

Experimental Evaluation of Floating Car Data Collection Protocols in Vehicular Networks

Ion Turcanu*, Florian Adamsky*†, and Thomas Engel*

*SnT, University of Luxembourg, Luxembourg

{name.surname}@uni.lu

†Hof University, University of Applied Sciences, Germany

{name.surname}@hof-university.de

Abstract—The main objectives of the Intelligent Transportation Systems (ITS) vision is to improve road safety, traffic management, and mobility by enabling cooperative communication among participants. This vision requires the knowledge of the current state of the road traffic, which can be obtained by collecting Floating Car Data (FCD) information using Dedicated Short-Range Communication (DSRC) based on the IEEE 802.11p standard. Most of the existing FCD collection protocols have been evaluated via simulations and mathematical models, while the real-world implications have not been thoroughly investigated. This paper presents an open-source implementation of two state-of-the-art FCD collection algorithms, namely BASELINE and DISCOVER. These algorithms are implemented in an open-source vehicular prototyping platform and validated in a real-world experimental setup.

I. INTRODUCTION

Recent advances in the research and standardization of Vehicle-to-Everything (V2X) communication technologies are bringing connected cars and Intelligent Transportation Systems (ITS) closer to reality. Vehicular Ad Hoc Networks (VANETs), defined as networks of vehicles that can communicate with each other and/or with the infrastructure, are among the main enablers of connected and cooperative mobility, aiming at improving the safety, efficiency, and comfort of future ITS. In a VANET, every vehicle becomes a network node acting as both provider and consumer of information. The wide range and variety of on-board sensors installed in today's cars allows them to sense not only their internal status, but also the surrounding environment, becoming moving hubs of useful data. The information provided by vehicles is known in the literature as Floating Car Data (FCD) and can enable a broad variety of new ITS services and applications, such as collaborative automated driving and network-assisted Advanced Driver-Assistance Systems (ADAS).

Periodically collecting FCD information from vehicles is a challenging task mainly due to the highly dynamic nature of VANETs and the unreliability of the communication links. Plenty of research has been carried out in recent years proposing different solutions for efficient FCD collection [1]. Some of these works propose using Dedicated Short-Range Communication (DSRC), based on the IEEE 802.11p wireless communication standard, to collect real-time FCD information [2]–[5]. Given the scarce deployment of DSRC-

enabled infrastructure nodes, known as Road Side Units (RSUs), most of these works focus on multi-hop communications to allow FCDs propagate towards the nearest RSU. In addition, since FCD collection protocols have to share the same bandwidth with high-priority vehicular safety applications, the main challenge faced by the proposed solutions is to limit the impact on the DSRC communication channel while maximizing the amount of collected FCDs.

Other works propose to collect FCD information using the cellular networks [6], [7]. However, such solutions depend on the availability of the mobile network operators. Also, the current architecture of the cellular networks was not designed to support frequent and periodic data transmissions from vehicles, which will degrade the normal operation for the traditional users. For this reason, some works propose the use of heterogeneous DSRC/cellular networks to collect FCD information [8]–[10], where DSRC is typically used to offload data from the cellular network.

All of the above mentioned FCD collection protocols are evaluated using computer simulations using state-of-the-art simulation models and tools. Among the main advantages of simulation-based evaluations are availability, flexibility, and scalability: the possibility to perform relatively accurate evaluations with different vehicular densities, parameters, measured metrics, network loads, etc. However, an experimental evaluation of FCD collection protocols would further help their validation and pave the way towards their actual adoption on the roads. To the best of our knowledge, none of the existing FCD collection solutions has been implemented in software and validated via experimental evaluations on real hardware.

In one of our previous works we propose DISCOVER [3], a DSRC-based FCD collection protocol for VANETs that aims at maximizing the number of collected FCDs while limiting the impact on the communication channel by selecting a subset of relay nodes to be in charge of the collection process. These relay nodes exploit timers and data merging to reduce the impact of the packet header overhead. In this paper, we evaluate the performance of this protocol via laboratory experiments using commodity hardware and an open-source implementation of the European Telecommunications Standards Institute (ETSI) ITS-G5 standard [11] for vehicular communications. In addition, we compare DISCOVER's performance with a basic FCD

collection protocol, named BASELINE, that exploits the same subset of relay nodes used by DISCOVER, but without merging the collected information. The main contributions of this paper can be summarized as follows:

- it presents two open-source implementations of FCD collection protocols based on OpenC2X [12] – an open-source vehicular prototyping platform;
- it presents a first experimental study of FCD collection protocols in multi-hop DSRC-based VANETs;
- it describes the design of an FCD collection service that can be integrated within the ETSI ITS-G5 protocol stack.

