# A Multi-Hop Broadcast Wave Approach for Floating Car Data Collection in Vehicular Networks

Ion Turcanu[a], Pierpaolo Salvo[b], Andrea Baiocchi[c], Francesca Cuomo[c], Thomas Engel[a]

[a]*SnT, University of Luxembourg, Belval, Luxembourg*
[b]*Fondazione Ugo Bordoni, Rome, Italy*
[c]*DIET, Sapienza University of Rome, Italy*

## Abstract

Inter-Vehicle Communication (IVC) is bringing connected and cooperative mobility closer to reality. Vehicles today are able to produce huge amounts of information, known in the literature as Floating Car Data (FCD), containing status information gathered from sensing the internal condition of the vehicle and the external environment. Adding networking capabilities to vehicles allows them to share this information among themselves and with the infrastructure. Collecting real-time FCD information from vehicles opens up the possibility of having access to an enormous amount of useful information that can boost the development of innovative services and applications in the domain of Intelligent Transportation System (ITS). In this paper we propose several solutions to efficiently collect real-time FCD information in Dedicated Short-Range Communication (DSRC)-enabled Vehicular Ad Hoc Networks (VANETs). The goal is to improve the efficiency of the FCD collection operation while keeping the impact on the DSRC communication channel as low as possible. We do this by exploiting a slightly modified version of a standardized data dissemination protocol to create a backbone of relaying vehicles that, by following local rules, generate a multi-hop broadcast wave of collected FCD messages. The proposed protocols are evaluated via realistic simulations under different vehicular densities and urban scenarios.

*Keywords:* Floating Car Data, VANET, Multi-Hop Broadcast, Vehicular Fog Computing, Intelligent Transportation Systems

## 1. Introduction

Dedicated Short-Range Communication (DSRC) [1] is among the key technologies for implementing the *connected car* paradigm. Indeed, DSRC-based technology is standard in many cars on the market and players in the automotive sector are providing new services in the areas of safety and entertainment. DSRC, which is based on the IEEE 802.11p amendment to the IEEE 802.11 standard, has been proposed as the main technology for Inter-Vehicle Communication (IVC) [1]. The primary motivation is to ensure safety on the roads by enabling Vehicle-to-Vehicle (V2V) communication and cooperative awareness. The latter is usually obtained through periodic exchange of beacon messages (i.e., Cooperative Awareness Message (CAM), Basic Safety Message (BSM)). DSRC operates in a dedicated spectrum in the 5.9 GHz frequency band, consisting of 75 MHz of bandwidth divided into seven channels of 10 MHz, with a 5 MHz guard band at the low end. Multi-hop communications are used to extend the coverage of the Vehicular Ad Hoc Network (VANET). Infrastructure nodes, namely Roadside Units (RSUs), are used as gateways of the VANET toward the fixed network, thus enabling Vehicle-to-Infrastructure (V2I) communications along with the V2V mode, or as nodes used as processing agents to support edge/fog computing architectures [2, 3].

The DSRC technology is mature enough to provide a complete networking architecture for connected vehicles. Two challenging aspects that are driving the research in this field are the opportunity to collect almost real-time data from vehicles, and the possibility of reaching a wide area in the road network through multi-hop communications. The information provided by vehicles, known in the literature as Floating Car Data (FCD), can boost new applications and services that aim to improve the overall traffic safety and efficiency. The main challenges today are the lack of widespread DSRC-based infrastructure deployment and the limited bandwidth dedicated to IVC. The first challenge can be tackled by exploiting the VANET via multi-hop communication, while the second demands efficient IVC protocols that minimize the impact on the DSRC communication channel.

To decrease the load on the communication channel, existing FCD collection algorithms typically use a variety of clustering mechanisms [4] to select subsets of vehicles to be in charge of the collection process from the entire Region of

*Email addresses:* `ion.turcanu@uni.lu` (Ion Turcanu), `psalvo@fub.it` (Pierpaolo Salvo), `andrea.baiocchi@uniroma1.it` (Andrea Baiocchi), `francesca.cuomo@uniroma1.it` (Francesca Cuomo), `thomas.engel@uni.lu` (Thomas Engel)

*Article published in Journal of Vehicular Communications 24 (2020) 100232 (10.1016/j.vehcom.2020.100232)*

Interest (ROI). These algorithms are usually customized to the specific requirements of an application and their main goal is to support the FCD collection operation. On one hand, decreasing the number of vehicles allows to collect FCDs and to decrease the exchanged information and the load on the DSRC channel. On the other, using customized dissemination algorithms to select such a subset of vehicles adds additional load.

In this paper we propose and evaluate three DSRC-based multi-hop FCD collection protocols, designed to operate in challenging dynamic urban scenarios. We exploit the intrinsic characteristics of the European Telecommunications Standards Institute (ETSI) Intelligent Transportation System (ITS)-G5 GeoNetworking standard [5] and, in particular, of the Contention-Based Forwarding (CBF) dissemination algorithm – the current standard for VANET data dissemination. In contrast to other existing solutions, we do not design a customized dissemination algorithm to select the subset of relaying vehicles. Instead, we exploit the CBF algorithm, which is part of the ETSI ITS-G5 standard and was designed to disseminate messages in multi-hop VANETs, with a slight modification to limit the number of selected relay nodes.

The use case we are considering is of an originating node periodically triggering FCD collection from vehicles roaming in a ROI while disseminating messages to the vehicular nodes at the same time. The originator can be either a fixed or a mobile node. We address the former scenario, since it is the more likely and is practical for monitoring and informational services. This architectural model well fits a framework recently proposed in the literature that is *vehicular fog computing* [2, 6]. In this framework RSUs deployed in different areas of a city act as fog nodes. They collect data sent by smart vehicles, process the collected data, and report the (processed) data to the cloud servers. In view of cooperative fog servers co-located with the RSUs (as in the architecture of [6]) our dissemination and collection protocol can be used to have the servers geographically closer to the vehicle to serve and provide vehicle-based applications in almost real time or within some time constraints. Moreover, besides RSUs, vehicles themselves can also act as fog nodes and use our proposed protocol for collecting data from other vehicles in a given ROI. In our case, we consider an infrastructure node, i.e., an RSU, that disseminates messages using a modified CBF algorithm, by starting a *forward wave.* During the dissemination process, relay vehicle nodes that will act as cluster heads in the ensuing collection phase are elected. The collection phase realizes a *backward wave*, where the collected data is consolidated as messages hop through the VANET back to the originator node via the elected relay nodes.

A preliminary version of our ideas was proposed in [7, 8], where we describe DISCOVER, a timer-based FCD collection protocol. The main limitation of DISCOVER is the assumption that all the information to be collected must fit into a single Maximum Transmission Unit (MTU), which limits the maximum amount of data that can be collected. In a real-world scenario, the collected FCD information can easily exceed the size of an MTU, which implies packet fragmentation in reassembly. In this work we employ a conservative assumption according to which FCD records belonging to a fragmented message are received correctly at the receiver only if *all* fragments composing the relevant message are received. To solve this, we extend the idea behind DISCOVER by adding a backup mechanism based on eavesdropping and selective retransmissions, aiming to improve the reliability and efficiency of the protocol under different road network topologies and vehicular densities. This backup mechanism allows us to overcome DISCOVER's main limitation. We designate this protocol as DISCOVER–Timer-based Collection (D-TC). The second proposed algorithm, named DISCOVER–Reply-based Collection (D-RC), aims to reduce the overall FCD collection delay by exploiting the network graph obtained with CBF, which allows us to shorten the waiting timers used by D-TC. Finally, we compare these two solutions with a baseline protocol that represents a basic alternative where no message consolidation or retransmission are used. Like the previous protocols, this baseline solution exploits the same backbone network of relay nodes obtained with CBF.

The FCD collection techniques and the relevant analysis presented in this paper have the following innovative elements:

- In-network processing of data, with duplicate suppression and integrity checks, to make FCD delivery more reliable;
- A new performance evaluation, with new metrics (e.g., coverage of the ROI, vehicle density, amount of collected data) and entirely new simulation scenarios, including vehicular measurements that lead to a more realistic performance assessment;
- Comparison between three different approaches, specifically a baseline collection protocol, a timer-based and an event-based collection protocol.

The remainder of this paper is organized as follows. Related work is reviewed in Section 2. A detailed description of the proposed protocols is given in Section 3. Section 4 defines the simulation model and the adopted metrics. Simulation results are discussed in Sections 5 and 6. Section 7 provides concluding remarks and suggestions for future work.

## 2. Related Work

Many papers in recent years have been focused on the data dissemination in VANETs. A comprehensive survey of data dissemination approaches can be found in [9], where three basic models are presented: push-based, pull-based and hybrid. The survey first analyses the existing data dissemination techniques proposed for VANETs under these three dissemination models, providing, for each model, a set of representative examples. Then, the existing performance modeling approaches of data dissemination in VANETs are reviewed. The work in [9] also highlights the technical challenges related to the scalability, security, quality of service and cooperation that are fundamental for different VANETs applications using data dissemination. Another overview, mainly dealing with data collection and the key performance metrics to be adopted in the relevant analysis, is [10]. Data aggregation, latency, packet delivery ratio, scalability, security, overhead and vehicle density are recognized as important parameters that characterize data collection approaches.

In general, data dissemination schemes can be divided into two main classes: beacon-assisted and beaconless approaches. The former use beacon control messages to get current vehicles attributes (e.g., positions and velocities) and so select the best vehicles to act as dissemination entities. The paper [11] proposes an Adaptive Data Dissemination Protocol (AddP), which aims to provide reliability for message dissemination in an efficient manner. The protocol dynamically adjusts beacon periodicity in order to reduce the number of messages and beacons in the network, and to attain reliability and efficiency of the dissemination protocol.

While the use of beacons can provide vehicles and VANETs protocols with more insight into the network dynamics, the overhead due to beacons may be high and difficult to manage. Consequently, there are solutions providing data dissemination and routing without the use of beacons [12, 13]. These works demonstrate that, even without a knowledge of the current network topology and conditions, it is possible to exploit the intrinsic VANET configuration (e.g., vehicle density) to design protocols able to dynamically adapt their behavior to the system. The work in [13] shows and characterizes the behavior of the spurious forwarding arising in time-based dissemination approaches, giving guidelines to reduce the effect and improve throughput in data dissemination on highways.

