DISCOVER: A Unified Protocol for Data Dissemination and Collection in VANETs

Ion Turcanu
University of Rome Sapienza
Rome, Italy
ion.turcanu@uniroma1.it

Pierpaolo Salvo
University of Rome Sapienza
Rome, Italy
pierpaolo.salvo@uniroma1.it

Andrea Baiocchi
University of Rome Sapienza
Rome, Italy
andrea.baiocchi@uniroma1.it

Francesca Cuomo
University of Rome Sapienza
Rome, Italy
francesca.cuomo@uniroma1.it

ABSTRACT

Message dissemination and data collection from vehicles are two key enablers of Intelligent Transportation System services that can be offered by a Vehicular Ad-Hoc Network (VANET) technology. In this work we propose a fully distributed protocol for dissemination of query and collection of reply messages carrying information gathered from vehicles moving in a given target area, in an urban scenario. The key idea is to use the dissemination phase (forward wave) to create a network of relay nodes that are in charge of delivering reply messages back to the originating point (reverse wave). The proposed protocol is evaluated with reference to two real urban environments. Main parameters are dimensioned and an insight in the protocol working is given.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless Communications*; C.2.2 [**Computer Communication Networks**]: Network Protocols—*Routing protocols*

Keywords

Vehicular Ad Hoc Networks; IEEE 802.11p; Data collection; Data dissemination

1. INTRODUCTION

A key role of Vehicular Ad-Hoc Networks (VANETs) is to disseminate packets to a wide set of vehicles traveling along a road. This can be achieved through the use of vehicle-to-vehicle multi-hop communications, enabling the extension of the road span covered by Road Side Units (RSUs) or On Board Units (OBUs) generating the data. This dissemination function is of interest for both safety and infotainment applications [5]. 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.

In this paper, we propose DISCOVER, a data collection protocol for VANET in an urban environment. Our motivation is the support of urban sensing applications. In [6], we proposed a new dissemination protocol, identified as Vehicular Backbone Network (VBN) for highway. In this paper we extend the key idea of that protocol to disseminate and collect data in a complex urban scenario.

There are several contributions of this paper: i) we propose a new protocol able at the same time to disseminate and collect data by using a VANET in an urban scenario, maximizing the amount of the collected information and minimizing the total overhead; ii) we provide a solution that overcomes the radio blindness of vehicular nodes due to the buildings in an urban environment; iii) we build a set of realistic simulation scenarios, based on the urban maps of New York and Rome; iv) we give a performance evaluation of our protocol and an analytical comparison with the theoretical bound, as well as with other two baseline solutions, here referred to as RANDOM (eg. approaches in [17][9][14]) and FLOODING.

The rest of the paper is structured as follows. Section 2 provides an overview of the related work. Section 3 defines the theoretical performance bound. Section 4 describes our proposed protocol. Section 5 presents the performance results while conclusions are drawn in Section 6.

2. RELATED WORK

A VANET dissemination logic is used to select a sub-set of vehicles, that are situated along the road, to act as relay nodes. This is done to avoid the broadcast storm problem [18]. The aim here is to select as relaying vehicles those that are located at preferred positions, while inhibiting others. In [6], a new dissemination protocol, identified as Vehicular Backbone Network (VBN) has been proposed. Under VBN, the messages sent out by a RSU are forwarded 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 only for disseminating data on highways, we extended this protocol in order to disseminate and collect data in urban scenarios.

A number of works have investigated data collection schemes. In [16] the authors propose a protocol based on a distributed Qlearning technique to make the collecting operation more reactive to nodes mobility and topology changes. In [4] Brik et al. propose a Token-based Clustered Data Gathering Protocol (TCDGP). In this protocol the road is divided in segments, and for each segment a Cluster Head (CH) vehicle is elected based on its position from the center of the segment and on how much time it has has traveled on the middle lane. The two above mentioned solutions were designed to collect data on highways, while our protocol aims at collecting data in urban environments.

