

An integrated VANET-based data dissemination and collection protocol for complex urban scenarios

Ion Turcanu^a, Pierpaolo Salvo^a, Andrea Baiocchi^{a,*}, Francesca Cuomo^a

^a*Dept. of Information Engineering, Electronics and Telecommunications
Sapienza University of Rome
Via Eudossiana 18, 00184, Rome, Italy*

Abstract

Data dissemination and data collection to/from vehicles traveling on city roads are key features to fully enable the advent of Intelligent Transport Systems and Autonomous vehicles. Both Road Side Units and On Board Units need to disseminate different kind of data to vehicles or to collect data sensed by the vehicles themselves and transfer them to road monitoring and control centers. In this work we propose a protocol, named DISCOVER, that disseminates and collects the data of interest in a quite large city area efficiently and timely by using a single network structure, i.e., a multi-hop backbone made up only of vehicles nodes. DISCOVER is distributed and adaptive to the different traffic conditions, i.e., to the different levels of vehicular traffic density. Several numerical results show that it attains very good performance in different type of city maps (New York, Paris, Madrid and Rome) when compared with baseline approaches as well as when compared with a theoretical bound.

Keywords: Vehicular Ad-Hoc Networks, Data Collection, Data Dissemination, Vehicular Backbone Networks, Urban Sensing.

1. Introduction

Intelligent Transportation Systems (ITS) integrate Information and Communications Technology (ICT) with transportation engineering methods to get an improved knowledge of current and future states of the transportation system and, possibly, to react to unexpected perturbations in order to keep the system near a desired state of safety, efficiency and comfort. The transportation system is traditionally represented through the interactions between its elementary components: travelers, vehicles, and infrastructure. These interactions affect and, at the same time, are affected by the external environment, both at monadic level - that is, the single vehicle - and at aggregate level, represented by the traffic system. ITS applications enhance efficiency and effectiveness of these interactions thanks to a set of sensors that monitor the surrounding environment up to a certain extent, and a set of actuators that put in practice predetermined control rules.

Two key enablers of ITS services that can be supported by the Vehicular Ad-Hoc Network (VANET) technology are the data dissemination and the data collection from vehicles on the road. Data dissemination can be achieved through the use of vehicle-to-vehicle multi-hop communications, enabling the extension of the road span covered by

the Road Side Units (RSUs) or On Board Units (OBUs) generating the data. This dissemination function is of interest for both safety and infotainment applications [1]. Another interesting function is the collection of data from vehicles, through the VANET. Vehicles can be used as sensors that monitor traffic, roads, the environment and send their data to a collection center. In opposition to the dissemination, data collection aims at gathering data, relevant to safety, traffic information, infotainment, over a given area of interest.

This article is an extension of our preliminary paper [2], in which we propose DISCOVER, a protocol for VANETs, that can be used both for data dissemination and collection in a complex urban scenario. Since data collection in real time from vehicles is a qualifying new aspect of our proposal, we focus on this aspect in this paper. Nevertheless, the protocol we propose merges the dissemination and collection functionalities in a modular way, so that the amount of information to be disseminated and/or collected and the repetition rate of the procedure can be tailored to any vehicular application environment. The proposed protocol design induces a self-organized VANET backbone structure, with no prior knowledge of the road map or of the intersection positions and no need for any special networking equipment. The VANET backbone is composed on relay nodes that can be used flexibly both to disseminate data and to collect FCD from roaming vehicles in the target area. The flexible composition of the dissemination and collection functions in a single protocol is a distinctive

*Corresponding author

Email address: andrea.baiocchi@uniroma1.it (Andrea Baiocchi)

feature of DISCOVER.

The major contribution of this work are as follows:

- (i) The definition of an integrated protocol to disseminate and collect data in real time, based on a self-organized VANET.
- (ii) The evaluation of the protocol performance as for real time FCD collection with a set of realistic simulation scenarios, based on the urban maps of New York, Rome, Madrid and Paris.
- (iii) The definition of bounding benchmarks to qualify the performance achieved by the proposed protocol: on one side, we define a theoretical lower bound of the amount of overhead required to collect the data from the vehicles roaming in the Region of Interest (RoI) through the VANET; on the other side, we show results for two baseline approaches, RANDOM and FLOODING, that provide a gauge of the gap between ideal performance as predicted by the theoretical bound and of easily achievable performance with baseline strategies.

The rest of the paper is structured as follows. Sec. 2 provides an overview of the related work. Sec. 3 defines the theoretical performance bound. Sec. 4 describes our proposed protocol. Sec. 5 presents the performance analysis and the comparison with other solutions, while conclusions are drawn in Sec. 6.

2. Related work

Many research projects are in progress aiming at creating new and efficient data dissemination and collection schemes based on the VANET. The most widespread techniques to disseminate and collect data are broadcast-based, because of the advantage given by eliminating the complexity of route discovery, address resolution and topology management. The most of the proposed schemes that we find in the literature focus either on data dissemination, or on data collection.

A recent survey on dissemination protocols in vehicular networks is provided in [3]. One of the most used technique to implement an efficient data dissemination scheme is to identify only a small subset of vehicles responsible for re-broadcasting the information. We can find this idea in [4], where a new dissemination protocol, named Vehicular Backbone Network (VBN), has been proposed. Here the messages sent out by the RSU are forwarded only by those vehicles that are situated closest to *nominal relaying positions*, that are spaced out by a range D . The distance D is chosen so as to provide each receiving relay node with a SINR level that can support the intended packet transmission rate. Since VBN was designed especially for disseminating data in highways, we extended this protocol in order to disseminate and collect data in urban scenarios. In [5] Viriyasitavat et al. face the problem of data

dissemination in urban scenarios in the presence of disconnected networks. They propose UV-CAST (Urban Vehicular Broadcast), a completely distributed protocol which utilizes both direct relays through multi-hop transmissions and indirect packet relays through the *store-carry-forward* (SCF) mechanism.

A number of researches and implementation efforts have been spent to investigate data collection schemes. In [6] Brik et al. propose a Token-based Clustered Data Gathering Protocol (TCDGP). This protocol divides the interested road in two types of segments: Collection Segments (CS), where the vehicle-to-vehicle (V2V) communication is allowed, and Silent Segments (SS), where V2V communication is prohibited in order to avoid collisions between adjacent segments. A Cluster Head (CH) vehicle is elected for every CS, considering the vehicle's distance from the center of the CS and the time traveled on the middle lane, which is in charge of collecting data inside the CS. This solution was designed to collect data on highways, while our protocol aims at collecting data in urban environments. A similar approach can be found in [7], where Chang et al. propose TrafficGather. This protocol also divides the road into a series of contiguous clusters and for each cluster it elects a Cluster-head Vehicle (CV). Once the clusters are created, each vehicle sends its information to its own CV according to a TDMA Access Control Algorithm (TACA) proposed by the authors, in order to avoid collisions. In the data retrieval phase the CVs send their collected data towards the initiating vehicle using a flooding strategy, which generates a very large overhead.

A different approach for data collection is found in [8], where the authors propose ADOPEL (Adaptive Data Collection Protocol using Reinforcement Learning). It is a data collection technique designed to collect data on highways while making the collection operation more reactive to nodes mobility and topology changes. The protocol is based on a distributed Qlearning technique where a reward function is provided and defined to take into account the delay and the number of packets to aggregate. The collect operation is periodically started by a randomly selected node called *initiator*, which has to collect the traffic data from vehicles and deliver it to a Traffic Control Center (TCC). The selection of the best next relay is based on the Qvalue, determined by the Qlearning algorithm. Zhang and He [9] proposed a two-way data collection scheme for VANET. Their algorithm is completely stateless, thus vehicles neither store neighbor information locally nor detect neighbors change. The algorithm is divided in two phases: the dissemination phase and the collection phase. The process starts with a request message from BS (Base Station), which is further disseminated possibly to all vehicles using broadcasting. In order to deal with the broadcast storm problem [10], the authors propose two mechanisms: *Rebroadcast filtering* and *Duty Cycled Execution*. The first mechanism tries to reduce the redundant rebroadcasts by filtering the received messages: only on receiving a message for the first time vehicles have to rebroadcast. The

second tries to reduce the contention and collision by restricting the transmission frequency: vehicles are allowed to broadcast once in a fixed duty cycle. As for the data collection phase, the authors propose to use the same algorithm adopted for data dissemination phase. In particular, once a vehicle receives a request message, besides continuing the dissemination process, it also sends back a reply message containing the requested information. Vehicles in charge of returning the reply message back to the BS are selected based on their geographical proximity to the BS itself. Since this protocol basically uses a flooding approach to collect data, it induces a large overhead due to multiple re-broadcasting of messages. The problem of delay-constrained data aggregation in VANETs is faced in [11] by Zhu et al. They propose a two version (centralized and distributed) approach called *aTree*, which first constructs a data aggregation tree using a *flooding* approach, and then assigns a waiting time budget for each node on the tree. Their basic idea is to assign larger waiting timers to nodes closer to the collection node, in this way allowing child nodes to send their information earlier. Nodes aggregate all the data received from their child nodes in the tree before transmitting it towards the collection node.