Apart from straight roads, few papers have dealt with specific road geometries. An example is [14], which addresses relay-node selection on curve roads. Through the definitions of the optimal position in a general exponent-based partition approach, the proposed algorithm picks the relay node from candidate nodes on a curved road, allowing the message to be delivered along the road as fast as possible.

By increasing the complexity of the dissemination protocol, [15] proposes a hybrid scheme that attempts to improve the performance of VANETs over varying node densities, traffic load conditions and mobility speed scenarios. By utilizing a relay priority list constructed at the sender side, the relay-selection solution in [15] includes both the furthest distance (FD) approach and the Bi-Directional Stable Communication (BDSC) scheme, thus adapting the relay selection to the varying traffic load conditions.

Several DSRC-based data collection protocols can be found in the literature, some proposing distributed Q-learning reinforcement techniques, making the collecting operation more reactive to node mobility and topology changes [16]; and others with the aim of solving the data collection problem in VANETs under rapidly evolving traffic conditions [17]. Alternative approaches exploit the DSRC technology to create local clusters and select cluster head nodes, but the collected FCDs are then sent via the cellular network. For example, the authors in [18, 19] try to minimize the number of selected cluster heads in order to reduce the impact on the cellular network. Other works require the support of external agents to collect or disseminate data in a VANET. These may be mobile agents periodically injected into the VANET covering the area using LTE, as in [20]; or unmanned aerial vehicles (UAVs) assisting data dissemination through a scheduling strategy based on predicting vehicle mobility [21].

An high level intelligent vehicular traffic information system (not depending on the transmission technology and assuming ideal broadcasting) is presented in [22], where a model that can autonomously collect and disseminate congestion information in urban road networks is presented. The mechanism uses the concept of “link nodes”, specific nodes positioned according to some criteria to allow congestion information to be spread rapidly over the entire road network.

Unlike the above-mentioned papers, in a previous study [8] we instead propose dissemination and collection integrated in a unique protocol solution based on DSRC, with the aim of increasing the amount of collected information by decreasing the number of forwarding nodes. The proposed protocol, DISCOVER, is based only on VANET multi-hop communications without requiring any a priori knowledge of the road network. DISCOVER establishes a forward wave used for disseminating the information from one or more RSUs and a backward wave to collect FCD from the interested vehicles.

In this paper we extend the work of [8], by providing reliable data collection mechanisms not limited to contributions of few bytes from each vehicle. In more detail, the new contributions of this paper with respect to our previous work [8] are as follows:

- The collection phase of the protocol has been redesigned to provide reliable data aggregation and forwarding to the collection point. This is achieved by means of backup timers, eavesdropping and possibly retransmissions.

- The dissemination phase, which defines the nodes responsible for data aggregation in the subsequent collection phase, is based on the existing standard for data dissemination in VANETs, namely CBF, with a slight modification to account for a randomized component, in addition to the geographical one.

- Arbitrary-length data handling is introduced, so that any amount of data can be dealt with by the protocol at application layer. Fragmentation and reassembly is provided to overcome the limited length of MAC level frames.

- Instead of urban scenarios fed with artificially- generated vehicular traffic, we have use a publicly-available urban scenario dataset for which real vehicular traffic traces are available, so as to improve the significance of the results and to facilitate future comparison work.

## 3. FCD collection protocols

We define an FCD collection process in a DSRC-based VANET including an initial dissemination phase, in which a trigger node (i.e., either an On-Board Unit (OBU) or an RSU) polls the surrounding vehicles to collect data from them. If the target area is larger than the requesting node's communication range, the *Request* message must be disseminated by means of multiple hops. This process requires the defining of *relay nodes*. These nodes also act as relay nodes for the FCD collection phase in the backward direction. Thus dissemination and data collection realize a synergy, by defining an on-the-fly backbone network in the VANET, used to improve the efficiency of the dissemination and collection phases. Since the VANET topology is time-varying due to vehicle mobility, triggering the FCD collection by means of a *Request* message is necessary to probe the network and find relay nodes to be used for the current collection phase. This can be viewed as an adaptive way of finding cluster-head nodes.

The collection process is composed of four main steps:

1. The trigger node (i.e., RSU) starts the collection process by broadcasting a *Request* message.
2. The *Request* is propagated into the network via multi-hop communication (*Request* dissemination phase).
3. Vehicles (all or only the selected relay nodes, depending on the FCD collection protocol) receiving the *Request* broadcast a *Reply* message (immediately or after a computed timeout, depending on the FCD collection protocol). A single *Reply* can contain one or more FCD records (i.e., FCD related to a single vehicle).
4. The FCD information is propagated towards the originator of the *Request* through the relay nodes (FCD collection phase). Depending on the collection protocol, *Reply* messages can either be simply forwarded (i.e., intermediate relay nodes act as simple forwarders, without altering the payload), or merged in order to remove duplicates and reduce the packet header overhead (i.e., intermediate relay nodes can merge together the payload of multiple *Reply* messages).

In the following, we describe the dissemination process, as well as three algorithms for the FCD collection phase: Baseline, D-TC [7, 8] and D-RC. Table 1 contains the notation and description of the main parameters used in this work.

| **Parameter** | **Description** |
|---|---|
| $D$ | distance between sending and receiving vehicle |
| $D_{\max}$ | maximum theoretical communication range of the adopted wireless access technology |
| $H$ | current hop count |
| $H_{\max}$ | maximum number of hops the message is allowed to travel |
| $T_{\min}$ | minimum timeout value |
| $T_{\max}$ | maximum timeout value |
| $T_{\text{req}}$ | request timeout |
| $T_{\max}^{\text{req}}$ | maximum request timeout |
| $T_{\text{rep}}$ | reply timeout |
| $T_{\max}^{\text{rep}}$ | relative weight of the randomized component of the reply timeout |
| $T_{\text{tx}}$ | transmission time of a *Reply* message |
| $T_{\max}^{\text{tx}}$ | maximum transmission time of a *Reply* message |

Table 1: Description of the main parameters.

### 3.1. Dissemination Phase

For the dissemination phase we propose a modified version of CBF [5], a timer-based data dissemination protocol defined in the ETSI ITS GeoNetworking standard. This algorithm runs in the application layer of every OBU[1]. In the following we refer to the node originating the *Request* message as the *source* node.

The source node triggers the data dissemination and collection process every $T_{\text{col}}$ seconds, according to the required collection frequency defined by a central collection unit (e.g., traffic monitoring center), by broadcasting a *Request* message. The *Request* message header contains the following fields:

- `msg_ID`: unique identifier of the message, e.g., a sequence number and the source node address.
- `hop_span`: $D_{\max}$.
- `hop_max`: $H_{\max}$.
- `hop_count`: the number of hops $H$ that the message has traveled through the VANET. It is initialized by the source node to 0 and incremented by 1 by each relay node that forwards the message.
- `orig_xy`: coordinates of the position of the source node.
- `fwrd_xy`: coordinates of the position of the last forwarding node.

According to the original CBF algorithm [5], all the vehicles that receive this message and are within a distance $D_{\max}$ from the sending node, compute a timer value that is proportional to each vehicle's distance $D$ from the sender.

[1] CBF could actually be implemented in the network layer as well, immediately on top of the IEEE 802.11p protocol suite.

By denoting as $T_{\text{out}}$ this timer value, the standard CBF prescribes that:

$$T_{\text{out}} = \begin{cases} T_{\max} - \frac{T_{\max} - T_{\min}}{D_{\max}} D & , D \leq D_{\max} \\ T_{\min} & , D > D_{\max} \end{cases} \tag{1}$$

Here $T_{\min}$ and $T_{\max}$ are the minimum and maximum timer values, $D$ is the distance between the sending and receiving vehicles; $D_{\max}$ is a parameter representing the theoretical maximum communication range of the adopted wireless access technology. According to this timer setting, a vehicle further away from the sender and closer to $D_{\max}$ is more likely to become the next relay node, while the others are inhibited. The inhibition rule works as follows: a vehicle that receives a copy of the *Request* message with the same sequence number as the one for which its timer is still counting down, must cancel its timer and pending *Request* message. If the timer expires and no inhibition has occurred, the vehicle becomes a *relay node* and forwards the *Request* message. Since the main goal of CBF is to disseminate the data as widely as possible, all the vehicles outside $D_{\max}$, that are able to receive and correctly decode the message, will re-send the message after the minimum timer delay $T_{\min}$.

Our purpose is different, in that we aim to use message dissemination as a means to define as few relay nodes as possible covering the VANET in the ROI. Our proposed modification of CBF re-defines the timer calculation as follows:

$$T_{\text{req}} = \begin{cases} T_{\max}^{\text{req}} \left[ \alpha \left( 1 - \frac{D}{D_{\max}} \right) + (1-\alpha) \mathcal{U}(0,1) \right] & , D \leq D_{\max} \\ \infty & , D > D_{\max} \end{cases} \tag{2}$$

where the timeout value has been renamed as $T_{\text{req}}$ and we have set $T_{\max} = T_{\max}^{\text{req}}$ and $T_{\min} = 0$. We will further refer to this protocol as Reduced CBF (rCBF). The main difference with respect to the original CBF algorithm is that vehicles further away from the sender than $D_{\max}$ will simply ignore the *Request* message, even if the message was correctly received and decoded. This helps to keep the number of forwarding nodes to a minimum, while guaranteeing that they form a connected overlay network. This is consistent with the purpose of the dissemination phase in our scheme.

As a second modification, we introduce an additional uniformly-distributed random term to avoid simultaneous re-transmissions and ease the task of the IEEE 802.11p MAC protocol. The deterministic and random components can be flexibly mixed by using the weighting parameter $\alpha \in [0, 1]$. In particular, for the dissemination to be effective, i.e., for the distance-dependent bias to be dominant, we have to set $\alpha$ much bigger than 1/2. Our choice is to give 80 % of the weight to the distance-dependent component. To limit the dissemination of the *Request* message to a target area, we use the parameter $H_{\max}$, which denotes the maximum number of hops this message is allowed to travel.

Notice that the dissemination phase aims to create a temporary backbone network, composed of relay nodes and anchored to a specific node (an RSU in this case). The dissemination of the *Request* message is instrumental to create this backbone network, to be used for one data collection instance. The backbone network is refreshed by re-sending the *Request* message periodically.

