Virtual Edge Computing Using Vehicular Micro Clouds

Falko Dressler*, Gurjashan Singh Pannu*, Florian Hagenauer*, Mario Gerla†,
Takamasa Higuchi‡ and Onur Altintas‡
*Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
†Computer Science Department, University of California - Los Angeles, Los Angeles, CA
‡Toyota InfoTechnology Center U.S.A., Mountain View, CA
{dressler,pannu,hagenauer}@ccs-labs.org, gerla@cs.ucla.edu,
{ta-higuchi,onur}@us.toyota-itc.com

***Abstract*—We will discuss the challenges and opportunities of the connected cars vision in relation to the need for distributed data management solutions ranging from the vehicle to the mobile edge and to the data centers. As a novel concept, vehicular micro clouds have been proposed that bridge the gap between fully distributed vehicular networks based on short range vehicle to vehicle communication and cellular based infrastructure for centralized solutions. We will discuss the need for vehicular micro clouds, followed by the architecture, formation of micro clouds, and feasibility of micro clouds. Furthermore, we will cover aspects of efficient data upload and download between cars and a data center facilitated by our micro cloud concept.**

***Index Terms*—Vehicular Cloud, Mobile Edge Computing, Virtual Cloud Architecture, Vehicular Networking**

I. Introduction

Cars today are equipped with a rich set of computing, data storage, communication, and sensor resources in their on-board computer unit. Based on current vehicular networking standards, not only Vehicle to Infrastructure (V2I) but also Vehicle to Vehicle (V2V) communication are supported. Thus, it is believed that cars will play a major role in future Information and Communication Systems (ICT) systems for supporting applications like Intelligent Transportation Systems (ITS) and full-scale smart cities [1]–[3].

Recently, several groups of researchers independently proposed the concept of Vehicular Clouds (VCs), which brings the mobile cloud model (or the edge computing model [4]) to vehicular networks. Eltoweissy et al. [2] and Gerla [5] were among the first to propose using a VC as the backbone of ITS, smart cities, and smart electric power grids. Lee et al. [3] drafted important design principles of building VCs on top of Vehicular Ad Hoc Networks (VANETs) in combination with information-centric networking.

This concept has been refined by Dressler et al. [1] and Hagenauer et al. [6], [7]. They discussed how to provide cloud-like computation and networking service distributed among cars in both a parking lot as well as using geographic clustering techniques for moving vehicles. Independently, Florin et al. [8] and Arif et al. [9] also explored how to distribute MapReduce-like computation onto cars in a parking lot.

Conceptually, these works extend the Mobile Edge Computing (MEC) concept [4], [10], which is about bringing computing and storage capabilities from the cloud to the edge of the network, mainly to reduce latencies – MEC is now being standardized as a key component of upcoming 5G networks [11], [12]. Several advantages of VC can be identified: Vehicular Clouds allow diverse resources to be pooled and deployed dynamically to serve users with different needs (e.g., on-demand market for computation and communication service from nearby cars), they further enable autonomy in real-time service sharing and management with lower network latency (if a VC is deployed in close proximity), and VCs are typically decentralized and peer-to-peer, which avoids a single point of failures or market monopoly.

Despite all these efforts, there are still many challenges and open problems. Prior work has discussed issues ranging from networking [3] to engineering [13] to security [14]. In this paper, we focus on the issues in the intersection of VC and distributed computing, particularly on issues related to the dynamic and distributed nature of vehicles. To address some of the challenges, we introduce the new paradigm of a virtual cloud architecture. Building upon the Car4ICT concept proposed by Altintas et al. [15], which describes a V2V-based mechanism for providing services from individual cars, we define the concept of a hierarchical Macro-Micro-Cloud (MMC).

The main challenges in this context are the management of clusters of cars, i.e., the micro cloud, the data management within the micro cloud and the upload of context information to a backend data center, as well as the download of information from such backend and the dissemination to all interested cars in the geographically local vicinity. We present first solutions to these challenges, which we believe will pave the road for next generation virtual cloud processing using cars as a main ICT resource.

