Cooperative Downloading in Vehicular Heterogeneous Networks at the Edge

Takamasa Higuchi*, R. Vince Rabsatt†, Mario Gerla†, Onur Altintas* and Falko Dressler‡
*InfoTech Labs, Toyota Motor North America R&D, CA, U.S.A.
†NetLab, University of California, Los Angeles, CA, U.S.A.
‡Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
{takamasa.higuchi, onur.altintas}@toyota.com, {rrabsatt, gerla}@cs.ucla.edu,
dressler@ccs-labs.org

***Abstract*—Connected and automated driving vehicles are expected to generate an increasing amount of data traffic, possibly overloading vehicle-to-network (V2N) communication infrastructure in the long run. In this paper, we investigate the potential of local collaboration among vehicles to mitigate the load on V2N (*e.g.*, cellular) communication networks. Vehicles in the vicinity use vehicle-to-vehicle (V2V) communications to form a group, called vehicular micro cloud, and each of them downloads a subset of data segments that comprise an original data content. The downloaded data segments are cached and shared with other group members by way of V2V networks. This enables the group of vehicles to collectively serve as a virtual content delivery server, which complements cloud / edge computing infrastructure. In order to maximize the benefit of cooperation, we design a lightweight local coordination mechanism for vehicles to agree on non-overlapping subsets of data segments that they request from a remote server. Our simulation results show that coordination among vehicles improves the efficiency of cooperative download, reducing the data traffic on cellular networks.**

I. INTRODUCTION

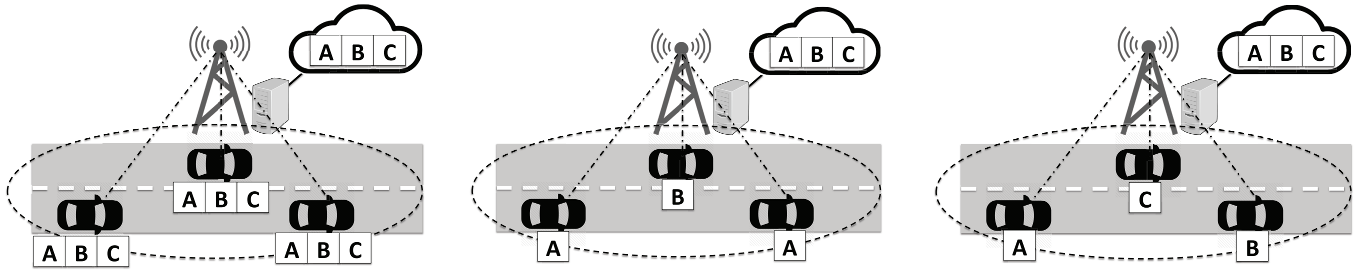
The demand on cellular network resources has tremendously increased over the recent years due to the growing popularity of data-hungry mobile applications (*e.g.*, video streaming). The emergence of connected and automated driving vehicles is expected to amplify the network load even further, as they often necessitate a large amount of data transfer between vehicles and remote cloud / edge computing infrastructure to collect, maintain and/or distribute up-to-date information about vehicles and the surrounding road environment [1]. Although the 5G mobile networks hold promise to achieve orders of magnitude higher network bandwidth, it would still take time until such new network infrastructure becomes prevalent.

In order to cope with the scalability challenge, some recent studies have pursued the possibility of composing *virtual edge servers* by aggregating on-board resources of a collection of connected vehicles [2]. Vehicles in the vicinity use V2V communications to form a group, called *vehicular micro cloud*, and collaborate with each other to perform resource-intensive computation, data storage, sensing and communication tasks. Hattab et al. [3] proposed a mechanism to optimally assign computational tasks to vehicles within a vehicular micro cloud, while some other work has shown the feasibility of forming a virtual cache storage server by data storage resources of multiple vehicles [4], [5]. Another potential use case of vehicular micro clouds is cooperative download, where each member vehicle downloads a subset of data segments, and exchanges those segments over V2V networks so that interested vehicles can reconstruct an original data content. The idea is motivated by the fact that data contents in vehicular networks often have limited geographical scopes of relevance. A typical example is a local dynamic map which contains time-varying information about road environment (*e.g.* lane closure, road traffic, etc.). Cloud / edge servers may collect sensor data from connected vehicles in each geographical area to update the corresponding regional map, and distribute it to vehicles heading to that region. In such cases, vehicles in the same geographical region are likely to request the same data content from the servers, bringing up the potential of collaboration.