We argue that this backbone network can be used to support different operations. In this paper, we investigate the use of this "relay node network" to sustain a data collection ITS application. The *same* backbone network of relay nodes could be used to disseminate a flow of data towards the vehicles in a ROI around the anchor node (the RSU in our case). This kind of application of the backbone has been investigated and proved feasible in [23]. Different applications could exploit the same backbone of relay nodes, thanks to multiple channels provided by the VANET. More in depth, the signaling message used to set-up the backbone network (the *Request* message) can be disseminated on the Control Channel (CCH), since it amounts to a relatively short message. More massive data can be disseminated by using a Service Channel (SCH). At the same time, another SCH could be used to perform the data collection using one of the protocols discussed in the present paper.

### *3.2. Collection Phase: Baseline Algorithm*

A simple protocol for collecting FCD information operates as follows: every time a vehicle node receives a *Request*, it sends in broadcast a *Reply* message containing its own FCD record. In case of vehicles roaming inside the communication range of the source node, their *Reply* messages can be received directly by the source node. If a vehicle receives the *Request* message from a forwarder node different from the source, then its *Reply* must be propagated back to the source node in a multi-hop fashion.

**Algorithm 1** Baseline operation: collection phase

```
relayNode: a boolean showing if the vehicle is a relay
    node in the current collection phase
myHopCount: vehicle's current hop count from the
    RSU defined during the dissemination phase
sentReplies: a local data structure containing the
    IDs of the vehicles whose Reply messages have been
    already sent within the current collection phase

upon event Reply received do
    if relayNode == TRUE then
        if myHopCount < Reply.getHopCount() then
            if Reply.getOriginator() ∉ sentReplies then
                broadcastMessage(Reply)
                sentReplies.insert(Reply.getOriginator())
            end if
        end if
    end if
```

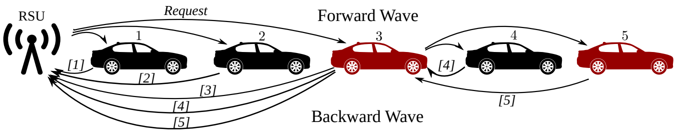


Figure 1: Baseline: example of a collection instance. The red vehicles represent the relay nodes, while $[x]$ indicates the FCD from vehicle $x$.

A straightforward way of doing this is to let every vehicle that receives a *Reply* message rebroadcast it, so that it will eventually reach the source node. This implies flooding the network with *Reply* messages, which can generate a broadcast storm [24]. To avoid this problem, we describe a Baseline algorithm (see Algorithm 1) that exploits the same relay nodes that are selected during the dissemination phase.

Baseline operates as follows: as soon as a vehicle receives a *Request*, it generates the *Reply* message containing its own FCD information and broadcasts it after a random delay uniformly distributed between 0 and $\theta_{\max}$. At this point, only the relay nodes previously selected with rCBF are allowed to forward *Reply* messages. In addition, to ensure that the message is propagated back to the source node, only the relay nodes having a hop count $H$ lower than the one indicated in the *Reply* message are allowed to re-broadcast it. Also, to avoid re-broadcasting the same message multiple times, a *Reply* is sent only when received for the first time. To be noted that in Baseline intermediate relay nodes act as simple forwarders, meaning that *Reply* messages are simply forwarded based on the networking layer forwarding rules described above (i.e., there is no aggregation/merging of the payload contained in the *Reply* messages).

An example of a collection instance using the Baseline algorithm is shown in Figure 1. In this example the source node starting the collection process is the RSU. Vehicles 1, 2 and 3 are in the RSU's communication range, hence, their *Reply* messages are received directly by the RSU. At the same time, according to the dissemination algorithm, vehicle 3 becomes the next relay node and forwards the *Request*. Vehicles 4 and 5 receiving the *Request*, broadcast their *Reply* messages after a random delay. Being the only relay node receiving these *Reply* messages and having a lower hop count, it is vehicle 3 that individually forwards them back to the RSU.

### *3.3. Collection Phase: D-TC Algorithm*

The Baseline algorithm does not fully exploit the features of the DSRC VANET, which is mainly concerned with safety and monitoring applications. These applications are based on an intermittent beaconing process, in which every vehicle periodically broadcasts a specific type of message, known in the literature as CAMs [25]. CAMs contain basic FCD information, such as location, speed, direction of travel, etc. Every vehicle saves the received CAM messages in a Local Dynamic Map (LDM) [26]. CAMs cannot be used for multi-hop FCD collection, since they are one-hop messages, i.e., the receiver must not forward them.

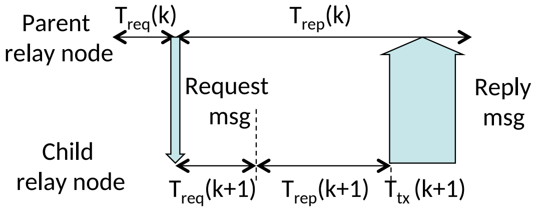


Figure 2: Relationship between timers of a parent and a child relay node.

In our previous work [8] we proposed DISCOVER, an algorithm that exploits the CAM exchange process, as well as the relay nodes selected during the dissemination phase, to collect FCD messages in a DSRC-based VANET. Here we extend this idea and propose D-TC, an FCD collection protocol designed to operate in complex urban scenarios with a wide range of vehicular densities. Unlike DISCOVER, which assumes that the size of a *Reply* message never exceeds the MTU of the IEEE 802.11p interface, D-TC handles arbitrary message sizes. In addition, it introduces a backup mechanism for message re-transmission.

The D-TC collection phase is initialized during the dissemination phase, when every newly-elected relay node sets up a local reply timeout $T_{\text{rep}}$ (see Figure 2). Specifically, we define as *parents* of a relay node $N$ all relay nodes that are one hop closer to the source node than $N$. $N$ is said to be a child relay node with respect to its parent relay nodes.

Let us consider a node $N$ becoming a relay (the parent relay) $k$ hops from the source node. When $N$'s timer $T_{\text{req}}(k)$ expires, $N$ sends out the *Request* message. Nodes receiving the *Request* message from $N$ execute the dissemination phase protocol. Eventually, one (or more than one) of those nodes, say $M$, forwards the *Request* message when its timer $T_{\text{req}}(k+1)$ expires. Then, $M$ becomes a relay node. Specifically, it is a child relay node of $N$ and $N$ is a parent relay node of $M$. $M$ is $k+1$ hops away from the source node, while its parent $N$ is $k$ hops away.

When FCDs are collected all the way from the furthest relay nodes back to the source node, the parent node waits for a timeout $T_{\text{rep}}(k)$ given by:

$$T_{\text{rep}}(k) \geq T_{\text{req}}(k+1) + T_{\text{rep}}(k+1) + T_{\text{tx}}(k+1) \qquad (3)$$

where $T_{\text{tx}}(k+1)$ is the time that the child relay node takes to transmit the collected FCDs back to the parent relay node. We decompose $T_{\text{rep}}(k)$ into the sum of two terms: (i) a cumulative timer $X_k$; (ii) a 'local' additional term $b_k \in [0, b_{\max}]$. The first term is intended to capture the accumulated wait for *Reply* messages coming from outer relay nodes. The second term is a randomized delay useful to de-synchronize replies from different relay nodes. Then, it is $T_{\text{rep}}(k) = X_k + b_k$.

Under this model, a sufficient condition for Equation (3) to hold is:

$$\begin{aligned} X_k = T_{\max}^{\text{req}} + T_{\max}^{\text{tx}} + X_{k+1} + b_{\max} = \\ = T_{\max} + X_{k+1}, \qquad k = H_{\max} - 1, \dots, 1, \end{aligned} \tag{4}$$

where we write $T_{\max} = T_{\max}^{\text{req}} + T_{\max}^{\text{tx}} + b_{\max}$ for ease of notation. Given the initial condition $X_{H_{\max}} = 0$, we have

$$X_k = T_{\max} \left(H_{\max} - k\right) \, , \qquad k = 1, \dots, H_{\max} \tag{5}$$

In the following, we set

$$b_k = T_{\max}^{\text{rep}} \, \left[1 + \mathcal{U}(0,1)\right] \left(1 - k/H_{\max}\right) .$$

The purpose of the randomized additive component $b_k$ is to decouple the transmission of the FCD sent by different child relay nodes of the same parent relay node. This is critical, given that it is probable that the child nodes of a given parent relay node are hidden from one another. The de-synchronization is more critical as we get closer to the RSU, since the size of the *Reply* message grows. This is why we choose a value of $b_{\max}$ that grows moving from the farthest relay node towards the RSU.

With our choices we find:

$$\begin{aligned} T_{\text{rep}}(k) = b_k + T_{\max}(H_{\max} - k) = \\ = \left[T_{\max}^{\text{rep}}(1 + \eta_k) + T_{\max} H_{\max}\right] \left(1 - \tfrac{k}{H_{\max}}\right) \end{aligned} \tag{6}$$

where $\eta_k \sim \mathcal{U}(0,1)$. According to the definitions above, we have also $T_{\max} = T_{\max}^{\text{req}} + T_{\max}^{\text{tx}} + 2T_{\max}^{\text{rep}}$.

An upper bound of the time to transmit the FCD can be found by considering the length of the MTU of the VANET, the FCD record length, and the number of nodes whose FCD are collected. Typical values range from one packet transmission time up to few tens of packet transmission times. With typical VANET data rates (6 Mbit/s), an MTU packet transmission time is about 2.5 ms. Thus, it is sensible to set $T_{\max}^{\text{tx}}$ on the order of tens of ms, e.g., $T_{\max}^{\text{tx}} = 25\,\text{ms}$.

Equation (6) is such that relay nodes with a smaller value of $H$ (i.e., closer to the source node) will have higher timeout values with respect to relay nodes having greater $H$ values (i.e., further from the source). This timeout setting ensures that inner relay nodes hold back for long enough to receive the *Reply* messages from outer relay nodes and are thus able to merge the received FCDs in their *Reply* message, before sending it. The merged information is saved into a local dataset, named $S_{\text{fcd}}$. The pseudo-code of the collection phase operation is given in Algorithm 2.

