Delay-Bounded Data Gathering in Urban Vehicular Sensor Networks

Claudio E. Palazzi[a], Fabio Pezzoni[b], Pedro M. Ruiz[c]

[a]*Dipartimento di Matematica Pura e Applicata, Università degli Studi di Padova, via Trieste, 63 - 35131 Padova, Italy*
[b]*CNR-IIT, via G. Moruzzi, 1 - 56124 Pisa, Italy*
[c]*Departamento de Ingeniería y las Comunicaciones, Universidad de Murcia, Campus de Espinardo - 30100 Murcia, Spain*

Abstract

Vehicular sensor networks are an emerging network paradigm, suitable for various applications in vehicular environment making use of vehicles' sensors as data sources and Inter-Vehicle Communication systems for the transmissions. We present a solution, based on vehicular sensor networks, for gathering data from a certain geographic area while satisfying with a specific delay bound. The method leverages the time interval during which the query is active in order to make the gathering process efficient, properly alternating data muling and multi-hop forwarding strategies like in delay-bounded routing protocols. Simulations show that our proposed solution succeeds in performing efficient data gathering outperforming other solutions.

Keywords:
Delay-bounded, Data Gathering, Vehicular Sensor Networks

Email addresses: `cpalazzi@math.unipd.it` (Claudio E. Palazzi), `fabio.pezzoni@iit.cnr.it` (Fabio Pezzoni), `pedrom@um.es` (Pedro M. Ruiz)

1. Introduction

Vehicular data communication has gained a lot for momentum within the automotive industry and the research community. In a few years from now, cars will be capable of communicating both with other cars and through roadside access points. Vehicular communications will provide additional safety and useful on-board services for drivers and passengers. Technologies and standards facilitating this revolution will be soon a real commodity: at the time of writing, the IEEE 802.11p radio technology, specifically designed vehicular communications, has been recently approved [1].

Inter-Vehicle Communication (IVC) systems rely on direct communication between vehicles, forming *Vehicular Ad-Hoc Networks* (VANETs), to satisfy the communication needs of a large class of applications including road and vehicle safety [2, 3], traffic coordination, infotainment [4], information gathering and dissemination [5], etc.

Some applications [6] need to handle data obtained by sensors which are mounted on vehicles' on-board computers and on roadside infrastructures. In this case resulting networks are called *Vehicular Sensor Networks* (VSNs). Unlike traditional sensor networks, VSNs do not have strong energy, storage and processing constraints; VSNs have an extreme mobility and they work in a scenario where the topology may often get partitioned. For these reasons traditional sensor network approaches, like Direct Diffusion [7], are unfeasible for vehicular applications.

In this work we focus on data gathering based on VSNs in an urban scenario. These applications can be used in order to obtain different information. For example, municipalities or other entities can be interested in gathering data about traffic conditions, air pollution, environmental noise. Information is obtained by the vehicles that have proper sensors and communication antennas on board; they can be public transportation vehicles, cabs, police cars, private cars, etc.

In our case study a fixed *base station* (BS) creates a message containing the details of the query. This message is propagated within a specified area of interest by a geocast protocol. Vehicles inside the target area that receive the query message gather requested data with their sensors and bring them back toward the querier BS. The collection process cannot last forever. Thus, we consider that the query includes a specified time interval after which the process terminates. The typical scenario is represented in Fig. 1.

In this work we focus on designing a solution being able to harvest data from the region of interest satisfying specified time constraints, while maintaining a low level of channel utilization. Our solution has to be integrated with a time-stable geocast protocol for the query propagation.

We propose a novel protocol called *Delay-Bounded Vehicular Data Gathering* (DB-VDG). The main novelty in our approach is the exploitation of the time interval during which the data gathering process is active. Properly managing available time it is possible to reduce significantly the bandwidth consumption and the occupation of the shared channel.

Indeed, DB-VDG uses the *delay-bounded* paradigm opting for either forwarding the data immediately or carrying them in memory while the vehicle moves, such as delay-bounded routing protocols [8]. This behavior aims at reducing forwarded messages and at aggregating data from different sources in a single node so as to include multiple data in a single message.

In order to improve the efficiency of communications, DB-VDG makes use of an embedded propagation model thanks to which it can infer the probability of successfully delivering a message to a neighbor based on its distance. According to the delivery probability, the protocol chooses whether to dispatch a message or wait for a better opportunity.

The rest of the paper is organized as follows. Previous studies are summarized in Section 2. Assumptions and preliminary definitions are stated in Section 3. The detailed description of DB-VDG protocol is in Section 4. In Section 5 we describe our simulation scenario and performance metrics. We analyze the performance of the proposed solution in Section 6. Finally, we summarize our main conclusions in Section 7.

2. Related Work

Our scenario has to be addressed in two steps. First, a query is generated by the BS and disseminated in the area of interest. Then, sensor data have to be collected and propagated back toward the BS before the deadline expiration while taking care of minimizing the number of transmissions. Related work about these two aspects is presented in the following two subsections.

2.1. Query Propagation

For the diffusion of the query message in the region of interest we need a time-stable geocast protocol which can deal with VANETs. Geocast protocols are basically a broadcasting solution in which the propagation on the message is restricted in a defined geographic area. We outline two of the most reliable and efficient broadcast protocols suitable for VANET that have been proposed in scientific literature: the PIVCA architecture [9] and the AckPBSM protocol [10].

PIVCA broadcasting solution uses an efficient priority scheme to choose the next-hop forwarder of a broadcast message based on the distance from the previous sender and on the expected transmission range. In this way, redundant transmissions are reduced. In order to compute the aforementioned priority, PIVCA provides a dynamic estimation of the transmission range that represents a fundamental feature in a highly dynamic scenario [9]. Extending PIVCA to scenarios with asymmetric wireless links, [11] proposes to optimize the broadcasting process by utilizing the node with ther farthest retransmission span as forwarder.

Acknowledgment PBSM (AckPBSM) is a fully-distributed adaptive algorithm for broadcasting that, unlike PIVCA, is suitable for different vehicular scenarios and traffic conditions, including urban contexts. Each vehicle decides by itself whether to forward a received broadcast message or not. The technique used by AckPBSM in order to reduce redundant transmissions is based on connected dominating sets (CDS). When a broadcast message is received, it is not immediately forwarded. Instead the node sets up a waiting timeout and monitors its neighborhood. When the timeout expires, the message is retransmitted only if the node still has uncovered neighbors [10].

The main contribution of this paper is the sensor data gathering protocol, whose literature will be reviewed in the next subsection. Yet, for the query propagation aspect we developed a simple solution, described in Section 3.4, which takes into account some aspects of PIVCA and AckPBSM and adapts them to the considered urban vehicular network scenario.

2.2. Sensor Data Gathering

A survey on urban vehicular sensing platforms is given in [6], however many of reported works are solutions to specific applications and cannot be directly applied to our case study.