A more recent trend in vehicular networking focuses on data dissemination and collection using a hybrid approach: vehicles use VANETs for V2V communication, and cellular networks for vehicle-to-infrastructure (V2I) communication. For example, the authors in [12] present a mechanism for information dissemination based on virtual infrastructure selection in combination with multiple communication technologies. They aim at increasing the penetration of information by defining an optimal dissimilarity relation among vehicles elected as virtual infrastructure. Sahli et al. [13] propose CGP (Clustered Gathering Protocol), a cross-layered gathering, dissemination and aggregation protocol. The idea of the protocol is to divide the road in virtual segments with the same length. In each segment a Cluster Head (CH) is elected, that gathers data from all nodes in its segment and aggregates them, before sending this result to the next segment or to the base station. Another framework that uses a centralized clustering mechanism to collect the data is LTE4V2X [14]. In particular, eNodeBs are responsible to organize vehicles into clusters, to elect Cluster Heads (CH), and broadcast the clusters topology to the vehicles. Then each CH collects data inside its own cluster using V2V communication, aggregates this information and sends it to the eNodeB. An extension based on the Clustered Gathering Protocol [13] of the original LTE4V2X framework is proposed as well, in order to use multi-hop communication to collect the data inside the areas where there is no LTE coverage.

Differently from the above mentioned works, our protocol aims to integrate the data dissemination and collection in one solution able to maximize the amount of the collected information while trying to minimize the overhead, relying only on the VANET multi-hop communication.

3. Data collection in a VANET

In this Section first we set out the system scenario; then, we derive performance bounds consistent with the assumed scenario. The bounds hold for any specific data collection protocol and provide us with a benchmarking tool to assess how close specific protocols match ideal performance targets.

The considered scenario comprises a RSU and a population of V vehicles moving in a given area around the RSU. We assume Cooperative Awareness Messages (CAMs) [15] are periodically exchanged among single hop neighbor vehicles, with a generation interval T_{CAM} (the standard specifies that $100\text{ ms} \leq T_{CAM} \leq 1000\text{ ms}$). The data extracted from the CAMs are stored in a Local Dynamic Map (LDM) [16] maintained by each vehicle On Board Unit (OBU). The LDM is updated every time a new CAM is received or when local information lifetime expires.

Floating Car Data (FCD) collection consists of periodic delivery of vehicle data to a traffic monitoring server via the RSU. Vehicles communicate by means of the VANET. Messages are generated by each vehicle with its respective FCD and sent to the RSU, via multi-hop communications through the VANET. The FCD record consists of vehicle's ID, position, speed, direction and a timestamp [15][17].

Let us freeze the picture of the system at the time the RSU issues a data collection *Request* message. Let L be the length of the FCD message. The net amount of data the RSU should receive is VL at most. At the time that the collection is done, triggered by the initial *Request* message emitted by the RSU, only $N \leq V$ out of V vehicles are connected, i.e., they belong to a connected VANET graph that includes the RSU. Hence, the minimum amount of data that must be received by the RSU does not exceed NL bytes.

The actual amount of bytes transmitted on the air through the VANET to deliver the FCD of the N vehicles to the RSU is bigger than NL , because of a number of reasons:

- (i) static overhead of the VANET protocol stack (including PHY, MAC, LLC, network and transport layers), denoted as H (i.e. a physical block of data carrying L bytes of data from the facility layer of the VANET has length $H + L$);
- (ii) multiple transmissions of a same FCD message due to the multi-hop networking;
- (iii) re-transmission of messages on each link, if ARQ mechanisms are provided;
- (iv) signaling messages required by the data collection protocol, besides data messages devoted to FCD transport.

A performance metric for the efficiency of the FCD collection protocol is the amount of bytes B_{tx} transmitted in the VANET channel to complete a single instance

of FCD collection from all vehicles in the RoI. Let B_{FCD} be the actual net amount of bytes of FCD messages delivered to the RSU. Let M be the number of vehicles monitored by the RSU at the end of the collection process: it is $M \leq N \leq V$. In general $B_{FCD} = ML \leq NL$, because not all data arrives at the RSU. We define a normalized metric by the ratio $\rho \equiv B_{FCD}/B_{tx}$. This is the ratio of the net FCD bytes successfully collected at the RSU in a collection instance and the overall amount of bytes transmitted on air to carry out that collection. The closer to 1, the more efficient the collection protocol in using the VANET bandwidth.

In the following we consider the connectivity graph \mathcal{G} , whose vertices (nodes) are the N vehicles' OBUs and the RSU. It comprises $N + 1$ nodes; the RSU is conventionally denoted as node 0. Two nodes i and j are connected by an undirected edge in \mathcal{G} if i can receive and decode successfully a frame sent by j and viceversa ($i, j = 0, 1, \dots, N$). Consistently with the definition of N , the number of vehicles that belong to the connected component of the VANET comprising the RSU, \mathcal{G} is a connected graph.

Let us consider a spanning tree of \mathcal{G} rooted at the RSU. There are many efficient algorithms to find such a tree (e.g., see [18, Ch. 5]). It must exist, since the graph \mathcal{G} is connected. Any spanning tree is made up of $N + 1$ nodes and N links. Let also h_j denote the number of links (hops) from the node j to the RSU, $j = 1, \dots, N$, i.e., if we consider a spanning tree with the RSU as source, then h_j represents the *depth* of the node j .

Given a specific spanning tree \mathcal{T} , we can evaluate the required minimum value of B_{tx} for that tree. We denote this value with $B_{tx}(\mathcal{T})$. An ideal protocol, i.e., with ideal communication channels (i.e., no re-transmissions, no contention overhead, only static overhead), a complete knowledge of the network topology (i.e., \mathcal{T}) and a perfect control of message scheduling (i.e., ideal centralized control) could operate as follows. Each node waits for all its child nodes to send their FCD messages. Then, it aggregates all the received FCD messages with its own one and packs the whole of these information into a single block of data, to minimize the impact of the static overhead. The aggregation principle just stated can be applied recursively by each node, starting with leaf nodes, that do not have to wait for any other node. As a consequence, the total amount of static overhead is NH . On the other hand, the FCD message of a node j must be transmitted h_j times in order to reach the RSU. Thus:

$$B_{tx}(\mathcal{T}) = NH + L \sum_{j=1}^N h_j \quad (1)$$

The number of hops h_j is lower bounded by h_j^* , i.e., the minimum number of hops from node j to the RSU. This is but the length of the shortest path from node j to the RSU *in the original graph* \mathcal{G} . The shortest path lengths h_j^* of the graph \mathcal{G} can be efficiently computed, e.g. by using

the Dijkstra algorithm. We finally get:

$$B_{tx}(\mathcal{T}) \geq B_{tx}^* = NH + L \sum_{j=1}^N h_j^*, \quad \forall \mathcal{T} \subseteq \mathcal{G} \quad (2)$$

This lower bound can be computed once the connectivity graph \mathcal{G} is given. It provides also an upper bound for the performance metric ρ , namely $\rho = B_{FCD}/B_{tx} = ML/B_{tx} \leq NL/B_{tx}^* \equiv \rho_{upper}$.

The graph model provides us with a bound also for the number of relay nodes, R , i.e., those nodes that forward FCD of other nodes, besides their own FCD. With reference to the shortest path tree rooted at the RSU, a lower bound of the number of relay nodes R that is consistent with the overhead bound B_{tx}^* is obtained by counting the intermediate nodes of the shortest paths without repetitions (if there are partially overlapped paths). The resulting lower bound R^* yields a lower bound of the fraction of relay nodes (FRN), i.e., $FRN^* = R^*/N$.

The performance bounds B_{tx}^* and R^* are used to qualify the proposed FCD collection protocol, given that any other protocol cannot improve our proposal further than the performance level predicted by the bounds.