A two-way data collection scheme is proposed by He and Zhang in [8]. Their algorithm is divided in two phases: the dissemination phase and the collection phase. A message sent by a BS (Base Station) is disseminated possibly to all vehicles using broadcasting. In order to deal with the broadcast storm problem [18], the authors propose two mechanisms: *Rebroadcast Filtering*, which tries to reduce redundant rebroadcasts by allowing vehicles to rebroadcast only when receiving a message for the first time, and *Duty Cycled Execution*, that aims at reducing contention and collisions by allowing vehicles to broadcast once in a fixed duty cycle. Once receiving a request message, besides continuing the dissemination process, each vehicle also sends back a reply message containing the requested information. The reply messages are forwarded back to the BS by vehicles selected based on their geographical proximity to the BS itself. Actually, there is no clear distinction between the dissemination and the aggregation phase, since the data to be collected is appended to the request messages sent during the dissemination phase. Since this protocol basically uses a flooding approach to collect data, it induces a large overhead due to multiple re-broadcasting of messages.

In [19] Zhu et al. propose a new approach for delay-constrained data aggregation in VANETs, named *aTree*. They propose a centralized, as well as a distributed version of *aTree*. The main objective of this protocol is to maximize the amount of information collected in an urban environment, while meeting some delay constraint and trying to reduce the transmission overhead during the aggregation process in the VANET. The basic idea of *aTree* is to first construct a data aggregation tree based on the shortest path tree and then to assign a waiting time budget for each node on the tree. The allocation of the waiting time budgets is done in such a way that parent nodes must have larger waiting time budgets in order to have enough time to collect and aggregate the information received from the child nodes. The construction of the tree is initiated by a collection node, which disseminates an *update* message to the whole network. This message is re-broadcasted at least once by every node in the network. Since the goal is to create a shortest path tree, if a node receives an *update* message with less hops and/or less forwarding delay to the collection node, then the message must be re-broadcasted again in order to update all the child nodes. The waiting time budget allocation is initiated by the leaves. Every parent node selects the maximum from all its child nodes as its waiting time budget. This operation is repeated until the collection node is reached, which computes the residual delay with respect to the time constraint and disseminates this information again to the whole network. Of course, all these operations generate a large overhead. Our solution, differently from *aTree*, selects only a subset of relay nodes during the dissemination phase to forward the request message and, simultaneously, allocates the waiting time budgets for every relay node. By minimizing the number of forwarding nodes, we aim at minimizing also the total overhead.

We choose to compare the performance of our protocol with two baseline solutions. The first one is FLOODING, since it is a simple and very robust solution. To implement FLOODING, we used the approach present in [8] and adapted it for collecting data from all vehicles inside an area of interest. The idea is that every vehicle broadcasts its own 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 information as possible from the network. The second solution is an improved version of FLOODING in which every node broadcasts its message in the network, but, differently from FLOODING, the message coming from a sending vehicle is re-broadcasted only by one randomly chosen receiving vehicle. We call this solution RANDOM [17][9][14]. We will show that our protocol is able to collect the messages from the majority of vehicles present in the network, with an overhead close to the minimum theoretical bound.

3. PROBLEM STATEMENT

The considered scenario comprises a RSU and a population of V vehicles moving in a given area around the RSU. Let us freeze the picture of the system at the time the RSU issues a data collection request message. Let N denote the number of vehicles connected to the RSU at that time, possibly through multi-hop relaying through other vehicles ($N \leq V$). Let M be the number of vehicles monitored by the RSU at the end of the collection process ($M \leq N$).

We assume Cooperative Awareness Messages (CAMs) [12] are periodically exchanged among vehicles (one hop message), with a generation interval $100\ ms \leq T_{GenCam} \leq 1000\ ms$, to create and maintain awareness of each other. CAMs are stored in a Local Dynamic Map (LDM) [11] by each vehicle, which is updated every time a new CAM is received.

The record of Floating Cara Data (FCD), contained in CAM, reported to the RSU by each vehicle consists of its ID, position and speed [12][10]. Overall, let L be the length of the FCD message. The net amount of information the RSU should receive is NL. The actual amount of bytes transmitted on the air in the VANET to deliver the FCD records is bigger 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 information from the facility layer of the VANET has length $H + L$); ii) multiple transmissions of a same FCD message due to 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.

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 round of collection. Let B_{FCD} be the actual net amount of bytes of FCD delivered to the RSU (notice that $B_{FCD} = ML$). Usually $B_{FCD} \leq NL$, because not all data arrives to the RSU. Then, a normalized metric can be obtained by considering the ratio $\rho \equiv B_{FCD}/B_{tx}$.