Considering sensor data gathering applications in VSNs, we have to mention *CarTel*, which is a mobile sensor computing system designed to collect, process, deliver and visualize data from sensors located on mobile units such as vehicles. It provides a simple query-oriented programming interface to handle large amounts of heterogeneous data from sensors and intermittent connectivity. Each node gathers and processes data before delivering them to a central portal, where the data is stored in a database. The project focuses on the system multi-layer architecture, the definition of a continuous query model and the protocols stack. However it does not define an efficient data routing method but, in the implementation, it implements only static opportunistic routing or flooding [12].

MobEyes is a middleware designed for proactive urban monitoring, that exploits node mobility to opportunistically diffuse sensed data summaries among neighbor vehicles and to create an highly distributed index to query sensor data. Despite its proactive vocation, MobEyes also allows on-demand data gathering called *on-demand harvesting* (ODH), in which a sink node queries all vehicles in the area of interest disseminating query messages and responses opportunistically in a Delay Tolerant Network style [13].

MobEyes ODH deals with the same problem treated in our work and can be considered the state of the art approach in vehicular sensor networks. For this reason we have consider it in our experimental evaluation. With this approach, vehicles that received the query reply forwarding immediately the response in the direction of the querier using opportunistic connections. Intermediate hops forward the message as soon as a connection with a node closer to the querier is available. Yet, no optimization is made to reduce the amount of generated data transmission.

Differently from MobEyes, our approach assumes that the gathering process can last for a specified time interval, therefore this period could be exploited to reduce the amount of forwarded messages while still guaranteeing the data delivery before the expiration time. A similar approach is used in delay-bounded routing protocol.

Delay-bounded routing protocols provide a routing scheme that satisfies user-defined delay requirements while maintaining a low level of channel utilization. On the contrary, other opportunistic routing protocols like VADD [14] aim at minimizing the transmission delay through immediate forwarding of any message received, whenever possible.

Delay-bounded Greedy Forwarding (D-Greedy) protocol, defined by [8], evaluates the current vehicle speed and the bounded delay-time to carefully opt between the *data muling* and *multi-hop forwarding* strategies to minimize communication overhead while delivering data to a static access point, satisfying with the delay constraints imposed by the application. D-Greedy chooses the shortest path to the destination AP using information from the city map. Then, it allocates the constrained delay-time to each street within the shortest path according to the length of streets. If packets can be delivered under the constrained delay-time in a street, data muling strategy is utilized, therefore packets are carried by a vehicle and forwarded at the vehicle's speed to the destination BS. Multi-hop forwarding strategy is applied if packets cannot be delivered within the constrained delay-time.

D-Greedy protocol is unicast, it has hence been developed with the goal of dispatching messages from a source to a destination. On the contrary we need a data gathering protocol that gathers messages from multiples sources toward an unique destination within a bounded time period. In order to success in designing an efficient delay-bounded data gathering protocol, we chose to use the alternation of data muling and multi-hop forwarding strategies like in D-Greedy.

3. Preliminaries

The goal of our work is to develop an efficient strategy to gather data from a certain area of interest in an urban scenario within a certain time interval. By efficient we mean that the data gathering process has to be able to collect a desirable amount of data, while limiting the bandwidth used for communication among nodes.

The collection process starts when a BS creates a query message (QUERY) that is diffused in the target area by a time-stable geocast protocol. Query message contains a unique ID, the type of required data, the position and the physical address of the BS, the creation and the expiration times of the query and the definition of the region of interest.

If, during the query lifetime, a vehicle transits within the area of interest, even for just a short time, it should receive the query message and take part in the data gathering process. The collection process has to be completed within the specified time interval. Data delivered after the deadline are ignored.

Before describing in detail DB-VDG protocols, in following subsection we state some important assumptions for our work and we present three mechanisms used in our solution: the beaconing scheme, the delivery probability estimator and the query propagation.

3.1. Assumptions

We assume that both the BS and vehicles are able to communicate with each other using IVC systems based on a standardized wireless protocol of the IEEE 802.11 family (e.g. IEEE 802.11p if available [1]). Each vehicle is equipped with a specific on-board antenna and a network device to communicate.

Data is to be sensed from a particular area included in the request from the BS. The position, form and size of this area can be chosen arbitrarily but the BS must be located within the area. In order to make it easier we assume that the region of interest is a circle, centered on the BS with a variable radius.

In order to know their geographical position, and to establish whether they are within or outward a data collection area, we assume that vehicles have a GPS receiver and a navigation system with city maps.

3.2. Beaconing Scheme

DB-VDG protocol requires a beaconing scheme so that each node could obtain knowledge about neighbors. Since geocast protocol for the query propagation (described in Section 3.4) exploits beacon messages too, the beaconing scheme can be shared between the two protocols.

Beacon messages (HELLOs) are broadcast periodically by each node, both vehicles and BSs. HELLO messages must include the geographic coordinates $\boldsymbol{x}$ of the sender and the time t when the message is generated. Nodes use these messages, to keep an updated *neighbor-table* containing the physical addresses and the positions of their neighbors.

In order to succeed in efficient data gathering, HELLO messages have to include an *aggregation level*, that is the amount of data that the node is currently carrying relative, as we will explain in detail in Section 4. Additionally, query propagation process may need other fields in HELLO. For example, in the solution proposed in Section 3.4, a query ID field is required.

When an entry in the neighbor-table, relating to a generic neighbor E is updated, the scheme estimates the speed vector $\boldsymbol{s_E}$ of the neighbor through (1) where $\boldsymbol{x_E}$ and t_E are the speed and the generation time included in the last received HELLO and $\overline{\boldsymbol{x_E}}$ and $\overline{t_E}$ are the previous values in memory.

$$\boldsymbol{s_E} = \frac{\boldsymbol{x_E} - \overline{\boldsymbol{x_E}}}{t_E - \overline{t_E}}. \tag{1}$$

Relying on $\boldsymbol{s_E}$ and the point of time t_E when the last HELLO was received, the scheme can predict the current position $\boldsymbol{x_E^{now}}$ at time t^{now} through (2).

$$\boldsymbol{x_E^{now}} = \boldsymbol{x_E} + \boldsymbol{s_E} \cdot (t^{now} - t_E) \tag{2}$$

3.3. Delivery Probability Estimator

Proposed protocol tries to avoid unreliable packet transmissions to save bandwidth. In order to success in it, DB-VDG relies on the *delivery probability estimator* (DPE) which, given the positions of any two vehicles A and B, estimates the delivery probability P_{AB} of successfully delivering a message between them.

DPE is based on an embedded propagation model which can be defined basing on experimental tests which can reveal what is, on average, the maximum transmission range and what is the delivery probability at various distances.