**Algorithm 2** D-TC operation: collection phase

```
replied: a boolean showing if the vehicle sent its own Reply or not in the current collection phase
uniqueID: a unique message identification
MaxRTX: a parameter defining the maximum number of Backup retransmissions allowed
backupRTX: a variable showing the current remaining retransmissions for a given Backup message
receivedFCDSet: a local data structure containing the FCDs received from other neighboring vehicles and extracted from the corresponding Reply messages
localFCDSet: a local data structure containing the FCD records extracted from the vehicle's LDM

upon event ForwardReply do
    replied = TRUE
    backupRTX = MaxRTX
    $S_{\text{fcd}}$ = merge(receivedFCDSet, localFCDSet)
    Reply.setID(uniqueID)
    Reply.setHopCount(myHopCount)
    Reply.setFCDs($S_{\text{fcd}}$)
    Backup = Reply
    broadcastMessage(Reply)
    scheduleEvent(ForwardBackup, $T_{\text{curr}} + T_{\text{backup}}$)
upon event Reply received do
    if relayNode == TRUE then
        if replied == FALSE then
            merge(receivedFCDSet, Reply.getFCDs())
        else
            if $S_{\text{fcd}} \subset$ Reply.getFCDs() then
                cancelEvent(ForwardBackup)
            end if
        end if
    end if
upon event ForwardBackup do
    if backupRTX > 0 then
        broadcastMessage(Backup)
        backupRTX = backupRTX −1
        scheduleEvent(ForwardBackup, $T_{\text{curr}} + T_{\text{backup}}$)
    end if
upon event Backup received do
    if replied == FALSE then
        merge(receivedFCDSet, Backup.getFCDs())
    else
        if myHopCount < Backup.getHopCount() then
            broadcastMessage(Backup)
            backupRTX = MaxRTX
            scheduleEvent(ForwardBackup, $T_{\text{curr}} + T_{\text{backup}}$)
        else
            cancelEvent(ForwardBackup)
        end if
    end if
```

When a vehicle's reply timeout expires, it attaches the FCD set, $S_{\text{fcd}}$, that it has accumulated up to that time, to its *Reply* message and broadcasts it on the DSRC channel. The set $S_{\text{fcd}}$ is obtained by merging the local FCD records in the vehicle's LDM with all those FCDs that are received from other relay nodes that sent their *Reply* messages earlier. Notice that only the relay nodes that are elected during the dissemination phase are allowed

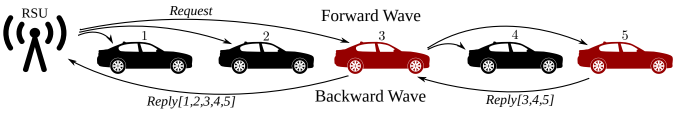


Figure 3: D-TC: example of a collection instance. Notice that vehicle 5 has the FCDs of 3 and 4 in its LDM thanks to the CAM exchange process.

to send back their *Reply* messages. Notice also that $T_{\text{rep}}$ is never canceled before the expiration, meaning that each relay node will eventually send its *Reply* message. An illustrative representation of a complete collection phase is shown in Figure 3.

Despite the algorithm's effort to minimize the number of vehicles participating in the FCD collection process, the collection itself is still challenging. The main issue comes from the fact that, in the collection phase, we have the problem of many nodes sending data to one sink. The problem is more challenging as the information approaches the RSU, since more data is being merged and sent, meaning that the size of the *Reply* messages to be sent is greater at each step. In addition, because of the increasing amount of merged and collected information, the size of the *Reply* messages can easily exceed the IEEE 802.11p MTU, which means that the message must be fragmented and multiple packet transmissions are required. The increased local load on the wireless interface brings a higher probability of collision.

To cope with this issue, we propose to add a backup mechanism in the collection phase. This mechanism is based on eavesdropping on the communication channel and allows a relaying vehicle to retransmit its *Reply* up to a predefined number of times. Whenever the reply timeout expires, meaning that the vehicle has to broadcast its *Reply* message, it creates a local copy of this message, named *Backup*, and schedules a backup timer, $T_{\text{bkp}}$, computed as:

$$T_{\text{bkp}} = \frac{T^{\text{rep}}_{\text{max}}}{H_{\text{max}}} + T^{\text{rep}}_{\text{max}} \left(1 + \eta\right) \tag{7}$$

where $\eta \in \mathcal{U}\left(0, \eta_{\text{max}}\right)$ is a uniformly-distributed random value between 0 and $\eta_{\text{max}}$.

If $T_{\text{bkp}}$ expires, the vehicle broadcasts *Backup* and reschedules $T_{\text{bkp}}$ provided that it has not exceeded the maximum allowed *Backup* retransmissions. However, if a relay node $V$, while waiting for its $T_{\text{bkp}}$ to expire, eavesdrops on a *Reply* or *Backup* message containing its $S_{\text{fcd}}$, and that message has been sent by another relay node $U$, having a hop count smaller than or equal to the hop count of $V$ (i.e., a relay node closer to the source node), then $V$ cancels its $T_{\text{bkp}}$ to avoid unnecessary re-transmissions.

### *3.4. Collection Phase: D-RC Algorithm*

The main challenge in the D-TC algorithm is to properly set the parameters so that inner relay nodes hold back long enough to receive the *Reply* messages from all outer relay nodes before their own reply timers expire. On one hand, setting short reply timers can lead to faster FCD collection, but also to a smaller timer difference between two consecutive relay nodes, leading to a higher probability that a relay node will send its *Reply* before receiving the *Reply* messages from all its child relay nodes. On the other hand, setting longer reply timers ensures enough time between two consecutive relay nodes, but also increases the overall collection delay. In addition, according to D-TC, a relay node sends its *Reply* only after its timer expires, even if the node has already received the *Reply* messages from all its potential child relay nodes.

**Algorithm 3** D-RC operation: collection phase

```
relayNode: a boolean showing if the vehicle is a relay node in the current collection phase
myHopCount: vehicle's current hop count from the RSU defined during the dissemination phase
sentReplies: a local data structure containing the IDs of the vehicles whose Reply messages have been already sent within the current collection phase

upon event Request received do
    if relayNode == TRUE then
        if Reply.getParentID() == myID then
            childList.insert(Reply.getParentID())
        end if
    end if

upon event ForwardRequest do
    relayNode = TRUE
    scheduleEvent(EnableReplies, $T_{\text{curr}} + T_{\text{enable}}$)
    scheduleEvent(ForwardBackup, $T_{\text{curr}} + T_{\text{bkp}}$)

upon event EnableReplies do
    replyEnabled = TRUE
    if receivedFromAllChildren() == TRUE then
        broadcastMessage(Reply)
    end if

upon event Reply received do
    if relayNode == TRUE then
        updateRepliesList(Reply.getSenderID())
        if replyEnabled == TRUE then
            if receivedFromAllChildren() == TRUE then
                broadcastMessage(Reply)
            end if
        end if
    end if
```

The D-RC algorithm, described in Algorithm 3, aims to speed up the collection process by allowing relay nodes to send their *Reply* messages as soon as they receive the *Reply* messages from all their child relay nodes. We extend the *Request* messages with a new field, the `parent_ID` field. To build its list of child nodes, during the dissemination phase a relay node $N$ inserts its ID in the `parent_ID` field, then eavesdrops on the communications channel. Whenever it captures a new *Request* that has the same `parent_ID` as its own ID, it deems that the node which sent this message is a child node and saves it into a local list. At this point, $N$ will send its *Reply* message as soon as it receives all the replies from all its child nodes, but not before it is sure that all the potential child relay nodes, as well as their child nodes in their corresponding subtrees, have been selected. To this end, $N$ schedules a local timer that depends on the *Request* message relay time from $N$ down to the relay nodes $H_{\max}$ hops away from the source node, computed as

$$T_{\text{enable}} = (T^{\text{req}}_{\max} + \delta)\,(H_{\max} - H)\,, \tag{8}$$

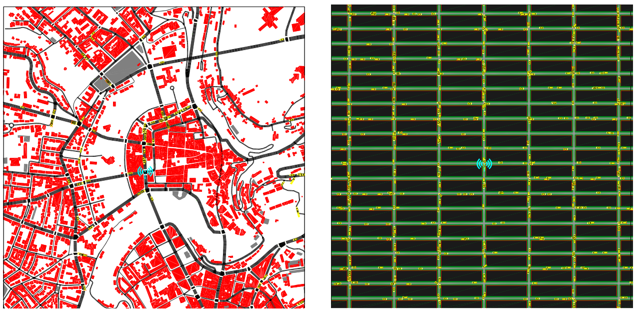
where $\delta$ is the time it takes the lower layers (MAC and PHY) to deliver the *Request* message from a relay node to the next one and, on the way back, to deliver the *Reply* message from a relay node to the next one. Thus, $T_{\text{enable}}$ is the minimum time required for the *Request* message to travel all the way from the current relay node down to the last relay node ($H_{\max}$ hops away from the RSU) and the *Reply* message to travel the way back, in case there is no delay except the timer associated to the *Request* message and the lower layers delay. Typical values range between one to several ms, depending on the length of the data to be transmitted and on the air bit rate. To be conservative, we set $\delta = 10\,\text{ms}$. To summarize, a relay node can only send its *Reply* message after its local $T_{\text{enable}}$ timer has expired and as soon as all its children relay nodes have sent their own *Reply* messages.

We also define a backup mechanism to avoid deadlocks, i.e., to prevent a relay node $N$ from waiting an indefinite amount of time for a potential child node to send its *Reply*, when that child node moves out of $N$'s communication range. To this end, whenever a node becomes a relay, besides scheduling $T_{\text{enable}}$, it also schedules a backup timer, defined as follows:

$$T_{\text{bkp}} = (T^{\text{req}}_{\max} + 2\delta)\,(H_{\max} - H + 1)\,, \tag{9}$$

The mechanism that determines the cancellation of $T_{\text{bkp}}$ is similar to that defined for D-TC. In particular, a node cancels its $T_{\text{bkp}}$ as soon as it deems that its own $S_{\text{fcd}}$ has been sent by another relay node that is closer to the source node.

To be noted that with both D-TC and D-RC algorithms the timers are computed locally and individually by each vehicle. Also, the process of sending *Request* and *Reply* messages is broadcast-based, hence all the decisions related to retransmitting or ignoring a message are taken at the receiver side. Although no directionality is specifically defined for the *Request* and *Reply* messages, the mechanism that governs the dissemination and collection process in D-TC and D-RC indirectly creates a two-way wave for each collection cycle: a *forward wave* that propagates the *Request* message up to $H_{\max}$, and a *backward wave* that starts from the outer nodes (i.e., the ones with $H = H_{\max}$) and propagates back to the source node in a multi-hop fashion.


(a) LuST.

(b) Manhattan.

Figure 4: The simulated scenarios. The represented areas illustrate the ROI of each scenario.

