corresponding VITP request.

The interaction between VITP and the routing protocol of the underlying VANET is as follows: whenever a VITP message arrives at a VANET node, the network layer *always* makes a call to the local VITP peer (See Figure 3). This call is made even if the peer is an intermediary, i.e., it is not placed inside the target location of the message. If the node is not VITP-enabled or if its VITP peer is busy or down, the call will fail; in that case, the network layer will simply retransmit the message to a neighboring node. Otherwise, the peer will receive and parse the message. Depending on the active VITP phase and the semantics of the message, the peer may re-write the message before retransmission. The peer will also signal the VANET routing module about the routing method to be used when transmitting the outgoing message (unicast or broadcast).

6) Caching: An intermediary peer that receives a VITP request can search into its local cache for a matching reply. The matching test should take into account both the semantics of the VITP query (as described by the query's `uri`) and the specification of the target region. In the case of a match, the peer can send the cached reply back to the VITP client and either complete the VITP transaction or retransmit the incoming message toward its target location. This decision affects the Return Condition of the VITP request and must be based on the semantics of the incoming VITP message. Therefore, VITP provides cache-control headers that can be included in VITP messages and act as directives to VITP-peer caching decisions.

7) Message Identifiers and Driver's Privacy: To achieve the delivery of a VITP reply to the peer that requested it and to preserve the correctness of a VAHS computation, we must ensure that: (i) A VITP peer can match incoming replies against its pending requests and can detect replies belonging to other peers. (ii) A VITP peer will not act again on the same VITP request even if it receives this request multiple times. To this end, a unique, random identifier (`msgID`) is attached to every new request. The same identifier is also attached to messages derived from the original VITP request, that is, to modified requests exchanged during a VAHS-computation phase and to the resulting VITP replies. VITP peers maintain a cache with recently received `msgID`'s. Every time a peer receives a VITP message, it compares its `msgID` against cached `msgID`'s in order to determine how to handle the incoming message. New identifiers are cached for a default period of time.

The `msgID` can be produced by hashing a combination of the Vehicle Identification Number, current time at the creation of the corresponding request, and vehicle location. This approach allows for the protection of driver privacy because the messages exchanged in the context of a VITP transaction do not carry any information identifying the driver or the vehicle that initiated the transaction.

8) Dissemination of Traffic Alerts: The VITP features described so far support a pull-based model of vehicular service provision. However, vehicular applications can benefit equally from a push-based model of information provision. For example, in the motivating scenario described in Section II-A and depicted in Figure 1, the vehicle moving in JFK Dr may detect a slippery road. Information about such a dangerous condition (depicted with an asterisk in Figure 1) should be propagated to other vehicles moving into the area. To this end, the vehicle that detects this condition must generate an alert message and transmit it via the underlying VANET. To support the transmission of traffic-alert information, VITP provides a special message-type dedicated to information dissemination ("push"). A "push" message carries a special VITP method, the VITP representation of a target-location area, a description of the alert, a unique identifier, and other VITP-specific attributes (expiration time, caching directive, etc). The protocol treats VITP alerts similarly to VITP replies: an alert message is transported to its target-location via geographic routing; upon arrival to it target-location, the message is broadcast to all vehicles in that area. Alternative implementations could combine geographic routing with broadcast inside areas along the way between the source and the target location.

Finally, VITP can be easily extended to support *persistent queries*, which allow the implementation of traffic-information dissemination schemes. A persistent VITP query arrives in its target location area and "stays" there for a pre-defined amount of time, re-calculating its answer and sending back a new reply at periodic intervals. This can be achieved through the service-migration mechanism described in [19], which allows the service end-point of the persistent query to migrate when the car hosting it leaves the target area.

9) Security: Potential requirements for security, depending on the kind of vehicular service supported over VITP, could be data verification, non-repudiation, confidentiality, authenticity, integrity, availability, privacy, etc. Although VITP does not provide built-in security features, overall security can be provided by other protocol layers and mechanisms, such as tamper-proof devices on vehicles, public/private key pairs and certificates issued by a trusted third party, secure geographic routing and location validation protocols [20], [21], [22].

III. VITP-MESSAGE SPECIFICATION

Here, we present a brief overview of the Vehicular Information Transfer Protocol specification, focusing on the format of VITP messages. More details can be found in [7]. The

TABLE I

STRUCTURE OF VITP REQUESTS.

VITP message syntax	
Line 1:	`METHOD <uri> VITP/<version_number>`
Line 2:	`Target: [rd_id-dest,seg_id-dest]`
Line 3:	`From: [rd_id-src,seg_id-src] with <speed>`
Line 4:	`Time: <current_time>`
Line 5:	`Expires: <expiration_time>`
Line 6:	`Cache-Control: <directive>`
Line 7:	`msgID: <unique_key>`
Line 8:	`Content-Length: <number of bytes>`
Line 9:	`CRLF`
Line 10:	`<message body>`

URI syntax
`/<type>/<tag>?[<rc_expr>,...]&<param_expr>&...`

Examples of VITP requests
`GET /vehicle/traffic?[cnt=10&tout=3000msec]&tframe=3min`
`POST /vehicle/alert?[cnt=*&tout=*]&type=slippery_road`

syntax of a generic VITP message is given in Table I. VITP provides two types of messages, distinguished by the `METHOD` entity placed at the beginning of each message (see Line 1 in Table I). `METHOD` takes the values `GET` and `POST`: `GET` represents a VITP request that queries the attributes of some geographic area (pull model); `POST` is used for VITP replies and for messages that disseminate an attribute of some particular location toward some other geographic area (push model). The information requested by or transported through a VITP message is specified further by the `<uri>` attribute (see Table I, Line 1).

The `Target` and `From` headers of Lines 2 and 3 in Table I specify the target and source-location areas of a VITP message, respectively. Locations are formatted according to a standard scheme that specifies the road and segment identifiers, as retrieved by an on-board navigation and positioning system. The `Time` header (Line 4 in Table I) carries a timestamp specifying the point in time at which the VITP message was generated by its originating peer. The different peers maintain their clocks synchronized by retrieving the time value from the GPS.