The signal propagation loss model used in our simulations and the derived model used by DPE are defined in Section 5.2.

3.4. Query Propagation

Proposed solution for data gathering needs to integrate DB-VDG protocol with a time-stable geocast protocol which deals with the diffusion of the query message from the querier BS within the region of interest. There some efficient time-stable broadcast solutions in the literature such as [15]. Other good alternatives can also be derived by adapting broadcast solutions for VANET such as PIVCA [9] and AckPBSM [10].

In our experiments we have implemented a simple solution for query propagation, which is outlined at a high-level below. We needed it just to disseminate a query before testing the the sensor data gathering protocols, which are the real contribution of this paper. Yet, nothing impedes in future to adapt one of the mentioned protocols or to devise a new one in order to optimize even the query propagation aspect.

Our approach for query propagation is schematized in Fig. 2 and it leverages on the use of beacon messages and of the DPE mechanism. In order to simplify the exposition we assume that vehicles can handle only one query propagation at a time.

If a vehicle is not managing any query, it is called *query free* node. When a query free node receives a QUERY message, it stores the query only after checking that it is inside the area defined in the message. Storing the query, a vehicle becomes a *query owner* node and it has to cooperate with other nodes in order to diffuse the query within the region, before it expires. When a node becomes query owner it has to notify this fact to its neighbors, therefore it immediately broadcasts a HELLO message which contains the ID of the query it is handling.

A query owner node schedules the dispatches of a QUERY message only in case it senses the presence of a query free node in its neighborhood and the probability to successfully dispatch the message to it, estimated through DPE, is high (above a certain threshold). The sensing of query free nodes is possible because they report their status in their beacon messages.

Queries are not immediately dispatched in order to avoid multiple transmissions from different nodes. On the contrary the dispatch is scheduled with a delay time inversely proportional to the distance with the sensed query free node.

If, during the waiting time, a QUERY transmission is overheard, the node estimates through DPE the probability that the overheard message reaches the target query free node. If the probability is high, it increase the wait time. This is in order to allow the target node to process the QUERY and to notify the reception with a HELLO message which has the effect of abort all the scheduled query dispatch. On the contrary, if the probability that the overheard QUERY message reaches the target query free node is low, the dispatch timer is not altered.

When the dispatch timer expires the QUERY message is broadcast and it should reach one or more query free nodes while query owners abort scheduled dispatches.

4. DB-VDG Protocol

The crucial task of collecting the data and carrying them toward the BS is managed by our novel *Delay-Bounded Vehicular Data Gathering* (DB-VDG) protocol. In order to succeed in efficient data gathering, DB-VDG takes advantage of the opportunistic data aggregation, the delivery probability estimation and of the carry-and-forward paradigm.

When a vehicle receives a query, it reads requested data via its sensors and stores them in its memory. Data are forwarded toward the BS though multiple hops; hence, each node can receive data from other vehicles during the data gathering period. If this happens, the receiver saves new data in its memory and, if necessary, aggregates them with data already stored. The amount of data from different sources carried by a vehicle represents the value of the *aggregation level* included in HELLO messages. If a vehicle is not carrying data its aggregation level is zero. When the aggregation level changes the node has to notify this to its neighbors though a new HELLO message.

When a node forwards data to another node, it dispatches a DATA message that includes all the stored data. In order to avoid data loss, a simple acknowledgment scheme is provided. When a node receives a DATA message, it immediately replies with an ACK that contains the sequence number of the received message. If the data sender receives the ACK, it can delete data from its memory. If the ACK timer expires the sender can make other attempts to dispatch data to the same node up to a maximum number of attempts.

Knowing the expiration time of the query, vehicles can decide whether to retain the data they are carrying (*data muling*) or to forward the data (*multi-hop forwarding*) depending upon the remaining time. Carrying the data is convenient because it can reduce bandwidth consumption while bringing the data closer to the requesting BS. Moreover when a vehicle is carrying data, it can receive and aggregate data from other vehicles.

DB-VDG incorporates a *strategy selection* component that handles the selection between the data muling and the multi-hop forwarding strategies. The goal is to manage the query life-time in order to improve the aggregation of data and to reduce the number of messages forwarded.

Another important component of DB-VDG is the *next-hop selection algorithm*. It is invoked when the multi-hop forwarding strategy is adopted for the selection of the next-hop node among all the neighbors closer to the BS than the current node. The algorithm takes into account information about delivery probability and the data aggregation level of each neighbor. The goal is to select a node with enough probability of successful delivery and that, at the same time, allows to aggregate data.

4.1. Strategy Selection

At regular intervals the algorithm checks the geographic position of the node and calculates the length $dist_{toBS}$ of the shortest path between the vehicle and the BS. The position of the BS is known because it is included in the query message. Shortest path can be calculated exploiting the city map by invoking each time the Dijkstra algorithm or by differential approaches (the used method is not relevant).

The strategy selection algorithm can use other available information in order to make its decision such as the total life-time Q_{life} of the query and the time left Q_{left} for a valid delivery; both values are easily obtainable from the query message. Based on available information, DB-VDG chooses what strategy to adopt at each invocation of the algorithm.

In order to achieve low transmission overhead, the protocol favors to carry the data using data muling strategy, while the Multi-hop strategy is only selected when the risk of not delivering the data before the query expiration is high.

We have developed two different methods for strategy selection: SBSS and DBSS, presented in following subsections.

4.1.1. Speed-Based Strategy Selection method (SBSS)

Speed-Based Strategy Selection (SBSS) method may be seen as similar to D-Greedy presented in Section 2.2 as they are both delay-bounded routing protocols. However, our SBSS goes beyond simple unicast communication and to this aim also exploits more information than D-Greedy.

More in detail, SBSS selects the strategy to adopt according to the vehicle's speed. At each invocation of the algorithm, $dist_{toBS}$ (the length of the shortest path to the BS) is calculated, therefore, from the second evaluation on, it is possible to determine the speed with which the vehicle is approaching the BS. The variable spd_{toBS} represents the average speed of the vehicle calculated during a k-second historical window. If the vehicle is moving away from the BS, spd_{toBS} is negative.

At the same time, taking into account $dist_{toBS}$ and Q_{left}, the algorithm calculates a value called spd_{thr}:

$$spd_{thr} = \frac{dist_{toBS}}{Q_{left}}. \tag{3}$$

spd_{thr} value is the minimum approaching speed to the BS necessary to carry the data to the base along the best path before the query expires.

- If $spd_{toBS} > spd_{thr}$, SBSS selects to adopt the data muling strategy;
- if $spd_{toBS} < spd_{thr}$, the protocol selects multi-hop forwarding.

We can notice that spd_{thr} grows when Q_{life} increases and $dist_{toBS}$ decreases, more easily satisfying the condition for data muling selection. For this reason we can presume that the efficiency of SBSS grows for high life-time Q_{life} and small regions of interest.