The collection phases of D-TC and D-RC have several main differences with respect to Baseline:

- While in Baseline every vehicle that receives the *Request* generates its own *Reply*, in D-TC and D-RC only relay nodes generate *Reply* messages.
- Unlike Baseline, where a *Reply* message contains only one FCD record, in D-TC and D-RC a *Reply* can contain multiple FCD records (i.e., the ones in the LDM and the ones received from other child relay nodes via their *Reply* messages).
- In Baseline, intermediate relay nodes act as simple forwarders for the *Reply* messages. In D-TC and D-RC, however, *Reply* messages received from child relay nodes are not forwarded. Instead, they are locally processed – they are unpacked and their payload is merged with the local LDM in order to eliminate duplicated FCD records. The entire processed information is then packed into one *Reply* message and sent in broadcast (in one or multiple fragments) when the reply timeout expires.

To be noted that data integrity and privacy are paramount while periodically collecting FCD information. Although in this study we do not explicitly address these issues, the proposed protocols are compliant with the ETSI ITS-G5 standard, as we experimentally demonstrated in a recent study [27]. For this reason, all the security and privacy mechanisms standardized by ETSI in [28, 29] can be applied to our solutions.

## 4. Simulation setup

To validate our proposed solutions, we consider two simulation scenarios: Manhattan Grid (see Figure 4b) and Luxembourg Simulation of Urban Mobility (SUMO) Traffic (LuST) (see Figure 4a). Manhattan Grid represents a 2 km × 2 km square area with orthogonal bidirectional roads and regular buildings as radio propagation obstacles. We generated 11 horizontal and 31 vertical intersections. Each building is represented by a rectangular shape of 300 m × 100 m. The vehicular traffic model in Manhattan Grid is realized according to the “random trips” model. The movement of the vehicles is governed by the car-following model with a target speed of 50 km/h. The actual realized velocity may be lower than the target and depends on the vehicle density in each road lane.

The second and more realistic scenario represents Luxembourg City, a typical mid-size European city with typical characteristics in terms of road topology and mobility patterns. LuST [30, 31] is a realistic vehicular traffic scenario that was specifically built and tailored to support the evaluation of vehicular networking protocols and applications. In particular, LuST covers 932 km of roads and an area of 156 km$^2$, containing 38 different bus routes with 563 bus stops. For our simulations, we identified a 2 km × 2 km square area in the city center of Luxembourg, illustrated in Figure 4a. Road topology and segments, building geometry, points of interest, traffic signals, and other environment information have all been extracted from OpenStreetMap[2]. The buildings and the points of interest are represented by red polygons. The vehicular traffic model in LuST is based on a realistic mobility study that describes the traffic characteristics of Luxembourg City over recent years. LuST models the traffic pattern over a 24 h time period.

The mobility of vehicles is generated by the micro-mobility simulator SUMO [32]. SUMO is coupled with the OMNeT++[33] simulation tool and Veins [34], which are used to simulate the communication process, including the operations of the PHY, MAC, and network layers, as well as our protocol implementation. In this study, we employ the IEEE 802.11p vehicular communication technology to evaluate the performance of our protocols. However, it can be noticed that our proposed solutions and algorithms are media independent (i.e., they are placed above the current 802.11p standard). This means that any other Device-to-Device (D2D) communication paradigm, such as future 5G Vehicle-to-Everything (V2X), could be easily employed (for example, to reduce the data load at the eNodeB). To model the impact of buildings and other obstacles to signal propagation, we have used two attenuation models in tandem: the Two-Ray Interference model [35, 36] with $\epsilon_\mathrm{r} = 1.02$, and the Obstacle Shadowing model [37], which reproduces in Veins the shadowing effect of a real urban environment by describing the attenuation as a function of the depth of the buildings traversed by radio links.

An RSU, placed in the middle of the most central road of each scenario, acts as source node. It periodically triggers the FCD collection process by broadcasting *Request* messages every $T_\mathrm{col} = 5\,\mathrm{s}$. A simulation run lasts 100 s and every run is repeated 15 times for statistical confidence. 95 % confidence intervals are also computed. Three different vehicular densities have been considered in both scenarios: high, medium, and low. To have a fair comparison, the vehicle-per-lane density values in the two considered scenarios have been chosen to be similar (see Figure 5). In the LuST scenario, the 100 s simulation time is placed at three different points in time from the entire 24 h range, so as to cover three different vehicular densities. In particular, we identify a high vehicular density scenario at approximately 8:00 a.m., a medium-density scenario at 1:00 p.m., and a low-density scenario at 11:00 p.m.. The main simulation parameters are displayed in Table 2.

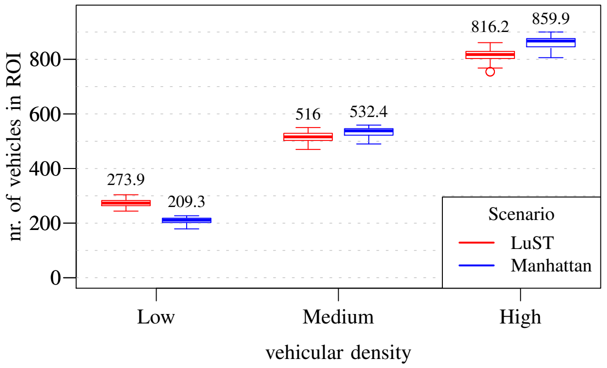


Figure 5: Number of vehicles in ROI for the low, medium, and high density scenarios.

| **Parameter** | **Value** |
|---|---|
| Simulation duration | 100 s |
| IVC technology | DSRC/IEEE 802.11p |
| DSRC maximum transmit power | 100 mW |
| DSRC beaconing frequency | 1 Hz |
| DSRC bitrate | 6 Mbit/s |
| Carrier frequency | 5.89 GHz |
| FCD size | 32 B |
| Beacon size | 32 B |
| $T_\mathrm{col}$ | 5 s |
| $D_\mathrm{max}$ | 100–800 m |
| $H_\mathrm{max}$ | 2–20 |
| $\alpha$, $\eta_\mathrm{max}$ | 0.8 and 0.2 |
| $\delta$ | 0.01 s |
| $T^\mathrm{req}_\mathrm{max}$, $T^\mathrm{rep}_\mathrm{max}$ | 0.1 and 0.5 s |
| MaxRTX | 0, 1, 2, 3, 4 and 5 |

Table 2: Main simulation parameters

## 5. Dissemination Phase Evaluation

The purpose of the data dissemination phase is for the *Request* message sent by the RSU to reach as many vehicles as possible within the target ROI. The dissemination procedure follows the specifications of the rCBF protocol that has been presented in Section 3.1. Notice that we make no distinction between the three presented FCD collection algorithms here, since they all exploit the same dissemination mechanism. In the following, we show only the results obtained when considering the LuST scenario, since the results for Manhattan are similar.

[2] www.openstreetmap.org

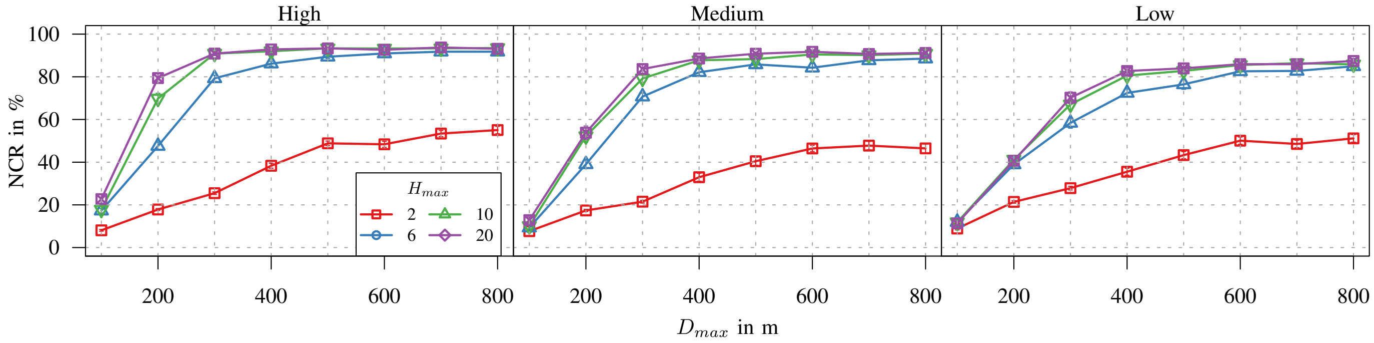


Figure 6: Mean Node Coverage Ratio (NCR) for different $D_{\max}$ and $H_{\max}$ values, and for different vehicular densities (LuST).

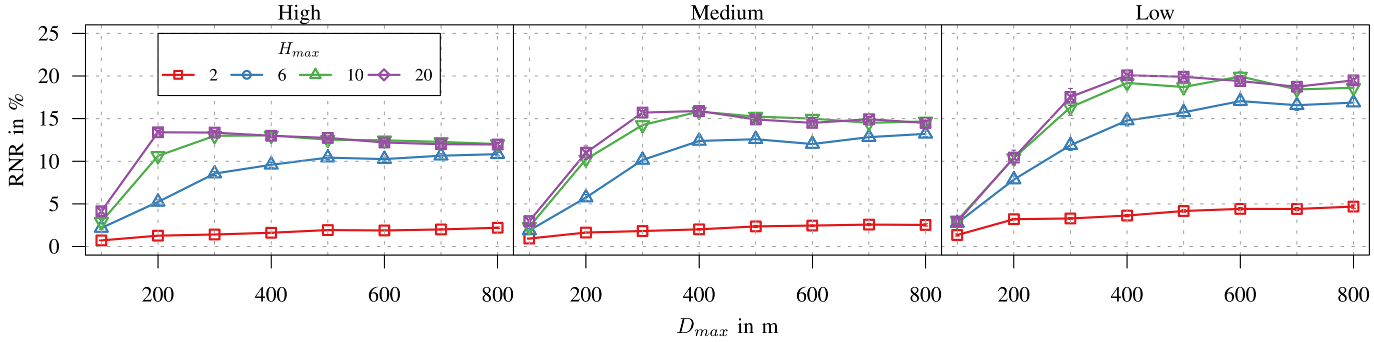


Figure 7: Mean RNR for different $D_{\max}$ and $H_{\max}$ values, and for different vehicular densities (LuST).

As key performance indicators, we define the NCR as the ratio of the number of vehicles that receive the *Request* message to the total number of vehicles roaming inside ROI at the time the *Request* message is issued by the RSU. From the protocol description we can see that $D_{\max}$ and $H_{\max}$ are the main parameters that can affect the NCR metric, which is why we vary these two parameters in our analysis.