The `Expires` header of a VITP message (Line 5 in Table I) specifies a point in time after which the corresponding VITP transaction has to be terminated. In the case of a `GET` request, the expiration time indicates that the originating peer wants to receive a reply before the specified expiration time. Consequently, any peer that receives a VITP message (request or reply) as part of the transaction after this time, can drop the message and not propagate it further. In the case of a `POST` request, the expiration-time header indicates when the VITP peers should stop propagating the corresponding alert. The `Cache-control` header of Line 6 defines the caching directives that should apply to a VITP request.

The `msgID` header is used to carry the unique numerical message identifier that is assigned to every VITP query. The remaining lines (Lines 8-10 in Table I) are used only in VITP requests that carry intermediate results during VAHS-computation phase or in `POST` messages. The `Content-Length` header declares the size of the data carried by the request. The actual data follow after the `CRLF` character.

The syntax of the `<uri>` attribute is presented in the second part of Table I. The `<type>` field of the `uri` specifies the classes of VITP-enabled physical-world entities involved in the resolution of a `GET` message or in the generation of a `POST` message. Currently, we anticipate two types of entities: vehicles (`type=vehicle`) and roadside facilities that offer services to vehicles (`type=service`). The `<tag>` field describes the actual information sought or disseminated by a VITP request. For example, a tag value `traffic` indicates that the request queries for information about road-traffic conditions expressed in terms of the average speed of vehicles in the area of interest. A tag value `alert` indicates that the request is trying to post a new alert, when used with a `POST` command. A special type of VITP query, defined with the tag-value `index`, returns the types of queries supported by VITP peers at the query's destination area.

The "`?`" character that follows the tag field in the URI syntax of Table I separates the request specification from its parameter list, which is a series of `(name, value)` expressions separated by the "`&`" character. VITP allows for two types of parameter expressions:

- Those that define the *Return Condition* of VITP requests (`rc_expr` in Table I). These expressions are placed inside a pair of brackets immediately after the "`?`" character. There are two default return-condition parameters, `tout` (time-out) and `cnt` (count). `tout` specifies the maximum lifetime of a GET-request resolution (VAHS-computation phase); `cnt` specifies the required number of peers that should contribute to the resolution of a request.
- Expressions placed after the Return Condition brackets. These, specify parameters that should be passed to the actual query that is to be executed on the VITP peer (`<param_expr>` field in Table I).

Two typical examples of VITP requests are given in Table I. The first query requests the average speed of 10 vehicles moving in the area of interest and specifies that this computation should be completed within 3000 msec. Upon arrival to a VITP peer, the query will retrieve from the OBD the speed of the vehicle, averaged over the last 3 mins; the 3min averaging is specified with the `tframe=3min` request parameter, which is specific to the `traffic` query. The second request generates a "slippery road" alert, which is dispatched as a POST message and for which the sender is not interested in getting back any information. Therefore, there is no Return Condition specified and it is up to the recipients of the alert to cache and/or keep posting it for as long as they see fit. The VITP implementation can impose a default timeout Return Condition of several hours, if the `tout` field of a query is set to "`*`."

IV. VITP EVALUATION

In this section, we present our simulation experiments with VITP. Our goal is to investigate the feasibility of VITP and to analyze its performance in large-scale vehicular networks, under realistic traffic conditions. To simulate VITP, we use mobility traces that correspond to two alternative road-traffic patterns: (i) The first is a *highway-traffic pattern* comprised of vehicles that move along a highway; the vehicles dispatch queries to investigate traffic conditions along the highway.

(ii) The second is a *city-traffic pattern* comprised of vehicles moving inside a city neighborhood and using VITP to query the traffic conditions in various road-segments of this neighborhood. To derive these mobility traces, we use two different traffic generators: (i) a highway traffic generator that we developed to model traffic patterns of cars traveling along a highway; (ii) *SUMO* ("Simulation of Urban MObility"), a powerful, open-source traffic generator from the Institute of Transport Research of the German Aerospace Center [23], [24]. For the simulation of wireless-network and VITP traffic, we use the *NS-2* simulator [25].

A. Highway Traffic Scenario

1) Simulation Setup: To conduct an early assessment of VITP performance, we developed a traffic generator that models cars moving along a highway. The generator accepts as parameters the simulation time, road length in meters, number of lanes per road, average speed of the vehicles in meters/sec, average gap distance between vehicles on same lane, number of service nodes on the road, and the number of users on the road. The tool simulates a simplified traffic model where: (i) vehicles may enter or leave the road through evenly distributed entries and exits located along the road every 1000 meters; (ii) vehicles can change their speeds and lanes independently of other vehicles, and (iii) vehicles are evenly distributed on the road; once a vehicle leaves the road, a new vehicle enters the road randomly. In our simulation scenarios we generated traffic for a 25km-long highway with 3 lanes and $375-1500$ vehicles. Vehicle speed is distributed uniformly over the range $[15, 25]m/s$ with an average of $20m/s$ to avoid the long term slowdown behavior of the random waypoint model of NS-2 as described in [26], [27]. The simulation time is set to 500 seconds.

For the purposes of this scenario, we choose as wireless medium an 802.11-compliant network with a data transmission rate of 11Mbps and a transmission range of 250 meters [1]. To allow vehicles to maintain neighbor connectivity, each vehicle broadcasts a `Hello` packet every period selected randomly from the range of 0.75 to 1.25 seconds. The received signal-strength threshold used in maintaining information about neighbors is set to distances below 200 meters in order to accommodate with the fast dynamics of the network and to maintain consistency in neighborhood information. Once a vehicle enters the road, it initiates its query at a random time selected uniformly over its remaining simulation time. The vehicle re-sends the query if no answer is received within a specified timeout of 10 seconds.

Current VITP implementation routes packets within vehicular networks using geographical routing and a packet forwarding strategy similar to MFR [28]. In this scheme, a packet is forwarded to the node that makes the most progress towards the destination in order to minimize the number of hops a packet has to traverse. If a vehicle fails in transmitting the packet to its next hop, the packet is dropped, indicating a failure of the query-dispatch phase and, consequently, a failure of the VITP transaction. Several other forwarding and routing strategies could be adopted and used within VITP.