4.1.2. Distance-Based Strategy Selection method (DBSS)

Distance-Based Strategy Selection method (DBSS) simply selects the strategy according to the distance to the BS $dist_{toBS}$, ignoring the approaching speed. The idea is that data are forwarded gradually from the limit of the target area toward the center, during the whole collection period. The gradual centralizing of data should guarantee a good level of data aggregation.

The algorithm needs to know an upper-bound $dist_{max}$ for $dist_{toBS}$. The upper-bound is calculated considering the shortest paths between the BS and all the points within the region of interest.

Knowing $dist_{max}$ and Q_{life}, DBSS calculates $dist_{thr}$ that is the maximum distance (shortest path) to the BS from which a vehicle can be, until which it can keep carrying the data.

$$dist_{thr} = dist_{max} \cdot \frac{Q_{left}}{Q_{life}}. \tag{4}$$

- If $dist_{toBS} < dist_{thr}$, DBSS selects to adopt the data muling strategy;
- if $dist_{toBS} > dist_{thr}$, the algorithm selects multi-hop forwarding.

Analyzing the trend of the function in (4) we can notice that:

- if $Q_{left} = Q_{life}$ (when the collection starts) then $dist_{thr} = dist_{max}$. Given that $dist_{toBS}$ is always smaller that $dist_{max}$, all the vehicles within the target area that have received the query, are adopting data muling strategy;
- if $Q_{left} = 0$ (when the collection ends) then $dist_{thr} = 0$ and all vehicles adopt multi-hop forwarding.

The first nodes that would change their state from data muling to multi-hop forwarding are those closest to the limit of the region of interest; the last ones are the nodes closest to the BS.

It is important to consider that the query propagation process requires a certain amount of time in order to deliver the query all over the region of interest. In particular, the nodes that are near the limit of area, probably are the last nodes which receive the message.

For this reason it is possible that the nodes near the limit receive the query when $dist_{thr}$ is already smaller than $dist_{toBS}$. This influences negatively the efficiency because they have to select immediately the multi-hop forwarding strategy without waiting for incoming data to aggregate.

In order to avoid this problem a certain delay called Q_{prop} can be set. When the data collection starts, an interval defined by Q_{prop}, is reserved for the query propagation: during this period all the nodes within the region of interest adopt the data muling strategy. When Q_{prop} expires, $dist_{thr}$ begins to decrease starting from $dist_{max}$ to zero.

Considering (4), the nodes close to the BS select the multi-hop forwarding strategy for the final time period only. If during this short period there are temporary communication voids, nodes can fail to deliver the data before query expiration.

A solution is to set an interval called Q_{final} in which all nodes in the region of interest adopt multi-hop forwarding. This is in order to allow the nodes nearby the BS to find a next-hop, although it is not immediately available.

Taking into account the last two observations, the new function that determines the values of $dist_{thr}$ is:

$$dist_{thr} = dist_{max} \cdot \frac{Q_{left} - Q_{final}}{Q_{life} - Q_{final} - Q_{prop}}. \tag{5}$$

If we assume that, when any node switches to multi-hop forwarding, it can select as next-hop a neighbor that is closer to the BS than itself, the next-hop should be still adopting data muling strategy therefore data are aggregated at each hop and it is pushed toward the BS with the minimum overhead because each node sends exactly one message. This situation should occur in presence of a high vehicle density.

4.2. Next-hop Selection

When a vehicle is adopting multi-hop forwarding, it has to forward the data it is carrying to another node selected among all its neighbors. The decision is based on the local knowledge contained in its *neighbor-table*.

The next-hop selection algorithm considers as relevant only the neighbors that are closer to the BS than the local node, so as to bring the data nearer to their final destination like in basic geographic routing protocols. For this reason it is necessary to calculate the length of the shortest path to the BS for each neighbor. In order to make the calculation more accurate, it is possible to predict the current position of the neighbors through (2).

If there are no neighbors closer to the BS than the local node, forwarding of data is not possible. Conversely, the algorithm estimates, through DPE, the probability P_{CT} to deliver successfully a DATA message for each relevant neighbor T.

The algorithm considers as *next-hop candidates* only the neighbors T' for which $P_{CT'}$ is greater than a fixed threshold P_{thr}, otherwise the neighbors are ignored. If there are not candidates, the algorithm selects as next-hop the node with the highest delivery probability. On the contrary, if there are multiple candidates, the algorithm selects the next-hop according to the aggregation level.

The aggregation level is considered important in order to improve the data aggregation and to reduce the number of DATA messages forwarded. Neighbors that are *data carriers* (that have an aggregation level greater than zero) are preferred over the neighbors that are *data free* (that are not carrying any data) because if a data carrier is selected as next-hop, it can merge new data with the data that it is already carrying and forward them together in a single DATA message.

However, among different data carrier nodes, the next-hop selection algorithm prefers the ones that have the smallest data aggregation level in order to avoid a large concentration of data in a single node. If there are more than one neighbor with the minimum level of data aggregation, the procedure selects as next-hop the closest to the BS.

Considering that the goal is forwarding the data toward the BS, the next-hop selection algorithm has to select the BS if it is in the neighborhood. In order to do this it is sufficient that the BS included in its HELLO messages the maximum aggregation level while its $dist_{toBS}$ is zero.

5. Experimental Assessment

We have performed a thorough performance analysis of the proposed protocol using the network simulator ns-3.6 [16]. In order to arrange a realistic testbed we have also implemented proper mobility and propagation models, which also consider signal fading due to buildings' edges. Adopted models and the setting used in our simulation are discussed in the following subsections.

5.1. Mobility Model

The mobility model we chose in order to measure the performance of DB-VDG protocol is the Manhattan grid model. Despite Manhattan is a very simple model, tests demonstrate that, at least for data gathering applications, there are not significant performance variations choosing a more realistic model [13].

The ns-3.6 simulative tool does not provide a mobility model specifically tailored for the scenario we are considering; nevertheless, we can resort to a smart configuration of the random-walk mobility model, defined in ns-3.6 by the object ns3::RandomWalk2dMobilityModel.

In our mobility model vehicles are constrained to move along streets that form a square grid in which street sections have the same length: 100 m. Vehicles travel along any street section at different speeds. When a crossroad is reached, vehicles randomly select a new direction and a new speed in the range between 10 m/s and 20 m/s.

5.2. Signal Propagation Loss Model

In our simulations we used a combination of the ns-3.6 built-in propagation loss model objects ThreeLogDistance and Nagakami, properly configured in order to simulate the 802.11p technology behavior considering the wireless fast-fading process. Additionally, in order to consider the physical obstacles that can obstruct communications in the real world, we implemented a novel propagation loss model called CityGrid that is stacked over other used models.