In Figure 6 we show the performance in terms of NCR when varying $D_{\max}$ and $H_{\max}$. For low values of $D_{\max}$ rCBF is less effective, especially for the low vehicular density scenario. The main reason is that the algorithm is designed to allow only vehicles within $D_{\max}$ of the sending vehicle to participate in the relay node selection process. The smaller the value of $D_{\max}$, the higher the probability that there is no available vehicle within that distance. This means that dissemination is more likely to be stopped for lack of a next hop relay node for low values of $D_{\max}$. For a given value of $D_{\max}$, this is the more true the smaller the vehicle density. For $D_{\max} \geq 300\,\text{m}$ in case of high and medium densities, and $D_{\max} \geq 500\,\text{m}$ in case of low density, the dissemination algorithm is able to reach on average more than 80 % of vehicles roaming inside the ROI, provided $H_{\max}$ is high enough. While the lower bound of $D_{\max}$ for which the dissemination covers the entire ROI is dependent on the vehicle density, it is apparent from Figure 6 that NCR performance is robust with respect to the choice of $D_{\max}$, provided it is large enough. This suggests that higher values of $D_{\max}$ should be preferred.

As for the maximum number of hops, lower values of $H_{\max}$ lead to fewer vehicles being reached by the *Request* message: by increasing the value of $H_{\max}$ the *Request* is allowed to be propagated further away from the RSU, hence covering a larger number of vehicles. However, since the ROI area is limited, there is a threshold for $H_{\max}$ beyond which further hops have a negligible effect on the NCR level attained. In our case, we noticed that this threshold is $H_{\max} = 10$ for both LuST and Manhattan Grid scenarios, which is why we see almost no difference when increasing the value to $H_{\max} = 20$.

Another metric of interest to dissemination performance is the Relay Nodes Ratio (RNR), defined as the ratio of the number of relaying vehicles to the total number of vehicles roaming inside ROI. The smaller the number of relay nodes for a given scenario, the lower the contention on the radio interface. This is beneficial both for the dissemination protocol itself and for any co-existing service that shares the same channel as the VANET. Figure 7 illustrates the RNR metric for different values of $D_{\max}$ and $H_{\max}$, as well as for the three considered vehicular densities. As expected, the RNR values are smaller for lower values of $H_{\max}$, since fewer vehicles are being reached by the *Request* message. Also, RNR is smaller for high vehicular density scenarios and larger for low-density scenarios. This is simply because rCBF tends to elect relaying vehicles that are geographically separated by a distance $D_{\max}$, independently of how

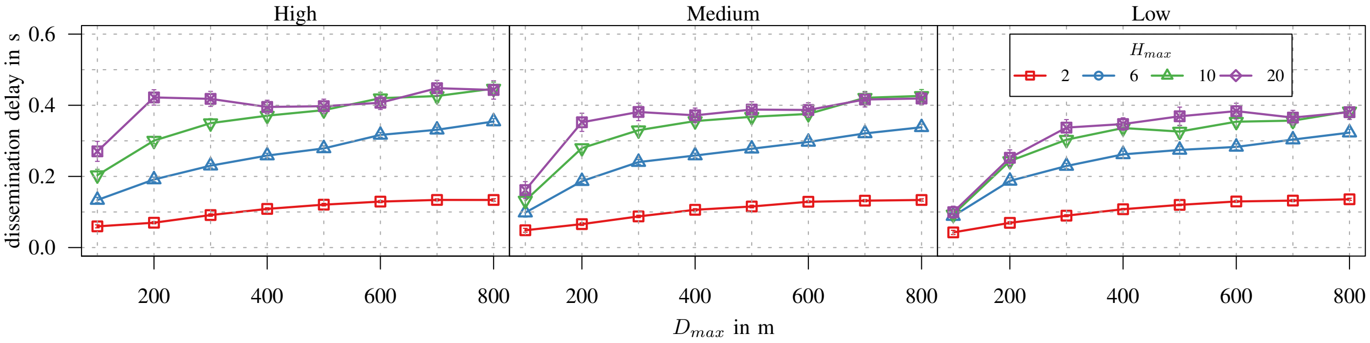


Figure 8: Mean dissemination delay for different $D_{\max}$ and $H_{\max}$ values, and for different vehicular densities (LuST).

many vehicles are in the area. This means that even if the total number of vehicles in the scenario increases, the number of selected relay nodes remains more or less the same. As a consequence, rCBF is actually more efficient in terms of RNR in high-density scenarios, with RNR $\approx 14\,\%$, with respect to low-density scenarios, where RNR $\approx 20\,\%$. We observed that the RNR values are slightly higher for Manhattan Grid (results not shown), because it presents more buildings and sharp corners blocking the communication, leading to more relay vehicles being selected.

When varying $D_{\max}$ the RNR values initially grow up to a certain point, and then start decreasing as $D_{\max}$ increases further. The very low RNR values when $D_{\max}$ is low are due to the fact that the dissemination process is interrupted prematurely, as can be seen from Figure 6. They reach a peak when $D_{\max}$ is great enough to cover the entire ROI (i.e., $D_{\max} = 400\,\mathrm{m}$ for low-density scenario). Since the selected relay vehicles tend to be separated by a distance $D_{\max}$, further increasing the values of $D_{\max}$ slightly decreases the number of selected relay nodes. This depends in turn on the radio coverage range of the DSRC equipment in the considered scenario.

Finally, Figure 8 shows the dissemination delay, measured as the time interval between the moment when the RSU issues the *Request* message and the time when the last vehicle inside our monitored area receives it. We can see that, in the worst case scenario, it takes roughly 0.5 s for the dissemination phase to reach all possible vehicles inside our area of interest. The results are consistent with the fact that increasing $H_{\max}$ allows us to reach more vehicles, hence increases the dissemination delay. At the same time, low values of $D_{\max}$ lead to smaller dissemination delays, but only because fewer vehicles are reached by the *Request* message. The dissemination delay does not change much once $D_{\max}$ values are large enough to cover the entire ROI.

Summing up, the analysis of the performance of the dissemination phase suggests that a good trade-off is achieved by setting $D_{\max}$ in the range 400 m-600 m, and $H_{\max}$ between 8 and 10.

## 6. Collection Phase Evaluation

The purpose of the collection phase is to gather FCD information from all vehicles roaming inside a ROI, striking a balance between reliability, timeliness and load on the DSRC communication channel. In the following, we evaluate the performance of the three protocols described in Sections 3.2 to 3.4, namely Baseline, D-TC, and D-RC. Since the results for the two considered simulation scenarios are similar, we show only the ones for LuST.

The performance metrics that we consider in this section characterize the efficiency of the FCD collection, the time needed to perform one collection cycle, and the impact of the collection procedure on the DSRC communication channel. To measure the efficiency of the FCD collection process, we define the Monitored Vehicles Ratio (MVR) metric, which is computed as the ratio of the number of vehicles whose FCDs arrived at the RSU at the end of a collection phase (i.e., vehicles monitored by the RSU) to the total number of vehicles that received the *Request* message. The collection delay, that is the time needed by the algorithm to complete a single collection cycle, is measured as the time elapsing since the moment when the RSU starts the collection process by issuing the *Request* message, until when the RSU receives the last non-duplicate FCD message triggered by the *Request* message. Finally, to measure the DSRC channel congestion level, we define the Channel Busy Ratio (CBR) metric, computed as the ratio of the amount of time the DSRC channel is busy around a vehicle to the total simulation time related to that vehicle. It is an estimate of the probability that a vehicle node finds the DSRC channel busy.

### 6.1. Collection efficiency analysis

Figure 9 illustrates the performance of D-TC, D-RC, and Baseline in terms of MVR metric when varying $H_{\max}$ and the vehicular density scenario. To evaluate the impact of the backup mechanism in case of D-TC and D-RC, we also vary the maximum number of backup retransmissions (MaxRTX). There is no backup mechanism for Baseline, hence only one curve is shown (i.e., the black solid line). We set $D_{\max} = 600\,\mathrm{m}$, since this is a reasonable value that

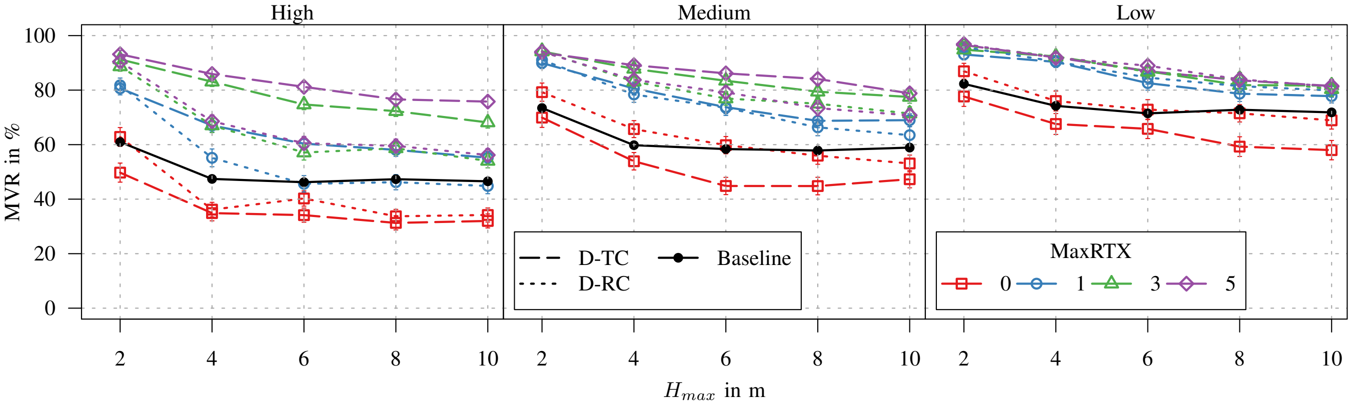


Figure 9: Mean MVR for different $H_{\max}$ and MaxRTX values, and for different vehicular density scenarios (LuST).

guarantees a good coverage for *Request* message dissemination (see Figure 6).

It is apparent that the backup mechanism significantly improves the efficiency of the FCD collection. To be noted that MaxRTX = 0 means there is no backup mechanism, which, in case of D-TC, is basically the original DISCOVER algorithm proposed in [7, 8]. We can see that with no backup mechanism the MVR values obtained with both D-TC and D-RC are generally worse than Baseline, and can drop to approximately 30 % (see D-TC in high density scenario). Such poor performance is a result of the high number of *Reply* packet collisions. Remember that the size of the *Reply* messages increases as they get closer to the RSU, since more information is being merged together. Hence, without a backup mechanism, the collisions have a significant impact on the protocol performance. This is not true for Baseline, where the *Reply* messages contain only one FCD record and they are forwarded individually, i.e., their size remains constant, since no data merging is performed.