4. The DISCOVER protocol

Let us consider the graph \mathcal{G} formed by the RSU and the N vehicles that are connected among themselves and to the RSU at time t . The main idea of DISCOVER is to select a sub-set of vehicles to act as *Relay Nodes (RN)*, thus creating a temporary backbone network that will be used for data dissemination and collection. In terms of the graph \mathcal{G} , the backbone of RNs should ideally form a minimum covering node set \mathcal{C}_{RN} of \mathcal{G} .

The operations of DISCOVER are organised in two phases (Fig. 1): a so called *forward wave*, characterized by the outbound propagation of a triggering message issued by the RSU. The forwarding process is used to elect the RNs, while the message dissemination expands over the target RoI centered at the RSU (hence the name of forward wave). A request message broadcasted by the RSU is received by vehicles traveling in the RSU coverage area and then forwarded across the network in a multi-hop fashion. The message dissemination phase (*forward wave*) goes on up to a given number of hops, defined according to the desired RoI. When the forwarding has reached the target number of hops, a *reverse wave* begins. From peripheral nodes of the RoI *Reply* messages crawl back inbound towards the RSU, hopping through the designated RNs, this time following a backward path to the RSU. The *Reply* messages carry the FCD in their payloads.

Periodic exchange of single hop hello messages (CAMs) among neighboring vehicle nodes is assumed, as envisaged by the ETSI standard [15]. This background beaconing process guarantees that local databases of vehicle nodes, the so called LDMS, are updated regularly.

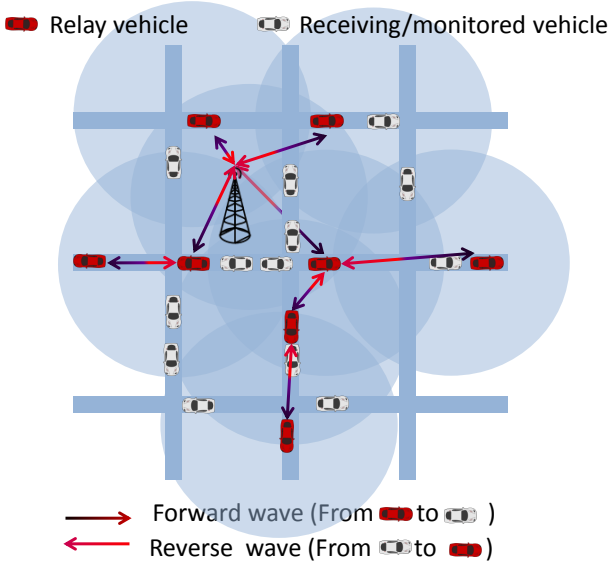


Figure 1: Example of forward wave and reverse wave of DISCOVER.

Both the forward (dissemination) and reverse (collection) waves are carried out by following local, autonomous rules. Each vehicle node exploits local positioning information, as provided by its local sensors, the information stored in its LDM, and the information received in the messages exchanged in the DISCOVER protocol operations. Two types of messages are defined:

1. *Request*: message originated by the RSU and sent during the forward wave; these messages create the backbone network by triggering the election of the RNs.
2. *Reply*: message sent by the RNs back to the RSU; these messages contain the FCD collected over the RoI spanned during the forward wave.

A Request message is represented by the tuple $\langle TM, ID, PO, HL, HC \rangle$, where:

- TM* Type of Message: discriminates between Request and Reply messages;
- ID* Identification: is a unique message identifier (e.g., a counter);
- PO* Position: gives the geographical coordinates of the transmitting vehicle node;
- HL* Hop Limit: a positive integer set by the RSU according to the extension of the desired RoI;
- HC* Hop Countdown: non negative integer field, initialized at *HL* by the RSU and decremented by each forwarding node; it is used to count down the hops traveled by the message.

A Reply message is represented by the tuple $\langle TM, ID, B \rangle$, where B is a data structure containing the reported FCD; those are obtained by merging information belonging to more than one LDM, as explained in Sec. 4.2.

4.1. The forward wave phase of DISCOVER

The basic idea is to select RNs so that any two closest RNs are nominally spaced out evenly by a target distance D . Next we describe a generic step of the forward wave. Let \mathcal{N}_V denote the set of neighbouring vehicle nodes of V ; this is but the set of vehicle nodes whose data are stored in the LDM of V . Let V_{TX} be a sending vehicle node and V_{RX} be a generic neighbor vehicle node of V_{TX} , i.e., $V_{RX} \in \mathcal{N}_{V_{TX}}$. Assume V_{TX} has just elected itself as a RN at time t_0 ; hence V_{TX} sends out a *Request* message instance.

Upon reception of the instance of the *Request* message coming from V_{TX} , V_{RX} checks whether it has already processed a *Request* with the same ID (sequence number). Formally this corresponds to a flag¹ $\phi(ID) = 1$. In that case, the message is discarded.

Otherwise, V_{RX} checks whether it is positioned in a desired location with respect to V_{TX} , i.e., it verifies that $D - \beta D \leq \overline{P_{V_{TX}} P_{V_{RX}}} \leq D + \beta D$, where $\beta \in (0, 1)$ and $\overline{P_A P_B}$ denotes the Euclidean distance between the points P_A and P_B , corresponding to the positions of the vehicle nodes A and B . If that is not the case, V_{RX} discards the message and does no further action.

Otherwise, V_{RX} calculates the quantities $d_0 = |\overline{P_{V_{RX}} P_{V_{TX}}} - D|$ and $d_v = |\overline{P_v P_{V_{TX}}} - D|$ for each $v \in \mathcal{N}_{V_{RX}}$. Note that V_{RX} can calculate those distances thanks to the information about its neighbors stored in its LDM. The vector made up by the distances d_0 and $d_v, v \in \mathcal{N}_{V_{RX}}$ is sorted in ascending order. Let r be the ranking of the distance d_0 . The vehicle node V_{RX} gets elected as a RN if $r = 1$, i.e., if $d_0 \leq d_v$ for all $v \in \mathcal{N}_{V_{RX}}$. In that case, V_{RX} sets a timer with a value chosen randomly in the interval $[0, T_d]$. As the timer expires, V_{RX} passes its instance of the *Request* message to its VANET communication interface for transmission. It also sets $\phi(ID) = 1$ and deems itself as elected to be a Relay Node; then its participation to the current forward wave as concluded. If instead it turns out that $r > 1$, V_{RX} waits for a time $(r - 1)T_d$ (*back-up timer*), to verify if one of its neighbouring nodes that are more qualified to take the role of RN do actually send out the *Request* message. If that is the case, V_{RX} is inhibited, since one of its neighbors has elected itself as a RN. Then, V_{RX} sets $\phi(ID) = 1$ and considers itself a plain (non Relay) node; then its participation to the current forward wave as terminated. If instead V_{RX} does not receive any *Request* issued by any of its $r - 1$ neighbor nodes having higher ranks within a time interval $(r - 1)T_d$, then V_{RX} takes over as the elected RN, draws a random timer value in $[0, T_d]$ and schedules its *Request* message.

¹At the start of the forward wave triggered by the *Request* message labelled with ID , each vehicle node has $\phi(ID) = 0$.

The algorithm run by any vehicle node during the forward wave is stated formally in Alg. 1.

Algorithm 1 RELAY NODE ELECTION

```

1:  $ID$ : ID of the Request message received from  $V_{TX}$ 
2:  $RelayNode$ : a boolean which is set to TRUE if the vehicle elects itself as a relay node and FALSE otherwise
3:  $V_{TX}$ : the transmitting vehicle
4:  $V_{RX}$ : the receiving vehicle
5:  $Vector$ : a vector made of tuples  $\langle v, d \rangle$  where  $v$  is a vehicle and  $d$  is a distance
6:  $T_d$ : a parameter denoting the maximum delay needed by a vehicle to broadcast a message

7: Receive  $Request(ID, HC)$ 
8: if  $flag(ID) = 0$  then
9:    $edist_0 = \text{computeEuclideanDistance}(V_{TX}.coord(), V_{RX}.coord())$ 
10:  if  $edist_0 \geq (D - \beta * D) \wedge edist_0 \leq (D + \beta * D)$  then
11:     $d_0 = |edist_0 - D|$ 
12:     $Vector.add(V_{RX}, d_0)$ 
13:    for all  $v \in LDM_{V_{RX}}$  do
14:       $edist_v = \text{computeEuclideanDistance}(V_{TX}.coord(), v.coord())$ 
15:       $d_v = |edist_v - D|$ 
16:       $Vector.add(v, d_v)$ 
17:    end for
18:    sort  $Vector$  in ascending order according to  $d$ 
19:     $r = Vector.getRank(V_{RX})$ 
20:    if  $r = 1$  then
21:       $RelayNode = TRUE$ 
22:       $replyTimer = \text{setReplyTimer}(T_{max} * (HC/HL))$ 
23:       $\text{schedule}(replyTimer)$ 
24:       $\text{send}(Request(ID, HC - 1))$ 
25:    else
26:       $bTimer = \text{setBackupTimer}(T_d * (r - 1))$ 
27:       $\text{schedule}(bTimer)$ 
28:    end if
29:     $flag(ID) = 1$ 
30:  end if
31: end if

32: upon  $\text{timerExpired}(bTimer)$ 
33: if  $\text{received}(Request(ID, HC - 1))$  then
34:    $RelayNode = FALSE$ 
35: else
36:    $RelayNode = TRUE$ 
37:    $replyTimer = \text{setReplyTimer}(T_{max} * (HC/HL))$ 
38:    $\text{schedule}(replyTimer)$ 
39:    $\text{send}(Request(ID, HC - 1))$ 
40: end if

