Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection

Florian Hagenauer
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
hagenauer@ccs-labs.org

Christoph Sommer
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
sommer@ccs-labs.org

Takamasa Higuchi
Toyota InfoTechnology Center, Co., Ltd.
Tokyo, Japan
ta-higuchi@jp.toyota-itc.com

Onur Altintas
Toyota InfoTechnology Center, USA, Inc.
Mountain View, CA, USA
onur@us.toyota-itc.com

Falko Dressler
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
dressler@ccs-labs.org

ABSTRACT

Automakers have already started rolling out cars with communication capabilities. They become more than simple means of transportation – their mobility combined with networking lets them interconnect people and machines. Looking at the network part, the most important aspect is to make the underlying network scaleable. Mobile edge computing has been introduced in 4G networks to achieve such scalability using caches and processing capabilities at eNodeBs, i.e., at the edge of the network. We introduce the concept of vehicular micro clouds as virtual edge servers. Micro clouds are conceptually clusters of cars, which help aggregating collected data that is transferred to some backend. In contrast to previous work on dynamic clustering in vehicular networks, we investigate the construction of such micro clouds using a map based approach.

ACM Reference Format:
Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. 2017. Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection. In *Proceedings of CarSys'17, Snowbird, UT, USA, October 20, 2017,* 5 pages.
https://doi.org/10.1145/3131944.3133937

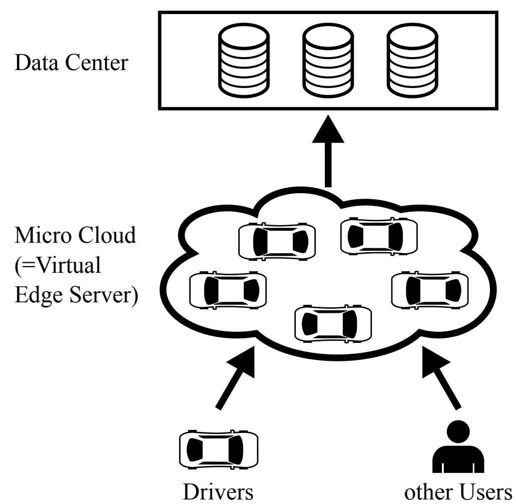


Figure 1: The system architecture for collecting data from the virtual edge servers connected to a data center.

1 INTRODUCTION

Cars equipped with vehicular networking capabilities are currently deployed all over the globe. Based on either the WLAN standard IEEE 802.11p or LTE/LTE-D2D, these networks not only allow Vehicle to Backend (V2B), but also Vehicle to Vehicle (V2V) communication. In the case of V2V, there is no need for a base station and the communication can be purely ad hoc. Both V2V and V2B are essential building blocks to enable cooperative driving in the near future, which in turn will completely transform transportation systems [9].

Being equipped with such networking capabilities, cars can be used for more than just transportation purposes. They become a gateway for interconnecting people and machines in future smart cities using Information and Communication Technology (ICT) [1, 5]. Furthermore, cars can be used to cooperatively collect data that can be used to provide services to drivers and non-drivers alike. Such services could range from simple driving route support to weather forecasts to multimedia content streaming. Especially for services that rely on larger amounts of data, new concepts and architectures need to be developed in order to cope with limited bandwidth and to enhance scalability. Such architectures are often hierarchical to scale better with the large amounts of cars.

CarSys'17, October 20, 2017, Snowbird, UT, USA

ACM ISBN 978-1-4503-5146-1/17/10…$15.00
https://doi.org/10.1145/3131944.3133937

In this paper, we establish the concept of vehicular micro clouds as virtual edge servers (cf. Figure 1) following the ideas developed in the context of mobile edge computing. To realize such virtual edge servers, we rely on clustering. Clustering in vehicular networks has been investigated before – using concepts from Mobile ad-hoc networks (MANETs). Unfortunately, such dynamic clustering solutions failed in general due to the very dynamic nature of Vehicular Ad Hoc Networks (VANETs) [3, 6]. Mostly used for safety related applications, clustering combines cars based on several parameters, e.g., position, direction, or connectivity. As a new concept, we propose the use of map information for the clustering process in order to have clusters (and the respective cluster heads) perfectly located at certain geographical features – in urban environments close to intersections. In brief, cars register with their locations at an Access Point (AP) or eNodeB, which, in turn, assigns cluster roles to the cars. We use the term Access Point to refer either to an IEEE 802.11p Roadside Unit (RSU) or an LTE cellular base station (eNodeB).