As soon as we set MaxRTX $\geq 1$ the MVR values are significantly higher and can get close to 100 % in some cases (e.g., MaxRTX = 5 and $H_{\max} = 2$). For high-enough values of MaxRTX, both D-TC and D-RC outperform Baseline in all of the considered scenarios and under all vehicular densities. Of the three algorithms, D-TC is the one that has the best performance in terms of MVR, especially under high vehicular densities, where we see D-RC being less effective. In low-density scenarios, D-TC and D-RC provide similar performance. Numerically, if we consider $H_{\max} = 10$ (which is enough to cover the entire ROI) and MaxRTX = 5 in LuST, we can see that MVR varies from 75 % to less than 50 % with Baseline, from 80 % to 55 % with D-RC, and from 80 % to 75 % with D-TC, when increasing the vehicular density.

Another thing that we can notice is that the FCD collection efficiency degrades when increasing $H_{\max}$. This happens mainly because of two reasons: (i) the amount of collected information increases with $H_{\max}$, and (ii) it is more likely to loose the information when it has to travel more hops. Remember that higher $H_{\max}$ values allow more vehicles in ROI to be reached (see Figure 6), which means that more FCDs have to make their way back to the RSU, increasing the size of the *Reply* messages. This is also confirmed by the fact that MVR decreases less when the vehicular density is low and drops much faster for higher vehicular densities when increasing $H_{\max}$. The first reason does not hold for Baseline, since the size of *Reply* messages, when considering this protocol, is constant. The second reason however does have a small impact on Baseline. In fact, we can see that for $4 \leq H_{\max} \leq 10$ MVR remains constant, while there is a slight improvement when $H_{\max} = 2$.

### *6.2. Collection delay analysis*

Figure 10 illustrates the overall collection delay of D-TC, D-RC, and Baseline. It can be seen that, although the backup mechanism increases the number of collected FCDs, in case of D-TC it also increases the collection delay. In fact, while for MaxRTX = 0 the collection delay varies between 0.5 s (for $H_{\max} = 2$) and 1.5 s (for $H_{\max} = 10$), independent of the considered scenario and/or vehicular density, for MaxRTX = 5 the collection delay can rise up to 4 s in the worst case. In addition, when considering the backup mechanism, the delay increases with the vehicular density, which means that more packet collisions occur, hence more information has to be retransmitted. In case of D-RC we see very little difference when increasing MaxRTX, especially in low vehicular density scenarios. There is only a slight increase with MaxRTX for high vehicular density (i.e., the collection delay goes from 1.25 s for MaxRTX = 0 to nearly 1.5 s for MaxRTX = 5 and $H_{\max} = 10$). Also, there is very little difference between low and high vehicular density scenarios.

For all three protocols, the collection delay grows also with $H_{\max}$, although with different slopes. This result can be expected, since the algorithms need more time to collect FCD information from a higher number of hops. The most efficient solution in terms of collection delay is Baseline, which is almost constant with a slight increase

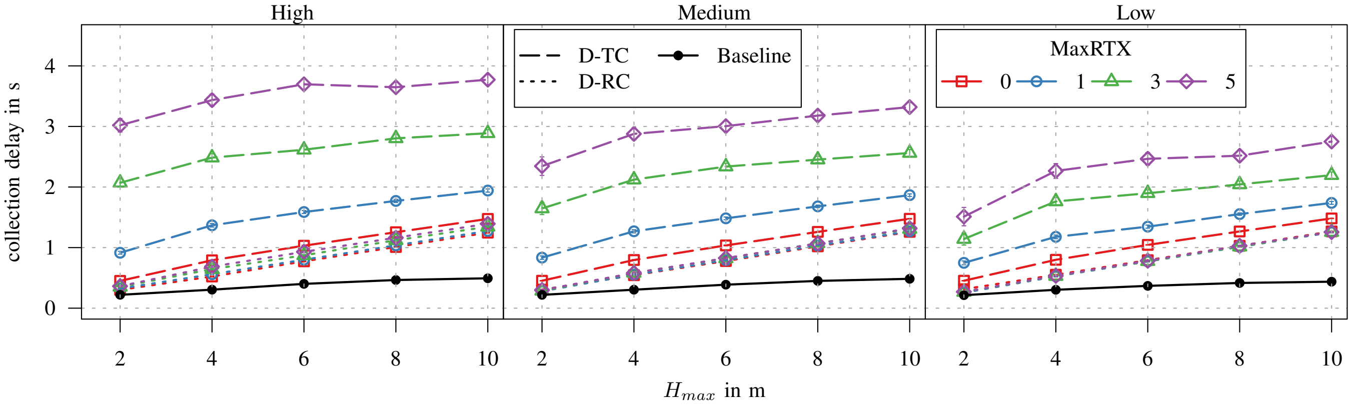


Figure 10: Total collection delay for different $H_{\max}$ and MaxRTX values, and for different vehicular density scenarios (LuST).

for higher values of $H_{\max}$, with a maximum value of 0.5 s. Among the three considered protocols, D-TC has the worst performance. Basically, the price D-TC pays in order to get better MVR is a higher delay, which can reach up to 4 s. In comparison, the highest collection delay observed with D-RC is 1.5 s. This suggests the fact that there is a trade-off between the collection efficiency and speed.

### *6.3. Impact on the communication channel*

Figure 11 shows the impact of D-TC, D-RC, and Baseline on the communication channel. As expected, having more vehicles sending messages on the same shared wireless communication channel increases congestion levels, hence the CBR values can reach 4.5 % (in case of D-TC) for high-density scenarios, while the CBR level drops to below 1 % for low-density scenarios. This is true for all three considered protocols.

In addition, the backup mechanism contributes to the CBR level, which increases with MaxRTX. This confirms our observations from Figure 10, according to which the backup mechanism generates more retransmissions, hence more load on the communication channel. In fact, the efficiency of the backup mechanism results from successfully retransmitting the lost information, which increases the amount of information collected. But this also introduces a higher load on the communication channel, which can be seen from the higher levels of measured CBR. These observation suggest that there is a trade-off between the efficiency of the FCD collection in terms of MVR and the impact the protocols have on the communication channel and on collection delay.

The highest CBR is introduced by D-TC, as we can see in Figure 11. D-RC's backup mechanism is less efficient, and so introduces less load on the communication channel. The most efficient algorithm in terms of measured CBR is Baseline, given the lack of a backup mechanism. However, we must stress that the relatively light load on the DSRC channel is mainly due to the poor performance of Baseline in terms of MVR: less FCD is collected, hence the smaller load.

To give an illustrative idea of what an RSU is able to collect after one collection cycle, in Figure 12 we show the result of one simulated collection instance with the D-TC algorithm divided into the two phases: dissemination (Figure 12a) and collection (Figure 12b). In Figure 12a vehicles colored in blue have received the *Request* issued by the RSU; green vehicles are roaming inside ROI but did not receive the *Request* message (e.g., because of collisions, of being temporarily disconnected from the DSRC graph, etc); while the brown vehicles are the relay nodes selected by the dissemination algorithm. Those relay nodes will be the ones participating in the collection phase. In Figure 12b cyan colored vehicles are the ones whose FCDs arrived at the RSU at the end of the collection phase, i.e., the monitored vehicles. As we can see, D-TC is able to collect the FCDs from most of the vehicles that received the *Request* message, that is those nodes that belong to the connected component of the DSRC graph centered on the RSU node.

## 7. Conclusion

Having access to real-time FCD information from vehicles enables a wide range of applications in the context of ITS. Today, the most mature vehicular communication technology able to support such collection operations is DSRC. In this paper we have designed, evaluated and compared three FCD collection protocols that aim to maximize the amount of the collected information while limiting the impact on the DSRC communication channel: Baseline, D-TC, and D-RC. The proposed solutions exploit a slightly modified version of the existing standard for data dissemination in VANETs, namely the ETSI ITS-G5 GeoNetworking CBF protocol, to create a backbone of relay nodes responsible for the collection operation. The collection process is triggered periodically by a fixed infrastructure node and is carried out in a multi-hop fashion up to a desired number of hops. The protocols are evaluated considering different vehicular densities and two different urban scenarios: LuST, a realistic scenario representing the traffic patterns and communication obstacles (e.g., buildings) in the city

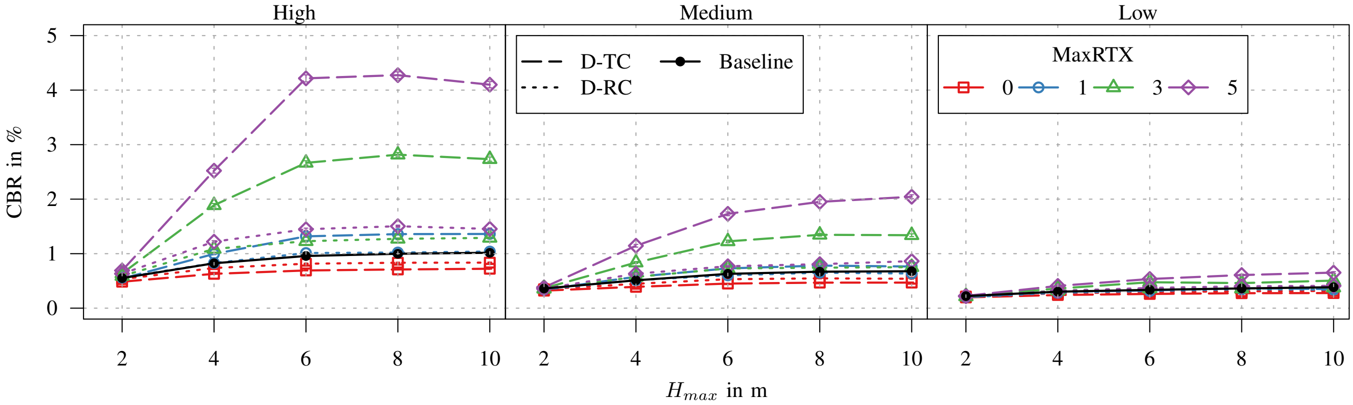
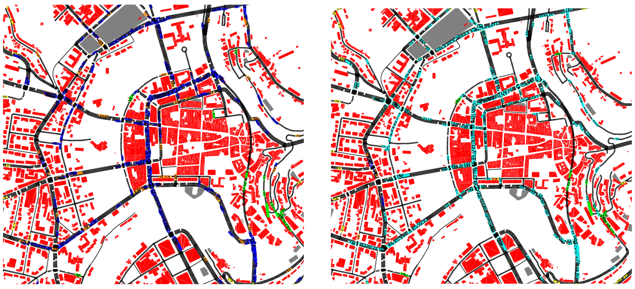


Figure 11: Mean CBR for different $H_{\max}$ and MaxRTX values, and for different vehicular density scenarios (LuST).