To evaluate VITP in the context of the highway-traffic scenario, we assume that, once a vehicle enters the highway, it initiates a `traffic` query at a random time selected uniformly over the remaining simulation time. The vehicle re-sends the query if no answer is received within a specified timeout of 10 seconds. The `traffic` query requests the average vehicular speed within a road segment of S meters long, in order to derive an estimate of the traffic-flow at that segment. We assume that the road segment in question is D meters away from the query sender vehicle (QS); we refer to D as the *query distance*. We use as Return Condition cnt, i.e., the total number of vehicles of the target road-segment that should be sampled for their speed. As mentioned earlier, a vehicle in the target segment may receive the same query message multiple times, but it participates in updating the query results only when it receives the query message for the first time.

2) Metrics: To describe the performance of VITP, we employ the following metrics:

- *Response time* is the average time of a *successful* VITP transaction. It measures the elapsed time between the time at which a query is initiated and the time at which its corresponding reply is received. The elapsed time takes into account both VITP processing and message transmission time.
- *Dropping rate* is the percentage of unsuccessful queries, i.e., queries for which a vehicle times-out before getting a valid reply.
- *Accuracy* measures how close the estimated average speed (calculated usually by a subset of the available vehicles in the target segment) is to the actual average speed in the region of interest (calculated by considering all present vehicles in the same area).
- *Efficiency* measures the percentage of the number of exchanged query messages that were actually employed in calculating a result over the total number of query messages exchanged both in geographic routing and inside the target location. A message is considered to have participated in the result computation when it is received by a vehicle in the target segment for the first time.

3) Effects of query distance D: First, we study the effects of query distance to the metrics defined above. For this study, we set the average gap between consecutive vehicles on the same lane to 100m. We vary the query distance D from 500 to 5000 meters, with the query's target segment length S fixed to 800 meters; this translates to an average of 30 vehicles moving inside the target segment. The value of cnt ranges from 1 to 20 vehicles. The computation time within a vehicle is assumed to be negligible.

Figure 4 plots the response time versus the query distance D for different cnt values. The response time increases almost linearly with D. However, as the value of cnt increases and becomes comparable to the total number of vehicles in the target road-segment, a VITP request would have to cover a large percentage of the target vehicles in order to satisfy the Return Condition. Therefore, the query message would have to

---

[1] In practice, the wireless transmission range is less than 250 meters. However, this transmission range could be restored with the use of external antennas.

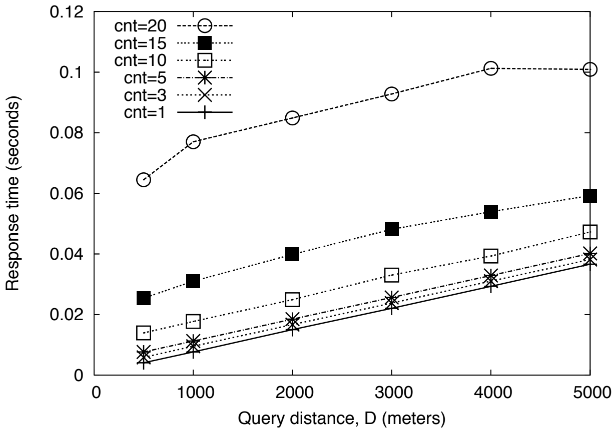


Fig. 4. Response time vs. query distance (D).

TABLE II

DROPPING RATES VS. QUERY DISTANCE (D).

Query distance (D)	Forward dropping rate (%)	Backward dropping rate(%)
500	11.84	0.47
1000	18.41	0.64
2000	36.06	1.52
3000	50.70	2.72
4000	60.69	3.65
5000	65.95	4.24

re-visit many vehicles before succeeding to discover unvisited vehicles. This translates to longer VAHS-computation times and, consequently, to longer response times. In our scenarios, with approximately 30 vehicles in the target road-segments, we observe that response time increases substantially for values of cnt greater or equal to 15 (see Figure 4).

Table II reports the dropping rates for different query distances. The forward-dropping and backward-dropping rates correspond to the query-dispatch and reply-delivery phases: the forward dropping rate is measured as the percentage of the failed queries due to the failure of the query-dispatch phase, over the total number of generated queries. The backward dropping rate is the percentage of the failed queries, due to failure of the reply-delivery phase, over the number of queries that successfully reach the region of interest. From Table II, we observe that both dropping rates increase with query distance D. For very distant queries, the forward dropping rate becomes prohibitively high (i.e., 65% for D=5000m), making it harder for queries to complete successfully. Nevertheless, once a query message finds its way to its target location, it is highly possible that the reply message will be routed successfully to the QS vehicle, since the connectivity between vehicles during the VAHS-computation phase remains stable. The data reported here correspond to a $cnt = 10$ (as we will see later, this value results to greater efficiency and a high accuracy for our simulation scenario). Using different values of cnt had no effect on both dropping rates.

Figure 5 plots the accuracy of the query results versus cnt for different query distances D. For our simulation scenario, we achieve a maximum accuracy of approximately 90% for cnt values greater than 5. The query distance has negligible effect on the accuracy.

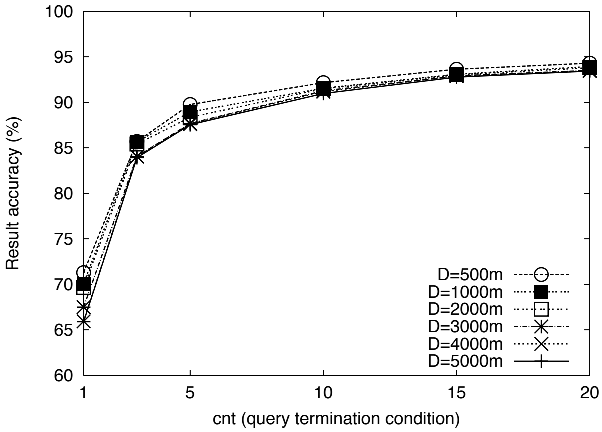


Fig. 5. Result accuracy vs. cnt for different query distances (D).