```

4.2. The backward wave phase of DISCOVER

RNs are in charge of sending the FCD listed in their respective LDMs to the RSU. To limit the replication of data and the transmission of many isolated small messages, a RN waits for further away RNs to send their data back to the RSU before sending its own *Reply* message. In this way, *Reply* messages roll back from the periphery of the covered RoI towards the RSU and get merged as they propagate through the network, hopping from RN to RN, back to the RSU.

As soon as a vehicle node elects itself as a RN, it schedules its *Reply* message with a delay $Reply_Timer$ com-

puted as:

$$Reply_Timer = T_{max} \frac{HC}{HL} \quad (3)$$

where T_{max} is the maximum value of the completion time of the whole DISCOVER operations. If the ordered distance vector set up during the forward wave can attain depth r_{max} at most, then a suitable value for T_{max} is $T_{max} = HL \cdot r_{max} T_d$.

According to eq. (3), the RNs that are closer to the RSU have a bigger timer with respect to further away RNs. In this way inner RNs hold back enough time to receive the *Reply* messages from outer RNs and are thus able to aggregate the received data before replying. This timer setting triggers a reverse wave of *Reply* messages. Let us consider a subtree \mathcal{ST}_X of \mathcal{T} (the spanning tree created in the dissemination phase) rooted at the RN X . Then the data structure collecting all the FCD that X sends to the RSU, i.e., B_X , is created by merging the LDMs of all vehicles belonging to \mathcal{ST}_X , including X itself. Thanks to merging, duplicated FCD are removed.

An example of how DISCOVER works is given in Figure 2. In particular, we can see that $V1, V2, V3$ and $V4$ are the vehicles receiving the Request message from the RSU, since they are in the RSU's transmission range. Upon receiving the Request, each of these vehicles triggers the RN election algorithm (Algorithm 1). According to the distance vector (left table in Figure 2), which is computed locally by each vehicle, $V3$ becomes a RN, hence it forwards the Request message. $V1, V2$ and $V4$ set up their corresponding backup timers according to their position in the distance vector. When they receive the Request sent by $V3$, they infer that $V3$ actually became a RN and cancel their backup timers. Once $V3$ forwards the Request, the process is repeated and $V5$ becomes the next RN. Besides forwarding the Request, every elected RN sets up a reply timer according to the eq. (3). In this example, $V3$ and $V5$ set up their reply timers. Thus $V5$ will send its Reply message before $V3$. In this way, $V3$ will aggregate the information received from $V5$ with its local information before transmitting its own Reply message. Suppose $LDM_{V3} = [V1, V2, V3, V4, V5, V6]$ and $LDM_{V5} = [V3, V4, V5, V6, V7]$. When $V5$'s reply timer expires, it creates a Reply message having $\langle TM, ID, B_{V5} \rangle$, where $B_{V5} = [V3, V4, V5, V6, V7]$ and broadcast it. Similarly, when $V3$'s reply timer expires, it creates a Reply message having $\langle TM, ID, B_{V3} \rangle$, where $B_{V3} = [V1, V2, V3, V4, V5, V6, V7]$. Notice that B_{V3} contains local information from LDM_{V3} , merged with the information contained in $V5$'s reply message and contained in B_{V5} .

4.3. DISCOVER+: Adding historical road traffic information

DISCOVER is purposely designed without requiring any a priori knowledge of the road network. If information about the road network were available, then the RN election could be designed so as to prefer those vehicles

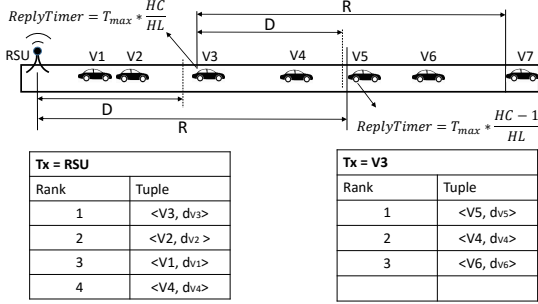


Figure 2: An example of RN election on a single road segment.

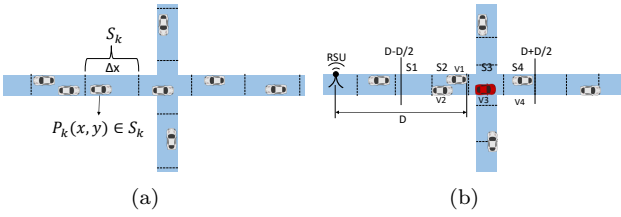


Figure 3: Example of road segmentation (a) and DISCOVER+ operation (b)

nodes that can cover potentially more vehicles residing on different roads. Such preferred RNs reside usually near intersections, squares, or on road segments with less obstacles around. The main idea is to divide the road network in segments and to assign to each segment a weight in order to identify such preferred road segments.

A given area of urban road map is considered. A space quantum Δx is defined and it is assumed that each road is segmented in slices with linear length Δx ². Let the road slices in the RoI be identified by an integer ranging from 1 to K and let S_k denote the k -th road slice, $k = 1, \dots, K$. A vehicle node V positioned at P_V is said to be in the k -th road slice if $P_V \in S_k$ (see Fig. 3(a)).

By collecting FCD of the vehicles moving in the RoI over an observation time window it is possible to collect the number of neighbouring nodes of each vehicle travelling in each S_k during the observation time window. Those numbers are appended to a list \mathcal{L}_k for the segment S_k , $k = 1, \dots, K$ (see Algorithm 2).

Once the observation time window is over, the information inside \mathcal{L}_k is used to compute a weight W_k for every road slice S_k . The weight is represented by the average of the elements of \mathcal{L}_k .

The assumption is that at the beginning of the data collection process every vehicle knows the road network segmentation, as well as the weights associated to every road slice³. This information is used in the RN election process. In particular, differently from the basic DISCOVER

Algorithm 2 RELAY NODE ELECTION

- 1: S_k : the k^{th} road slice.
- 2: \mathcal{L}_k : the list containing the saved data associated to the k^{th} road slice.
- 3: $P^t(x_v, y_v)$: the geographical position of vehicle v at time instance t .
- 4: $N^t(v)$: the number of neighbors of vehicle v at time instance t .
- 5: T : duration of the observation time window.
- 6: $t = 0$;
- 7: **while** $t \leq T$ **do**
- 8: $t = t + 1$;
- 9: **for all** vehicles v into the map **do**
- 10: identify $P^t(x_v, y_v)$;
- 11: find S_k such that $P^t(x_v, y_v) \in S_k$;
- 12: $\mathcal{L}_k = \mathcal{L}_k \cup N^t(v)$
- 13: **end for**
- 14: **end while**

election, where the vehicle closest to a certain distance D from the sender becomes the next RN, DISCOVER+ uses both the distance and the weight to select the RN. More in depth, the distance vector built by a tagged vehicle node V_{RX} in Alg. 1 is replaced by a the following tuple vector: (d_v, W_{k_v}) , where $v \in \{V_{RX}\} \cup \mathcal{N}_{V_{RX}}$, $d_v = |\overline{P_v P_{V_{TX}}} - D|$ and k_v is such that $P_v \in S_{k_v}$. Then, the list of nodes $v \in \{V_{RX}\} \cup \mathcal{N}_{V_{RX}}$ is sorted according to the weight values W_{k_v} . Vehicles sharing the same value of the weight (i.e., belonging to the same road slice) are ordered according to the distance value d_v . As a result, the elected RN is chosen as the vehicle node residing in the road slice with the highest weight; conditionally on the biggest weight criterion, the elected RN is the vehicle whose distance from the transmitting node V_{TX} is closest to D .

