V-Radar: A Vehicular Traffic Query Protocol for Urban Environments

Nicholas Loulloudes, George Pallis, Marios D. Dikaiakos
Department of Computer Science, University of Cyprus
Nicosia, CY1678, Cyprus
{loulloudes.n, gpallis, mdd}@cs.ucy.ac.cy

Abstract—Automobile congestions have an adverse effect in modern societies, causing the loss of billions of dollars and man-hours every year throughout the world. In this era of global economic recession, drivers will require the necessary solutions and driving aids that facilitate the improvement of daily road transport and minimize unnecessary expenditure. In this work, we lay the groundwork for *V-Radar*, a query protocol for retrieving vehicular traffic information using V2V communications. The advantage of V-Radar over related works is its ability to monitor using location-dependent queries the prevailing traffic conditions in a number of road-paths from a vehicle's current location towards its final destination. We introduce its modular architecture and provide preliminary evaluation results showing significant improvements over a similar scheme.

I. Introduction

Nowdays, the commodization of low-cost, high quality electronic circuitry, has realized the near-omnipresence of sensory and communication technologies in our daily lives. The transportation and more specifically the automobile environment is no exception to this norm, with in-vehicle sensors recording real-time information such as location, speed, acceleration, breaking, etc. Thanks to advances in vehicular technology, a significant percentage of money and man-hour waste due to traffic congestion can be ultimately reduced by disseminating such sensory information to drivers and/or navigation systems of other vehicles via wireless ad hoc inter-vehicle communication (IVC). The proliferation of initiatives and consortia from academia and industry in the likes of [1], [2] confirm the positive impact that Vehicular Ad Hoc NETworks (VANET) will have on modern societies.

This article lays the groundwork for *V-Radar*, a vehicular traffic information query protocol for urban environments based on V2V communication. In contrast to other approaches in the literature that aim to obtain traffic information on *singular roads*, V-Radar enables the querying and acquisition of traffic information along a composite *road-path*, starting from a vehicle's current position towards its final destination. Specifically, V-Radar is able to query not only the initally selected road-path, but also a number of alternate paths that lead to the vehicle's destination. This allows the driver or the in-vehicle navigation system to establish a more broad and complete view of the traffic conditions that will be encountered further ahead. Such knowledge can be extremely valuable in the process of calculating a more optimal route to the destination in terms of travel time. Furthermore, this work proposes

a number of components for V-Radar that can work in tandem so as to maximize the number or road-paths monitored, whilst keeping wireless transmissions and bandwidth utilization at the minimum. Specifically the contributions of this work are:

- We define the problem of identifying, in the absense of an infrastructure navigation service, the set of road-paths among any two road intersections, that if followed will result in reduced travel times. To identify such road-paths we utilize location-dependant queries.
- We propose V-Radar, a traffic information query protocol for urban environments using V2V communications. The advantage of V-Radar over related works is its ability to monitor the prevailing traffic conditions in a number of road-paths from a vehicle's current location towards its final destination using location-dependant queries.
- Using realistic trace-driven simulation studies, we show how V-Radar even in its simplest form has a significant performance over existing vehicular traffic query methods available in the related literature.

II. Related Work

A number of research works have proposed the utilization of in-vehicle sensors and V2V communications as means of capturing, monitoring and disseminating the dynamic vehicular traffic conditions in an urban road environment. In SOTIS [3] and TrafficView [4] vehicles periodically broadcast their position and speed information in order to enable other vehicles to estimate the existence of traffic congestion. While TrafficView focuses on the monitoring the congestion of the road directly ahead, SOTIS extends the idea to both sides of the road. StreetSmart [5] pertains to the above concept but uses clustering and epidemic communication to disseminate traffic information. Simulative and field evaluations have shown that these proactive traffic information dissemination techniques work on small and sparse VANETs. However, this is unlikely to be the case in geographically larger and more dense VANETs, since the amount of traffic data that vehicles will collect and consequently broadcast will increase quadratically [6].