- *ThreeLogDistance* propagation loss model is based on a function which is the weighted sum of the logs of the distance, divided in three fields: near (d_0), middle (d_1) and far (d_2), with different coefficients (n_0, n_1, and n_2). ThreeLogDistance propagation loss model is used in order to define the range of transmission and power loss at each distance. It is defined in [17].

- *Nakagami* propagation loss model is based on Nakagami's m-distribution [18] that is related to the gamma probability distribution. It is widely used to model fading process of wireless signals traversing multiple paths and it is defined in [19].

Setting the initial transmit power equal to 20 dBm, we choose to keep Nagakami at default configuration and to set ThreeLogDistance parameters as Table 1 shows. Considering the exchange of HELLO messages with variable distances we experimentally obtained the percentage of successful dispatches depicted in Fig. 3.

Since the embedded propagation model used by DPE has to be based on the implemented propagation loss model, we define it through Formula (6). It returns delivery probability 0.85 if distance $d = 300$ m and has a maximum transmission range of 500 m.

$$\begin{array}{rl} 1 + \frac{(0.85-1)\cdot d}{300} & \text{if } d < 300 \\ 0.85 - \frac{0.85\cdot(d-300)}{200} & \text{if } d \geq 300 \end{array} \tag{6}$$

5.2.1. CityGrid Propagation Loss Model

CityGrid is a novel propagation loss model which calculates the signal power loss considering the presence of obstacles between the sender and the receiver. CityGrid shares the same configuration as the implemented mobility model for making resulting topologies overlap.

If there are no obstacles between two nodes, CityGrid returns the input power unaltered; on the contrary, if a communication is totally obstructed by some buildings, it returns a value that makes the transmission unfeasible. Sometimes communications between two nodes are partially obstructed by the edge of a building. In these cases CityGrid values how much the edge obstructs the channel, and then returns a proper power loss value.

In details CityGrid calculates how much the straight line between two nodes penetrates within the edge through (7). Parameters Δx and Δy are the absolute distance between the nodes along the main axes, W is the width of the street that in our simulation is set to 10 m. p is the penetration through the edge in meters.

$$p = \sqrt{2} \cdot \left(\frac{\Delta y}{1 + \frac{\Delta y}{\Delta x}} - \frac{W}{2} \right). \tag{7}$$

If $p > 0$, CityGrid calculates the path loss L through (8), where Tr is the penetration threshold (in our simulations set to 5 m), E_{in} is the signal energy in input (resulting after ThreeLogDistance and Nakagami models process) and E_{Dthr} is the energy detection threshold (default value is -96 dB).

$$L = (E_{in} - E_{Dthr}) \cdot \frac{p}{Tr}. \tag{8}$$

5.3. Protocols Settings

DB-VDG protocols have a lot of parameters which affect its performance and which need to be tuned. We have set these parameters according to empirical tests which demonstrated that our configuration permits to balance performance and bandwidth consumption.

For the beaconing scheme we set an interval of 500 ms between beacon messages; the interval between two consecutive invocations of the strategy selection algorithm is 100 ms; the ACK time-out is 50 ms and the maximum number of ACK attempts is set equal to 2; probability threshold P_{thr} is set equal to 0.60. DBSS parameters are proportional with the current query life-time Q_{life} therefore, after empirical tests, we obtained the best performance setting $Q_{prop} = 0.3 \cdot Q_{life}$ and $Q_{final} = 0.15 \cdot Q_{life}$.

6. Analysis of Results

In order to evaluate the performance of DB-VDG protocol we have arranged various testbed configurations with different grid sizes, query life-time and the density of vehicles expressed in terms of number of vehicles per km^2.

Considered grids are square with different side widths: 1000 m (1 km^2), 2000 m (4 km^2) and 3000 m (9 km^2). The region of interest is a circle with a diameter set equal to 80% of the diagonal of the grid, so that the corners of the scenario are excluded from the region of interest.

For each scenario configuration we have performed 20 runs. Results of the tests are been summarized in graphs that indicate the averages and the 95% confidence intervals calculated with the Student's t-distribution.

In addition to measure the performance of DB-VDG protocol, we also performed some tests in order to evaluate how the density of vehicles influences the query propagation process.

6.1. Query Propagation Performance

Results proposed in Fig. 4 and in Table 2 measure the percentage of vehicles, within the region of interest, which receive the query and demonstrate that implemented propagation method is effective. The lower result in case of low density scenario is probably due to the high partitioning of the network and it should be independent on the used propagation method.

Low density also affects the number of query free nodes reached by each query forwarding, because it is related with the number of nodes in the transmission range. Results in Fig. 5 and in Table 3 show that implemented propagation method is rather efficient for high density scenarios. However a deeper analysis and the comparison with existent methods should be necessary in order to evaluate the goodness of our solution which is not the main contribution of this work.

6.2. Performance Metrics

In order to analyze the performance of DB-VDG we defined two different metrics which measure the effectiveness and efficiency of the protocol:

1. *Effectiveness* measures the ability of DB-VDG to forward the data from the involved nodes to the BS. Involved nodes are the vehicles in the region of interest which have received the query message. Effectiveness is defined as:

$$effectiveness = \frac{N_{delivered}}{N_{nodes}}\%. \tag{9}$$

   where $N_{delivered}$ is the amount of data delivered to the BS before the query expiration and N_{nodes} is the number of involved nodes.

2. *Efficiency* is an index that measures the level of bandwidth optimization. It is the reciprocal of the number of DATA messages sent, on average, by each involved node. Its definition is:

$$efficiency = \frac{N_{nodes}}{N_{sent}}. \tag{10}$$

   where where N_{nodes} is the number of involved nodes and N_{sent} is the total number of DATA messages sent during the propagation period.

6.3. General Behavior

In order to analyze how the scenario influences the performance of DB-VDG, we performed various tests with different parameters for the size of the scenario and the vehicle density. We consider the use of DB-VDG with DBSS strategy selection method and a query life-time of 50 s.

Tests summarized in Table 4 demonstrate that low vehicle density influences negatively the effectiveness index of the protocol. This result was predictable because in these cases connectivity is poor and vehicles experience difficulties in delivering data to the BS. For low density, large scenarios reduce the amount of data delivered since the paths between vehicles and the BS become longer. With high density the reduction of effectiveness caused by large scenarios is irrelevant.

Regarding the efficiency index of DB-VDG using DBSS, we noticed that it is quite stable, slightly decreasing for large scenarios and low densities. This is also correlated with poor connectivity: a certain amount of DATA messages are sent in order to forward data which do not reach the BS in time. Results are summarized in Table 5.

In order to study how variations of the query life-time influence effectiveness and efficiency of DB-VDG protocol we have performed some tests with variable query life-time (25 s, 50 s and 100 s) on a scenario with density equal to 50 vehicles/km^2 and a 4 km^2 grid. Results in Table 6 demonstrate that increasing the query life-time, the amount of data delivered to the BS grows, while the efficiency is stable. The increase of effectiveness was predictable because vehicles have more time to find a path for the data.