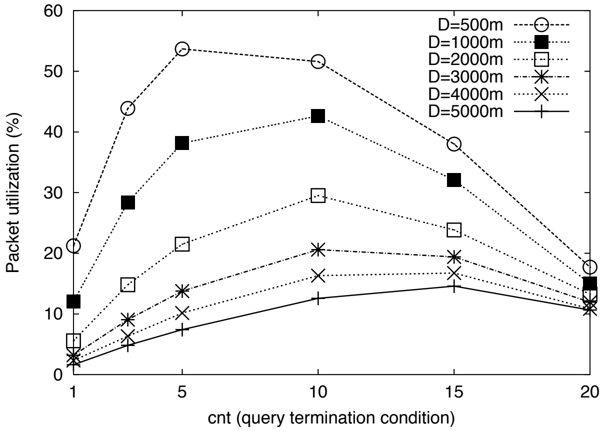


Fig. 6. Query efficiency vs. Return Condition (cnt) for different query distances (D).

Figure 6 plots the query efficiency versus cnt for different query distances. As expected, the efficiency is higher for smaller D's because, as D decreases, the number of forwarded query messages in the query-dispatch and reply-delivery phases becomes smaller. It is interesting to observe that for each query distance examined, there is an optimal value of cnt for which the efficiency is maximized. When using smaller cnt values, the overhead of forwarding the query messages during the query-dispatch and reply-delivery phases dominates. Adopting cnt values greater than the optimal value dictates the visit to a large number of the vehicles in the target segment and this results to a large number of query messages that have to be forwarded to previously visited vehicles. For the parameters of our simulation study and for most query distances examined, this optimum value of cnt is around 10.

4) Effects of vehicle density: We study the effects that vehicle density has on VITP performance. We use a simulation scenario similar to the one used when evaluating the effects of query distance D. However, we fix D to 2000 meters and change the vehicle density by changing the gap between consecutive vehicles on the same lane from 50 to 200 meters. The response time increases with the gap (see Figure 7) but this effect becomes pronounced for larger values of cnt.

Examining the dropping rates, we find that the forward and backward rates increase significantly with the gap, for

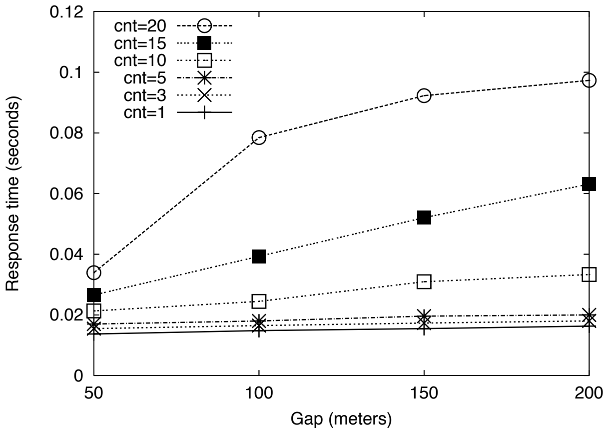


Fig. 7. Response time vs. gap between consecutive vehicles for different Return-Condition (cnt) values.

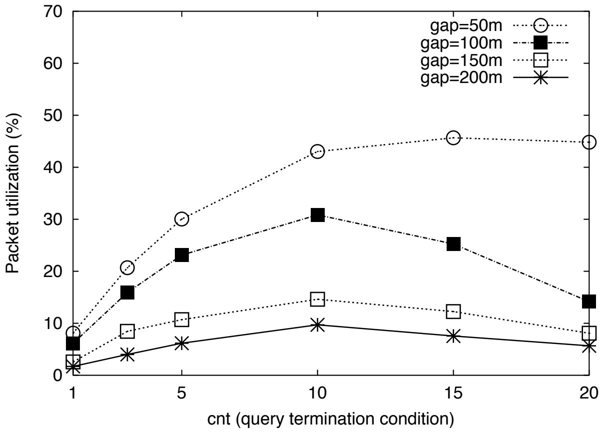


Fig. 8. Query efficiency vs. cnt for different gap values.

all values of cnt. For example, the forward dropping rate for $cnt = 10$ increases from 0.81% to 89.14% when the gap increases from $50m$ to $200m$; for the same increase in the gap, the backward dropping rate increases from 0.1% to 7.89%. Finally, an increase in inter-vehicle gap reduces significantly the measured efficiency (see Figure 8), especially for large cnt values. The reason behind this reduction is that as the vehicle density decreases when increasing the gap, it becomes more difficult to reach the required number of vehicles in the target region. This difficulty increases even more when cnt is large.

We have also studied the effects that the query request rate has on VITP performance. We measured the response time for the successful queries and found that changing the request rate did not have a proportional effect on response time. Increasing the request rate, however, results to an increase in the forward and backward dropping rates.

B. City Traffic Scenario

1) Simulation Setup: To further evaluate the functionality and performance of VITP under realistic urban-traffic conditions, we use *SUMO*, an open source, microscopic, space continuous, vehicular traffic simulator from the Institute of Transport Research of the German Aerospace Center [23], [24]. SUMO accepts as input a map of a road network, annotated with information about speed limits, priority junctions, operational traffic lights, lane-connections etc. It simulates the motion of each vehicle individually, according to models that take into account physical characteristics of vehicles (size, maximum speed), traffic behavior (acceleration, deceleration), surrounding traffic conditions, and the road network.

TABLE III

DROPPING RATES VS. REQUEST RATE FOR $cnt = 10$.

Request rate	Forward dropping rate (%)	Backward dropping rate(%)
1.5	50.70	2.72
2	53.36	3.19
3	62.09	3.86
4	72.29	4.52
10	81.79	4.81
30	87.75	4.98

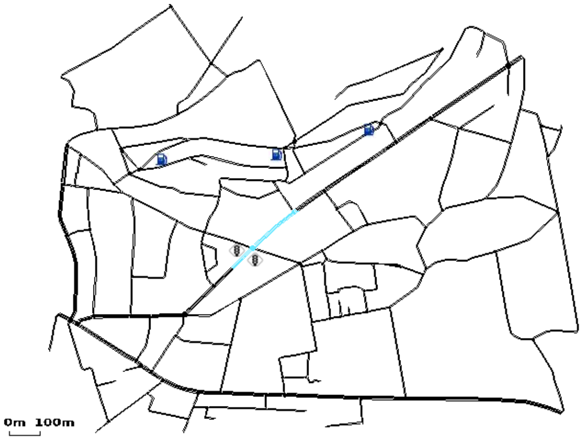


Fig. 9. Neighborhood city map used in simulations.