Gao et al [7] propose an adaptive query evaluation plan based on the structure of the underlying road topology. Specifically, traffic queries can be issued by a source vehicle towards individual target roads. The evaluation plan is a sub-tree of the road topology with its root being the location of the

source vehicle and leaf-nodes the query target road. Based on query construction rules known a priori by all vehicles, each vehicle residing on streets between the root and leaf-nodes, can autonomously decide whether it will participate in the query evaluation process and under which role. In addition, control messages are introduced to provide updates to vehicles on changes in the location of the query source.

However, the method proposed in [7] caters only for querying the traffic flow of one *single* target road. Consequently, if we are to apply this method in order to query the prevailing traffic conditions on a number of roads in a composite path towards the destination, the respective number of packets/queries need to be generated and transmitted. In turn, this imposes a significant overhead in the VANET, which subsequently leads to information loss due to packet collisions in the wireless channel. In addition, the frequent transmission of control packets provides an additional overhead to the already limited network capacity. Given that in a real urban environment a number of road-paths need to be monitored, it is evident that such an approach does not scale.
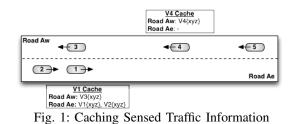
III. SYSTEM MODEL

A. Assumptions

Consider an urban vehicular environment where the driver and occupants of any vehicle would like to minimize the travel time en-route to their destination.

We assume that vehicles are able to communicate with each other (V2V) via wireless communication hardware that support the WAVE standard [8]. Each vehicle makes its presence known to neighboring vehicles by periodically broadcasting a beacon message of the form $(v_{uid}, pos, spd, hdg, r_{uid})$, where v_{uid} is the vehicle unique ID and *pos, spd, hdg* are respectively its current position, speed and heading. r_{uid} denotes the unique ID of the road the vehicle is currently on.

Information regarding the underlying road-network topology is provided to the vehicle through on-board preloaded digital maps, while its current geographic location is obtained via the Global Positioning System (GPS). Such maps are enriched with historical statistics that exhibit the traffic conditions of the road-network at different times of the day.

Fig. 1: Caching Sensed Traffic Information

Each vehicle encompasses a local cache where it temporarily maintains vehicular traffic information sensed through received beacons from the roads that have been traversed. For example, as shown on Figure 1, two new records are created in each vehicle's cache upon entering a new road. One record

is for Road segment A_e and one for Road segment A_w [1]. Whenever a beacon is received from a neighboring vehicle, the respective road records are updated based on the value of r_{uid} - vehicle's 1 cache contains entries for both Road A_e and A_w through the beacons of vehicles 3 and 2 respectively.

Obviously, the decision to follow the shortest road-path(s) in terms of geographic distance does not necessarily guarantee a reduction in the total commute time towards a specific destination, since the prevailing traffic conditions might dictate otherwise. For instance, if several roads in the shortest geographic road-path exhibit high vehicle density and low average speed, will eventually result in a higher total travel time than if following another road-path with sparse traffic.

Therefore, an IVC-enabled vehicle can provide its driver or the on-board navigation system with the necessary traffic information required to make the aforementioned decisions by issuing location-dependent traffic queries to other vehicles.
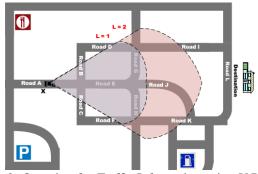

Fig. 2: Querying for Traffic Information using V-Radar

B. V-Radar Overview

The aim of V-Radar protocol is to support the generation, dissemination and resolution of such location-dependent traffic queries in a manner that resembles a forward-scan radar, similar in concept to the radars used in maritime or aviation activities to detect the presence of obstacles that might block the normal course of a vessel or an aircraft. In the context of V-Radar such obstacles are considered to be traffic build-ups or congestion, which as aforementioned can cause severe delays.