II. FCD COLLECTION PROTOCOLS

Broadly speaking, FCD collection consists of periodic delivery of vehicular data to a remote server. The FCD collection operation can be periodically triggered by a *source node* that can either be a vehicle or an infrastructure node, like an RSU. The source node initiates the collection process by periodically issuing a request. Every vehicle receiving this request replies with its own FCD information. The RSUs can act as source nodes, assuming they are deployed densely enough. Vehicles roaming in the wireless transmission range of these RSUs and receiving the request could simply broadcast their FCDs, directly reaching the nearest RSU that can deliver this information to the remote server via the backhaul network. However, it is unlikely to have an RSU in the immediate neighbourhood of every vehicle at any time. To overcome this issue, the requests coming from a generic source node and the respective generated FCDs could be propagated via multi-hop communication through the DSRC-based VANET.

To avoid the broadcast storm problem [13] when disseminating the requests, one could use any data dissemination algorithm that limits the number of relaying vehicles [14]. We choose a modified version of the ETSI ITS-G5 GeoNetworking Contention-Based Forwarding (CBF) protocol [15]: the same approach used in [3] to disseminate the requests. Contention-Based Forwarding is a data dissemination protocol that uses timers as the prime mechanism to select a backbone of relaying vehicles to participate in the message dissemination process while inhibiting the others. These timers are proportional to each vehicle's distance D from the sending vehicle, and are computed as follows:

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

where T_{\max}^{req} is a parameter defining the maximum timeout value, D_{\max} is the desired distance separating two consecutive relay nodes, D is the current distance from the sending vehicle, α is a weighting parameter, and $\mathcal{U}(0, 1)$ is a uniformly distributed random value needed to avoid simultaneous retransmissions. According to Equation (1), the shorter the distance from the sender, the longer a vehicle waits. The idea is to let vehicles that are further away from the sender forward the request, hence becoming relay nodes. A vehicle receiving a second

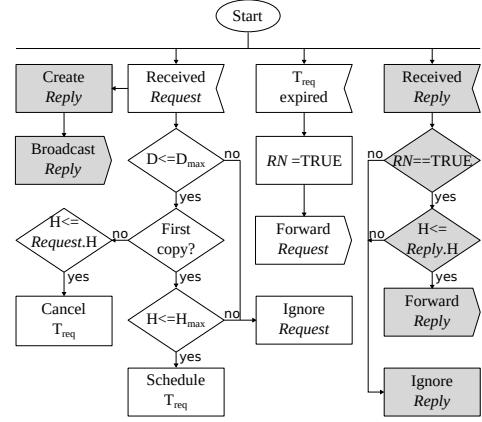


Figure 1. BASELINE FCD collection protocol. White boxes are related to the request dissemination process. Gray boxes illustrate the FCD collection operation.

copy of the same request, while still counting down its T_{req} , will cancel its respective timer.

The purpose of these relay nodes is not only to disseminate the requests sent by the source node, but also to forward the replies containing every vehicle's FCD. A basic FCD collection protocol, named BASELINE, that exploits the relay nodes in order to propagate the replies towards the source node is presented in Figure 1. The main idea behind this protocol is the following: whenever a vehicle V receives an FCD collection request, it broadcasts a reply containing its own FCD information. If vehicle V is in the direct communication range of the source node, then its FCD will be directly received by the latter. Otherwise, we assume the source node defines a hop count $H = 0$ for the FCD collection request and that every relay node increments this hop count by one. A simple rule to let FCDs propagate towards the source node is to let other relay nodes having a smaller hop count H than vehicle V to retransmit its FCD. In this way, without explicitly defining a propagation direction for the single FCDs (all transmissions are broadcast-based), they are still being propagated towards the source node.

Notice that with BASELINE every vehicle receiving the FCD collection request will generate and broadcast its own FCD, meaning that every vehicle has to access the communication channel at least once. In addition, the relay nodes have to access and retransmit on the DSRC communication channel every time a new FCD is being received from another vehicle with a higher hop count H . Assuming a certain relay node i receives N FCDs to be retransmitted and that the header length of a data packet is L , then the additional load on the communication channel is NL .