The rest of this paper is structured as follows: We first introduce our virtual cloud architecture in Section II. In Section III, we discuss how vehicular micro clouds, i.e., clusters of cars, are maintained both in a stationary case using parked cars and in a very dynamic environment using map-based clustering techniques. We also present concepts for the efficient data download and upload in Sections IV and V , respectively. We finally provide some conclusions in Section VI.

II. Virtual Cloud Architecture

A. Motivation

The core mechanism of existing vehicular clouds is as follows (cf., e.g., [3], [15]): (i) A user sends a service request message to a car passing by using V2V or V2I communication; (ii) the request is then forwarded over the communication network to discover a communication path to a vehicle offering the desired service; and (iii) if the user successfully finds a service provider, s/he uses this communication path to exchange service-related data thereafter. Although the mechanism works well for a variety of services as shown in prior works, it is non-trivial to adapt the mechanism for many other services, e.g., long-lasting services tied to a certain geographical area. Due to vehicle mobility, a user needs to find an alternative service provider whenever the current one leaves the neighborhood. This has the potential to cause huge communication overhead and poor quality of service.

For this reason, we developed a hierarchical approach, named Macro-Micro-Cloud (MMC). The main goal of MMC is to reduce the communication complexity and to improve the quality of service for long-lasting location-based services. Our design is based on the following observations:

- For location-based services, nearby cars can coordinate with each other to provide the service more efficiently (compared with an opportunistic approach in recent works such as [15]),
- the hierarchical structure of MMC allows to deploy edge services more flexibly and efficiently, and
- additional hand-over mechanisms are necessary for long-lasting services (to handle cases when there are no cars available in the neighborhood).

B. Proposed Architecture

Our virtual edge computing architecture is following a *hierarchical cloud structure* as depicted in Figure 1. Closely following the concepts of MEC, we add an additional layer between the backend data centers or cloud servers on the top and the individual users on the bottom of this figure. In MEC, this edge server layer is provided by compute and storage servers that are pre-deployed to 4G or 5G base stations, thus, requiring additional deployment and management costs. In contrast, we make use of available compute and storage resources in modern cars, which are also equipped with a variety of network capabilities for V2I and V2V communication. As there is no physical deployment necessary anymore, we call this layer now *virtual edge computing*, or the macro cloud. Such macro clouds can cover rather large areas, such as cities, because they are not necessarily directly connected by means of V2V communication and may also make use of Internet connections via the backend.

On the lower layer, cars form multiple local clusters, namely vehicular micro clouds. Such micro clouds behave as special nodes of the macro cloud. In micro clouds, cars entering a designated region (e.g., an intersection) dynamically join a micro cloud using a membership management protocol. The

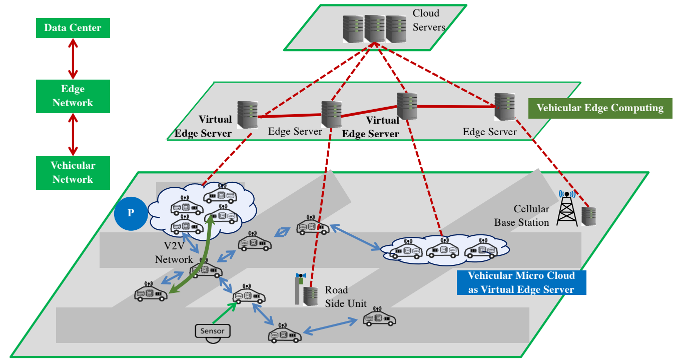

Figure 1. Elements of the vehicular cloudification architecture

micro cloud is logically fixed and *tied to that region*, and cars belonging to the same micro cloud collaborate with other micro cloud members to provide services to other users (e.g., monitoring, storage, or distributed computing). While individual cars may leave or join the designated region in a short period of time, the micro cloud can keep providing its services. To achieve this, other cars or nearby micro clouds in the region take over the tasks using a suitable hand-over mechanism.

In MMC, we use V2V communication to discover and provision services among micro clouds and individual cars and/or users that access the service using other devices like smart phones and smart watches. The main features of our MMC architecture are the following: The static nature of micro clouds is especially beneficial to long-lasting services, as a user does not need to repeatedly send service request messages as in [3], [15]; through the macro cloud, wide-area service discovery and provisioning becomes possible; and the hierarchical structure makes the deployment very flexible.