The key challenges on cooperative download in a vehicular micro cloud are two-fold: (i) how to form a group of vehicles that facilitates efficient data download and (ii) how to coordinate the group members so as to reduce redundant data download via V2N communications while keeping the channel resource overhead for control messages as low as possible. In this paper, we design a distributed protocol, called *Cooperative Downloading Vehicular Cloud (CDVC)*, which addresses these research questions. In the CDVC, vehicles that have similarity in their travel paths and common interests in data contents form a vehicular micro cloud on an on-demand basis. The member vehicles collaboratively maintain a distributed hash table (DHT), which keeps track of their interests in data segments. We employ the DHT as a lightweight coordination mechanism for the members to agree on non-overlapping subsets of data segments that they are responsible for downloading on behalf of other members. The simulation results show that our V2V coordination mechanism improves the efficiency of cooperative download, reducing the data traffic on cellular networks.

II. RELATED WORK

A possible approach to reducing the load on cellular networks is to offload data traffic to other types of communication media such as IEEE 802.11p-based vehicle-to-infrastructure (V2I) networks. Ota et al. [6] proposed a so-called max-throughput and min-delay cooperative downloading (MMCD) algorithm, which prioritizes vehicles' data requests based on designated delivery deadline. Their goal is to minimize the average response time to data requests while offloading data



(a) LTE-only: each vehicle downloads all the data segments individually

(b) Random: each vehicle downloads a random subset of data segments

(c) CDVC (proposed): each vehicle downloads a non-overlapping subset of data segments

Figure 1. System architecture with three possible download policies

traffic from cellular base stations to roadside units (RSUs) as much as possible. Malandrino et al. [7] proposed an algorithm for RSUs to predict future paths of vehicles and prefetch data contents before those vehicles reach their communication range. They also considered using cars parked at the curbside as additional relays [8]. Shen et al. [9] proposed a scheduling algorithm to efficiently disseminate data contents through RSUs. Although extensive simulations and pilot studies have proved the potential of V2I networks, the availability of RSUs is still limited due to the cost for infrastructure deployment.

Peer-to-peer content sharing has been proved to be effective to improve the scalability of content delivery networks. BitTorrent is one of the most common peer-to-peer content sharing systems that enable a group of network nodes (*i.e.,* peers) to cooperatively download data contents over the Internet. A centralized server is required to maintain a list of trackers, which assist peers to locate contents and other peers in the network. Each peer attempts to download data segments from other peers that already have those segments. The original content can be reconstructed once they complete all the data segments [10]. Although it has been successfully applied to the Internet-based content sharing, it may not fit well in vehicular networks, where connectivity between network nodes frequently changes over time due to vehicle mobility.

Connected vehicles generate a huge amount of sensor data while traveling on a road. Part of the vehicle-generated data may be uploaded to remote cloud / edge servers for further analysis. Some recent work has explored the potential of V2V collaboration to facilitate efficient data collection. Hagenauer et al. [2] leveraged a vehicular micro cloud to aggregate sensor data from multiple vehicles in the vicinity, while Turcanu et al. [11] designed a mechanism to select the best vehicle to upload the aggregated data via a cellular base station. Both of these solutions focus on uplink data traffic from vehicles to remote servers. In this work, we extend the concept of vehicular micro clouds to downlink communications to efficiently distribute data contents of common interest from the servers to vehicles.

V2V communications have been proved to be a powerful means to assist distribution of data contents over vehicular networks. Trullols-Cruces et al. [12] employed a store-carry-forward mechanism to extend the coverage of RSUs. Each RSU predicts future paths of vehicles within its communication range, and selects a *carrier* vehicle that is likely to encounter a destination vehicle. The selected carrier vehicle receives that content from the RSU, and forwards it over V2V networks upon encountering the destination vehicle. Deng et al. [13] designed an incentive mechanism to motivate drivers to contribute to cooperative download. Nandan et al. [14] proposed the concept of tracker-less BitTorrent, which is tailored to cooperative download among vehicles. Vehicles that have downloaded data segments from RSUs cache them in on-board data storage. They periodically advertise the list of cached data segments over V2V networks while traveling a road, and forward them in reaction to data requests from neighboring vehicles. The existing solutions above either rely on network infrastructure (*e.g.*, RSUs) to coordinate data download strategies [12], [13], or do not perform any coordination among vehicles [14]. The CDVC introduces a fully-distributed coordination mechanism to avoid redundant data download without depending on infrastructure.