To limit this additional overhead, the authors of [3] propose DISCOVER, a timer-based FCD collection protocol that allows every relay node to aggregate the received FCDs and transmit only one packet containing all the merged information. The main operation of DISCOVER is illustrated in Figure 2. The authors assume that vehicles periodically broadcast Cooperative Awareness Messages (CAMs) [16]—one-hop beacon messages containing basic vehicle status information, such as motion

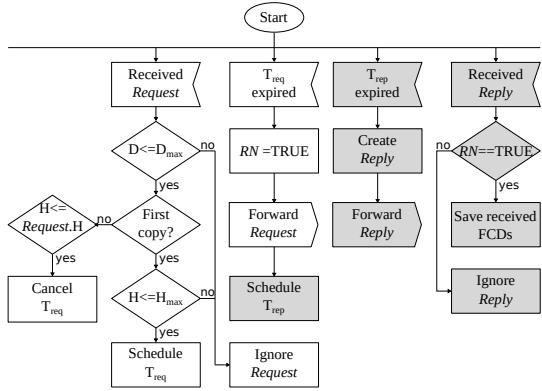


Figure 2. DISCOVER FCD collection protocol. White boxes are related to the request dissemination process. Gray boxes illustrate the FCD collection operation.

(speed, acceleration, heading), geolocation, and vehicle characteristics. It is safe to assume that such a process is in place, given the main motivation behind vehicular networks, which is enabling safety on the roads through vehicular cooperative awareness. The data extracted from incoming CAMs is stored and periodically updated by every vehicle in a local database, named Local Dynamic Map (LDM) [17].

Similarly to BASELINE, DISCOVER exploits the same chain of relay nodes created by the request dissemination process to forward the replies. However, while in BASELINE every vehicle generates and broadcasts its own FCD, DISCOVER allows only relay nodes to send their replies. Another major difference is that in DISCOVER the relay nodes send not only their own FCDs, but all the information contained in their LDMs. In this way, by exploiting the CAMs exchanged for safety applications, DISCOVER aims at collecting FCDs from all the vehicles roaming in a target area with less overhead and physical accesses on the DSRC communication channel. The timers that grant access to the communication channel are computed locally by every relay node as follows:

$$T_{\text{rep}} = (T_{\text{rep}}^{\max} + H_{\max} T_{\text{req}}^{\max}) \left(1 - \frac{H}{H_{\max}} \right) \quad (2)$$

Here T_{rep}^{\max} is a time bound, H is the current hop count, and H_{\max} is the maximum number of hops the request is allowed to be retransmitted. This timeout setting ensures the fact that inner relay nodes hold back long enough to receive the reply messages from outer relay nodes and are thus able to merge the received FCDs before replying. When T_{rep} expires, the relay node generates a reply containing as payload the extracted set of FCDs from the LDM, as well as all the received FCDs coming from other relay nodes whose T_{rep} expired before. We refer interested readers to [3] for a complete description of the DISCOVER algorithm.

An illustrative example of how these protocols work is shown in Figure 3, where the FCD collection operation is carried out up to a maximum number of three hops. Here, the RSU acts as source node, while the blue vehicles represent the relay nodes in charge of forwarding the *Request* and *Reply* messages, selected according to Equation (1). In case of BASELINE, every vehicle

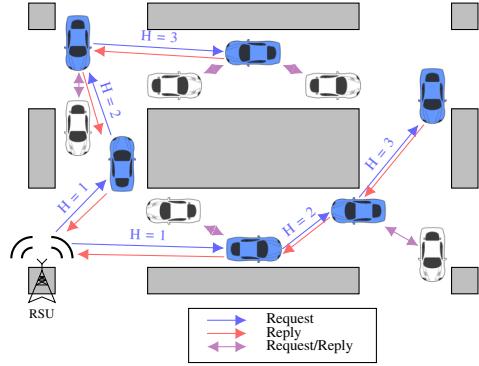


Figure 3. System perspective with $H_{\max} = 3$.