C. Research Questions

In order to enable the virtual edge computing concept using our MMC architecture, a number of research challenges need to be solved. The following list is by no means meant to be complete, but rather outlines some of the first steps towards a virtual edge cloud for ITS.

Cluster Management: Before we can use the micro clouds, these need to be formed and continuously maintained. This means that geographic regions need to be identified that will have to be covered by a micro cloud. Cars in this region may join this micro cloud and help providing services in this area.

Data management and upload to the backend: A core function of the virtual edge computing concept is to store and maintain data, i.e., providing caching capabilities for simple and most importantly fast access by other users. Thus, data replication for high reliability is required in combination with the possibility to upload data to the macro cloud in order to keep data consistent even if, temporarily, no (or too few) cars are available to form a micro cloud.

Content download and data dissemination: In order to provide more efficient data download from the backend, the micro cloud also aims at reducing the need for unnecessary direct downloads (individual concurrent 4G/5G connections reduce the download speed and add to the overall costs for the underlaying data plans). Having the micro cloud downloading content once (or with a certain, but minimal, degree of redundancy), and then distributing the data locally helps solving this problem.

In general, the cars' mobility is the key challenge in most of these steps. Furthermore, IT security questions need to be taken care of so that the MMC cannot be attacked or data be falsified. In the following, we go through selected challenges and discuss first solutions.

III. Vehicular Micro Cloud Management

For connecting users to the MMC and collecting data from sensors and users, we rely on micro clouds acting as virtual edge servers of the MMC. These virtual edge servers not only collect data from users but provide an access point to the macro cloud. In their role as edge servers, they may furthermore preprocess and aggregate data. To setup and maintain these micro clouds, we rely on clustering concepts.

As a general concept, clustering creates groups of nodes based on certain parameters (e.g., position, speed, moving direction, interest). Clustering potentially enables an architecture to better manage its nodes and in turn increases the scalability.

All solutions are based on two core components:

- Positioning: To make clustering work, we require every car to be able to determine its own geographical position, for example using GPS or by means of neighborhood information.
- Networking: To be able to interact with other cars as well as users, cars need a short-range communication technology (e.g., IEEE 802.11p, LTE C-V2X, Wi-Fi). For communication to the macro cloud, uplink communication is needed (e.g., LTE, Wi-Fi via an AP).

Cluster as a methodology has been explored in-depth in the scope of sensor networks and mobile ad-hoc networks. Yet, many of these solutions fail in the scope of vehicular networks due to the very high degree of mobility. Very recently, two alternative solutions have been proposed to overcome these problems, namely using clusters of parked cars and map-based clustering using dedicated geo-positions such as intersections as a reference.

A. Clusters of Parked Cars

After some early works on integrating parked cars as an infrastructure element in vehicular networks [16], [17], many follow-up studies helped completing this picture. Parked cars still provide all necessary communication capabilities and may contribute to the moving network as a reference point. Calculations have shown that the energy footprint is very low, if used just for a few days (after which cars usually move again and can re-charge the battery) [18].

In recent work, we proposed a Distributed Hash Table (DHT)-based clustering solution for parked cars [19]. This concept combines routing capabilities with distributed data management. Conceptually, such DHT-based routing and data management solutions have been developed in the scope of ad hoc networks for deeply integrated efficient routing with low complexity DHT integration. The resulting protocols such as Virtual Ring Routing (VRR) [20] and Virtual Cord Protocol (VCP) [21] can easily be re-used in other situations such as our parked car clustering. These protoocols guarantee reachability of all nodes and provide a DHT capable of storing and retrieving data using its hash code.