III. PROTOCOL DESIGN

A. Overview

We assume that each data content is equally divided into multiple data segments, which can be either stored at a remote back-end server or cached at vehicles' on-board data storage devices. Vehicles are equipped with a GPS receiver as well as V2N (*e.g.*, cellular) and V2V (*e.g.*, DSRC) communication modules. We also assume that each of the data segments is assigned a unique identifier, and vehicles know the identifiers of the data segments that constitute data content(s) relevant to their current geographical region.

Figure 1 shows the system architecture with three different download policies. In these examples, we assume that the back-end server divides an original data content into multiple data segments A, B and C. The three vehicles form a vehicular micro cloud and attempt to download this data content. Each vehicle can reconstruct the original data content only if they retrieve all of the data segments either from the back-end server or another vehicle in the same vehicular micro cloud. A straightforward approach is to have every vehicle download all the data segments individually, as illustrated in Figure 1a. Obviously, this approach incurs non-negligible load on cellular networks especially if vehicle density is high. A possible alternative is to have each vehicle download a random subset of data segments, and subsequently forward them to other members over V2V networks (see Figure 1b). Although it helps reduce the cellular data traffic, it cannot guarantee that every data segment is downloaded by at least one member. In this example, no vehicle can reconstruct the original data content even after V2V data forwarding because the segment C is missing. Furthermore, the same data segment still may be downloaded by multiple member vehicles, incurring redundant data traffic on cellular networks. Our CDVC protocol is designed to allocate non-overlapping subsets of data segments to each member vehicle as shown in Figure 1c. The members are responsible for caching the allocated data segments, and download them over cellular networks in the case that no member caches them in their data storage.

The CDVC protocol basically consists of three phases: *interest phase*, *download phase*, and *distribution phase*. The members in a vehicular micro cloud perform these phases every time slot with a constant time duration. We will describe details of each phase in the following sections.

B. Interest Phase

The CDVC introduces a distributed coordination mechanism for member vehicles to agree on the distinct sets of data segments. This is enabled by periodic broadcast of beacon messages over V2V networks. In order to save the channel access overhead, the control information is embedded into standard V2V messages (*e.g.*, Basic Safety Messages (BSMs) [15] in the U.S. or Cooperative Awareness Messages (CAMs) [16] in Europe) instead of sending it as separate data packets. At regular time intervals (*e.g.*, 1 second), the protocol appends three additional data fields to the standard BSM / CAM. Two of these data fields describe a bit vector encoding (i) a list of data segments that the vehicle is interested in and (ii) a list of data segments currently cached by the vehicle, respectively. Each bit position in the bit vector corresponds to a data segment that constitutes an original data content. A bit is set to 1 if the vehicle is interested in / caches the corresponding segment. The remaining data field contains identifiers of vehicles currently belonging to the vehicular micro cloud.

Each vehicle monitors extended BSMs / CAMs received from neighboring vehicles. Each of these messages contains the sender vehicle's position and orientation along with the additional data fields for coordination. If a vehicle finds a neighboring vehicle that (i) travels on the same road segment, (ii) is heading to the similar direction and (iii) has *common data interests*, it forms a new vehicular micro cloud by inviting that vehicle as a new member over V2V networks. We define that a pair of vehicles have common data interests if their lists of interested data segments share at least N common elements. A new vehicle may join a vehicular micro cloud in the vicinity if it receives extended BSMs / CAMs from all the existing members and the conditions (i) - (iii) above are met for all the members. Note that a vehicle may belong to multiple vehicular micro clouds at the same time. If a vehicle has not received any extended BSMs / CAMs from another member vehicle for a designated time period, it removes that vehicle from the list of members.