We use the vehicular micro cloud concept to optimize the processing and aggregation of data before sending it to a backend, e.g., a central data center. By doing this, the amount of data sent to the data center is reduced which increases the scalability. Data collection is performed using short-distance device-to-device communication technologies. Not only does processing the data at these virtual edge servers reduce the load on the wireless link between cars and the AP but also at the data center.

The contributions of our paper can be summarized as follows:

- We present the concept of vehicular micro clouds as virtual edge servers.
- We introduce a map based clustering algorithm to create such vehicular micro clouds.
- We evaluate the proposed system comparing the different aggregation rates at the vehicular micro cloud and different technologies for the backend communication.

2 RELATED WORK

Clustering and aggregation have been identified by many authors as key enablers for vehicular networks. By organizing the nodes in a hierarchical network and reducing the amount of transmitted data, both allow vehicular networks to scale better. Usually, clustering and aggregation are discussed separately from each other while implicitly most algorithms do both.

Cooper et al. [6] outline the basic scheme of a cluster creation algorithm as follows:

- *Gather control data:* To calculate the clusters, control information has to be exchanged. Such an exchange can be between potential Cluster Members (CMs), or between cars and the AP. Information included in these messages are the necessary parameters for cluster creation (e.g., positional data, direction, interest).
- *Select Cluster Heads (CHs):* Based on the received control data, one car per cluster is selected as CH. These CHs are usually the ones which collect data from the cluster and exchange it with other clusters or the AP. In a completely distributed approach, this step can be rather error prone, especially if all cars have to agree on the CH.
- *Distribute control information:* The information which car is a CH and which one a CM has to be sent to all cars. To do this, the broadcast nature of wireless networks can be exploited to reach as many cars as possible. If the cluster consists of cars which are farther away than a single hop, forwarding algorithms have to be used to make sure all cars have the same view of the cluster. Otherwise, the cluster might become unstable and extra effort is needed for its correction.
- *Gather data:* This final step consists of CMs sending their data to the CH for further processing.

The most crucial step in the process is the cluster creation and especially the selection of the CH. A failure in this step (e.g., having multiple CHs or no CH at all) lets the whole process fail.

Clustering algorithms can be categorized in multiple ways, e.g., based on scenario, based on application, or based on coordination. The dominant scenario is a freeway where clustering algorithms are able to exploit the predictable movement patterns of driving cars [2, 4, 13, 14]. For clustering algorithms in urban scenarios, movement patterns become less predictable [16, 17]. These clustering algorithms try to solve this most of the time by supporting the clustering process with an RSU.

We further differentiate between application-specific clustering algorithms (e.g., the dissemination of safety messages [2, 10, 14, 16, 17] or vehicle tracking [13]) and application-agnostic generic algorithms (e.g., combining channel management and clustering [7]).

Regarding coordination, the usual focus of clustering algorithms is on completely distributed algorithms [3, 6, 10, 13]. Still, certain proposed algorithms rely on a centralized coordinator node [16–18]. As distributed algorithms in urban scenarios induce a larger overhead for achieving stable clusters, we also rely on a centralized coordinator node, e.g., an RSU or a cellular base station.

Data aggregation has been explored in depth by Dietzel et al. [8]. The authors formulated a generic model of aggregation:

- *Decision:* Decide what data to aggregate.
- *Fusion:* Actually aggregate the received data.
- *Dissemination:* Send the aggregated data.

One of the main takeaways is that aggregation is very important for vehicular networks.