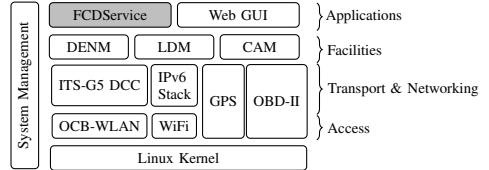


Figure 4. Integration of FCDService in the OpenC2X architecture [12].

will broadcast its own *Reply* upon receiving a *Request*, and only the relay nodes having a smaller hop count H will be allowed to forward the *Reply* messages. However, if a certain relay node overhears more than one *Reply*, it will forward each and every one individually. In case of DISCOVER, the intermediate relay nodes wait and merge the received *Reply* messages thanks to the timeouts defined in Equation (2). This allows them to send the collected information only once, reducing the packet header overhead.

III. IMPLEMENTATION

The described FCD collection protocols have been implemented in OpenC2X [12]—one of the most complete open-source experimental and prototyping platforms for vehicular networking solutions. OpenC2X supports most of the ETSI ITS-G5 [11] features—the European standard for short-range vehicular communications. Figure 4 presents the high-level architecture of OpenC2X in which we have integrated the FCD collection protocols. Our implementation follows the same modular approach as OpenC2X. In particular, we create a new dedicated module, named *FCDService*, which is located in the Applications layer of the protocol stack. The *FCDService* module interacts with the LDM module to get the local information to be attached to the *Reply* message. It also interacts with the Decentralized Congestion Control (DCC) module to send and receive packets from the lower layers. The source code of our implementation is open-source and can be downloaded¹.

From Section II, we can observe that two types of messages have to be defined in order to support the described FCD collection protocols:

- *Request*: message originated periodically by the source node to start the FCD collection process; these messages

¹<https://github.com/ion-turcanu/OpenC2X-FCDService.git>

F	C	D	vers=1	reserved	type=1
Src ID			Message ID		
D_{\max}	H_{\max}	H	T_{req}^{\max}	T_{rep}^{\max}	
Latitude			Longitude		
Longitude	...optional padding				

Figure 5. *Request* message structure.

F	C	D	vers=1	reserved	type=2
Src ID			Message ID		
S_{FCD} Payload					

Figure 6. *Reply* message structure.

create a backbone network by triggering the selection of relay nodes.

- *Reply*: message sent by vehicles/relay nodes; these messages contain FCD information.

We implemented these two message types in OpenC2X. The structure of the *Request* message is shown in Figure 5. In particular, the first three bytes identify the FCD message type from all other message types (e.g., CAM), the forth byte represents the version of the messages, the fifth is a reserved byte, and the sixth is used to differentiate between *Request* and *Reply* messages. Other important fields are *SrcID*, the ID of the sending vehicle, and *MessageID*, a unique message identifier (i.e., a counter). D_{\max} , H_{\max} , H , T_{req}^{\max} , and T_{rep}^{\max} have been previously described. A length field is not necessary, since the packet header is static. Finally, the *Latitude* and *Longitude* fields contain the geographical position of the vehicle.

Figure 6 illustrates the structure of the *Reply* message. The first fields are identical to the ones of the *Request* message. However, this message has a variable-length field, named S_{FCD} , to include the aggregated FCDs. Notice that in case of BASELINE, S_{FCD} contains only the FCD of the vehicle generating it.

IV. PERFORMANCE EVALUATION

A. Experimental Setup

To validate our implementation, we evaluated DISCOVER and BASELINE in a real-world experimental setup. This setup contains five Alix2d3² single-board computers that play the role of On-Board Units (OBUs), where we integrated our software. Each device uses a WLM200NX Wireless Network Interface Card (WNIC) containing Atheros 9k chipsets. OpenC2X comes with kernel patches for this chipset that allows sending and receiving signals on the 5.9 GHz frequency band, which is specifically assigned for ITS applications in Europe. We installed GNU/Linux Ubuntu 16.04 with a patched 3.18 Kernel³ on the Alix boards. In addition, we used gpsd, a software

²<https://www.pcengines.ch/>

³<https://ctu-iig.github.io/802.11p-linux/>



Figure 7. Placement of the five Alix devices in our office environment. The red lines show the DSRC communication links between the devices.

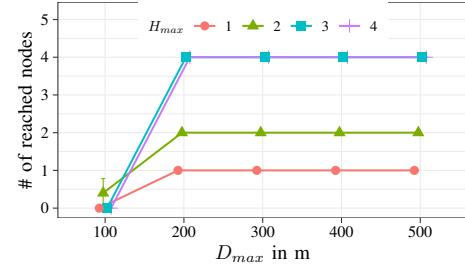


Figure 8. Mean nr. of reached nodes for different values of D_{\max} and H_{\max} .

module that communicates with a USB Global Positioning System (GPS) dongle and gets the coordinates.