We fed SUMO with the road network of a medium-scale neighborhood extracted from a data-set with a detailed map of the German city of Osnabrück in Lower Saxonia [29]. By pruning this map, we got a medium-scale neighborhood of $2.38Km^2$ (see Figure 9). We used SUMO to define a few "hot spots" inside the neighborhood, i.e., locations that attract a lot of traffic when people go to or return from work, shopping etc. For the experiments presented below, we assume that there are no operational traffic lights unless stated otherwise. Therefore, the traffic flow is determined by the speed limits of streets, priority junctions, and SUMO's model for the motion of individual vehicles. With this configuration setup, the vehicular traffic generated by SUMO comprises around 200 vehicles that move simultaneously in the neighborhood of our simulation.

The output of SUMO is translated into input supplied to the NS-2 simulator in order to describe the movement of nodes that take part in our simulation. For the wireless-network simulation we assume an 802.11 network with 11Mbps links and 250 meters range. Each node broadcasts a `Hello` packet containing the node's identity and current location. This action takes place every 1 second plus a random "jitter," uniformly distributed over a 0-0.5 seconds interval. We apply a 200-meters threshold when selecting the neighbors of a given node. The `Hello` message exchange allows the nodes to maintain knowledge of the location of their neighbors, which is vital for geographic routing.

We simulate VITP `traffic` and `alert` requests in the context of the scenarios derived with SUMO. Again, we assume the existence of a geographic routing layer that pushes VITP messages towards their destination area. To simulate this effect, we apply greedy forwarding of packets to nodes that are always progressively closer to the destination [11]. In regions of the network where such a greedy path does not exist, we drop the message indicating that the corresponding VITP transaction cannot complete.

Before starting the simulation of wireless communications and VITP traffic, we allow for the vehicular-traffic simulation to run for some time so that traffic circulation inside the simulation area reaches a steady state. After that time, cars start exchanging messages and sending VITP requests. We assume that vehicles initiate VITP requests at random moments, distributed uniformly over the remaining simulation time.

To evaluate the performance and functionality of VITP, we use again the following metrics: *Response time*, *Dropping rate*, *Accuracy*, and *Efficiency*. We choose randomly the vehicles that issue `traffic` requests. These requests query the speed of cars that move inside randomly selected road segments with length S that varies between $400-800$ meters. The query distances used are smaller than in the highway scenario because, with D ranging from 400 to 1200 meters, we can effectively query every segment of our map. We run simulations with cnt values between $5-20$. The maximum observed density of vehicles inside a single query resolution area is approximately 25 vehicles.

2) Response time and Dropping rate: City-traffic simulations show that the topology of the road network does not affect the response-time of VITP requests: in agreement to the highway-traffic simulation of Section IV-A, response-time increases almost linearly with distance D and with cnt, and the response-time values measured remain below $0.1sec$ even for large cnt values ($cnt = 20$).

From the simulation results, however, we observe that both the forward and the backward dropping rates of VITP messages range between $10-20\%$ in every route, depending on factors like node density and query distance. This is in contrast to the highway scenario, where the backward dropping rate is much smaller than the forward dropping rate. The reason behind this difference is that, in the city-scenario, the query-area is large enough to lead the routing protocol to return the VITP reply through a different route than the one taken by the VITP request.

3) Result accuracy: In Figure 10, we plot the accuracy of the `traffic` query results versus cnt, averaged over different values of D in the $400-1200$-meter range. As we can see from this plot, VITP returns an accurate estimate of the average speed of vehicles that move inside the query destination area, even for small cnt values. In particular, we achieve an accuracy of 90% for $cnt = 5$; when increasing the value of cnt to 20, the accuracy increases to more than 95%. The accuracy in the city scenario is somewhat smaller than the one observed in the highway scenario for values of cnt smaller than 5 and plotted in Figure 5. This is attributed to the fact that there is a larger variation in the traffic behavior of vehicles that move inside a city.

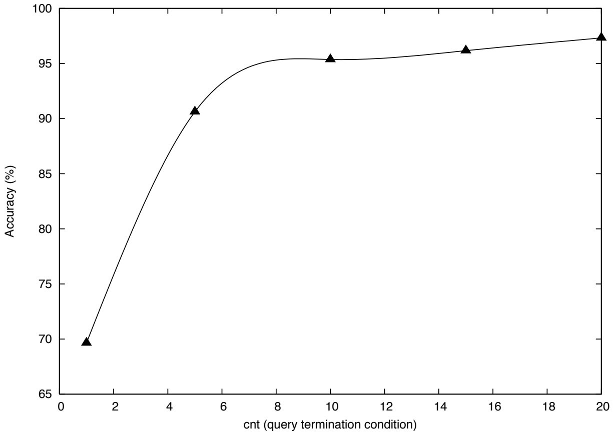


Fig. 10. Result accuracy vs. Return Condition (cnt).

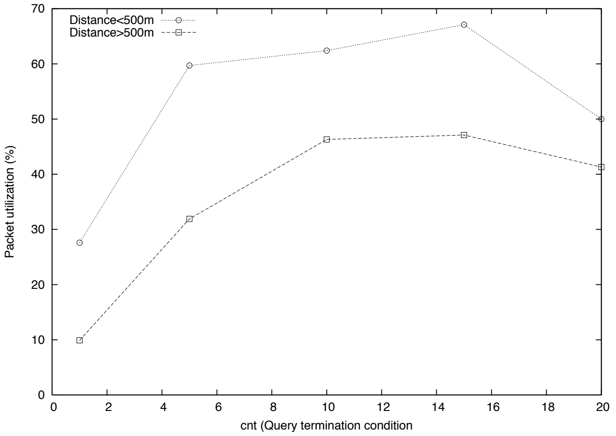


Fig. 11. Query Efficiency vs. Return Condition (cnt) for different query distances.

4) Query efficiency: Figure 11 plots the query efficiency versus cnt for different query distances. As expected, and in agreement with the results of Section IV-A, the efficiency is higher for smaller values of D. We observe also that there is an optimal value of cnt for which the efficiency is maximized: similarly to Section IV-A, query efficiency is maximized for cnt having a value between 10 and 15.