For safety applications, Raya et al. [15] investigate aggregation in combination with clustering. The main objective is not reducing the size of the content, but rather reducing the overhead of cryptography by grouping cars based on location and then combining signatures and keys. Another example is by Ibrahim and Weigle [12] who propose a highway traffic information system. Their system is based on aggregation to allow information cover large distances even in dense road traffic conditions. In their scheme, they reduce the amount of data to transmit by sending only deltas and not absolute values. Taherkhani and Pierre [16] propose an aggregation algorithm which uses clustering to control data congestion at intersections. It is a good example of how complex handling clustering and aggregation in urban environments can get as the architecture consists of three separate units for detecting congestion, clustering messages, and controlling data congestion. An interesting detail is that not the cars themselves are clustered, but rather the transmitted messages.

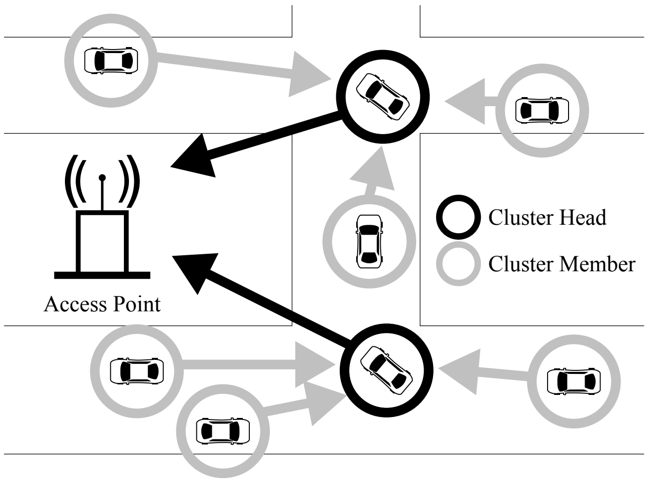


Figure 2: Overview of the map-based clustering algorithm with two clusters.

3 MICRO CLOUD ARCHITECTURE

Our focus in on the cars forming a micro cloud, which acts as a virtual edge server. As shown in Figure 1 these micro clouds consist of multiple cars forming a cluster. We assume these cars to be equipped with GPS and at least one short-range radio communication module. For heterogeneous vehicular networks, we also assume that cars are equipped with an additional cellular networking module. The clusters are coordinated by an AP. To enable communication between cars and the AP, the AP has to be equipped with the same networking technologies. Furthermore, the AP is assumed to be connected to the data center.

The role of a virtual edge server consists of several tasks: First, it acquires the data from surrounding cars. This includes all cars inside the cluster, but might potentially also be extended to surrounding cars. Second, the cluster processes the acquired data by aggregating it and in the process reducing its size. Third, the processed data is sent to an AP (and then transferred to the data center).

For actual clustering, we propose a *map-based scheme* that takes geographical features into account. The map is assumed to be available in form of street segments and intersections. The idea is to have Cluster Heads located at positions suited best for communication to (a) the AP and (b) to other cars, i.e., the Cluster Members. In a first step, in urban environments we selected intersections as such positions. This is done because intersections provide line-of-sight into multiple directions and therefore potentially provide better connectivity between a CH and its CMs.

The general clustering steps given in [6] can be nicely mapped onto our concept of virtual edge servers (shown in Figure 2):

(1) *Gather control data:* Control data required is information about the cars' geographical position as well as their ID. This control data is sent periodically by all cars to the AP.
(2) *Select Cluster Heads (CHs):* Based on this control information, the AP periodically (every CH interval) calculates optimal clusters. In our case, the car closest to an intersection is selected as an CH. For this, the AP creates and maintains a list of geographical coordinates stored in the AP. A single AP is able to create multiple clusters, i.e., vehicular micro clouds as virtual edge servers.
(3) *Distribute control information:* All other cars are made CMs and associated to the CH that is closest to their geographical position. The AP now distributes the cluster structure via broadcast to all cars. All cars are now aware to which virtual edge server they belong and to which CH they send their data.
(4) *Gather data:* All cars start gathering data and forward this data to their CH. Before a CH sends the collected data to the AP, it processes and aggregates the data. This is done to reduce the processing load of the data center as well as reduce the amount of data which has to be transmitted to the AP. For our proposed virtual edge servers, especially the aspect of reducing the amount of transmitted data is important.