An example of how the protocol works can be seen in Figure 3(b). Let the RSU be the sender and $V1, V2, V3, V4$, be the vehicles that received the Request packet and that reside inside $[D - D/2; D + D/2]$ interval ($\beta = 1/2$). Also, suppose $W_{S3} > W_{S2} > W_{S4} > W_{S1}$. According to DISCOVER, $V1$ would be elected as the next RN, since it is the vehicle whose distance from the RSU is closest to D among the set of nodes that have received the message from the RSU. On the other side, according to DISCOVER+, the road slice $S3$ has the highest weight, thus $V3$ is elected as the next RN. In this case it appears that $V3$ will cover more vehicles than $V1$.

5. Performance analysis

The performance analysis of our protocol is divided into the data dissemination phase and data collection phase evaluation. For the first phase we perform a parametric study of DISCOVER with respect to the parameters D and HL . As for the second phase, we compare the performance of our protocol with THEO, which denotes the performance bounds defined in Section 3 (B_{tx}^* and R^*),

²Intersection areas may be considered as a unique space quantum, irrespective of their sizes; this could be applied to squares as well.

³As a matter of example, the RSU could update the weights by

using the FCD collected via DISCOVER and then disseminate them to the roaming vehicles.

as well as with two baseline solutions. The first one is FLOODING: it is a simple and very robust solution. To implement FLOODING, we follow [9]. The idea is that every vehicle broadcasts its own FCD message in the network and rebroadcasts an incoming message only when receiving it for the first time. The goal is to allow the RSU to collect as much FCD as possible from the network. The second baseline reference algorithm is an improved version of FLOODING in which every node broadcasts its FCD message in the network, but, differently from FLOODING, the message coming from a sending vehicle is re-broadcasted only by one randomly chosen neighbor vehicle. We call this solution RANDOM [10][19][20].

We used a multi-layer simulation tool composed by Veins [21], SUMO [22] and OMNET++ [23]. For this purpose, we have configured the simulation tool to employ two main building blocks: the vehicular micro-mobility simulator and the communication network simulator.

5.1. Simulation Stack

We consider a set of real urban maps, obtained by OpenStreetMap [24], with a RSU located at the most central junction. Mobility of vehicles is generated by the micro-mobility simulator SUMO, according to the so called "random trips" model. A flow of vehicle is fed to the map. The vehicle trip start and exit points are selected at random among all road edges (the span of road between two consecutive junctions), with a probability proportional to the number of lanes of the edge. Vehicle routing follows the shortest path between the start and exit points. The movement of the vehicles is governed by the car-following model. The target speed assigned to each vehicle has a Gaussian distribution with mean of 50 km/h and a standard deviation equal to 0.1 times the mean value. The actual speed realized by each vehicle in the simulation results from the interaction of the vehicle's target speed level and the conditioning of its neighboring vehicles.

The OMNET++ simulation tool is used to simulate the behavior of the communications process, including the operations of the Physical (PHY), MAC and network layers. The MAC and PHY parameters are set equal to those specified by the IEEE 802.11p standard. The network layer embeds the implementation of DISCOVER and DISCOVER+.

To model the impact of buildings and other obstacles to signal propagation, we have used jointly two attenuation models: the Two Exponents Model (TEM) [25] and the Simple Obstacle Shadowing Model (SOSM) [26]. The TEM represents the distance dependant component of the power loss: it assumes that the attenuation is $A(d) = \kappa d^{\alpha_1}$, for distances d up to a break point value d_{bp} . For $d > d_{bp}$, it is $A(d) = \kappa d_{bp}^{\alpha_1 - \alpha_2} d^{\alpha_2}$. Typical values of the path loss parameters are $d_{bp} = 120$ m, $\alpha_1 = 2$, and $\alpha_2 = 4$. The SOSM reproduces in Veins the shadowing effect of a real urban environment: it describes the attenuation as a function of the depth of the buildings crossed by radio links.

Table 1: Notation and simulations parameter values

Parameters	Values
$\lambda_{Manhattan}$	70, 110 [veh/km ²]
λ_{Rome}	70, 90 [veh/km ²]
λ_{Paris}	70, 100 [veh/km ²]
λ_{Madrid}	70, 90 [veh/km ²]
Link Rate	6 [Mbit/s]
Node tx power level	27 [dBm]
D	200, 250, 300...750 [m]
β	1/2
$HopLimit$	4, 6, 8, ..., 20
T_d	5 [ms]
T_{max}	900 [ms]
L	14 [byte]
H	40 [byte]
Urban Area	12 [km ²]
Vehicle target mean speed v_{mean}	50 [km/h]
Vehicle speed st. dev. (σ)	$0.1 \cdot v_{mean}$
Propagation Model	TEM + SOS
MAC, PHY parameters	IEEE 802.11p
Carrier frequency	5.9 [GHz]
Noise Floor	-104 [dBm]

We invoke the packet broadcasting operations mode, under which no ACK frames are produced at the MAC layer, as conducted under the IEEE 802.11p MAC specification. Numerical values used to configure the simulation parameters are listed in Table 1.

5.2. Simulation Scenarios

Four real urban scenarios have been created.

The first considered scenario is the district of Manhattan in the city of New York (see Figure 4(a)). This map is mainly characterized by a regular grid of avenues and streets that originate a considerable number of junctions.

The second considered scenario is the neighborhood of Termini Central Station in the city of Rome (Figure 4(c)). Differently from the first scenario, this one is characterized by a high level of irregularity, being composed by roads with widely differing shapes. Also, a chaotic orientation of the streets can be observed. Obviously, this map is much more complex than the first one.

The third simulation scenario is the area around Place Charles de Gaulle in Paris (see Figure 4(b)). This map includes the famous Arc de Triomphe and the Avenue des Champs-Élysées, and is split between the 8th, 16th, and the 17th arrondissements of Paris. The square is surrounded by two streets forming a circle around it. Also, the square is symmetrical and there are six main roads intersecting in this point, splitting the map in *sections*. However, inside those sections there are a plenty of different streets, making this map a typical example of a mixed map having both regular and irregular roads.

The fourth and last considered scenario is the central part of the city of Madrid around the Buen Retiro Park. An interesting characteristic of this map is that, besides being a typical European city with many different roads and many irregularities, a big area without buildings and less obstacles (the Buen Retiro Park indeed) is present in the middle of the map (see Figure 4(d)).

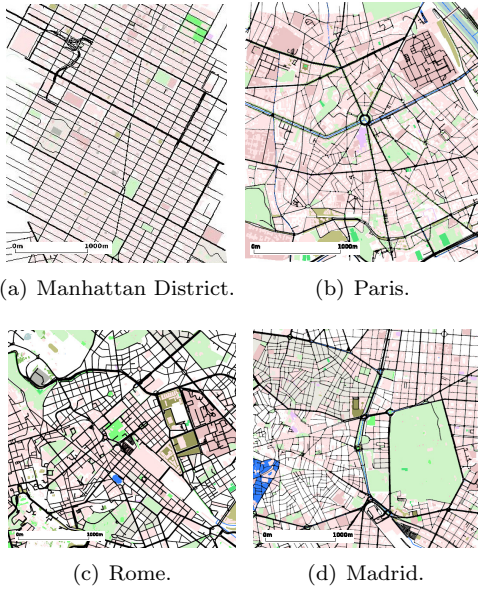


Figure 4: Urban scenarios

5.3. Performance Metrics

Every considered scenario has been analyzed under two different vehicle densities λ (see Tab. 1). We evaluate the following performance metrics by considering all vehicles in the scenario at the time that the RSU issues the *Request* message:

- *FRV* - the fraction of reached vehicles, that is the ratio of vehicles that receive the *Request* message to the total number of vehicles connected to the RSU;
- *FMV* - the fraction of monitored vehicles, that is the ratio of vehicles whose FCD messages arrive at the RSU to the total number of vehicles connected to the RSU;
- *FRN* - the fraction of relay nodes (vehicles that forward the *Request* message);
- *TWT* - two-way time: is the time needed to entirely complete the data dissemination and collection phases;
- *OWT* - one-way time: is the time needed to complete only the data dissemination phase;
- B_{tx} - the total amount of bytes transmitted on air by the vehicles during the whole procedure, including FCD and any overhead;

The performance analysis is focused on the two main phases of DISCOVER: data dissemination and data collection. For each metric we have estimated the 95% level confidence interval normalised with respect to the estimated value (relative confidence intervals). Relative confidence intervals are well below 10%, so they are not shown in the figures.