In the following we consider the network graph $\mathcal{G}$ of the N vehicles and the RSU. It comprises $N+1$ nodes; the RSU is conventionally denoted as node 0. An edge exists in this graph between node i and node j iff it is possible for j to receive and decode correctly a block of data transmitted by i ($i, j = 0, 1, \ldots, N$). Let us consider a spanning tree of $\mathcal{G}$ rooted at the RSU. There are many efficient algorithms to find such a tree. It must exist, since we assume the graph $\mathcal{G}$ is connected. Any spanning tree is made up of $N+1$ nodes and N links. We consider only directed links, from a child node to the corresponding parent node. Let also h_j denote the number of links (hops) from the node j to the RSU, $j = 1, \ldots, N$, 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 centralised control) could operate as follows. Each node waits for all its child nodes to send their FCD data. Then, it aggregates all the received FCD records with its own one and packs the whole of these information into a single block of data, to minimise 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 record of a node j must be transmitted h_j times in order to reach the RSU. Thus:

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

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:

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

This lower bound can be computed efficiently, once the connectivity graph $\mathcal{G}$ is given. It provides an ideal bound for performance for both the absolute metric B_{tx} and the normalized one $\rho \leq NL/B_{tx}^*$.

4. THE DISCOVER PROTOCOL

The main idea is to select a sub-set of vehicles to act as *Relay Nodes* (*RN*), thus creating a temporary backbone network that will be used also for data collection. If we consider the network graph $\mathcal{G}$ of the N connected vehicles to the RSU and the RSU itself at time t, then this backbone network represents a spanning tree $\mathcal{T}$ of $\mathcal{G}$ rooted at the RSU. A message broadcasted by the initial message source (e.g., a RSU) is received by vehicles traveling in the source's 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 area of interest. Then a *reverse wave* starts, where relay nodes are responsible for generating and sending data collection messages back to the source.

Two types of packets are defined:

1. *Request*: packet originated by the RSU and sent during the forward wave; these packets create the backbone network by triggering the relay node election.

2. *Reply*: packet sent by the relay nodes towards the RSU; these packets contain the data collected over the region of interest spanned during the forward wave.

A Request packet is represented by the tuple $\langle ID, POS, HL, HLC \rangle$, where ID is the identifier of the packet, POS is the geographical position of the sender, HL is the hop limit defined by the RSU according to the desired area of interest, and HLC is a variable, decremented by each forwarding node, used to count down the hops traveled by the message. A Reply packet is represented by the tuple $\langle ID, B \rangle$, where ID is the identifier of the packet, and B is a data structure containing merged information belonging to more than one LDM.

The process is started by the RSU by sending in broadcast a Request message. Each vehicle has an updated version of its LDM thanks to the background exchange of CAMs. The key idea of the DISCOVER dissemination phase is to divide the service area in circular, partially overlapped sub-region with radius D (see Figure 1), where D is any fixed distance between 0 and the maximum transmission range R of a node (RSU or vehicle), and to elect as RNs the vehicles that reside closest to the center of each sub-region (see Algorithm 1). This special position is referred to as Nominal Relaying Position (NRP).

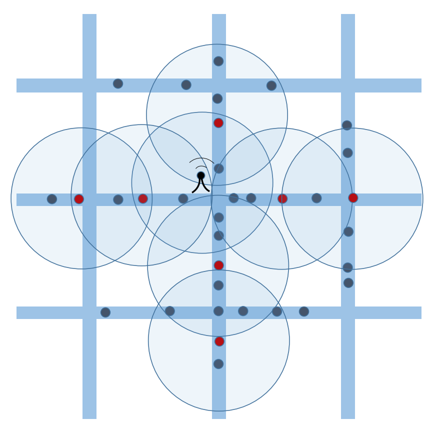


Figure 1: Election of RNs. The circular areas have radius *D*. The red points represent the elected RNs.