4 EVALUATION

To better understand our clustering algorithm in an urban scenario, we evaluated the scenario shown in Figure 3. For this, we used the Veins LTE simulation framework [11], which provides an integration of the road traffic simulator SUMO with the network simulators Veins and SimuLTE. All simulation parameters are set to common baseline assumptions; they are summarized in Table 1. In particular, we want to highlight the use of an application-agnostic LTE scheduling algorithm.

Cluster structure: We first investigate the cluster structure. Figure 5 illustrates that the majority of cars are CH exactly for the duration of one single interval. This underlines the dynamicity of the network topology in the urban scenario. One example for a car being a CH longer than a single interval was a car waiting at a traffic light and being the car closest to the intersection for multiple intervals.

Performance with IEEE 802.11p-based Access Point: We next investigate the success rate of periodically collecting data from all cars. For this set of experiments all vehicles relied purely on short-range radio communication using IEEE 802.11p – referred to as Dedicated Short-Range Communication (DSRC) in the following – both for transferring data from CMs to CHs and from CHs to APs. We conduct this investigation for different aggregation rates

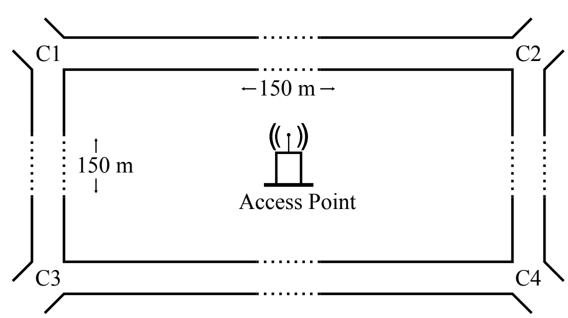
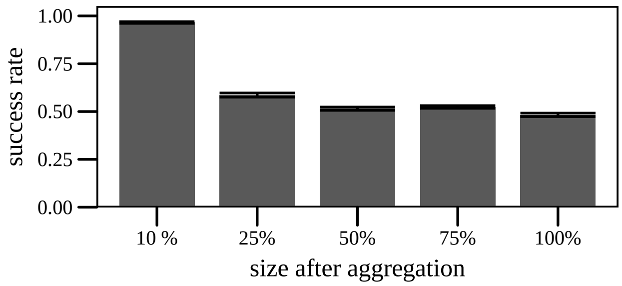
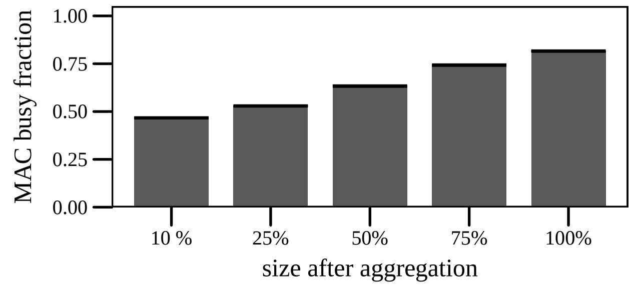


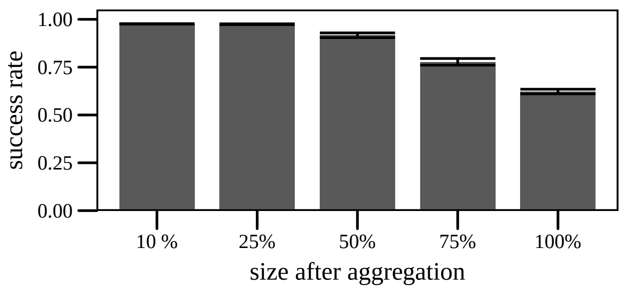
Figure 3: Simulation scenario showing four intersections (clusters C1–C4) and an access point.