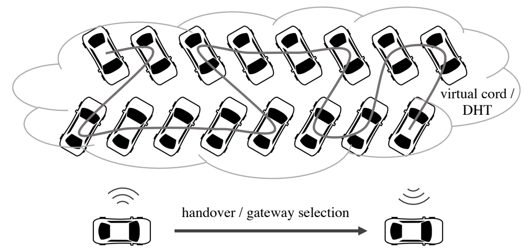

Figure 2. Cluster of parked cars with gateways and necessary handover

The process of joining or creating a cluster starts when a car is being parked and locked. Afterwards, the car starts listening for existing clusters. After a certain time, the car detects clusters in the surroundings and is able to join one. If no clusters are found, the car can start a new one.

We assume that the underlying scheme is able to handle nodes leaving (e.g., indicated by a person boarding the car). If that happens, data stored locally can be moved to other cluster members in the DHT. Note that leaving events do happen on a timescale in the order of tens of seconds to minutes, which allows to re-organize the cluster without too much impact on the system.

Based on the cluster formation, the cluster can now act as a *virtual road side unit* [6]. Users connecting to the cluster can connect to any of the Cluster Members (CMs) and continue transfering data while moving along the cluster (cf. Figure 2). To reduce network load, not all cars need to be able to communicate with the user of the system, such as the moving car. Exploiting the cluster structure, it is enough if a subset of cars, so called gateways, is reachable. We recently proposed an algorithm to select such gateways and were able to show in our evaluation that this selection does indeed reduce the load on the network while maintaining its performance [22]. We will discuss this in more detail in Section IV.

B. Map-Based Clusters of Moving Cars

In order to also cover moving cars and to provide micro cloud functionality, we developed a map-based clustering solution [7]. The initial concept was to use clusters of driving cars to collect data from outside users (e.g., other cars, pedestrians, IoT devices) and pre-process this data before sending it (via the macro cloud) to the backend data center. In this role, these clusters act as virtual edge servers and reduce the load on the cloud as well as increase the overall scalability of the system.

The map-based approach helps overcoming problems of other clustering solutions due to the dynamic topology of vehicular networks. A map is used to cover suitable spots to maintain geo-information – we identified intersections as such spots. All

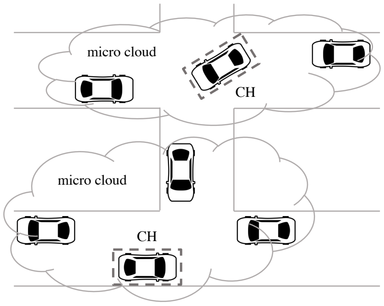

Figure 3. Cluster of moving cars with dedicated cluster heads close to the center of an intersection

clusters are created and maintained by a central entity (e.g., an access point or a cellular base station – we use the term access point to refer to both of them). This way, perfect mapping of cars to clusters can be realized, also including (approximate) information about the planned mobility pattern of the cars.

The concept is depicted in Figure 3. First, one car is selected as a Cluster Head (CH), which is in charge for all maintenance and operational tasks of the micro cloud. All other cars become CMs. The process is initiated by all cars sending beacons about their position (and partial mobility information) to the access point, which, in turn, can identify the car closest to the intersection and all other cars in proximity of it. The outline of the algorithm to do this is as follows:

1) Gather control data: All cars periodically send their current position to the AP.
2) Select CHs: Periodically, the AP calculates the new clusters. For each intersection, the car closest to the intersection is selected as a CH. All other cars are now associated as CMs to their closest CH.
3) Distribute control information: All cars are informed by the AP of their role and their associated CH.

IV. Data Management and Upload

The first application domain for our MMC approach is to maintain data at a certain geo-location to provide continuous access for other users in the area without the need of forwarding all data to a backend data center and having all users individually downloading. The micro cloud, thus, performs one of the classical MEC functions, namely local caching.

Data to be stored in the micro cloud can be traffic control information (e.g., Floating Car Data (FCD)), multimedia data (e.g., images like Google StreetView), or other data related to the location of the cluster. Assuming the parked cars cluster concept, which also provides DHT capabilities, this is a straightforward internal functionality.

Using the stationary micro cloud by moving cars maintained by map-based clustering, however, this needs additional extensions. The micro cloud has to distribute all data to achieve a sufficient degree of redundancy, so that leaving cars do not reduce data availability. The idea is, thus, to distribute the data among all CMs so that the data remains in the cluster with a high probability even though cars are coming and going, i.e., joining and leaving the micro cloud.