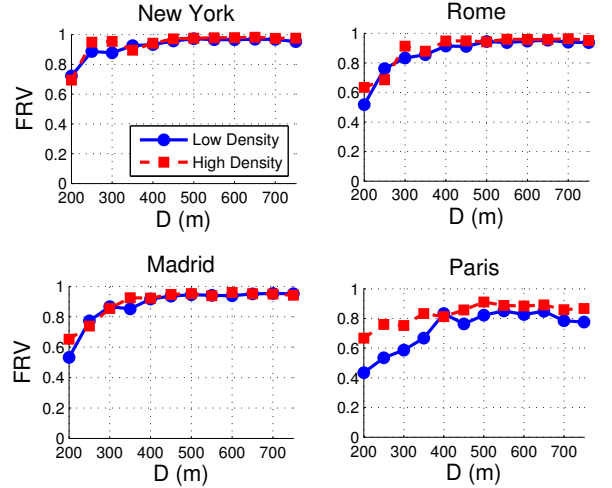


Figure 5: The Fraction of Reached Vehicles when varying the parameter D .

5.4. Data dissemination: the forward wave

In the data dissemination phase the message sent by the RSU should reach as many vehicles as possible within the target RoI, independently of the considered scenario. We show that the proposed dissemination scheme gives good results in term of coverage in widely different urban scenarios. DISCOVER has been tested under different vehicle densities, investigating the effect of the parameter D over the metrics relevant for the dissemination phase, namely *FRV*, *FRN* and *OWT*. Since simulations are run on a delimited area, defined by the considered urban map, and assumed to coincide with the target RoI, we do not consider the HopLimit in this first set of simulations. In Table 1 the average values of vehicular density for New York, Rome, Madrid and Paris for two different traffic congestion scenarios can be found.

In Figure 5 the results in terms of *FRV* are presented, when varying the parameter D . It can be noted that the *FRV* curve has a constant trend for values of D lying in the interval $450 \text{ m} \leq D \leq 700 \text{ m}$, with values of *FRV* around 0.9 for all the considered scenarios and vehicular densities. For small D levels, e.g., for $D = 200 \text{ m}$, it can be observed that $FRV \approx 0.75$ for *Manhattan* and $0.45 \leq FRV \leq 0.65$ for the other considered scenarios. A reason for such result is that for small values of D , the small distance between consecutive RNs might increase the number of packet collisions, thus obstructing the dissemination process. Another reason is that the algorithm is designed to allow only vehicles whose distance from the sending node falls into the interval $[D - \beta D, D + \beta D]$ to participate in the election process. Thus, there might be such cases for small λ and small D when there is no vehicle inside this area and the dissemination process stops.

As for the fraction of nodes that elect themselves as RNs, a constant trend is observed by varying D for each value of λ (see Figure 6). However, *FRN* varies noticeably

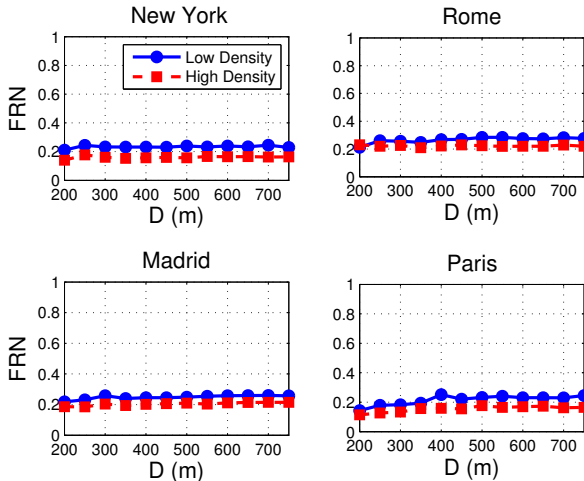


Figure 6: The Fraction of Relay Nodes when varying the parameter D .

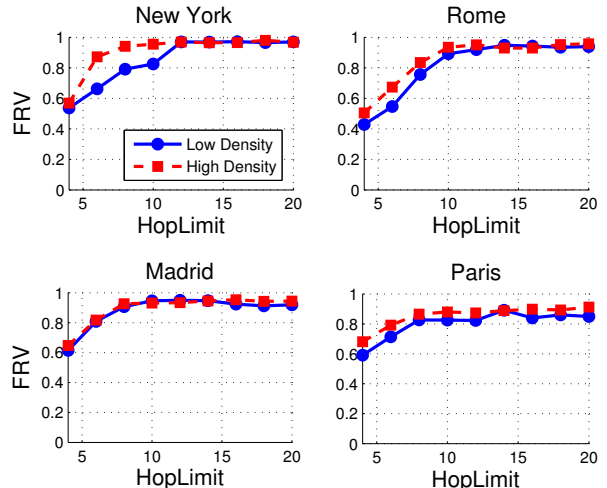


Figure 8: The Fraction of Reached Vehicles when varying the parameter HL .

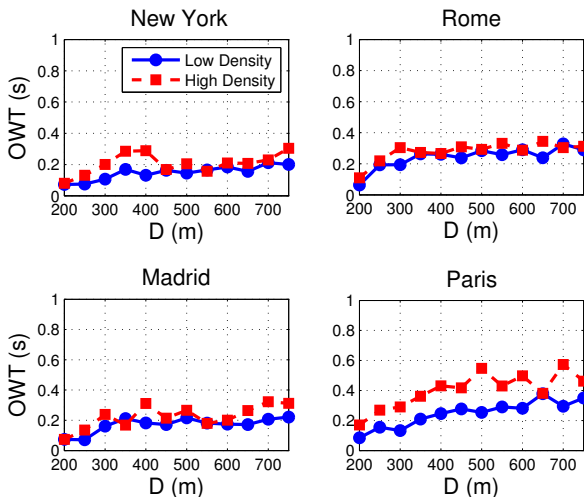


Figure 7: DISCOVER dissemination duration.

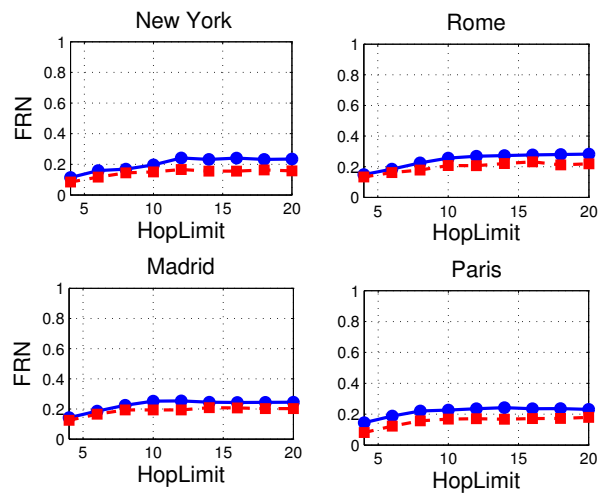


Figure 9: The Fraction of Relay Nodes when varying the parameter HL .

with the mean vehicle densities. In particular, $0.23 \leq FRN \leq 0.28$ for Low Density and $0.15 \leq FRN \leq 0.20$ for High Density, for all the considered scenarios. This means that for high densities the percentage of RNs is lower with respect to small densities, i.e., the dissemination process gets more efficient.

Figure 7 shows that the time needed to complete the dissemination phase depends on the parameter D . In particular, for lower values of D the OWT is smaller, which is intuitive, since there are less reached vehicles. However, what we are interested in is to know the performance in terms of OWT when the FRV is at the maximum. In this case the values of OWT ranges from a minimum of 0.21 s for *Manhattan* to a maximum of 0.57 s for *Paris*.

After performing simulations for different values of D we found that DISCOVER gives the best results for 450 m \leq

$D \leq 750$ m, independently of the simulation scenario. We set $D = 500$ m to evaluate all the other metrics below.

Another parameter that might influence the protocol performances is the HL . We simulated DISCOVER for different values of HL and for $D = 500$ m. The results in terms of FRV are shown in Figure 8. It appears that the FRV metric is constant for $HL \geq 12$. This means that for $HL \geq 12$ and $D = 500$ m the message sent by the RSU reaches around 90% of vehicles for all the considered scenarios, an expected outcome considering the results of the dissemination phase. On the other hand, for $HL < 12$ a reduction of FRV can be seen. However, this result does not mean that the algorithm is weak for small HL , but it simply reflects a reduction of the target RoI.