The concept of V-Radar is explained with the following example. We assume that vehicle X on Figure 2 is driven eastbound on *Road A* and wishes to follow the road-path to its destination Y with the minimum travel time. We also assume that X currently follows the shortest geographic path towards Y suggested by the on-board navigation system. However, X would also like to know the prevailing traffic conditions for the other available road-paths (with larger travel distance) to Y, in case a route change is required. Through its knowledge of the road network, the on-board computer of X can calculate and consequently rank the K possible road-paths in terms of the geographic distance from the next intersection to the intersection which is closest to Y.

---

[1] The subscript next to each road denotes traffic direction: "e" for eastbound, "w" for westbound.

Upon entering *Road A*, X issues *LookAhead (L)* traffic information queries at a selected rate (R) towards all the calculated K road-paths. We envision *LookAhead* queries as radar pulses that travel towards a specific direction and as soon as they hit on a surface (i.e. a moving or stationary object) they are reflected back towards the source. *Hence, LookAhead queries are propagated to a certain depth L in each of the identified road-paths and collect the traffic conditions of all the roads up to and including the specified depth.*

Therefore, if X would like to know the traffic conditions in all the road-paths up to 2 roads ahead of its current position, then the look ahead value will be set to $L = 2$ and the following traffic queries (Q) will be generated: $Q_1 : \{Roads\ B, D, I\}$, $Q_2 : \{Roads\ E, G, I\}$, $Q_3 : \{Roads\ E, J, K\}$, $Q_4 : \{Roads\ E, H, K\}$ and $Q_5 : \{Roads\ C, E, K\}$ [2].

Consequently, each query is propagated in a multi-hop fashion to each individual road in a given path, where the required traffic information is retrieved either with on-the-fly cooperation of other vehicles on location (i.e. the concept of VAHS in [9]) or from vehicles' cache. Upon reaching depth L and retrieving the required traffic information query replies are routed back to X. This query cycle process iterates until X reaches its destination Y.

C. Problem Formulation

A road-network can be considered as a directed graph $G = (V, A)$, where intersections or end-points (dead-ends) correspond to the set of vertices's $V = \{v_i\}$ and roads to the set of arcs $A = \{a_{ij}\}$. Two nodes v_i and v_j can communicate directly with each other, if they are connected by an arc a_{ij} and no other intersection or end-point exists in-between them. A path P_{ij} is a unique, alternating sequence of connected nodes and arcs in G that starts from v_i and ends at v_j. $D(P_{i,j})$ is the length of path P_{ij}. Consequently, P denotes a set of paths that can be defined in $G = (V, A)$ from v_i to v_j. To this end, a road-path is considered as the list of roads - and by definition, intersections - that a vehicle has to traverse in order to travel from one intersection to another intersection. Taking under consideration the historical traffic statistics, each arc is attributed with the average speed $\bar{u}$ and average vehicle density $\bar{\rho}$ of the respective road.

Let $t_{P_{ij}}$ be the time it takes to travel from node v_i to node v_j in any path P_{ij}. Since travel times are influenced from static parameters such as the road length but also by the dynamic traffic conditions (average speed, vehicle density, traffic light queues, etc.) vehicles would like to identify the set of road-paths from node v_i to node v_j $S_{ij} = \{P_{ij}^1, P_{ij}^2, P_{ij}^3, \ldots, P_{ij}^K\} \subseteq P$ towards their destination such that: $t_{P_{ij}^n} \le t_{P_{ij}^{n+1}}$, for any $n \in \{1, \ldots, K - 1\}$.

A naive assumption would be that to discover these road-paths and construct S_{ij}, a vehicle should firstly calculate *all*

---

[2]Note: Although overlapping road-paths are possible and will be identified (i.e $\{B, D, I\}$ and $\{B, D, G\}$), in the example we deal only with distinct, non-overlapping paths.

the possible geographic shortest paths and subsequently generate and transmit the necessary traffic queries towards them so as to acquire the necessary information for the calculation of $t_{P_{ij}}$. However, due to various constraints such as query-reply delay thresholds, size of the road topology, vehicle density and the VANET connectivity status, querying the traffic conditions of: i) all paths $P_{ij} \in P$ and ii) consequently all roads (whole length D) of any individual path P_{ij}, is by no means scalable and with very high probability will lead to inaccurate or out-of-time results. Therefore vehicles should be able to estimate the number K of paths to be queried and additionally up to what depth L of each P_{ij}, *LookAhead* queries should be propagated in order to meet constraints such as the above.