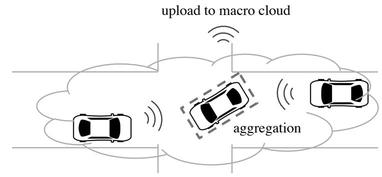

Figure 4. Data aggregation and information to the backend data center via selected cars in the micro cloud

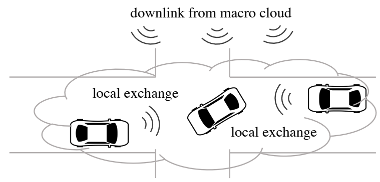

Figure 5. Cooperative download by three vehicles in the micro cloud

Our algorithm as presented in [7], already provides some minimal functionality here. All cars periodically report locally stored data to the access point (e.g., by means of its hash value). The access point, in return, can calculate which car is missing which data segment, and send a list to each car with the results. Cars can now request missing fragments from any of the cars providing them. Optimization potential lies in the network load vs. data reliability, as mechanisms such as overhearing, reduced replication rates in high vehicle density scenarios, and store-carry-forward concepts can be exploited.

Furthermore, the micro cloud can facilitate the upload process of all local data to the macro cloud and to the backend data center as depicted in Figure 4. Here, one or multiple dedicated upload points need to be identified. These will be cars that have direct access to an access point (or that provide LTE-based Internet uplinks). Cars now use these selected uplink points to forward their local data (such as traffic information) to a backend data center. Before uploading to the backend, aggregation techniques can be applied as many cars will likely report the same information at a similar time. Thus, the amount of data to be forwarded can be substantially be reduced [7].

V. Download and Data Dissemination

The second major application of our MMC concept is to facilitate the download process from the backend data center and local data dissemination of the received results. This way, cooperative download functionality can be provided by the micro cloud.

Again, following the idea of MEC, a gigantic distributed cache is established in the macro cloud, realized by the many associated micro clouds. Our goal is to minimize the number of requests to the cellular network required by the vehicular cluster to retrieve the segments of a file, and at the same time reduce the cellular resources utilized by vehicles interested in the same content. Making use of the data replication ideas discussed in Section IV, such download can be realized using both a parked cars cluster as well as a map-based cluster.

The principles are shown in Figure 5. A file is segmented from the backend data center in a distributed manner. A, B, and C represent different segments of a file. In the best case, vehicles request unique data segments from the cloud and later share their data segments via local communication. This way, vehicles can cooperatively download while reducing the cellular resources used. The challenge is to manage the selection of data segments in a distributed manner.

We argue that each vehicle should select which data segment to request from the server in a fully distributed manner. As a novel concept, we use an efficient approach of requesting data segments by making use of consistent hashing within a cluster so that vehicles can determine which data segments to request. The micro cloud can coordinate on this by following a three-phase model: (1) An interest phase is facilitated by the micro cloud, (2) all segment downloads are done individually during the download phase, and finally, (3) in the distribution phase, all required data segments are sent to interested cars.

Further optimization is possible based on the integration of adequate network coding concepts. Such ideas have been explored first in the scope of ad hoc and opportunistic networks [23]. Given the high degree of mobility and the limited continuous connectivity to a central server, it makes sense to use coding concepts to overcome connectivity holes and to improve the reliability of the data transfer as a whole. In the scope of VANETs, network coding has been already been suggested as a tool [24] and there are promising new application domains. In particular, we are planning to integrate network coding into the download as well as the dissemination process to reduce possible retransmissions (or even worse, duplicate transmissions) while still substantially improving the communication reliability with every successfully transmitted (now coded) data segment.

VI. Conclusions

We discussed the need for mobile edge support for and by vehicles on the road as a general ICT infrastructure. Our virtual edge architecture using the novel Macro-Micro-Cloud (MMC) concept allows to hierarchically integrate computing and storage resources from cars into such edge computing system. Instead of having to pre-deploy edge support at LTE base stations, we make use of available resources in cars, which can make these available to others in a crowd-sourced approach. The result is a fully distributed system, which uses clusters of cars – now called micro clouds – on the lowest layer, which themselves integrate to macro clouds covering entire cities. The macro cloud, in turn, provides access to backend data centers. Research challenges include set up and maintenance of clusters, data management within these clusters and the upload of such data to the backend data centers, as well as download and data dissemination from the backend.