Let V_{TX} be a sending vehicle and V_{RX} be a receiving vehicle. Then, V_{RX} checks whether it is the closest to the NRP with respect to V_{TX} among all its neighbors, i.e., it checks whether $dist(V_{RX}, NRP) < dist(v, NRP)$ for any

$v \in LDM_{V_{RX}}$, where $LDM_{V_{RX}}$ is the LDM of the receiving vehicle V_{RX}. If that is the case, V_{RX} elects itself as RN, decrements the value of HLC, updates the POS value with its own position and only if $HLC > 0$ it forwards the Request with a delay chosen randomly between 0 and a maximum delay time T_d. Otherwise, one of V_{RX}'s neighbors is closest to the NRP and hence it is in a better position to take the role of RN. If for some reason the designated RN does not forward the Request, there is a backup mechanism that permits to the next closest to NRP vehicle to become the new RN. The backup mechanism provides that each V_{RX} waits for the potential RN to forward the Request message, to verify that the election process has ended successfully. To this end, every V_{RX} creates a distance vector of its neighbors (including itself) sorted according to their distance from the NRP and sets a timer according to its ranking order in the distance vector multiplied by T_d. If no other copy of the Request message has been received by V_{RX} before its timer expires, V_{RX} infers that no other neighbor closer to NRP has taken the role of RN; hence, V_{RX} elects itself as RN.

Let us consider a tagged RN, say V_{RN}. The Reply is scheduled with a delay computed as:

$$RepTimer_{V_{RN}} = T_{max}\frac{HLC}{HL} \qquad (2)$$

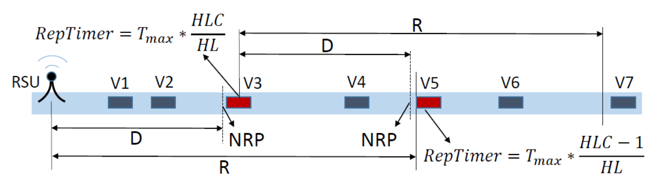
where T_{max} is the maximum value of delay. According to Eq. (2), the RNs that are closer to the RSU have a bigger timer with respect to further RNs. In this way inner RNs have enough time to receive the Reply packets from outer RNs and are thus able to aggregate the received information before replying. This timer setting triggers a reverse wave of Reply packets. If we consider a subtree of $\mathcal{T}$ (the spanning tree created in the dissemination phase), say $\mathcal{T}_{V_{\mathcal{RN}}}$, rooted at V_{RN}, then $B_{V_{RN}}$ (the data structure B of V_{RN}) is created by merging the LDMs of all vehicles belonging to $\mathcal{T}_{V_{\mathcal{RN}}}$.

Algorithm 1 RELAY NODE ELECTION

```
V_TX: the transmitting vehicle
V_RX: the receiving vehicle
NRP: the Nominal Relaying Position, it's the position
     at distance D from V_TX
Vector: a distance-based vector made of tuples <
     v, dist(v) > where v is a vehicle and dist(v) is the dis-
     tance of v from NRP
T_d: a parameter denoting the maximum delay needed
     by a vehicle to broadcast a message

dist(V_RX) = computeDistance(V_RX.coord(), NRP);
Vector.add(V_RX, dist(V_RX));
for all v ∈ LDM_{V_RX} do
    dist(v) = computeDistance(v.coord(), NRP);
    Vector.add(v, dist(v));
end for
sort Vector in ascending order according to dist
if Vector.getFirst() == V_RX then
    return TRUE;
else
    setBackupTimer(T_d * posInVector(V_RX));
    return FALSE;
end if
```

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 trigger the RN election algorithm (see Algorithm 1). According to the distance vector (left table in Figure 2), which is computed locally by each vehicle, $V3$ becomes a RN (it is the closest to the NRP with respect to the RSU), hence it forwards the Request. $V1$, $V2$ 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 set up a reply timer according to the equation 2. In this example, $V3$ and $V5$ set up their reply timers. T_{max} and HL are constant, the only variable is HLC, which is decremented at each hop. This means that $V5$ will send its Reply message before $V3$. In this way, $V3$ will aggregate the information received from $V5$ with the 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 $< ID_{V5}, B_{V5} >$, where $B_{V5} = [V3, V4, V5, V6, V7]$ and broadcast it. Similarly, when $V3$'s reply timer expires, it creates a Reply message having $< ID_{V3}, B_{V3} >$, 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.



Tx = RSU	
Position	Tuple
0	<V3, dist(V3)>
1	<V2, dist(V2)>
2	<V1, dist(V1)>
3	<V4, dist(V4)>

Tx = V3	
Position	Tuple
0	<V5, dist(V5)>
1	<V4, dist(V4)>
2	<V6, dist(V6)>

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

5. SIMULATION RESULTS

We evaluate the DISCOVER performance using a multi-layer simulation tool composed by Veins [3], SUMO [13] and OMNET++ [1]. 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 [2], 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 vehicles is fed into the map. The vehicle trip starting and exiting 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 starting and exiting points. The movement of the vehicles is governed by the car-following model. The target speed has a Gaussian distribution with mean value 50 km/h and standard deviation equal to 0.1 times the mean value.