The five Alix devices were placed in different offices on the university campus, as can be seen in Figure 7. The reason for placing these devices in different offices is to create a multi-hop communication scenario by artificially creating obstacles to avoid all devices being in each other's communication range. In particular, we set Alix 5 to be the source node periodically triggering the FCD collection. This is done by broadcasting a *Request* message every 5 s. Notice that the only node in Alix 5's communication range is Alix 3, so the only way Alix 5 can collect all the FCDs from the network is via multi-hop communication.

At the same time, since the algorithms need the geographical location of each device to compute the distance D between sending and receiving nodes, each device has been placed close to the window. This setup allows the USB GPS dongle to receive the actual latitude and longitude of the device. T_{req}^{\max} and T_{rep}^{\max} values have been set to 0.1 and 0.4 s respectively, while $\alpha = 0.8$.

B. Experimental Evaluation

The first thing we evaluate is the dissemination algorithm. To this end, we vary D_{\max} and H_{\max} parameters and count the number of nodes that receive the *Request* message. The results are shown in Figure 8. The values of D_{\max} represent typical communication ranges in vehicular networks and are consistent with [3]. Given the fact that the network is static (i.e., the nodes are not moving), there is a clear breakpoint between no nodes received the *Request*, and all nodes receiving it. In particular, this breakpoint is noticed when we increase D_{\max} from 100 m to 200 m. Basically, the real distance between Alix 5 and Alix 3 is bigger than 100 m and setting $D_{\max} = 100$ m is not enough

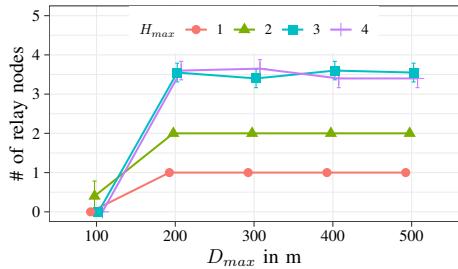


Figure 9. Mean nr. of relay nodes for different values of D_{\max} and H_{\max} .

to let the *Request* message propagate. However, we can notice that in very few cases the *Request* was able to propagate even with $D_{\max} = 100$ m. This can be explained by the fact that we are using real GPS coordinates, which may vary in time even if the nodes are not moving. This GPS error lead to some cases when the distance between Alix 5 and Alix 3 was less than 100 m.

For $D_{\max} \geq 200$ m the number of nodes receiving the *Request* is constant and depends mainly on H_{\max} . Remember that H_{\max} defines the maximum number of hops the *Request* is allowed to be retransmitted. This can be clearly seen in Figure 8, where we notice that the number of nodes receiving the *Request* increases when incrementing H_{\max} . For example, with $H_{\max} = 1$ only Alix 3 receives the *Request*, while with $H_{\max} = 2$ Alix 2 also receives the *Request* since Alix 3 acts as relay node. Finally, with $H_{\max} > 2$ all the nodes in the network receive the *Request*.

Figure 9 measures the number of relay nodes that are selected according to Equation (1). Given the network topology, for $H_{\max} \leq 2$ all the nodes receiving the *Request* become relay nodes. The only case when the inhibition mechanism can actually be validated is for $H_{\max} > 2$. In particular, when the *Request* is retransmitted by Alix 3, both Alix 4 and Alix 1 receive it. At this point, only one of the two should become relay node: the one having the highest distance D from Alix 2. For simplicity, let's denote D_{24} the distance between Alix 2 and Alix 4, and D_{21} the distance between Alix 2 and Alix 1. Also, let T_{req}^1 and T_{req}^4 be the calculated request timers of Alix 1 and Alix 4 respectively. Again, given the GPS error, D_{24} and D_{21} are not always constant, meaning that T_{req}^1 and T_{req}^4 are also changing. When the difference between these two timers is high enough, the node with the smallest T_{req} value broadcasts first the *Request*, and the other is inhibited. When the values of T_{req}^1 and T_{req}^4 are too close, then both devices broadcast their *Requests* before eavesdropping each other's retransmission. This is the main reason why on average we have 3.5 relay nodes instead of 3.