As for the FRN , the results in Figure 9 are fully consistent with those presented in Figure 6. In particular, for

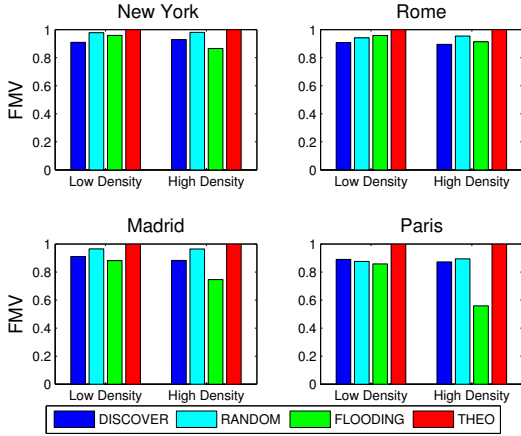


Figure 10: Fraction of Monitored Vehicles 2 different vehicular densities in the district of Manhattan, NY, Rome, Madrid and Paris.

$HL = 12$ and $D = 500$, the values of FRN range from 0.23 to 0.28 for Low Density and from 0.15 to 0.20 for High Density, for all the considered scenarios.

5.5. Collection phase: the reverse wave

The goal of this second phase is to collect data from the vehicles roaming in the RoI. We investigate the case where the collected data contains vehicles' geographical position. In general, information can be collected on vehicles' motion parameters (e.g., for vehicular traffic monitoring and controlling purposes) or from the on board sensors (e.g., to estimate pollution, quality of the street surface, etc.)

The dissemination process creates a *backbone network*, formed by the elected RNs, that will be responsible for sending data back to the RSU. In particular, every RN will send information about its own neighbors, collected during the beacon exchange phase.

In this Section we compare DISCOVER with THEO, FLOODING and RANDOM. Since THEO provides a bound to the best performance obtainable as for FMV , FRN and B_{tx} , we intend to show that DISCOVER gets close to THEO, rather than comparing DISCOVER with any of the different heuristic algorithms that can be found in the literature. Notice that the performance bound given by THEO in terms of FMV is always equal to 1. On the other hand, achieving performance metrics close to THEO is non trivial as shown by the results provided by two baseline heuristics, namely RANDOM and FLOODING. Those two simple protocols will be shown to yield much worse performance results than THEO (and DISCOVER).

The Figure 10 shows the fraction of vehicles whose FCD messages arrived to the RSU at the end of the data collection phase (FMV) in New York, Rome, Madrid and Paris. The value of D is set equal to 500, while $HL = 12$. It can be noticed that DISCOVER, FLOODING and RANDOM are able to reach similar performance results

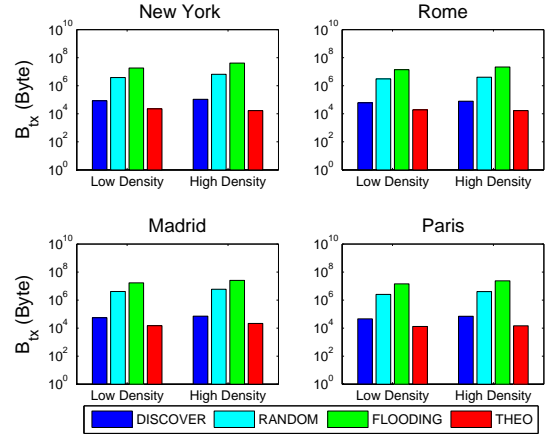


Figure 11: B_{tx} for 2 different vehicular densities in the district of Manhattan, NY, Rome, Madrid and Paris.

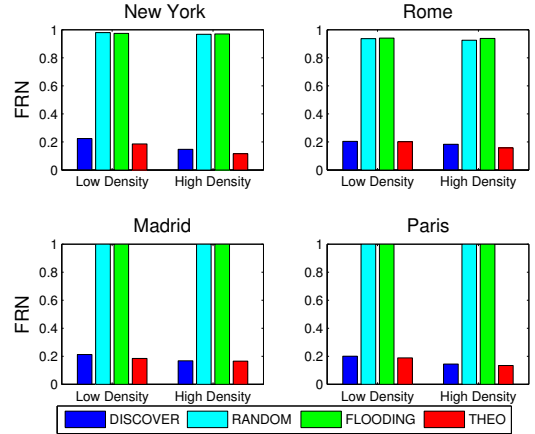


Figure 12: Fraction of Relay Nodes and Measured Overhead for 2 different vehicular densities in the district of Manhattan, NY, Rome, Madrid and Paris.

and in average very close to the optimal results represented by THEO, independently of the considered scenario.

However, the real performance difference introduced by DISCOVER is perfectly depicted in the other metrics. Firstly, Figure 11 shows how DISCOVER is able to reduce the total amount of bytes transmitted (B_{tx}) during the whole data collection process. As we expected, FLOODING and RANDOM are very distant from THEO; due to their intrinsic simplicity and distributed feature, these algorithms do not use any kind of information about topology (i.e., neighborhood knowledge) and vehicles (i.e., vehicle position), they are perfectly capable to collect data, as depicted in Figure 10 in terms of FMV , but not in an efficient way: the total amount of bytes transmitted is orders of magnitude greater than the optimal (theoretical) value. On the other hand, it can be noticed that the total amount of bytes transmitted by DISCOVER is very close to the theoretical optimal value.

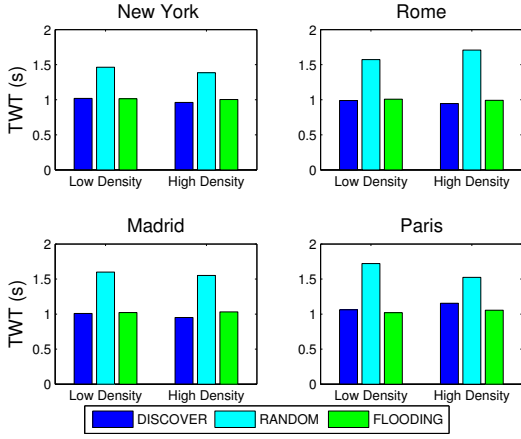


Figure 13: Two-way time for 2 different vehicular densities in the district of Manhattan, NY, Rome, Madrid and Paris.

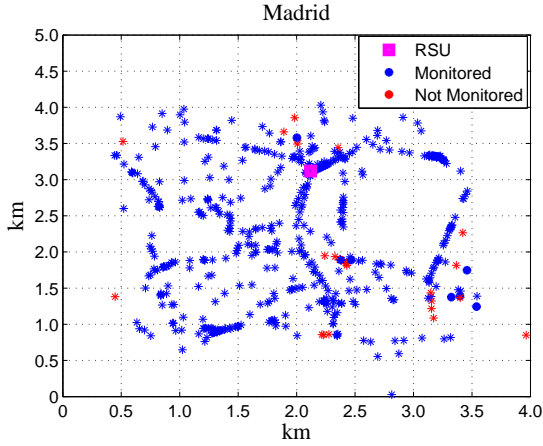


Figure 14: Example of the RSU's view of the vehicles in the target area of Madrid using DISCOVER.

The metric in Figure 12, FRN , gives us a measure of the redundancy level obtained by the different algorithms. Figure 12 shows that, with FLOODING and RANDOM, almost all nodes are involved in the forwarding operation of at least one packet, while DISCOVER, with its preliminary data dissemination phase, is able to maintain the number of RNs very close to the correspondent THEO value.

Another aspect that we must consider is the total time needed to collect FCD data (TWT). The result shown by Figure 13 is that DISCOVER has almost the same performance as FLOODING in terms of TWT , while RANDOM, due to its timers used to reduce the B_{tx} , appears to be the worst in terms of delay.

Finally, in Figure 14 we show the final result obtained with DISCOVER. The figure represents the RSU view, at the end of the DISCOVER collection procedure, related to the vehicular traffic for the target area Madrid. In this particular case, it is shown that only few vehicles are not monitored at the RSU location, but despite this fact, the

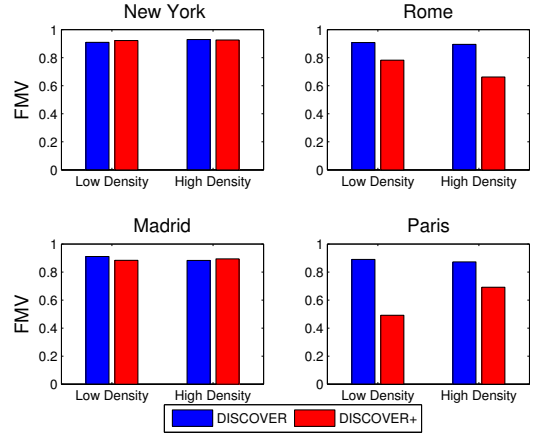


Figure 15: Fraction of Monitored Vehicles for two different traffic congestion scenarios: a) Low Density; b) High Density.