6.4. Performance Comparison

In order to analyze the performance of DB-VDG we can compare it with MobEyes protocol with on-demand harvesting. Cited protocol can be easily implemented as a strategy selection that always selects the multi-hop forwarding strategy and never the data muling like in opportunistic routing. In this way the data are dispatched with the minimum delay, therefore the effectiveness of MobEyes ODH can be considered as an upper-bound for other strategy selection methods of DB-VDG. On the other hand, the lack of any optimization of bandwidth consumption, ignoring the aim of taking advantage of data aggregation, should makes MobEyes ODH method very inefficient therefore its efficiency index can be considered a lower-bound.

We choose to arrange a testbed with a medium vehicle density (50 vehicles per km^2), query life-time set to 50 s and variable grid size (1 km^2, 4 km^2 and 9 km^2).

In results summarized in Fig. 6 and Fig. 7 it is possible to notice that SBSS method has an effectiveness value close to the upper-bound obtained by MobEyes ODH for any grid dimension. Regarding the efficiency, SBSS achieves good results in case of small scenarios while it falls toward the lower-bound for large grids. In any case, SBSS is a preferable solution with respect to MobEyes ODH.

DBSS proves to be the most efficient method for any grid dimension. In case of small scenarios it also delivers to the BS almost the same amount of data than other methods. Increasing the dimension of the scenario, the effectiveness of DBSS, compared to the MobEyes ODH upper-bound, slightly decreases while the efficiency grows. Analyzing the results contained in Table 7, regarding the testbed with grid size equal to 4 km^2, we can state that in this scenario the loss of collected data is fully compensated by the better efficiency. In fact, against a loss of 10 unit of data collected, DBSS saved about 180 DATA message dispatches.

Other performed tests show that SBSS is very close to MobEyes ODH in term of effectiveness in every scenario, while its efficiency is always higher, especially for long query life-times. DBSS method is always preferable with respect to SBSS because a little loss of collected data is always compensated by a higher efficiency. The benefits of DBSS method, with respect to SBSS, grow for scenarios with high vehicles densities. This is confirmed by the results obtained using a high density scenario summarized in Table 8.

7. Conclusions and Future Work

In this work we have presented a solution for collecting data in an urban scenario using Vehicular Sensor Networks. In order to achieve this, we have proposed a integrated solution that combines the use of a generic time-stable geocast protocol with a novel protocol: DB-VDG, that manages the collection of data and its forwarding toward the requesting base station.

Tests demonstrate that data gathered by vehicles can be collected and forwarded to the requesting BS in an effective and efficient way. A vehicle density of 50 vehicles per km^2 is sufficient in order to collect data from almost all the involved vehicles, however good results are also achieved with lower densities.

The main mechanism of our solution is the strategy selection algorithm that selects either to keep carrying the data or to forward it to a neighbor.

We have proposed two different strategy selection methods that DB-VDG can adopt according to specific applications: SBSS and DBSS. The SBSS method is suitable for applications in which it is important to collect the maximum amount of data regardless of efficiency; its bandwidth consumption is slightly better than simply greedy approaches (e.g. MobEyes ODH).

The DBSS method allows collecting a significant amount of data while optimizing bandwidth consumption. Moreover, in the case of high density scenarios and/or long data gathering periods, DBSS succeeds in delivering to the BS almost the same amount of data as SBSS; therefore in these cases its adoption is the best choice.

In conclusion we can state that the proposed work reaches the intended goal. DB-VDG protocols can be successfully used in order to collect data in an urban area, within a certain time deadline, also reducing the total bandwidth consumption in the vehicular network. The availability of different strategy selection methods makes the DB-VDG protocol suitable for different applications.

As future work, it would be interesting to add a method to optimize the merging of data within the DB-VDG protocol so as to reducing memory consumption; this should make the DATA message lighter and reduce message loss, improving the overall performance. Furthermore, a void handling method should be integrated with data gathering protocol in order to manage communication voids.

Finally, although we have empirically investigated how to configure parameters for our protocols we would also like to further investigate this aspect through analytical evaluation. To this aim, we deem that interesting inspiration can be taken from [13].

References

[1] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. amendment 6: Wireless access in vehicular environments, 2011.

[2] S. Biswas, R. Tatchikou, F. Dion, Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety, Communications Magazine, IEEE 44 (2006) 74–82.

[3] F. Ye, M. Adams, S. Roy, V2v wireless communication protocol for rear-end collision avoidance on highways, in: Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on, pp. 375 –379.

[4] M. Roccetti, G. Marfia, C. E. Palazzi, Entertainment beyond divertissement: Using computer games for city roads accessibility, ACM Computers in Entertainment (2010). To appear.

[5] L. Wischhof, A. Ebner, H. Rohling, Information dissemination in self-organizing intervehicle networks, Intelligent Transportation Systems, IEEE Transactions on 6 (2005) 90 – 101.

[6] U. Lee, M. Gerla, A survey of urban vehicular sensing platforms, Computer Networks 54 (2010) 527 – 544. Advances in Wireless and Mobile Networks.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: MOBICOM, ACM, 2000, pp. 56–67.

[8] A. Skordylis, N. Trigoni, Delay-bounded routing in vehicular ad-hoc networks, in: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '08, ACM, New York, NY, USA, 2008, pp. 341–350.

[9] C. E. Palazzi, M. Roccetti, S. Ferretti, An intervehicular communication architecture for safety and entertainment, Trans. Intell. Transport. Sys. 11 (2010) 90–99.

[10] F. Ros, P. Ruiz, I. Stojmenovic, Reliable and efficient broadcasting in vehicular ad hoc networks, in: Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th, pp. 1 –5.

[11] A. Amoroso, G. Marfia, M. Roccetti, Optimal assessments in vanet: The oracle., in: Wireless Days'10, pp. 1–5.

[12] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, S. Madden, Cartel: a distributed mobile sensor computing system, in: In 4th ACM SenSys, pp. 125–138.

[13] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, A. Corradi, Dissemination and harvesting of urban data using vehicular sensing platforms, Vehicular Technology, IEEE Transactions on 58 (2009) 882 –901.

[14] J. Zhao, G. Cao, VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks, in: Proceedings IEEE INFOCOM 2006. 25TH IEEE

International Conference on Computer Communications, IEEE, 2006, pp. 1–12.

[15] S. Hermann, C. Michl, A. Wolisz, Time-stable geocast in intermittently connected ieee 802.11 manets, in: Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th, pp. 1922 –1926.

[16] `http://www.nsnam.org`, 2011.

[17] `http://www.nsnam.org/doxygen-release/classns3_1_1_three_log_distance_propagation_loss_model.html`, 2011.