OMNET++ is used to simulate the behavior of the communications process, including the operations of the Physical, 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.

To model the impact of buildings and other obstacles to signal propagation, we have used jointly two attenuation models: the Two Exponents Model (TEM) [7] and the Simple Obstacle Shadowing Model (SOSM) [15]. 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.

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.

Table 1: Notation and simulations parameter values

Parameters	Values
D (m)	500
$T_{max}(ms)$	900
$T_d(ms)$	5
$L(byte)$	14
$H(byte)$	40
Urban Area (km^2)	12
Target veh mean speed (v_{mean}, km/h)	50
Vehicle speed st. dev. (σ)	$0.1 \cdot v_{mean}$
MAC, PHY parameters	IEEE 802.11p
Transmission Range (LOS) (m)	827

5.2 Simulation Scenarios

Two real urban scenarios have been created. The first considered scenario is the district of Manhattan in the city of New York (see Figure 3(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 3(b)). Differently from the first scenario, this one is characterized by a high level of irregularity, being composed by roads different in shapes and dimensions.

5.3 Performance Evaluation

Every considered scenario has been analyzed under two different vehicle densities $\lambda(veh/km)$, measured as the mean number of vehicles per lane unit distance. We evaluate the following performance metrics by considering all vehicles in the scenario:

- FRV - the fraction of reached vehicles (N/V);

(a) Manhattan District. (b) Rome.

Figure 3: Urban scenarios

- FMV - the fraction of monitored vehicles (M/N);
- FRN - the fraction of relay nodes (vehicles that forwarded the packet);
- TWT - two-way time: is the time needed to entirely complete the data dissemination and collection phases;
- B_{tx} - the total amount of Bytes transmitted by the vehicles during the whole data collection procedure;
- TH - the ratio between B_{tx} and the duration of the whole data collection procedure (TWT).

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

5.4 Data Dissemination: the Forward Wave

In the data dissemination phase the message sent by the RSU must reach as many vehicles as possible within the target region of interest, independently from 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, first investigating the effect of the parameter D, that can affect the performance results significantly. Then, we present the results for the metrics relevant for the dissemination phase, namely FRV.

After performing simulations for different values of D we found that DISCOVER gives best results for $400\ m \leq D \leq 750\ m$, independently from the simulation scenario. We have set $D = 500\ m$ to evaluate the metrics below. HL is set so as to cover an area of 12 km^2 centered at the RSU.

Figure 4 shows the performance in terms of FRV in New York and Rome considering two different traffic densities. In Table 2 the average values of vehicular density and speed for New York and Rome for two different traffic congestion scenarios can be found. Considering that in the theoretical case (TEO) FRV is always 100%, we show that DISCOVER is very close to this result and, in particular, that its FRV values are always higher than the 93%, independently from the considered scenario and/or vehicular density. Since RANDOM and FLOODING were not designed to disseminate data, but only to collect it, we do not compare DISCOVER with these two solutions in the data dissemination case.

Table 2: Average values of vehicular density (λ) and speed

	New York	Roma
Low	18 veh/km, 28 km/h	55 veh/km, 37 km/h
High	28 veh/km, 14 km/h	70 veh/km, 30 km/h

5.5 Collection Phase: the Reverse Wave

The goal of this second phase is to collect data from the vehicles roaming in the region of interest. 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, which are responsible for sending the Reply packets back to the RSU.

The Figure 5 shows the percentage of the vehicles in New York and Rome that are monitored (FMV) by the RSU using the different data collection algorithms. It can be noticed that DISCOVER, FLOODING and RANDOM are able to reach similar performance results and in average very close to the optimal results represented by TEO, independently from the considered traffic congestion scenario.

However, the real performance difference introduced by DISCOVER is perfectly depicted in the other metrics. Firstly, Figure 6 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 TEO; 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 5 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 way, it can be noticed that the total amount of bytes transmitted by DISCOVER is very close to the theoretical optimal value.