To investigate the efficiency of the VAHS computation phase in the city-traffic scenario, we define as *VHAS efficiency* the ratio of cnt over the total number of VITP messages that are exchanged between vehicles taking part in the VAHS computation phase. We calculate the average value of this ratio over different values for D (query distance between $400-1200$ meters) and S (query-destination area with a radius between $400-800$ meters). A diagram of the *VAHS efficiency* is presented in Figure 12. As expected, VAHS efficiency drops for larger cnt values because the VITP request keeps moving inside the destination area until it discovers the requested number of vehicles or until its timeout expires; thus, the request stays for a longer time in its destination area and ends-up visiting the same vehicles more than once. The percentage of "overhead" messages, however, remains less than 20% for cnt values that provide high-accuracy results ($cnt = 10, 15$).

5) Traffic Lights: To assess the effectiveness of VITP in the presence of traffic lights, we add two operational traffic lights near a junction of our map, and one fixed node that dispatches

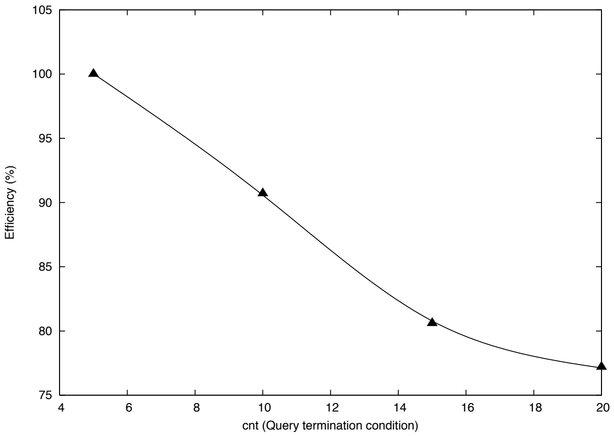


Fig. 12. VAHS Computation Phase Efficiency vs. Return Condition (cnt).

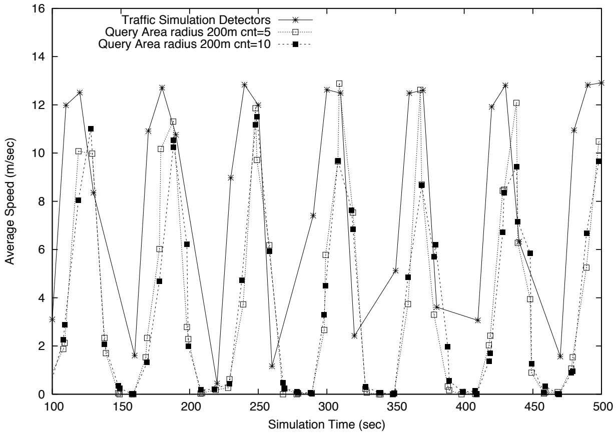


Fig. 13. Speed estimation in the presence of traffic lights (Return Condition: $cnt = 5$ and $cnt = 10$).

`traffic` queries towards a road-segment near that junction. We also add a "detector" on this road segment in order to monitor the average speed of vehicles that move through it.

The diagram of Figure 13 presents the average speed of vehicles in the road segment of interest as measured by the SUMO detector, along with the average speed reported in the VITP reply messages delivered to the fixed node of our VITP simulator. We provide different plots for different values of cnt. As we can see from this diagram, the VITP replies accurately follow the rapid changes in the average speed of vehicles caused by the traffic light in the road segment of interest.

6) Alert Propagation and Gas Station Discovery: As described earlier, the VITP protocol supports the push-based propagation of messages in order to allow vehicles to disseminate alert conditions and to inform their neighbors of accidents or other conditions of danger. To evaluate this functionality, we add to our traffic-generation model a vehicle that breaks down, stops, and then produces a VITP `alert` message, which is disseminated in a target area surrounding the vehicle. We examine the extent of the alert propagation and observe that all vehicles in the destination area do receive the `alert` message. The probability that a vehicle in the target area would miss the alert message is minimal, due to the fact that vehicles inside the destination area re-broadcast the alert message when they receive it for the first time; the vehicles will drop this message upon subsequent receipts. We also note that the alert is propagated to vehicles outside the destination area, which leads to an overhead, but this is the price we pay if we want to make sure that everybody in the alert area will get the message.

TABLE IV

DROPPING RATES AND TIMEOUTS VS. THE RETURN CONDITION (cnt).

Return Condition	Success rate (%)	Timeout rate(%)
1	76.2	0
2	38.8	36.3
3	17.0	42.1

Finally, we examine a scenario where VITP is used to retrieve information about VITP-enabled road-side services. To this end, we place three VITP-enabled gas stations along a $450m$-long street segment of our city map (see Figure 9). Randomly chosen vehicles dispatch VITP `service` requests in order to query an $800m$-long area covering the designated street segment about the existence of gas stations. We set the Return Condition (cnt) to 1 through 3. Here, cnt represents the maximum number of gas-stations that the VITP request is supposed to find and poll. We evaluate two metrics: (i) The *success rate*, that is the percentage of `service` queries that locate the requested number of gas stations; (ii) The *timeout rate*, that is the percentage of VITP `service` queries that return with an incomplete reply to the originating vehicle because of VHAS timeout expiration.

Table IV shows the success and timeout rates for the three return conditions examined ($cnt = 1, 2, 3$). The success rate decreases substantially for values of cnt greater than 1, while the timeout rate increases. This reflects the difficulty of locating very specific targets (gas stations) in a large query area. A user may influence these rates by decreasing the query area or defining a bigger timeout value.

V. RELATED WORK

VITP can be seen as a framework for context-aware services, where location is used as the primary attribute for defining "context." Recently, several projects have proposed the adoption of context-awareness to build context-aware services on top of mobile ad-hoc networks. The Sentient Model [30] abstracts context-aware applications in pervasive ad-hoc environments as a large collection of software components called Sentient Objects. These objects accept input via a variety of sensors and autonomously react by acting upon the environment through a variety of actuators. Sentient Objects were used to build sentient vehicles, which are context-aware vehicles cooperating over mobile ad hoc networks [31]. Julien et al. have proposed the EgoSpaces model [32] and demonstrated how context-awareness can be employed to abstract resources available in an ad-hoc network into a data structure. EgoSpaces consist of mobile agents that operate over mobile nodes. The agents can specify which data have to be included in their operating context by means of declarative specifications. Unlike VITP, none of these context-aware schemes has been designed with vehicular network characteristics in mind.