Moreover, due to the high dynamics of vehicular environment where the conditions on any given path can change abruptly, one may assume that a high query generation rate R would be required to capture traffic accurately. However, in an urban setting where a large number of vehicles are continuously competing for the VANET resources, it is crucial to be prudent in the use of the wireless channel and thereby refraining from transmissions in a selfish manner.

D. Objective

The objective of the V-Radar query protocol is thus to provide the necessary mechanisms that enable: *i) the sustainability of an acceptable traffic query-reply delivery rate and ensuring any delay thresholds are met, ii) the maximization of the number of alternate road-paths $(max(K))$ and roads $(max(L))$ to be monitored, whilst iii) minimizing wireless transmissions and bandwidth utilization.*
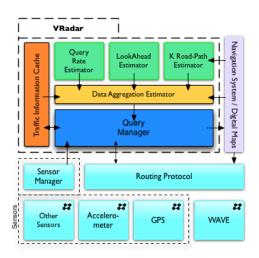

Fig. 3: V-Radar Architecture

IV. ARCHITECTURE

This section introduces the V-Radar architecture. Its modular design is such that it allows the use of various components that can collectively realize the V-Radar objective. As Figure 3 depicts, V-Radar runs on the application layer of the vehicle's on-board computer therefore it can utilize the underlying routing and transport protocols. Below the function of its individual modules is explained:

Road-Path Span (K) Estimator Module

It is evident that monitoring the traffic conditions of all available paths towards the destination does not scale in the real-world. Besides the unknown number of road-paths which might be available, the resulting overhead that will be imposed in the VANET from such an attempt will be forbidding. The Road-Path estimator function is to identify the critical value K, that is which and how many of the available road-paths towards the destination will be queried. It interfaces with the Navigation system in order to be aware of the road topology and various driver preferences (e.g. maximal deviation in terms of geographic distance from the initially selected road-path, willingness to use road-paths with bridges or tolls, etc). The K estimator can additionally utilize information items from the Traffic Information Cache (see below) to assist in the estimation process. For instance, knowing from past queries that the traffic conditions of a particular road-path exhibit variability below a certain threshold, then that road-path can be queried on a less frequent basis.

LookAhead (L) Estimator Module

The L value and specifically how deep a road-path will be queried has a crucial role on the correctness of the information that V-Radar will provide to the driver. Querying too shallow might lead to a horizon effect where a congestion further down the path might not be identified. On the other hand, querying too deep in the path introduces the risk of a packet loss due to network fragmentation or the violation of query-reply delay thresholds.

Query Rate (R) Estimator Module

To maintain an update view of the traffic conditions in the identified K road-paths, new query messages need to be dispatched periodically. Nevertheless, the new query generation rate depends on the existence of several constraints that must be taken under consideration. For instance, vehicle speed and/or current road length impose a time constraint on when the next junction will be reached. In a higher vehicle speed and small road length setting, queries need to be generated at an increased rate. On the other hand, consideration needs to be given on the number of neighboring vehicles that compete for the wireless medium and which can influence the number of message lost due to possible packet collisions.

Data Aggregation Module

It is possible that several of the identified K road-paths will be overlapping. In such cases "duplicated" query messages will be generated, causing unnesecary utilization of the wireless medium. The function of the Data Aggregation module is to identify such situations and provide the necessary mechanisms that facilitate the aggregation of query information in a single message such that duplicates are avoided.