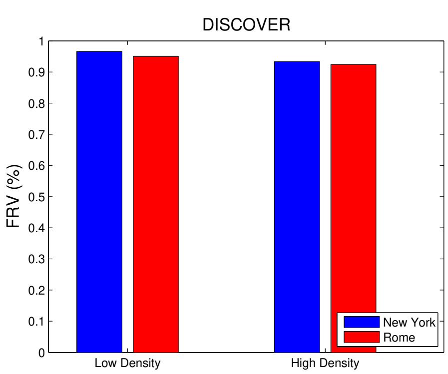


Figure 4: Fraction of Reached Vehicles for 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

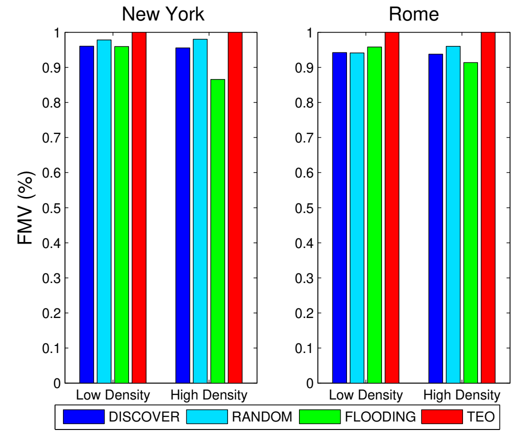


Figure 5: Fraction of Monitored Vehicles 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

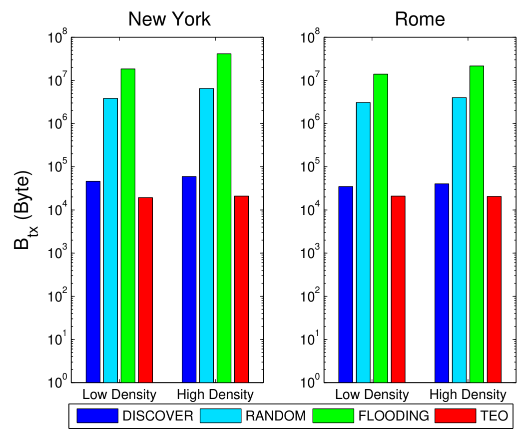


Figure 6: B_{tx} for 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

The metric in Figure 7, FRN, gives us a measure of the redundancy level obtained by the different algorithms. We can note that in this figure TEO represents the minimum number of forwarders (the total number of vehicles in the spanning tree $\mathcal{T}$) needed to minimize B_{tx}, maximizing at the same time the number of monitored vehicles FMV. Figure 7 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 TEO value.

Other aspects that we must consider are the total time needed to collect all data (TWT), and the bandwidth needed to transmit the data (TH). The first result shown by Figure 8 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}, is proved to be the worst in terms of delay. Moreover, Figure 9 shows the TH measured by the different protocols; in particular it is possible to note

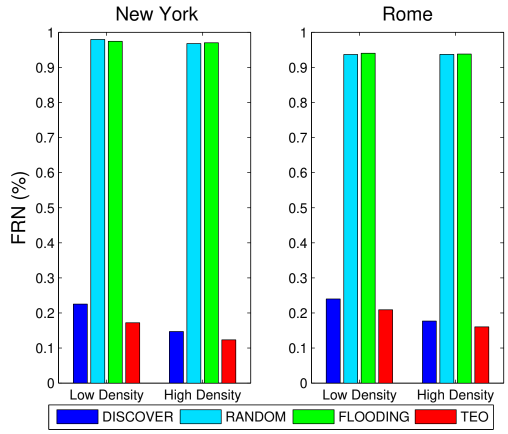


Figure 7: Fraction of Relay Nodes and Measured Overhead for 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

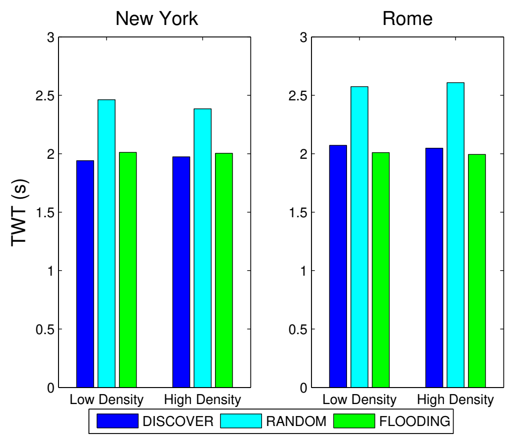


Figure 8: Two-way time for 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

that DISCOVER experiences a better (lower) TH, having to transmit a considerably less amount of data than other protocols, and in lower time respect to the simpler algorithms FLOODING and RANDOM.