A number of recent works, such as context-aware migratory services [19], virtual mobile nodes [33], and "follow-me" services [34], propose platforms for ad-hoc and vehicular services. Unlike these proposals, where service code is capable of migrating to different nodes in the network in order to effectively accomplish the service task, VITP does not transfer any code between VITP peers and, therefore, it is a more secure and easier to deploy and adopt protocol.

Few attempts have been done so far to deploy real services over mobile ad hoc networks: CarTel [35] focuses on building a delay-tolerant mobile sensing architecture based on opportunistic communication. User queries are answered by a continuous query processor that runs on a central portal. MetroSense [36] proposes three-tier architecture for scalable support of concurrent people-centric applications. Urban Sensing [37] seeks to build short-term, community-oriented urban sensor networks. All these projects assume central collection points, which perform data and task management and act as mediators between users and the network. VITP, on the other hand, presents a complementary decentralized mechanism for services over vehicular networks.

Vehicular networks and correspondind applications are closer to reality than ever before. Academia, industry, and goverments worldwide are pushing for vehicular technologies and services to improve transportation and quality of life. For example, the U.S. DOT Intelligent Transportation System Joint Program Office (ITS JPO) has launched several initiatives for improving transportation safety, such as the *Integrated Vehicle Based Safety Systems* and the *Vehicle Infrastructure Integration*. Within these initiatives, a network-centric architecture for information flow is proposed to provide a connection between users and data sources with a minimum interface. VITP could easily fit in this architecture as the underlying communication protocol that supports and facilitates information queries about the vehicular network. Moreover, significant efforts have been made to identify which communication-enabled vehicular safety applications will provide the greatest benefits. The deliberations by the US National Highway Traffic Safety Administration (NHTSA), the US Department of Transportation (USDOT), and the Vehicle Safety Communications Consortium (VSCC) have identified eight applications [38], [39]. Responding to these identified applications, the Society of Automotive Engineers [40] has defined over seventy vehicle Data Elements (e.g. heading, acceleration, headlight status and brake status) as being needed to support vehicular safety applications. Extending VITP to accommodate those data elements is straightforward.

VI. CONCLUSIONS

In this paper, we studied the problem of providing vehicle drivers with time-sensitive information about traffic conditions and roadside services. Such information can serve drivers who are interested in adjusting their route dynamically according to nearby traffic conditions and/or roadside services. Moreover, it can be fused with existing GPS navigation information, extending the functionality of state-of-the-art on-board navigation systems.

To address this problem, we introduced the *Vehicular Information Transfer Protocol* (VITP), a location-aware, application-layer communication protocol designed to support the establishment of distributed, ad-hoc, best-effort service infrastructures over Vehicular Ad-hoc Networks. We described the system architecture of the VITP-based infrastructure, the semantics of VITP interactions, and the specification of VITP messages. We introduced the concept of the *Vehicular Ad-Hoc Server*, which is established on-demand as an ad-hoc collection of VITP peers that collaborate to resolve incoming VITP requests. The Vehicular Information Transfer Protocol has the expressive power to define location-aware queries seeking and integrating information from vehicle sensors and roadside facilities, taking advantage of on-board GPS navigation systems. VITP is simple, stateless and lightweight; therefore, it can be easily implemented on embedded processors and resource-limited computing devices that are found on-board of modern vehicles.

To evaluate the performance of VITP, we conducted simulation studies of large-scale vehicular networks that are representative of realistic highway and city traffic conditions. We also investigated the effectiveness of the protocol and the accuracy of its results, when used for the resolution of traffic queries, the monitoring of traffic conditions, the distribution of alert requests, and the discovery of road-side services. Simulation results demonstrate the viability and effectiveness of VITP under different assumptions for vehicular traffic and wireless-network performance, and prove the feasibility of our approach in vehicular ad-hoc networks.

ACKNOWLEDGMENTS

This work has been supported in part by NSF under grants ANI-0121416 and CNS-0520123, and by the European Union under projects CoreGrid (contract number 004265) and EGEE-II (contract number INFSO-RI-031688). The authors would like to thank the anonymous reviewers for their helpful comments and suggestions on improving this paper.

REFERENCES

[1] W. Kellerer, "(Auto)Mobile Communication in a Heterogeneous and Converged World," *IEEE Personal Communications*, vol. 8, no. 6, pp. 41–47, December 2001.

[2] "General motors collaborative laboratory website," http://gm.web.cmu.edu/.

[3] Z. Chen, H. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad-hoc Networking and Computing*, Long Beach, CA, Oct 2001, pp. 247–250.

[4] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *IEEE Vehicular Technology Conference (VTC)*, Birmingham, AL, May 2002.

[5] C. Schwingenschloegl and T. Kosch, "Geocast enhancements for aodv in vehicular networks," *ACM Mobile Computing and Communications Review*, vol. 6, no. 3, July 2002.

[6] E. Welsh, P. Murphy, and P. Frantz, "A mobile testbed for gps-based its/ivc and ad hoc routing experimentation," in *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Honolulu, HI, Oct 2002.

[7] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "Vitp: an information transfer protocol for vehicular computing," in *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, 2005, pp. 30–39.

[8] J. Luo and J.-P. Hubaux, "A survey of inter-vehicle communication," School of Computer and Communication Sciences. EPFL, Lausanne, Switzerland, Tech. Rep. IC/2004/24, 2004.

[9] "U.S. Census Bureau. Topologically Integrated Geographic Encoding and Referencing system," http://www.census.gov/geo/www/tiger/ (accessed Nov 2006).

[10] T. Nadeem, S. Dashtinezhadd, C. Liao, and L. Iftode, "Trafficview: Traffic data dissemination using car-to-car communication," *ACM Sigmobile Mobile Computing and Communications Review, Special Issue on Mobile Data Management*, vol. 8, no. 3, pp. 6–19, July 2004.

[11] B. Karp and H. Kung, "Greedy perimeter stateless routing for wireless networks," in *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*. ACM, August 2000, pp. 243–254.