Traffic Information Cache (TIC)

The cache module is used to store traffic information (average speed and vehicle density) about various roads. Such information is obtained as explained in Section III or through the contents of a received query-reply. Each cached item is a 4-value tuple *[id, timestamp, data-type, value]*. Specifically, the index value $id = r_{uid}$, $timestamp$ determines the point in time where the provided information becomes stale, $datatype$ is the type of cached information (speed, density, etc) and $value$ is the actual value of the sensed or received information. A Cache Replacement Module (CRM) is used to define and enforce the necessary policies (e.g. LRU, MRU, etc) that dictate the eviction of stale items and maintaining the cache freshness.

Query Manager

The Query manager orchestrates the operation of the V-Radar protocol components on each vehicle. Specifically, it collects the values calculated by the K, L, R estimators and is responsible to construct the respective V-Radar query messages for each of the identified road-paths. Furthermore, it is responsible for resolving incoming traffic queries.

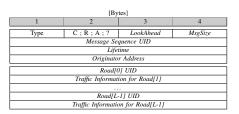| [Bytes]			
1	2	3	4
Type	C ; R ; A ; ?	*LookAhead*	*MsgSize*
Message Sequence UID			
Lifetime			
Originator Address			
Road[0] UID			
Traffic Information for Road[1]			
...			
Road[L-1] UID			
Traffic Information for Road[L-1]			

TABLE I: V-Radar Query Message Format

As Table I illustrates, the V-Radar query message consists of a 16-byte fixed header and a variable-sized payload (data) section. The header fields have the following functions:

- *Type:* Indicates what type of traffic information the query message contains. For instance, a value of 1 can be used for average vehicle speed, while 2 for vehicle density.
- *Boolean Bit Flags:* The "C" flag indicates whether cached traffic data can be appended to the reply message. The "R" flag indicates whether the message is a query-reply. A query-reply message is considered to be the message which includes traffic information for at least one road in the road-path being queried. The "A" flag indicates whether a receiving vehicle is allowed to append traffic information to the message. The remaining bits are reserved for future use.
- *LookAhead:* Indicates at what depth in the road-path being queried the message currently is.
- *MsgSize:* A count of the total number of bytes contained in header and data sections. As the header length is a fixed size, this field effectively tracks the length of the variable-sized payload. In effect it indirectly reports the number of roads in a path that the message will visit.
- *Sequence ID:* A sequence number uniquely identifying the particular query message when taken in conjunction with the originating node's address.
- *Lifetime:* The time in milliseconds for which vehicles receiving the query message consider it to be valid and are allowed to forward it or append traffic information to its payload section.
- *Originator address:* The address of the node from which the query message originates.

The payload section of the V-Radar query message contains (L), 16-byte fields. The first 4-bytes of each pair are used for

storing r_{uid} of the roads in a path to be queried. The sequence of r_{uid} in the message is also indicative of the order that each road in the path will be traversed by the query. The remaining 4-bytes are used for storing the queried traffic information for the respective road.
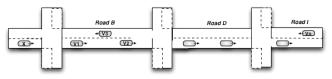

Fig. 4: Query Resolution Illustration

A. Query Resolution

Figure 4 depicts how a V-Radar query is generated and resolved in the context of the example presented earlier in Section III. Vehicle X selects to query and obtain the average vehicle speed on road-path P_1 : $\{Roads\ B_e, D_e, I_e\}$. V constructs a new query message Q (as per Table I), inserts in the payload the UID's of the roads to be queried and sets flags $C = 1$ and $A = 1$. Once the V-Radar Query manager in V_1 receives Q, it parses the message and consults the Navigation System whether the vehicle's current position qualifies participation in the query resolution. Since V_1 is the first vehicle on $RoadB_e$ that can join the query resolution, the manager updates the LookAhead field in the header to reflect the current road-path depth ($L = 1$). Consequently, it piggybacks its current speed in the appropriate payload field for $RoadB_e$ and passes the message on to any available neighbor. If Q is received by a vehicle on $RoadB_w$ (V_3), the query manager aggregates traffic information for $RoadB_e$ stored in the TIC and appends them to the query. It then sets flag $A = 0$ to avoid duplicate information from other vehicles further down the road (e.g. V_2). Once Q approaches the end of $RoadB_e$, the receiving vehicle checks if it is allowed to append its own speed information, sets flag $R = 1$ to indicate that the query contains traffic information for at least one road, sets flag $A = 1$ and forwards the message to $RoadD_e$ where the above process is repeated. In the case that Q is received by a vehicle such as V_n, where TIC information indicate that no vehicle was sensed in the opposite traffic direction (i.e average speed for $RoadI_e = 0$), the query manager may decide to forward Q back to V based on various rules [3]. Finally, when Q traverses the last road in P_1 and records the necessary information, it is routed back to vehicle X.