6. CONCLUSIONS

We designed a new protocol for VANETs, named DISCOVER, which encompasses dissemination and collection features. We provide both the protocol description and the simulation of its behavior in real urban scenarios, where real building impairments are considered for the radio propagation. A single roadside unit is deployed in the urban area. We analyzed the downstream dissemination by evaluating performance metrics like the fraction of reached vehicles. Data collection performances are instead evaluated in terms of fraction of monitored vehicles, fraction of relay nodes, delay, overhead and throughput. The main conclusion is that, thanks to a suitable election of Relay Nodes, we are able to collect always more than the 90% of the vehicles information, in different cities and in different vehicular density conditions, minimizing the total amount of overhead transmitted, using only the VANET.

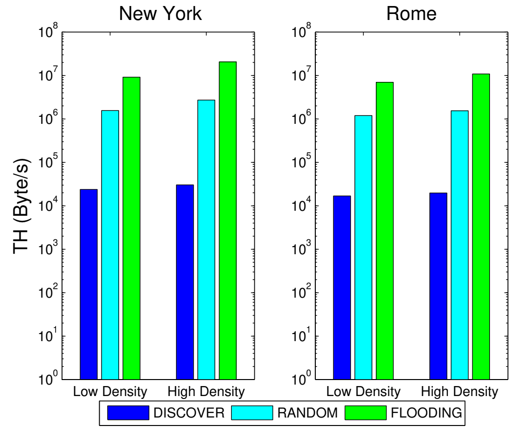


Figure 9: Collection Capacity for 2 different vehicular densities in the district of Manhattan, NY, and in Rome.

7. REFERENCES

[1] *OMNeT++ Network Simulation Framework*, 2001. Available at: http://www.omnetpp.org/.

[2] *OpenStreetMap*, 2001. Available at: http://www.openstreetmap.org/.

[3] *Veins: vehicular network simulation framework*, 2008. Available at: http://veins.car2x.org/.

[4] B. Brik, N. Lagraa, H. Cherroun, and A. Lakas. Token-based clustered data gathering protocol(tcdgp) in vehicular networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1070–1074, July 2013.

[5] H. T. Cheng, H. Shan, and W. Zhuang. Infotainment and road safety service support in vehicular networking: From a communication perspective. *Mechanical Systems and Signal Processing*, 25(6), 2010.

[6] F. Cuomo, I. Rubin, A. Baiocchi, and P. Salvo. Enhanced vanet broadcast throughput capacity via a dynamic backbone architecture. *Ad Hoc Netw.*, 21:42–59, October 2014.

[7] K. L. H. Hartenstein. Vanet vehicular applications and inter-networking technologies (intelligent transport systems). In *John Wiley & Sons*, March 2010.

[8] Z. He and H. Zhang. Density adaptive urban data collection in vehicular sensor networks. *Journal of Networks*, 9(8), 2014.

[9] Q. Huang, Y. Bai, and 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*, pages 1–5, September 2006.

[10] 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.

[11] 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.

[12] 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.

[13] D. Krajzewicz and C. Rossel. *Simulation of Urban MObility (SUMO).* German Aerospace Centre, 2002. Available at: http://sumo.sourceforge.net/index.shtml.

[14] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and 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, pages 151–162, New York, NY, USA, 1999. ACM.

[15] C. Sommer, D. Eckhoff, R. German, and 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*, pages 84–90, January 2011.

[16] A. Soua and H. Afifi. Adaptive data collection protocol using reinforcement learning for vanets. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1040–1045, July 2013.

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

[18] N. Wisitpongphan and et al. Broadcast storm mitigation techniques in vehicular ad hoc networks. *Wireless Communications, IEEE*, 14(6), 2007.

[19] Y. Zhu, Q. Zhao, and Q. Zhang. Delay-constrained data aggregation in vanets. *Vehicular Technology, IEEE Transactions on*, 64(5):2097–2107, May 2015.