[12] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," in *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*. USENIX Association, May 2005.

[13] Y. Ko and N. Vaidya, "Location-aided routing in mobile ad hoc networks," in *ACM/IEEE Mobicom*, October 1998.

[14] J. Li, J. Janotti, D. D. Coutu, and R. M. D. Karger, "A scalable location service for geographic ad hoc routing," in *Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, August 2000.

[15] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, 2005.

[16] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftode, "EZCab: A Cab Booking Application Using Short-Range Wireless Communication," in *3rd IEEE PerCom 2005*, Kauai Island, Hawaii, USA, March 8–12, 2005.

[17] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: A Scalable Traffic Monitoring System," in *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, January 2004, pp. 13–26.

[18] T. Nadeem, P. Shankar, and L. Iftode, "A Comparative Study of Data Dissemination Models for VANETs," in *Proceedings of the 3rd ACM/IEEE Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS 2006)*, July 2006, pp. 1–10.

[19] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Context-aware Migratory Services in Ad Hoc Networks," to appear in IEEE Transactions on Mobile Computing.

[20] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks." *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[21] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles." *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[22] G. Xu, C. Borcea, and L. Iftode, "Trusted Application-Centric Ad-Hoc Networks," in *Proceedings of the Fourth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2007, to appear.

[23] Centre for Applied Informatics (ZAIK), Institute of Transport Research, German Aerospace Centre, "Sumo - simulation of urban mobility," http://sumo.sourceforge.net/ (last accessed January 2007).

[24] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility); an open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling (MESM2002)*. SCS European Publishing House, September 2002, pp. 183–187.

[25] "The network simulator ns-2," http://www.isi.edu/nsnam/ns/.

[26] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of Infocom 2003*, vol. 2. IEEE, 2003, pp. 1312– 1321.

[27] J. Y. L. Boudec and M. Vojnovic, "Perfect Simulation and Stationarity of a Class of Mobility Models," in *Proceedings of Infocom 2005*, vol. 4. IEEE, 2005, pp. 2743–2754.

[28] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. Commun.*, vol. 32, no. 3, pp. 246–257, 1984.

[29] "Frida: Free Vector-Geodata Osnabrück," http://frida.intevation.org/index.en.html (accessed Dec 2006).

[30] M. Wu, A. Friday, G. Blair, T. Sivaharan, P. Okanda, H. D. Limon, C.-F. Srensen, G. Biegel, and R. Meier, "Novel component middleware for building dependable sentient computing applications," in *ECOOP 2004, Workshop on Component-Oriented Approaches to Context-Aware Computing*, Oslo, Norway, June 2004.

[31] T. Sivaharan, G. Blair, A. Friday, M. Wu, H. Duran-Limon, P. Okanda, and C.-F. Srensen, "Cooperating sentient vehicles for next generation automobiles," in *MobiSys 2004 1st ACM Workshop on Applications of Mobile Embedded Systems (WAMES 2004)*, Boston, USA, June 6 2004.

[32] C. Julien and G. Roman, "Active coordination in ad hoc networks," in *6th International Conference on Coordination Models and Languages*, Pisa, Italy, February 2004, pp. 199–215.

[33] S. Dolev, S. Gilbert, N. A. Lynch, E. Schiller, A. A. Shvartsman, and J. Welch, "Virtual mobile nodes for mobile ad hoc networks," in *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, 2004, p. 385.

[34] R. Handorean, R. Sen, G. Hackmann, and G.-C. Roman, "Context aware session management for services in ad hoc networks," in *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, 2005, pp. 113–120.

[35] "Cartel project," http://cartel.csail.mit.edu.

[36] "Metrosense project," http://metrosense.cs.dartmouth.edu.

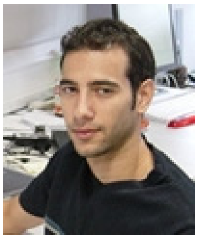
[37] "Urban sensing project," http://research.cens.ucla.edu/projects/2006/Systems/Urban Sensing.

[38] A. Carter, "The status of vehicle-to-vehicle communication as a means of improving crash prevention performance," NHTSA, Tech. Rep., 2005, http://www-nrd.nhtsa.dot.gov/pdf/nrd-01/esv/esv19/05-0264-W.pdf.

[39] "Vehicle safety communications project-final report," April 2006, http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html.

[40] DSRC Technical Committee, "Dedicated short range message set (DSRC) dictionary," Society of Automotive Engineers, Tech. Rep. J2735, December 2006.

Marios D. Dikaiakos is an associate professor of Computer Science at the University of Cyprus, Nicosia, Cyprus. He received his Ph.D. in Computer Science from Princeton University, USA in 1994. He also holds a Dipl.-Ing. degree in Electrical Engineering from the National Technical University of Athens, Greece and an M.A. in Computer Science from Princeton University. His research interests include network-centric computing with an emphasis on Grid computing, vehicular computing, and World-Wide Web techologies.



Andreas Florides received the Dipl. Ing. degree in Electrical and Computer Engineering from the National Technical University of Athens, Greece (2005) and an MSc in Data Communications, Networks and Distributed Systems from the University College London, UK (2006). He is working with the High Performance Computing Systems Laboratory (HPCL) at the Computer Science Department of the University of Cyprus as a research scientist since October 2006. His interests include Vehicular ad-hoc networks, protocols and sensor networks.

Tamer Nadeem is a research scientist at Siemens Corporate Research. He received his Ph.D. degree from University of Maryland in 2006. His research interests include wireless networks management, mobile ad hoc networks, vehicular networks, sensor networks, peer-to-peer systems, and pervasive computing. He is a member of IEEE and ACM. He is an elected member for the honor society Phi Kappa Phi and for the honor society Sigma Xi, and a member of the Association of Egyptian American Scholars.

Liviu Iftode is an associate professor of computer science at Rutgers University, New Jersey. He received his Ph.D. in Computer Science from Princeton University in 1998. His research interests include distributed systems, operating systems, vehicular networks, mobile and pervasive computing. He is a senior member of the IEEE Computer Society and a member of the ACM. Liviu Iftode is the vice-chair of IEEE Technical Committee on Operating Systems and a member of the editorial board of IEEE Pervasive Computing.