[18] N. Nakagami, The m-distribution, a general formula for intensity distribution of rapid fading, in: W. G. Hoffman (Ed.), Statistical Methods in Radio Wave Propagation, Oxford, England: Pergamon, 1960.

[19] `http://www.nsnam.org/doxygen-release/classns3_1_1_nakagami_propagation_loss_model.html`, 2011.

Table 1: ThreeLogDistance propagation loss model parameters.

$d_0 = 1$	$n_0 = 2.0$
$d_1 = 150$	$n_1 = 5.0$
$d_2 = 350$	$n_2 = 10.0$

Table 2: Percentage of vehicles, within the region of interest, which receive the query message for query life-time = 50 s and different vehicle densities and grid sizes.

	25 v/km^2	**50 v/km^2**	**100 v/km^2**	**200 v/km^2**	**400 v/km^2**
1 km^2	88.6%	96.4%	98.3%	98.4%	98.1%
4 km^2	90.4%	97.9%	98.6%	98.8%	n/a
9 km^2	90.7%	98.5%	99.1%	n/a	n/a

Table 3: Number of query free nodes reached by each query forwarding for query lifetime = 50 s and different vehicle densities and grid sizes.

	25 v/km^2	**50 v/km^2**	**100 v/km^2**	**200 v/km^2**	**400 v/km^2**
1 km^2	1.10	1.24	1.67	2.22	2.41
4 km^2	1.11	1.39	1.93	2.58	n/a
9 km^2	1.14	1.44	2.05	n/a	n/a

Table 4: Effectiveness index of DB-VDG with DBSS method for query life-time = 50 s and different vehicle densities and grid sizes.

	25 v/km^2	**50 v/km^2**	**100 v/km^2**
1 km^2	79.6%	96.7%	99.4%
4 km^2	61.3%	89.7%	97.5%
9 km^2	32.0%	82.7%	95.1%

Table 5: Efficiency index of DB-VDG with DBSS method for query life-time = 50 s and different vehicle densities and grid sizes.

	25 v/km^2	**50 v/km^2**	**100 v/km^2**
1 km^2	0.62	0.61	0.66
4 km^2	0.55	0.58	0.67
9 km^2	0.53	0.56	0.67

Table 6: Effectiveness and efficiency indexes of DB-VDG with DBSS method for density = 50 vehicles/km^2, 4 km^2 grid and different query life-times.

	25 s	**50 s**	**100 s**
effectiveness	59.0%	89.7%	98.3%
efficiency	0.60	0.58	0.59

Table 7: DB-VDG results for density = 50 vehicles/km^2, query life-time = 50 s and 4 km^2 grid size with different strategy selection methods.

	MobEyes	**SBSS**	**DBSS**
N_{sent}	490.8	440.9	311.2
$N_{delivered}$	177.9	177.4	168.1
effectiveness	96.2%	96.0%	91.2%
efficiency	0.38	0.42	0.59

Table 8: DB-VDG results for density = 100 vehicles/km^2, query life-time = 50 s and 4 km^2 grid size with different strategy selection methods.

	MobEyes	**SBSS**	**DBSS**
N_{sent}	928.6	793.0	554.6
$N_{delivered}$	371.1	371.1	364.8
effectiveness	99.3%	99.3%	97.7%
efficiency	0.40	0.47	0.67

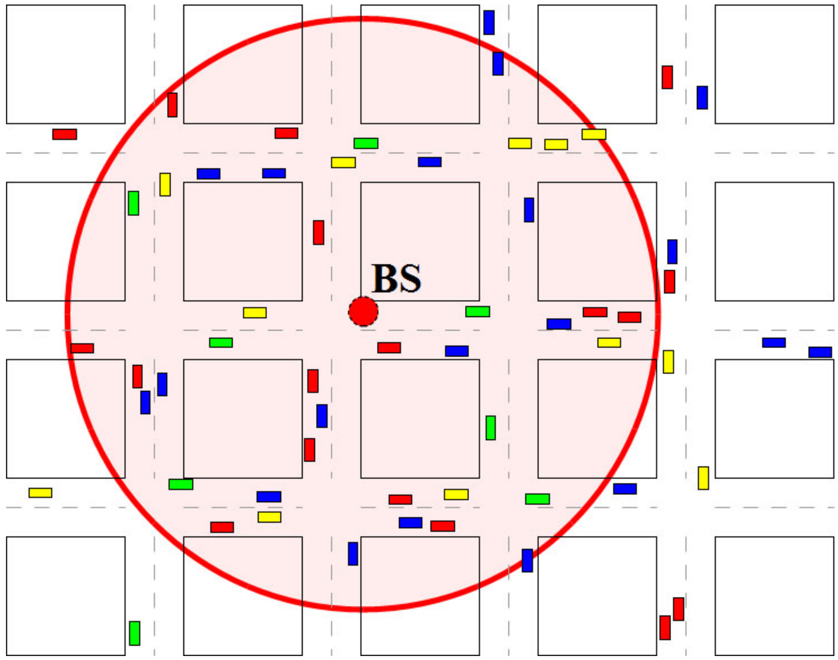

Figure 1: A base station installed in an urban scenario with the region of interest of a query.

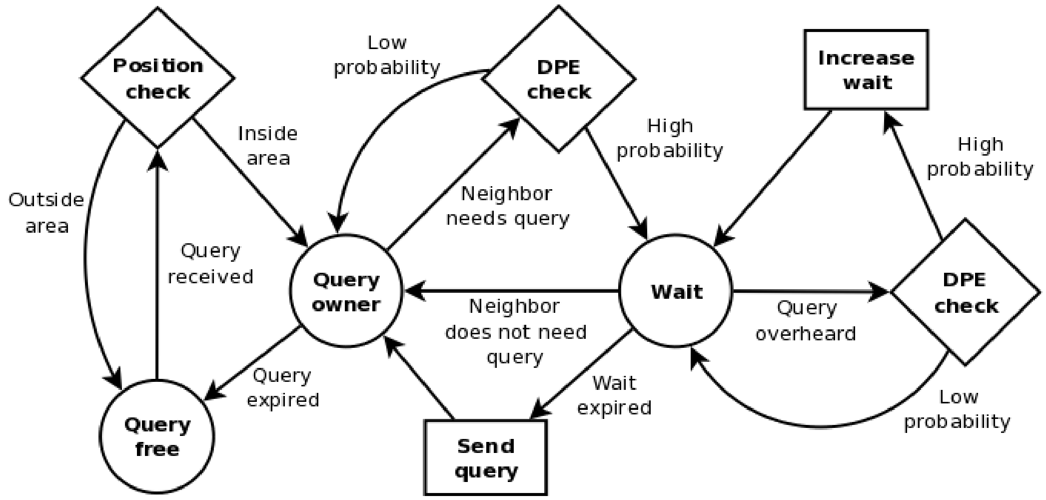

Figure 2: Diagram of the implemented query propagation method.