References

[1] F. Dressler, P. Handle, and C. Sommer, "Towards a Vehicular Cloud - Using Parked Vehicles as a Temporary Network and Storage Infrastructure," in *ACM MobiHoc 2014, WiMobCity Workshop*, Philadelphia, PA: ACM, 2014, pp. 11–18.

[2] M. Eltoweissy, S. Olariu, and M. Younis, "Towards Autonomous Vehicular Clouds," in *ADHOCNETS 2010*, J. Zheng, D. Simplot-Ryl, and V. C. M. Leung, Eds., Victoria, BC, Canada: Springer, 2010, 1–16.

[3] E. Lee, E.-K. Lee, M. Gerla, and S. Oh, "Vehicular Cloud Networking: Architecture and Design Principles," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 148–155, 2014.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] M. Gerla, "Vehicular Cloud Computing," in *IFIP/IEEE Med-Hoc-Net 2012*, Ayia Napa, Cyprus: IEEE, 2012, pp. 152–155.

[6] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Parked Cars as Virtual Network Infrastructure: Enabling Stable V2I Access for Long-Lasting Data Flows," in *ACM MobiCom 2017, CarSys Workshop*, Snowbird, UT: ACM, 2017, pp. 57–64.

[7] ——, "Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection," in *ACM MobiCom 2017, CarSys Workshop*, Snowbird, UT: ACM, 2017, pp. 31–35.

[8] R. Florin, P. Ghazizadeh, A. G. Zadeh, S. El-Tawab, and S. Olariu, "Reasoning About Job Completion Time in Vehicular Clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1762–1771, 2017.

[9] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the Airport: Reasoning about Time-Dependent Parking Lot Occupancy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2067–2080, 2012.

[10] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[11] ETSI, "Mobile Edge Computing (MEC), Framework and Reference Architecture," ETSI, Tech. Rep. MEC 003 V1.1.1, 2016.

[12] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing - A key technology towards 5G," ETSI, ETSI White Paper No. 11, 2015. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf.

[13] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[14] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, "Security challenges in vehicular cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 284–294, 2013.

[15] O. Altintas, F. Dressler, F. Hagenauer, M. Matsumoto, M. Sepulcre, and C. Sommer, "Making Cars a Main ICT Resource in Smart Cities," in *IEEE INFOCOM 2015, SmartCity Workshop*, Hong Kong, China: IEEE, 2015, pp. 654–659.

[16] D. Eckhoff, C. Sommer, R. German, and F. Dressler, "Cooperative Awareness At Low Vehicle Densities: How Parked Cars Can Help See Through Buildings," in *IEEE GLOBECOM 2011*, Houston, TX: IEEE, 2011.

[17] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," in *IEEE INFOCOM 2011, Mini-Conference*, Shanghai, China: IEEE, 2011, pp. 431–435.

[18] C. Sommer, D. Eckhoff, and F. Dressler, "IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1733–1745, 2014.

[19] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Using Clusters of Parked Cars as Virtual Vehicular Network Infrastructure," in *IEEE VNC 2016, Poster Session*, Columbus, OH: IEEE, 2016, pp. 126–127.

[20] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual Ring Routing: Network routing inspired by DHTs," in *ACM SIGCOMM 2006*, Pisa, Italy: ACM, 2006.

[21] A. Awad, R. German, and F. Dressler, "Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1214–1226, 2011.

[22] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular Micro Cloud in Action: On Gateway Selection and Gateway Handovers," *Elsevier Ad Hoc Networks*, vol. 78, pp. 73–83, 2018.

[23] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[24] M. Gerla, C. Wu, G. Pau, and X. Zhu, "Content distribution in VANETs," *Elsevier Vehicular Communications*, vol. 1, no. 1, pp. 3–12, 2014.