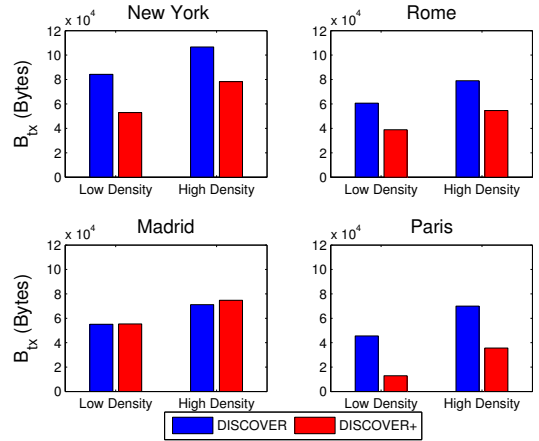


Figure 16: B_{tx} for two different traffic congestion scenarios: a) Low Density; b) High Density.

RSU is perfectly able to build a map of the vehicles located in a target area, well beyond its physical radio coverage area.

5.6. Analysis of DISCOVER+

In the previous subsection we have seen that the performances of DISCOVER are similar for all the considered urban maps and/or vehicular densities. Here we are interested in examining if there is any benefit in adding road network and historical traffic information to the protocol operation logic. The performances of DISCOVER+ are compared with the original DISCOVER protocol.

Figure 15 presents the results in terms of FMV for *New York*, *Rome*, *Madrid* and *Paris*. It can be noted that DISCOVER+ has similar performance as the original DISCOVER for *New York* and *Madrid*, while for *Rome* and *Paris* the FMV decreases significantly. As for the overhead metric, the only urban area that highlights some improvement in terms of B_{tx} is *New York*, as can be seen from

Fig. 16, while there is no gain for all the other considered scenarios. Notice that the difference in terms of B_{tx} between DISCOVER and DISCOVER+ for *Rome* and *Paris* is not related to a better performance of the latter, but to a reduced number of monitored vehicles as clear from the *FMV* levels.

The indication emerging from these results is that DISCOVER+ gains some overhead performance improvement, while achieving the same level of vehicle monitoring in the target area as DISCOVER does, in regular road grid maps as *New York*. On the contrary, it behaves much worse when considering more complex urban maps, such as *Madrid*, *Rome* and/or *Paris*. This puts a strong warning on the value of leveraging from time series statistics of the road traffic to improve FCD collection protocols.

6. Conclusions

In this work we propose DISCOVER, a protocol for VANETs, that can be used both for data dissemination and collection in a given city area. The novelty of the proposed protocol is that, in a distributed and adaptive way, it forms a backbone structure, constituted by some selected vehicles, through which a high percentage of vehicles traveling the city roads can be reached by data sent by a Road Side Unit. In the opposite direction, from vehicles to a collection center, the same structure once formed can be used to send information in a very quick and efficient way. The evaluation, via simulations, of DISCOVER shows that in quite different cities (New York, Madrid, Paris and Rome) the percentage of reached vehicles is always around 90% of those roaming in the target RoI. When compared to other protocols it results that DISCOVER uses a reduced number of relay nodes (vehicles that forward the protocol messages), thus resulting in a low load on the wireless communication channel. A suitable usage of the data collection tree formed by DISCOVER also allows to keep the number of transmitted bytes quite low, close to the theoretical lower bound for the given VANET connectivity. As for the delay performance, the two-way time (i.e., the time needed to entirely complete the data dissemination and collection phase) turns out to be quite low (below 1.5 seconds) thus giving the possibility to use DISCOVER for several future ITS applications. Moreover, we proved that adding road network and historical traffic information in the protocol operation logic does not bring any performance gain, except in case the considered urban map exhibits a highly regular road grid. On the contrary, for more complex urban scenarios, such approach reduces the number of the monitored vehicles.

References

[1] H. T. Cheng, H. Shan, W. Zhuang, *Infotainment and road safety service support in vehicular networking: From a communication perspective*, Mechanical Systems and Signal Processing 25 (6).

[2] I. Turcanu, P. Salvo, A. Baiocchi, F. Cuomo, *DISCOVER: A Unified Protocol for Data Dissemination and Collection in VANETs*, in: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN '15, ACM, New York, NY, USA, 2015, pp. 25–32.

[3] M. Chaqfeh, A. Lakas, I. Jawhar, *A survey on data dissemination in vehicular ad hoc networks*, Vehicular Communications 1 (4) (2014) 214 – 225.

[4] I. Rubin, A. Baiocchi, F. Cuomo, P. Salvo, *GPS aided inter-vehicular wireless networking*, in: Information Theory and Applications Workshop (ITA), 2013, 2013, pp. 1–9.

[5] W. Viriyasitavat, F. Bai, O. Tonguz, *UV-CAST: An urban vehicular broadcast protocol*, in: Vehicular Networking Conference (VNC), 2010 IEEE, 2010, pp. 25–32.

[6] B. Brik, N. Lagraa, H. Cherroun, A. Lakas, *Token-based Clustered Data Gathering Protocol (TCDDGP) in vehicular networks*, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, 2013, pp. 1070–1074.

[7] W.-R. Chang, H.-T. Lin, B.-X. Chen, *TrafficGather: An Efficient and Scalable Data Collection Protocol for Vehicular Ad Hoc Networks*, in: Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, 2008, pp. 365–369.

[8] A. Soua, H. Afifi, *Adaptive data collection protocol using reinforcement learning for VANETs*, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, 2013, pp. 1040–1045.

[9] Z. He, H. Zhang, *Density Adaptive Urban Data Collection in Vehicular Sensor Networks*, Journal of Networks 9 (8).

[10] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu, *The broadcast storm problem in a mobile ad hoc network*, in: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, 1999, pp. 151–162.

[11] Y. Zhu, Q. Zhao, Q. Zhang, *Delay-Constrained Data Aggregation in VANETs*, Vehicular Technology, IEEE Transactions on PP (99) (2014) 1–1.

[12] P. M. d'Orey, N. Maslekar, I. de la Iglesia, N. K. Zahariev, *NAVI: Neighbor-Aware Virtual Infrastructure for Information Collection and Dissemination in Vehicular Networks*, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), 2015, pp. 1–6.

[13] I. Salhi, M. Cherif, S. Senouci, *A New Architecture for Data Collection in Vehicular Networks*, in: Communications, 2009. ICC '09. IEEE International Conference on, 2009, pp. 1–6.

[14] G. Remy, S.-M. Senouci, F. Jan, Y. Gourhant, *LTE4V2X - Collection, dissemination and multi-hop forwarding*, in: Communications (ICC), 2012 IEEE International Conference on, 2012, pp. 120–125.

[15] E. T. S. Institute, ETSI TS 302 637-2 v1.3.2; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service (November 2014).

[16] E. T. S. Institute, ETSI EN 302 895 v1.1.1; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM) (September 2014).

[17] E. T. S. Institute, ETSI EN 302 637-3 v1.2.0; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service (August 2013).

[18] D. Bertsekas, R. Gallager, *Data Networks*, 2nd Edition, Prentice Hall, 1992.

[19] Y.-C. Tseng, S.-Y. Ni, E.-Y. Shih, *Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network*, Computers, IEEE Transactions on 52 (5) (2003) 545–557.

[20] Q. Huang, Y. Bai, L. Chen, *Efficient Lightweight Broadcasting Protocols for Multi-Hop Ad Hoc Networks*, in: Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, 2006, pp. 1–5.

[21] C. Sommer, R. German, F. Dressler, *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Anal-*

- ysis, IEEE Transactions on Mobile Computing 10 (1) (2011) 3–15.
- [22] D. Krajzewicz, C. Rossel, Simulation of Urban Mobility (SUMO), German Aerospace Centre, available at: <http://sumo.sourceforge.net/index.shtml> (2002).
 - [23] OMNeT++ Network Simulation Framework, available at: <http://www.omnetpp.org> (2001).
 - [24] OpenStreetMap, available at: <http://www.openstreetmap.org> (2001).
 - [25] H. Hartenstein, K. Laberteaux, *VANET Vehicular Applications and Inter-Networking Technologies*, Intelligent Transport Systems, John Wiley and Sons, 2009.
 - [26] C. Sommer, D. Eckhoff, R. German, F. Dressler, *A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments*, in: Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on, 2011, pp. 84–90.