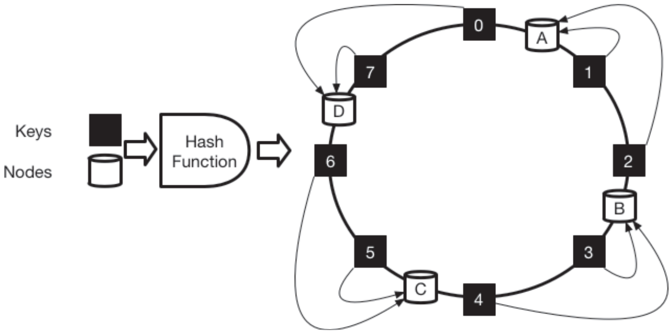


Figure 2. Example of DHT-based segment allocation

C. Download Phase

A vehicle that wants to download a certain data segment first checks the extended BSMs / CAMs recently received from other members, and checks if the segment of interest is already cached by any micro cloud member. If the vehicle finds a match, it requests that segment over V2V networks. Otherwise, one member in the vehicular micro cloud downloads the segment from a remote back-end server. The members employ *consistent hashing* to agree on non-overlapping sets of data segments to download, while keeping communication overhead for coordination as low as possible.

Consistent hashing [17] is a distributed hashing scheme that maps data elements and network nodes to virtual positions along a hash ring (*i.e.*, DHT) using a common hash function. In the CDVC, the network node corresponds to a member of a vehicular micro cloud, while a data element signifies a single data segment. Each member of a vehicular micro cloud maintains the DHT in a distributed manner based on the extended BSMs / CAMs received from other members. The hash function is applied to unique identifiers associated with the vehicles and data segments. When a new member joins a vehicular micro cloud, each member calculates a hash value of this vehicle's identifier, and registers it at the corresponding position along the DHT. Likewise, all the requested data segments that are not cached by any micro cloud member are also mapped to the DHT. For each of these missing segments, the first member in the counter-clockwise direction along the DHT will be responsible for downloading from a back-end server.

Figure 2 shows an example DHT with eight missing data segments and four member vehicles A, B, C, and D. The data segments 1 and 2 will be downloaded by vehicle A, because it is the first vehicle in the counter-clockwise direction with respect to the segments' virtual positions on the DHT. If vehicle A leaves the vehicular micro cloud, the remaining members delete this vehicle from the DHT. It triggers vehicle D to take over vehicle A's responsibility, and download the data segments 1 and 2 if it is not already cached in its data storage.

Table I
SIMULATION PARAMETERS

Parameter	Value
Simulation area	800 m × 800 m
Average vehicle density	12 veh/km/lane
Simulation time	300 s
DSRC transmission power	20 mW
Beacon frequency	1 Hz
LTE scheduler	MAXCI
eNodeB transmission power	45 dBm
UE transmission power	26 dBm
Number of RBs	100

D. Distribution Phase

Once a vehicle has downloaded the missing data segments from the back-end server, other members will be notified in subsequent extended BSMs / CAMs. This allows other members to request those data segments over V2V networks.

IV. EVALUATION

A. Simulation Setup

To evaluate the performance of our CDVC protocol, we conducted simulation experiments using Veins LTE [18]. We model an urban traffic scenario with a grid-shaped road network. Each road segment consists of two lanes, and the average vehicle speed is set to 40 km/h. Based on the well-known road traffic characteristic that the maximum lane capacity is reached at a density between 13-17 vehicles per kilometer per lane [19], we set the average vehicle density to 12 vehicles per kilometer per lane. All the simulations are repeated 10 times with an independent random seed for each run. We assume that the entire simulation field is covered by an LTE eNodeB, and all the vehicles are equipped with LTE and DSRC interfaces. Vehicles request a new data content every three seconds. Each data content has the size of 100 kbytes, and is equally divided into 20 data segments. Other simulation parameters are listed in Table I.

We compare the performance of CDVC (referred to as *Hashing*) with two baseline download policies. The *LTE-only* approach has every vehicle download all the data segments over cellular networks, while the *Random* download policy randomly selects a subset of data segments that each member of a vehicular micro cloud is responsible for downloading.

We investigate the following metrics to evaluate the performance:

- *Cellular network load*: the ratio of downlink resource blocks (RBs) utilized within an LTE cell
- *Average download time*: average time required for a vehicle to download a complete data content
- *Cellular download ratio*: the proportion of data segments that are downloaded via cellular networks, which indicates the efficiency of traffic offloading

B. Simulation Results

Figure 3 shows the cellular network load, which is indicated by the ratio of utilized LTE resource blocks. As expected, the *LTE-only* approach incurs a non-negligible amount of cellular data traffic, because multiple vehicles download the same data segments from a remote server. The *Random* policy significantly reduces the cellular network load by mitigating such redundant data requests from the server. The *Hashing*-based download policy can further save the cellular network resources by 75% as compared to the *Random* scheme by coordinating vehicles such that they download non-overlapping subsets of data segments.