V. EVALUATION

A. Simulation Setup

This section presents how V-Radar in its simplest form performs against other related works. For the purposes of the evaluation, V-Radar was implemented as an application module under the ns-3 [10] network simulation framework. Since we are still researching on the techniques that will eventually be used for the estimators modules presented in

---

[3]Due to space limitations here we leave the formal definition of such rules in a future work.

Section IV, in the following evaluation we are simulating their existence by using different values for K, L and R. We compare V-Radar against the road network based adaptive query evaluation (RNBAQ) method proposed in [7].

Realistic urban mobility traces were utilized from an improved version [11] of the TAPAS-Cologne [12] dataset. We extract the mobility traces of all vehicles moving within a 4Km X 3Km rectangular area surrounding the Cologne city center.

200 randomly selected vehicles issue queries for 800s. At each intersection each vehicle calculates the Top-K shortest paths towards its destination with $K = 3$. Each road-path is monitored up to a depth of 7. Therefore, depending on the experiment, queries can take a value $1 \leq L \leq 7$. For V-Radar, simple information caching is used. A vehicle can provide a reply to a received message if it knows traffic information for one or more roads that will be queried. Cached items TTL is set to 128sec. For routing queries to destination roads, we utilize VADD [13] as the underlying routing protocol. For RNBAQ, we extend VADD to facilitate the broadcast of control messages used for location change notification. Each vehicle broadcasts a HELLO beacon at a rate of 10Hz. Data-rate was set to 3Mbit/s, and all the PHY and MAC properties conform to IEEE 802.11p [8].

Vehicle Transmission Range	300m (802.11p)
Simulation Time	1000s (200s warm-up)
Routing Protocol	VADD (H-VADD)
Beacon Generation Rate	10Hz with some jitter
Query Generation Rate (pkt/sec)	0.1 - 1.0
Query TTL	64 hops
Cache Items TTL	128 sec
Query Sources	200
K-Paths	3

TABLE II: Network Simulation setup parameters

Note that the results depicted in this section, account for the average values of each of the studied metrics, calculated over 20 runs for each simulation scenario, with different random number seeds. Finally, Table II presents all the parameters that were utilized in the simulation.

B. Results

The following metrics are used in the evaluation:

- *Packet Delivery Ratio:* the ratio of queries successfully delivered back to the source vehicle to those generated by the source vehicle.
- *Network Overhead:* the total number of KBytes transmitted. The total number of KBytes is inclusive of any control messages and the underlying routing protocol headers.
- *Accuracy:* denotes whether the retrieved traffic information is close to the real value. Real values are obtained from traffic statistics computed by SUMO.

As depicted by Figure 5a the network overhead imposed on the VANET by the V-Radar query protocol is substantially lower than the in the case of RNBAQ. While V-Radar uses a single packet for querying the traffic conditions of all the roads in a path, RNBAQ is required to generate and inject in the network one packet per road. In addition, due to its single packet/query design, V-Radar keeps the network

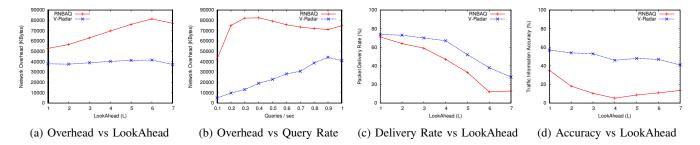| (a) Overhead vs LookAhead | (b) Overhead vs Query Rate | (c) Delivery Rate vs LookAhead | (d) Accuracy vs LookAhead |

Fig. 5: Evaluation Results

overhead lower than RNBAQ as the query generation rate increases (Figure 5b). Nevertheless, the query generation rate depends on various constraints such as query-replay delay thresholds and minimum information accuracy levels. It is crucial, therefore to investigate intelligent techniques that can be utilized the V-Radar for estimating the query rate R in order to provide even better utilization of the wireless medium.