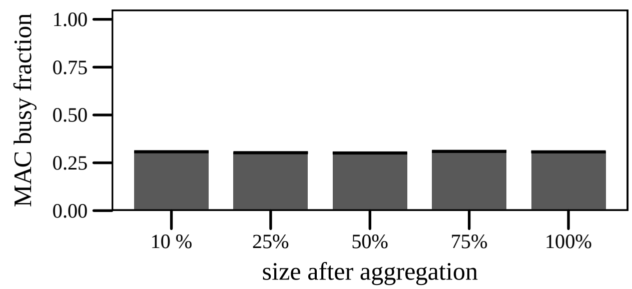
(a) Success rate of collecting 10 kB every 2 s from the Cluster Members (CMs) and aggregating the data before sending it to the Access Point (AP).



(b) MAC busy fraction of collecting 10 kB every 2 s from the Cluster Members (CMs) and aggregating the data before sending it to the Access Point (AP).



(c) Success rate of collecting 10 kB every 2 s from the Cluster Members (CMs) and aggregating the data before sending it to the Access Point (AP).



(d) MAC busy fraction of collecting 10 kB every 2 s from the Cluster Members (CMs) and aggregating the data before sending it to the Access Point (AP).

Figure 4: Performance with IEEE 802.11p-based Access Point (top row) and with LTE-based Access Point (eNodeB) (bottom)

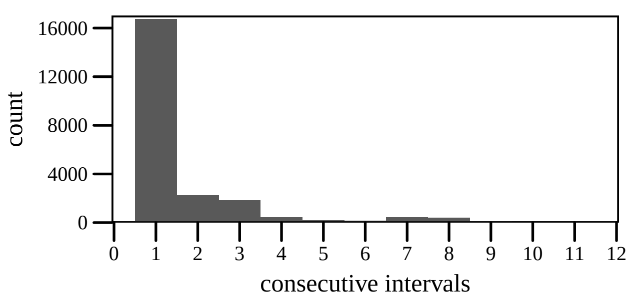


Figure 5: Distribution of the number of consecutive CH intervals for which a once-elected car remained CH.

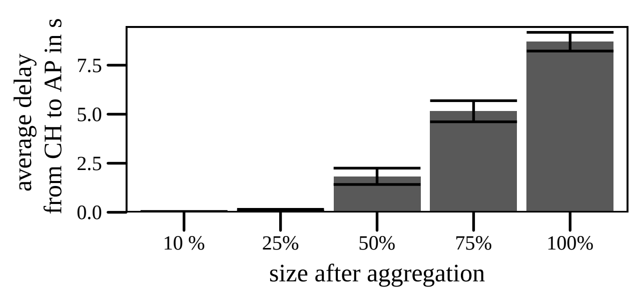


Figure 6: Average delay of transmitting the aggregated data from CH to Access Point (AP) via LTE.

of collected data at the CHs, expressed simply as a fraction of the size of data after aggregation in the range of 10 % to 100 %.

We observe that data collection works nearly all the time if the amount of periodically generated data is smaller than a few kilobytes. Nevertheless, even in such cases, reducing the load on the channel by performing aggregation might be useful because it frees resources for other purposes. Things change if we investigate the success rate at higher loads, such as periodically transmitting 10 kB every 2 s from all CMs. Figure 4a illustrates that the success rate is rather bad if no (or only a small amount of) aggregation is performed.

Figure 4b reveals the reason for this poor performance: With only DSRC technology used for all links and no aggregation being performed, the wireless channel is busy up to more than 75 %, which prevents the system from working properly. We can see that, by employing aggregation, the channel becomes less busy and the performance increases, i.e., it is possible to transmit the data gathered from the CMs.