To validate the implemented FCD collection algorithms, we measure the number of collected FCDs by the source node (Alix 5). The results are displayed in Figure 10, where we compare DISCOVER and BASELINE for different values of H_{\max} . The first thing we observe is that DISCOVER outperforms BASELINE in terms of collected FCDs when $H_{\max} \leq 2$. This result is explained by the fact that DISCOVER exploits the CAM exchange process and the information inside

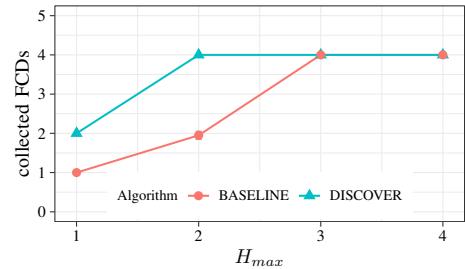


Figure 10. Mean nr. of collected FCDs for different values of H_{\max} .

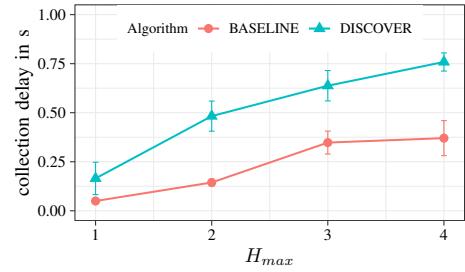


Figure 11. Mean collection delay for different values of H_{\max} .

LDMs. For example, when $H_{\max} = 1$ the *Request* is received by Alix 3 only, which means Alix 3 is the only device sending its *Reply*. However, with BASELINE Alix 3 will include only its own FCD in the *Reply* message, while with DISCOVER it will include all the FCDs in its LDM, i.e. the FCDs of Alix 2 and Alix 5. For this reason, with $H_{\max} = 1$ Alix 5 is able to collect only one new FCD if using BASELINE, while it collects two new FCDs when using DISCOVER. This is the same reason why DISCOVER needs only two hops to collect all the FCDs in the network given our topology setup. The second and most important result is that Figure 10 actually validates our implementation of the two considered FCD collection algorithms. In particular, the source node (Alix 5) is able to collect FCD information from devices outside of its direct communication range via multi-hop DSRC-based communications. If we assume Alix 5 being an RSU and other devices being vehicles driving on real roads, then the RSU would be able to periodically collect FCDs and monitor all the vehicles in a certain area around itself larger than its direct communication range.

Figure 11 illustrates the amount of time needed to complete one collection process (i.e., one cycle of sending a *Request* and receiving the *Reply* messages). Notice that DISCOVER is less efficient than BASELINE in terms of collection delay. In particular, we can see that in the worst case scenario ($H_{\max} = 4$) DISCOVER needs 0.75 s on average to complete one collection cycle, while BASELINE needs only 0.375 s. The additional delay in DISCOVER is caused by the reply timers computed by every relay node (see Equation (2)) and that are needed to enable the aggregation mechanism. Notice also that the overall collection delay increases when incrementing H_{\max} , which is consistent with the fact that more time is necessary to collect FCDs over a larger number of hops.

Finally, in Figure 12 we illustrate the total number of bytes received by every node for each FCD collection experiment.

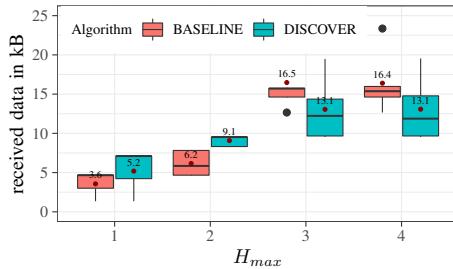


Figure 12. Amount of received data per node for different values of H_{max} .

In particular, every node summarizes the total number of bytes coming from other nodes' FCDService (i.e., *Request* and *Reply* messages). This is a measure of the additional load on the DSRC communication channel generated by each FCD collection protocol. We can notice that for $H_{max} \leq 2$ BASELINE generates less bytes than DISCOVER, but this is mainly due to the lower number of FCDs collected with BASELINE. When the number of collected FCDs is the same, we can see that DISCOVER is more efficient in terms of generated bytes. This is because with DISCOVER the relay nodes merge the *Reply* messages received from other relay nodes in one single *Reply* message, hence generating less packet header overhead.

V. CONCLUSION

Real-time vehicular FCD information represents an important enabler of a potentially large number of new ITS services and applications. Periodically collecting FCD from vehicles is a challenging task, which attracted many research efforts in recent years. Existing FCD collection protocols in the literature have been mainly evaluated using computer simulation models. However, before being deployed in cars, these protocols have to be properly validated via real-world experiments.