By examining Figure 5c we can observe that as the number of generated queries increase (due to the LookAhead parameter increase), the PDR of RNBAQ drops faster than in the case of V-Radar. This behavior can be attributed to packet collisions that take place at the busy wireless channel and cause several traffic query messages to be lost. Unavoidably, V-Radar also suffers from packet collisions however the inherently smaller number of generated messages mitigates the above side-effects.

Moreover, the improved PDR that V-Radar exhibits over RNBAQ is attributed to the existence of the Traffic Information Cache. There is a high probability for a query message to encounter an intermediary vehicle on a monitored road-path that can provide traffic information for one or more roads from its local cache. In turn, this overcomes the need for a query to traverse the whole road-path up to L in order to retrieve the required traffic information. Therefore, it enables the query return back to the source vehicle prior the message TTL expiration.

Although the retrieved traffic information accuracy is better than in the case of RNBAQ, Figure 5d indicates that it remains quite low and becomes worst as the LookAhead value increases. However this is expected since it takes more time to reach a road further away and return the result back to the source vehicle. Subsequently, the returned results do no reflect correctly the real situation in the road network. Here, clever traffic information caching techniques can be utilized in the vehicle cache which will allow a query to be answered faster and with better accuracy. Therefore it is a clear indication that the design and implementation of an adaptive CRM for V-Radar is important.

VI. CONCLUSIONS

This work introduces for *V-Radar*, a query protocol for retrieving vehicular traffic information using V2V communications. The advantage of V-Radar is its ability to monitor using location-dependant queries the traffic conditions in a number of road-paths from a vehicle's current location towards its final destination. Preliminary results have shown significant improvements over related schemes.

For future work, our plan is to investigate and evaluate adaptive techniques that will intelligently estimate key protocol parameters such as the number of monitored road-paths (K), the LookAhead (L) value and the query generation rate (R).

REFERENCES

[1] "Cooperative Vehicle-Infrastructure Systems (CVIS)," http://www.cvisproject.org/ (accessed Sept. 2012).
[2] "DRIVE C2X - Accelerate Cooperative Mobility," http://www.drive-c2x.eu/project (accessed Sept. 2012).
[3] L. Wischoff, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS - A Self-Organizing Traffic Information System," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4, April 2003, pp. 2442 – 2446.
[4] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: Traffic Data Dissemination Using Car-to-Car Communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 3, pp. 6–19, 2004.
[5] S. Dornbush and A. Joshi, "StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, April 2007, pp. 11 –15.
[6] C. Lochert, B. Scheuermann, and M. Mauve, "Probabilistic Aggregation for Data Dissemination in VANETs," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, ser. VANET '07. New York, NY, USA: ACM, 2007, pp. 1–8.
[7] J. Gao, J. Han, D. Yang, and T. Wang, "Road Network Based Adaptive Query Evaluation in VANET," in *Mobile Data Management, 2008. MDM '08. 9th International Conference on*, April 2008, pp. 49 –56.
[8] IEEE, "Draft Amendment to Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications-Amendment 7: Wireless Access in Vehicular Environment," *P802.11p/D3.0*, 2007.
[9] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 8, pp. 1590 –1602, October 2007.
[10] "The Network Simulator (ns-3)," http://www.isi.edu/nsnam/ns/.
[11] S. Uppoor and M. Fiore, "Large-Scale Urban Vehicular Mobility for Networking Research," in *Vehicular Networking Conference (VNC), 2011 IEEE*, November 2011, p. 62.
[12] "The TAPAS-Cologne Project," http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Data/Scenarios/TAPASCologne (accessed Sept. 2012).
[13] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–12.