Performance with LTE-based Access Point (eNodeB): To further reduce the load on the DSRC network and free up channel capacity for other applications (e.g., safety message dissemination) we now investigate a heterogeneous version of the system. For this, all V2V communication is done via DSRC while the V2B communication is happening via LTE. As can be seen in Figure 4c, this yields a much higher success rate. Nevertheless, there is a certain amount of unsuccessful transmissions even if we rely on LTE for V2B communication. As expected, the load on the DSRC channel could be observed to remain roughly constant, at 30 %, independent on the aggregation factor (cf. Figure 4d).

The underlying cause of the described effects can be tracked down to resource sharing in the LTE network. This is also observable in an increased delay when transmitting larger files as can be seen in Figure 6. This strongly indicates that it is important to focus very much on the last hop, i.e., the one between CH and AP. If this hop fails, far more data gets lost compared to a failing hop between a CM and its CH. Improvements can be done either by adapting the scheduling algorithm to this uneven distribution of priority or by adapting the upload process to work around a generic scheduling algorithm.

5 CONCLUSION

We propose the use of vehicular micro clouds as virtual edge servers for efficient connections between cars and backend infrastructure for future Intelligent Transportation System (ITS). In addition to the underlying micro cloud architecture, we investigate the use of map-based clustering techniques to cope with the dynamicity of vehicular networks. In our evaluations, we explored intersections as these positions to optimize the data flow between a Cluster Head (CH) and its Cluster Members (CMs). Initial simulation results indicate that the concept is sound and beneficial. Using aggregation functionality at the CH, we further demonstrated the need for such vehicular micro clouds.

Table 1: Simulation Parameters.

Parameter	Value
IVC technology	IEEE 802.11p
Channel	5.89 GHz
Transmission power	20 mW
Bandwidth	10 MHz
Cellular technology	LTE
Number of available RBs (Up- & Downlink)	15
LTE scheduler	MAXCI
UE transmission power	26 dBm
eNodeB transmission power	45 dBm
Investigated Area	150 m × 150 m
Average number of vehicles	60
CM to CH data	10 kB every 2 s
CH to AP upload interval	4.5 s
Simulation duration	120 s
Repetitions	200
Aggregation factors	0.1, 0.25, 0.5, 0.75, 1
Control information collection interval	1 s
CH interval	10 s

REFERENCES

[1] Onur Altintas, Falko Dressler, Florian Hagenauer, Makiko Matsumoto, Miguel Sepulcre, and Christoph Sommer. 2015. Making Cars a Main ICT Resource in Smart Cities. In *34th IEEE Conference on Computer Communications (INFOCOM 2015), International Workshop on Smart Cities and Urban Informatics (SmartCity 2015)*. IEEE, Hong Kong, China, 654–659. https://doi.org/10.1109/INFCOMW.2015.7179448

[2] Rachad Atat, Elias Yaacoub, Mohamed-Slim Alouini, and Fethi Filali. 2012. Delay efficient cooperation in public safety vehicular networks using LTE and IEEE 802.11p. In *9th Annual IEEE Consumer Communications and Networking Conference (CCNC 2012)*. IEEE, Las Vegas, NV, 316–320. https://doi.org/10.1109/CCNC.2012.6181109

[3] Rasmeet S Bali, Neeraj Kumar, and Joel J.P.C. Rodrigues. 2014. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. *Elsevier Vehicular Communications* 1, 3 (July 2014), 134–152. https://doi.org/10.1016/j.vehcom.2014.05.004

[4] Luciano Bononi and Marco Di Felice. 2007. A Cross Layered MAC and Clustering Scheme for Efficient Broadcast in VANETs. In *4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2007)*. Pisa, Italy. https://doi.org/10.1109/MOBHOC.2007.4428735

[5] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. 2011. Smart Cities in Europe. *Journal of Urban Technology* 18, 2 (2011), 65–82. https://doi.org/10.1080/10630732.2011.601117

[6] Craig Cooper, Daniel Franklin, Montserrat Ros, Farzad Safaei, and Mehran Abolhasan. 2017. A Comparative Survey of VANET Clustering Techniques. *IEEE Communications Surveys & Tutorials* 19, 1 (Feb. 2017), 657–681. https://doi.org/10.1109/COMST.2016.2611524