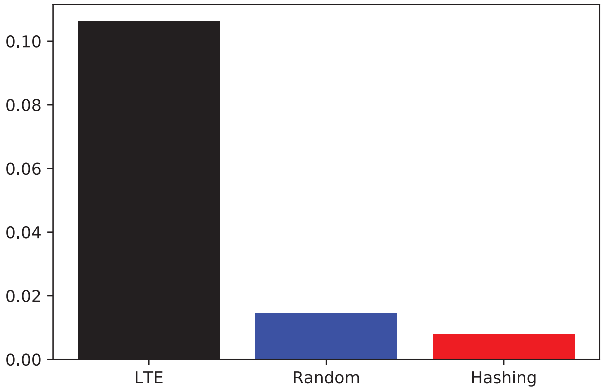


Figure 3. Cellular network load

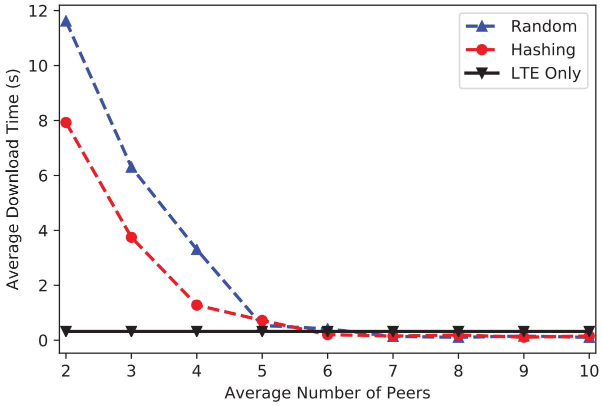


Figure 4. Average download time

Figure 4 shows the average download time with various numbers of micro cloud members. The results imply that the size of a vehicular micro cloud is an important factor that affects the efficiency of cooperative download. The download time of the *Random* and *Hashing*-based policies tends to be longer than that of the *LTE-only* scheme for small vehicular micro clouds, because of the limited likelihood of finding cached data segments. However, we can clearly observe that the download time significantly decreases as the number of micro cloud members increases. For micro clouds with more than five members, the download time can be even lower than the *LTE-only* scheme while saving the cellular network resources by more than 95%.

The similar trends can be also seen in Figure 5, which shows the cellular download ratio. While the significant ratio of data requests can be offloaded to V2V networks even with the smallest vehicular micro cloud consisting of two members,

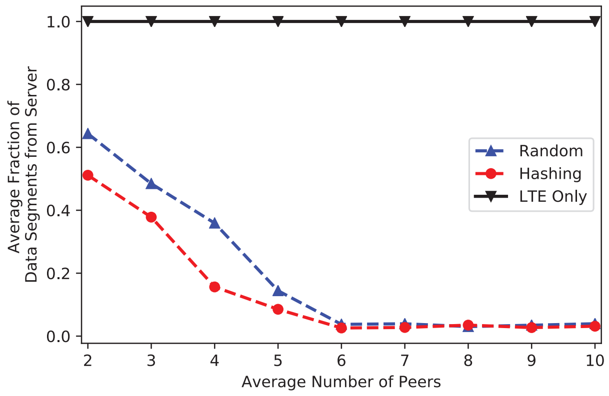


Figure 5. Cellular download ratio

much higher efficiency can be achieved as the size of vehicular micro clouds increases. In [20], the authors have analyzed realistic road traffic data in the city of Luxembourg and found that many regions across the urban road network consistently have more than 10 vehicles, especially during rush hours. This clearly indicates the potential that cooperative download in a vehicular micro cloud can significantly reduce the cellular network load.

V. CONCLUSION

In this paper, we have proposed a protocol for a group of connected vehicles to collaboratively download data contents of common interest. Our CDVC protocol forms a vehicular micro cloud among the vehicles with similar mobility characteristics and common data interests. The micro cloud members employ a distributed mechanism to maintain a DHT, which uniquely determines non-overlapping subsets of data segments that each member is responsible for downloading in the case that those segments are not cached within the micro cloud. The simulation results show that the DHT-based coordination mechanism can effectively reduce the redundant data requests from a remote back-end server, and thereby save the cellular network resources by up to 95%.

Our future work includes more detailed analysis on communication overhead associated with coordination among vehicles. The comparison with edge-assisted cooperative download where an edge server helps vehicles adjust their data download policies is also part of our next steps.