In this paper, we take a first step towards experimentally evaluating state-of-the-art FCD collection protocols. In particular, we considered DISCOVER, a multi-hop DSRC-based FCD collection protocol for VANETs that we initially described in [3], and implemented it in hardware using OpenC2X, an open-source vehicular experimentation platform. We designed an experimental setup and validated our implementation of DISCOVER against a basic FCD collection protocol, named BASELINE. Our results confirm the feasibility of collecting FCD information via multi-hop communications in DSRC-based VANETs. They also confirm the efficiency of DISCOVER with respect to BASELINE in terms of number of collected FCDs and generated packet header overhead. In addition, we show that FCD collection protocols can be integrated into the ETSI ITS-G5 protocol stack. We make our implementation open-source and available for download.

The main limitation of the proposed evaluation is the static network topology. For this reason, as future work we plan to evaluate these protocols using mobile nodes (i.e., vehicles). In addition, for a more realistic evaluation, we will artificially generate background cross-interfering data traffic on the same communication channel to evaluate the protocols under congested channel conditions.

ACKNOWLEDGEMENT

This work has been supported by the EU H2020 RAWFIE-OC3-SCI, under the EC Future Internet Research Experimentation (FIRE+) initiative.

REFERENCES

- [1] B. Pourghbleh and N. Jafari Navimipour, "Towards efficient data collection mechanisms in the vehicular ad hoc networks," *International Journal of Communication Systems*, e3893, 2019.
- [2] I. Turcanu, P. Salvo, A. Baiocchi, and 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.
- [3] ———, "An integrated VANET-based data dissemination and collection protocol for complex urban scenarios," *Elsevier Ad Hoc Networks*, vol. 52, no. Supplement C, pp. 28–38, 2016, Modeling and Performance Evaluation of Wireless Ad Hoc Networks.
- [4] Z. He and D. Zhang, "Cost-efficient traffic-aware data collection protocol in VANET," *Ad Hoc Networks*, vol. 55, no. Supplement C, pp. 28–39, 2017, Self-organizing and Smart Protocols for Heterogeneous Ad hoc Networks.
- [5] A. Soua and H. Afifi, "Adaptive data collection protocol using reinforcement learning for VANETs," in *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul. 2013, pp. 1040–1045.
- [6] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for Vehicular Networking: A Survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.
- [7] C. Ide, B. Dusza, M. Putzke, and C. Wietfeld, "Channel sensitive transmission scheme for V2I-based Floating Car Data collection via LTE," in *IEEE International Conference on Communications (ICC 2012)*, Ottawa, Canada: IEEE, Jun. 2012, pp. 7151–7156.
- [8] I. Turcanu, F. Klingler, C. Sommer, A. Baiocchi, and F. Dressler, "Duplicate suppression for efficient floating car data collection in heterogeneous LTE-DSRC vehicular networks," *Computer Communications*, vol. 123, pp. 54–64, 2018.
- [9] I. Turcanu, C. Sommer, A. Baiocchi, and F. Dressler, "Pick the Right Guy: CQI-Based LTE Forwarder Selection in VANETs," in *8th IEEE Vehicular Networking Conference (VNC 2016)*, Columbus, OH: IEEE, Dec. 2016, pp. 98–105.
- [10] S. Ancona, R. Stanica, and M. Fiore, "Performance boundaries of massive Floating Car Data offloading," in *11th Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014)*, Obergurgl, Austria: IEEE, Apr. 2014, pp. 89–96.
- [11] ETSI, "Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band," ETSI, EN 302 663 V1.2.1, Jul. 2013.
- [12] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer, and F. Dressler, "OpenC2X - An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5," in *8th IEEE Vehicular Networking Conference (VNC 2016), Demo Session*, Columbus, OH: IEEE, Dec. 2016, pp. 152–153.
- [13] O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, and V. K. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Networks," in *3rd International Conference on Broadband Communications, Networks and Systems*, Oct. 2006, pp. 1–11.
- [14] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," *Vehicular Communications*, vol. 1, no. 4, pp. 214–225, 2014.
- [15] 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," ETSI, EN 302 636-4-1 v1.2.1, May 2014.
- [16] ———, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," ETSI, Tech. Rep. 302 637-2 V1.3.2, Nov. 2014.
- [17] ———, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)," ETSI, EN 302 895 V1.1.1, Sep. 2014.