[7] Riccardo Crepaldi, Mehedi Bakht, and Robin Kravets. 2012. QuickSilver: Application-driven Inter- and Intra-cluster Communication in Vanets. In *Third ACM International Workshop on Mobile Opportunistic Networks (MobiOpp '12)*. ACM, Zurich, Switzerland, 69–76. https://doi.org/10.1145/2159576.2159591

[8] Stefan Dietzel, Jonathan Petit, Frank Kargl, and Björn Scheuermann. 2014. In-Network Aggregation for Vehicular Ad Hoc Networks. *IEEE Communications Surveys & Tutorials* 16, 4 (April 2014), 1909–1932. https://doi.org/10.1109/COMST.2014.2320091

[9] Falko Dressler, Hannes Hartenstein, Onur Altintas, and Ozan K. Tonguz. 2014. Inter-Vehicle Communication – Quo Vadis. *IEEE Communications Magazine* 52, 6 (June 2014), 170–177. https://doi.org/10.1109/MCOM.2014.6829960

[10] Efi Dror, Chen Avin, and Zvi Lotker. 2012. Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks. *Elsevier Ad Hoc Networks* 11, 7 (Sept. 2012), 2002–2015. https://doi.org/10.1016/j.adhoc.2012.02.006

[11] Florian Hagenauer, Falko Dressler, and Christoph Sommer. 2014. A Simulator for Heterogeneous Vehicular Networks. In *6th IEEE Vehicular Networking Conference (VNC 2014), Poster Session*. IEEE, Paderborn, Germany, 185–186. https://doi.org/10.1109/VNC.2014.7013339

[12] Khaled Ibrahim and Michele C. Weigle. 2008. CASCADE: Cluster-Based Accurate Syntactic Compression of Aggregated Data in VANETs. In *IEEE Global Telecommunications Conference (GLOBECOM 2008)*. New Orleans, LA, 1–10. https://doi.org/10.1109/GLOCOMW.2008.ECP.59

[13] Sanaz Khakpour, Richard W. Pazzi, and Khalil El-Khatib. 2017. Using clustering for target tracking in vehicular ad hoc networks. *Elsevier Vehicular Communications* 9 (July 2017), 83 – 96. https://doi.org/10.1016/j.vehcom.2017.02.002

[14] Aboobeker Sidhik Koyamparambil Mammu, Unai Hernandez-Jayo, and Nekane Sainz. 2013. Cluster-based MAC in VANETs for safety applications. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI 2013)*. IEEE, Mysore, India, 1424–1429. https://doi.org/10.1109/ICACCI.2013.6637388

[15] Maxim Raya, Adel Aziz, and Jean-Pierre Hubaux. 2006. Efficient Secure Aggregation in VANETs. In *3rd International Workshop on Vehicular Ad Hoc Networks (VANET '06)*. ACM, Los Angeles, CA, 67–75. https://doi.org/10.1145/1161064.1161076

[16] Nasrin Taherkhani and Ssamuel Pierre. 2016. Centralized and Localized Data Congestion Control Strategy for Vehicular Ad Hoc Networks Using a Machine Learning Clustering Algorithm. *IEEE Transactions on Intelligent Transportation Systems* 17, 11 (Nov. 2016), 3275–3285. https://doi.org/10.1109/TITS.2016.2546555

[17] Lung-Chih Tung, Jorge Mena, Mario Gerla, and Christoph Sommer. 2013. A Cluster Based Architecture for Intersection Collision Avoidance Using Heterogeneous Networks. In *12th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2013)*. IEEE, Ajaccio, Corsica, France. https://doi.org/10.1109/MedHocNet.2013.6767414

[18] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. 2016. Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination. *IEEE Transactions on Vehicular Technology* 65, 4 (April 2016), 2621–2636. https://doi.org/10.1109/TVT.2015.2421277