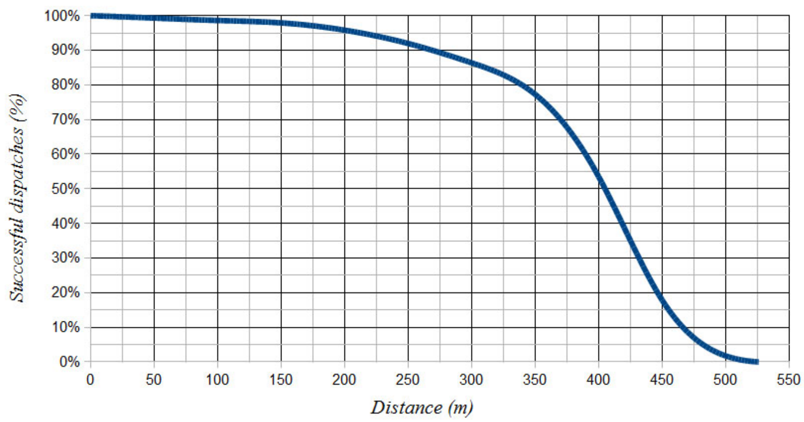

Figure 3: Delivery probability for implemented propagation model.

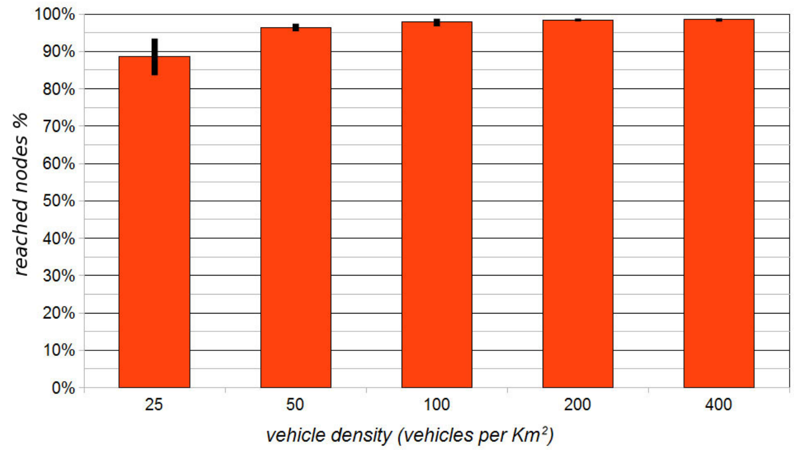

Figure 4: Percentage of vehicles, within the region of interest, which receive the query message for 1 km^2 grid with different vehicle densities and query life-time = 50 s.

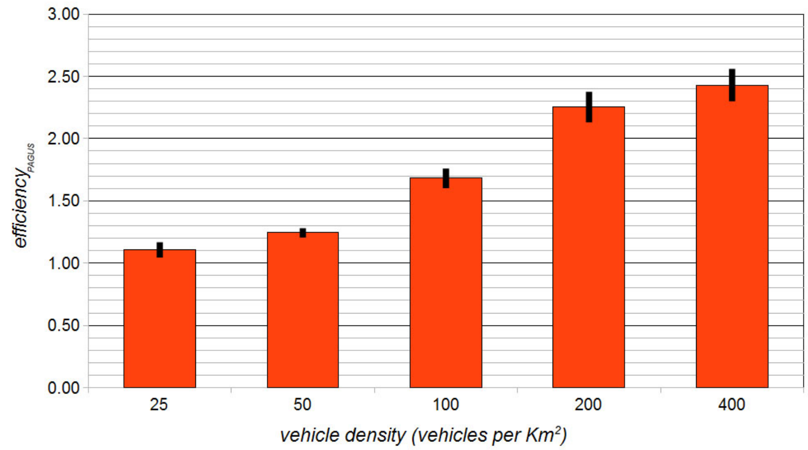

Figure 5: Number of query free nodes reached by each query forwarding for 1 km^2 grid with different vehicle densities and query life-time = 50 s.

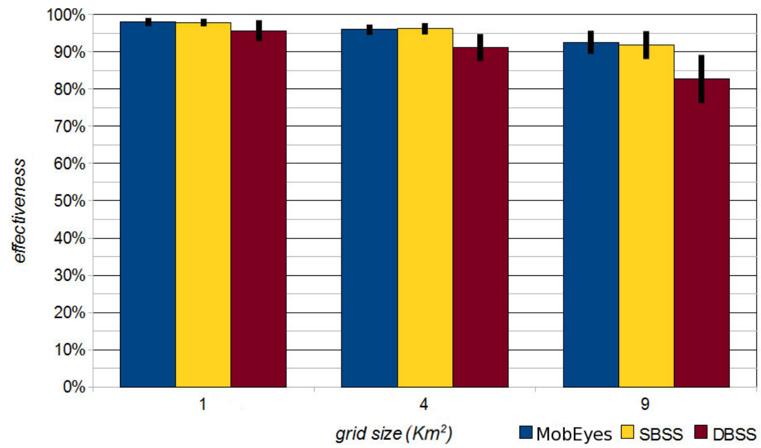

Figure 6: Effectiveness index of DB-VDG protocol with different strategy selection methods for density = 50 vehicles/km^2, query life-time = 50 s and variable grid size.

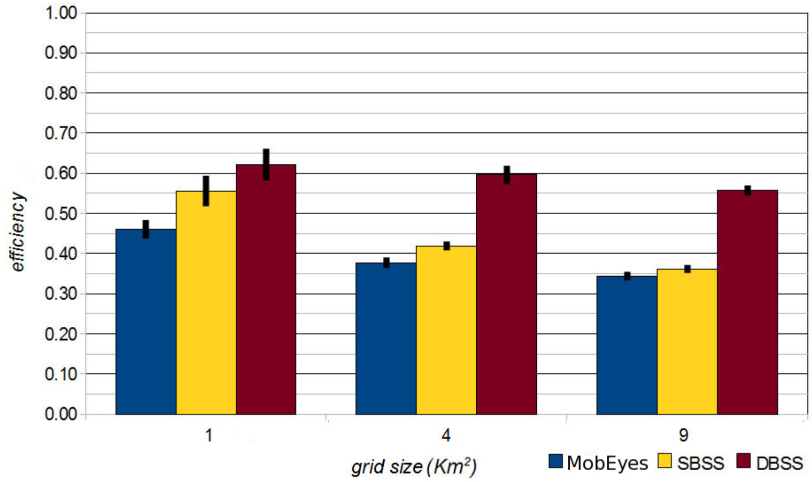

Figure 7: Efficiency index of DB-VDG protocol with different strategy selection methods for density = 50 vehicles/km^2, query life-time = 50 s and variable grid size.