(a) Dissemination phase.

(b) Collection phase.

Figure 12: An illustrative example of a collection cycle using D-TC.

of Luxembourg, and Manhattan Grid, an artificial scenario with synthetic mobility and obstacle definition.

The performance evaluation results obtained indicate that D-TC outperforms D-RC and Baseline in terms of collected FCDs, but at the price of a higher collection delay. D-RC performs quite well for low-density scenarios, but the amount of FCDs collected decreases faster for high-density scenarios when compared to D-TC. Both D-TC and D-RC outperform Baseline, which suggests that aggregating the collected information at every relay node brings a significant benefit both in terms of collected information and impact on the communication channel. Finally, we showed that increasing the desired area to monitor, i.e., raising the maximum number of hops up to which one would like to collect FCD information, decreases the efficiency of the collection operation. This is consistent with the fact that increasing the ROI span leads to greater amount of information being collected, stressing the communication channel more, hence increasing the probability of loosing some information.

As future work, we plan to study the stability of the backbone network under different vehicular traffic conditions. This will allow us to gain more insights into how often the relay nodes have to be re-selected before noticing a significant decrease of FCD collection performance. For example, the collection phase of the proposed FCD collection algorithms is always preceded by a dissemination phase that selects a new set of relay nodes. A potential optimization would be having multiple collection phases that exploit the same backbone network of relay nodes.

## References

[1] J. B. Kenney, Dedicated Short-Range Communications (DSRC) Standards in the United States, Proceedings of the IEEE 99 (7) (2011) 1162–1182. doi:10.1109/JPROC.2011.2132790.

[2] C. Huang, R. Lu, K.-K. R. Choo, Vehicular fog computing: architecture, use case, and security and forensic challenges, IEEE Communications Magazine 55 (11) (2017) 105–111. doi:10.1109/MCOM.2017.1700322.

[3] R. Soua, I. Turcanu, F. Adamsky, D. Führer, T. Engel, Multi-access edge computing for vehicular networks: A position paper, in: 2018 IEEE Globecom Workshops (GC Wkshps), IEEE, 2018, pp. 1–6.

[4] R. S. Bali, N. Kumar, J. J. Rodrigues, Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions, Elsevier Vehicular Communications 1 (3) (2014) 134–152. doi:10.1016/j.vehcom.2014.05.004.

[5] ETSI, Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality, EN 302 636-4-1 v1.2.1, ETSI (May 2014).

[6] W. Zhang, Z. Zhang, H.-C. Chao, Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management, IEEE Communications Magazine 55 (12) (2017) 60–67. doi:10.1109/MCOM.2017.1700208.

[7] 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, & Ubiquitous Networks, ACM, 2015, pp. 25–32.

[8] I. Turcanu, P. Salvo, A. Baiocchi, F. Cuomo, An integrated VANET-based data dissemination and collection protocol for complex urban scenarios, Elsevier Ad Hoc Networks 52 (Supplement C) (2016) 28–38, Modeling and Performance Evaluation of Wireless Ad Hoc Networks. doi:10.1016/j.adhoc.2016.07.008.

[9] M. Chaqfeh, A. Lakas, I. Jawhar, A survey on data dissemination in vehicular ad hoc networks, Vehicular Communications 1 (4) (2014) 214 – 225. doi:10.1016/j.vehcom.2014.09.001. URL `http://www.sciencedirect.com/science/article/pii/S2214209614000448`

[10] B. Pourghebleh, N. Jafari Navimipour, Towards efficient data collection mechanisms in the vehicular ad hoc networks, International Journal of Communication Systems (2019) e3893.

[11] R. Oliveira, C. Montez, A. Boukerche, M. S. Wangham, Reliable data dissemination protocol for VANET traffic safety applications, Ad Hoc Networks 63 (Supplement C) (2017) 30 – 44. doi:10.1016/j.adhoc.2017.05.002.
URL `http://www.sciencedirect.com/science/article/pii/S1570870517300835`

[12] L. W. Chen, P. C. Chou, BIG-CCA: Beacon-Less, Infrastructure-Less, and GPS-Less Cooperative Collision Avoidance Based on Vehicular Sensor Networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (11) (2016) 1518–1528. doi:10.1109/TSMC.2015.2504040.

[13] A. Baiocchi, P. Salvo, F. Cuomo, I. Rubin, Understanding Spurious Message Forwarding in VANET Beaconless Dissemination Protocols: An Analytical Approach, IEEE Transactions on Vehicular Technology 65 (4) (2016) 2243–2258. doi:10.1109/TVT.2015.2422753.

[14] D. Cao, Y. Liu, X. Ma, J. Wang, B. Ji, C. Feng, J. Si, A Relay-node Selection on Curve Road in Vehicular Networks, IEEE Access 7 (2019) 12714–12728. doi:10.1109/ACCESS.2019.2892979.

[15] O. Rehman, M. Ould-Khaoua, A hybrid relay node selection scheme for message dissemination in VANETs, Future Generation Computer Systems 93 (2019) 1 – 17. doi:10.1016/j.future.2018.10.042.
URL `http://www.sciencedirect.com/science/article/pii/S0167739X18320417`

[16] A. Soua, H. Afifi, Adaptive data collection protocol using reinforcement learning for VANETs, in: 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013, pp. 1040–1045. doi:10.1109/IWCMC.2013.6583700.

[17] Z. He, D. Zhang, Cost-efficient traffic-aware data collection protocol in VANET, Ad Hoc Networks 55 (Supplement C) (2017) 28 – 39, Self-organizing and Smart Protocols for Heterogeneous Ad hoc Networks. doi:10.1016/j.adhoc.2016.09.021.
URL `http://www.sciencedirect.com/science/article/pii/S1570870516302360`

[18] I. Turcanu, F. Klingler, C. Sommer, A. Baiocchi, F. Dressler, Duplicate suppression for efficient floating car data collection in heterogeneous LTE-DSRC vehicular networks, Computer Communications 123 (2018) 54 – 64.

[19] I. Turcanu, C. Sommer, A. Baiocchi, F. Dressler, Pick the Right Guy: CQI-Based LTE Forwarder Selection in VANETs, in: 8th IEEE Vehicular Networking Conference (VNC 2016), IEEE, Columbus, OH, 2016, pp. 98–105. doi:10.1109/VNC.2016.7835938.

[20] H. Huang, M. Hassan, G. Geers, L. Libman, CarAgent: Collecting and disseminating floating car data in vehicular networks, Vehicular Communications 13 (2018) 114 – 127. doi:10.1016/j.vehcom.2018.08.002.
URL `http://www.sciencedirect.com/science/article/pii/S2214209618300135`

[21] F. Zeng, R. Zhang, X. Cheng, L. Yang, UAV-Assisted Data Dissemination Scheduling in VANETs, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422219.

[22] Q. Zhang, H. Zheng, J. Lan, J. An, H. Peng, An Autonomous Information Collection and Dissemination Model for Large-Scale Urban Road Networks, IEEE Transactions on Intelligent Transportation Systems 17 (4) (2016) 1085–1095.

[23] P. Salvo, F. Cuomo, A. Baiocchi, I. Rubin, Investigating VANET dissemination protocols performance under high throughput conditions, Vehicular Communications 2 (4) (2015) 185–194.

[24] O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, V. K. Sadekar, On the Broadcast Storm Problem in Ad hoc Wireless Networks, in: 3rd International Conference on Broadband Communications, Networks and Systems, 2006, pp. 1–11. doi:10.1109/BROADNETS.2006.4374403.

[25] ETSI, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, Tech. Rep. 302 637-2 V1.3.2, ETSI (Nov. 2014).

[26] ETSI, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM), EN 302 895 V1.1.1, ETSI (Sep. 2014).

[27] I. Turcanu, F. Adamsky, T. Engel, Experimental Evaluation of Floating Car Data Collection Protocols in Vehicular Networks, in: 90th IEEE Vehicular Technology Conference (VTC2019-Fall), IEEE, Honolulu, Hawaii, USA, 2019.

[28] ETSI, Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management, TS 102 940 v1.2.1, ETSI (Nov. 2016).

[29] ETSI, Intelligent Transport Systems (ITS); Security; Trust and Privacy Management, TS 102 941 v1.2.1, ETSI (May 2018).

[30] L. Codeca, R. Frank, T. Engel, Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research, in: 7th IEEE Vehicular Networking Conference (VNC 2015), IEEE, Kyoto, Japan, 2015, pp. 1–8. doi:10.1109/VNC.2015.7385539.

[31] L. Codeca, R. Frank, S. Faye, T. Engel, Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation, IEEE Intelligent Transportation Systems Magazine 9 (2) (2017) 52–63. doi:10.1109/MITS.2017.2666585.

[32] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent Development and Applications of SUMO – Simulation of Urban MObility, International Journal On Advances in Systems and Measurements 5 (3&4) (2012) 128–138.

[33] A. Varga, R. Hornig, An overview of the OMNeT++ simulation environment, in: 1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008), ACM, Marseille, France, 2008, pp. 60:1–60:10.

[34] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, IEEE Transactions on Mobile Computing 10 (1) (2011) 3–15. doi:10.1109/TMC.2010.133.

[35] C. Sommer, F. Dressler, Using the Right Two-Ray Model? A Measurement based Evaluation of PHY Models in VANETs, in: 17th ACM International Conference on Mobile Computing and Networking (MobiCom 2011), Poster Session, ACM, Las Vegas, NV, 2011, pp. 1–3.

[36] C. Sommer, S. Joerer, F. Dressler, On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation, in: 4th IEEE Vehicular Networking Conference (VNC 2012), IEEE, Seoul, Korea, 2012, pp. 64–69. doi:10.1109/VNC.2012.6407446.

[37] C. Sommer, D. Eckhoff, R. German, F. Dressler, A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments, in: 8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011), IEEE, Bardonecchia, Italy, 2011, pp. 84–90. doi:10.1109/WONS.2011.5720204.