ACKNOWLEDGEMENTS

The core of the work was done when late Mario Gerla was with us. We honor his contributions by keeping his name in the list of co-authors.

REFERENCES

[1] “General Principle and Vision,” Automotive Edge Computing Consortium (AECC), White Paper 2.1.0, 12 2018. [Online]. Available: https://aecc.org/wp-content/uploads/2019/04/AECC_White_Paper_v2.1_003.pdf

[2] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, “Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection,” in *23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017), 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2017)*. Snowbird, UT: ACM, 10 2017, pp. 31–35.

[3] G. Hattab, S. Ucar, T. Higuchi, O. Altintas, F. Dressler, and D. Cabric, “Optimized Assignment of Computational Tasks in Vehicular Micro Clouds,” in *14th ACM European Conference on Computer Systems (EuroSys 2019), 2nd ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys 2019)*. Dresden, Germany: ACM, 3 2019, pp. 1–6.

[4] T. Higuchi, G. S. Pannu, F. Dressler, and O. Altintas, “Content Replication in Vehicular Micro Cloud-based Data Storage: A Mobility-Aware Approach,” in *10th IEEE Vehicular Networking Conference (VNC 2018)*. Taipei, Taiwan: IEEE, 12 2018.

[5] G. S. Pannu, F. Hagenauer, T. Higuchi, O. Altintas, and F. Dressler, “Keeping Data Alive: Communication Across Vehicular Micro Clouds,” in *20th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2019)*. Washington, D.C.: IEEE, 6 2019.

[6] K. Ota, M. Dong, S. Chang, and H. Zhu, “MMCD: Max-throughput and min-delay cooperative downloading for Drive-thru Internet systems,” in *IEEE International Conference on Communications (ICC 2014)*. Sydney, Australia: IEEE, 6 2014, pp. 83–87.

[7] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, “Offloading cellular networks through ITS content download,” in *9th IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2012)*, Seoul, South Korea, 6 2012, pp. 263–271.

[8] F. Malandrino, C. Casetti, C.-F. Chiasserini, C. Sommer, and F. Dressler, “The Role of Parked Cars in Content Downloading for Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4606–4617, 11 2014.

[9] X. Shen, X. Cheng, L. Yang, R. Zhang, and B. Jiao, “Data Dissemination in VANETs: A Scheduling Approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2213–2223, 10 2014.

[10] D. Qiu and R. Srikant, “Modeling and Performance Analysis of BitTorrent-like Peer-to-peer Networks,” in *ACM SIGCOMM 2004*. Portland, OR: ACM, 8 2004, pp. 367–378.

[11] 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, 12 2016, pp. 98–105.

[12] O. Trullols-Cruces, M. Fiore, and J. Barcelo-Ordinas, “Cooperative Download in Vehicular Environments,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 663–678, 4 2012.

[13] G. Deng, F. Li, and L. Wang, “Cooperative downloading in VANETs-LTE heterogeneous network based on Named Data,” in *35th IEEE Conference on Computer Communications (INFOCOM 2016), 3rd Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM 2016)*, San Francisco, CA, 4 2016, pp. 233–238.

[14] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, “Co-operative downloading in vehicular ad-hoc wireless networks,” in *2nd IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2005)*. St. Moritz, Switzerland: IEEE, 1 2005, pp. 32–41.

[15] “Dedicated Short Range Communications (DSRC) Message Set Dictionary,” SAE, Standard J2735-200911, 11 2009.

[16] ETSI, “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” ETSI, EN 302 637-2 V1.3.2, 11 2014.

[17] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, “Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web,” in *29th ACM Symposium on Theory of Computing*. El Paso, TX: ACM, 5 1997, pp. 654–663.

[18] F. Hagenauer, F. Dressler, and C. Sommer, “A Simulator for Heterogeneous Vehicular Networks,” in *6th IEEE Vehicular Networking Conference (VNC 2014), Poster Session*. Paderborn, Germany: IEEE, 12 2014, pp. 185–186.

[19] P. I. Richards, “Shock Waves on the Highway,” *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[20] T. Higuchi, J. Joy, F. Dressler, M. Gerla, and O. Altintas, “On the Feasibility of Vehicular Micro Clouds,” in *9th IEEE Vehicular Networking Conference (VNC 2017)*. Turin, Italy: IEEE, 11 2017, pp. 179–182.