Improving Data Consistency in Vehicular Micro Clouds

Gurjashan Singh Pannu*, Stephan Dunkel*, Seyhan Ucar†, Takamasa Higuchi†,
Onur Altintas† and Falko Dressler*
*School of Electrical Engineering and Computer Science, TU Berlin, Germany
†InfoTech Labs, Toyota Motor North America R&D, CA, U.S.A.
{pannu, stephan.dunkel, dressler}@ccs-labs.org,
{takamasa.higuchi, seyhan.ucar, onur.altintas}@toyota.com

***Abstract*—In the field of vehicular networks, multiple approaches have been proposed to share and reuse data acquired by participating hosts. In this context, vehicular micro clouds extend the concept of Mobile Edge Computing (MEC) and bring data storage and processing to the vehicles, solving application tasks that need to be done in real-time. A critical point in shared computing tasks and storage is to keep all nodes synchronized and to maintain consistency. For the first time, we study the relevance of data versions in micro clouds offering intersection management service at four-way stop intersections and we investigate how different versions of data affect both road traffic and wireless communications. Our results validate the intuition that an increase in the amount of different data versions in the micro cloud has negative effects on both road traffic and wireless channel usage. Yet, the choice of the data sharing algorithm can make a huge difference and reduce these effects on a large scale. We found that synchronicity of data versions can be increased by up to 20 % through small changes, e.g., keeping small amounts of data history in the applied algorithm.**

I. Introduction

The present day technologies for vehicle-to-vehicle (V2V) and vehicle-to-cloud (V2C) communication enable new horizon of applications for traffic management [1]. The concept of *vehicular micro cloud* extends the Mobile Edge Computing (MEC) paradigm by grouping cars into a small cluster [2] which virtually provides edge computing services on the road. Cars participating in such clusters share data among each other via direct V2V communication to solve different application tasks like intersection management, cooperative perception, distributed data collection and storage [3], [4].

While vehicular micro clouds can play a crucial role to provide edge services with less dependence on physical edge servers for efficient vehicular communications, they also impose several challenges that need to be addressed. In comparison to standard cloud operations, data must be processed and shared in real-time, since micro clouds are usually very dynamic networks. The participating cars remain only for a short time (lifetime) in the micro clouds, i.e., they join and leave the micro cloud often. Also, data in vehicular micro clouds often changes over time. The data can be synchronized among all micro cloud members with support from cloud infrastructure. However, communication with the cloud always comes with an additional delay. Thus, having perfectly synchronized data in a micro cloud is very challenging.

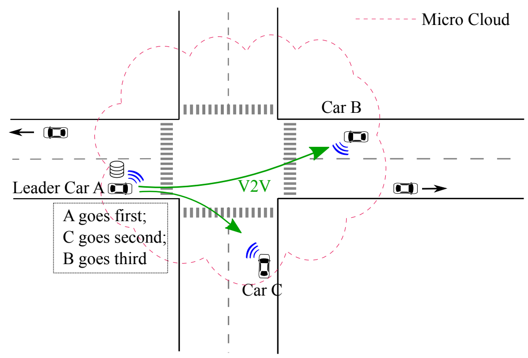

Figure 1. Overview of the SIM protocol. All cars share their arrival time at the intersection, and use this data for leader election. The leader car is responsible to find the intersection crossing priorities of all cars, and inform them about it.

In this paper, we are interested in the impact of multiple inconsistent data versions on the application performance; and how to improve the algorithms to deal with these issues. We assume that vehicles periodically exchange position updates and control messages. Some of the data messages may be lost due to network congestion, resulting in multiple versions of the same data coexisting in the micro cloud. We analyze the frequency of such data inconsistency and how it affects the application performance. This kind of data is small in size, but is very critical for proper operation, and management of micro clouds. For our investigation, we consider a simple, yet very important application for vehicular networks, i.e., a four-way stop intersection management.

II. Smart Intersection Management

Four-way stop intersections do not have any traffic light, and require all drivers to come to a stop; the one reaching first gets the right of way. Crossing such intersections is often complicated when several drivers approach the intersection from different directions at approximately the same time. We make use of a Smart Intersection Management (SIM) protocol using vehicular micro clouds – a practical demonstration of a simplified solution has been performed to show the conceptual feasibility [5].

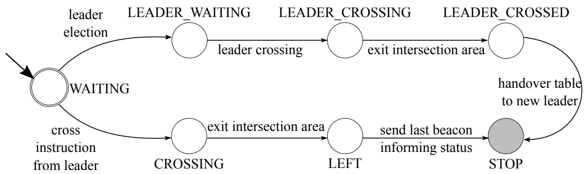

Figure 2. Different states of car at the intersection.

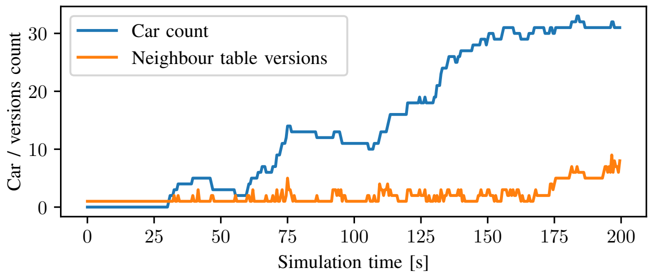

Figure 3. Car count and neighbor table versions at the intersection for a sample simulation.

Figure 1 shows an overview of the SIM protocol at a four-way stop intersection [5]. The basic idea of the SIM protocol is to let the cars cross the intersection by the first-come, first-served principle, which is a common way to cross at four-way stop intersections. In our protocol, the earliest arriving car becomes the leader at the intersection. The leader maintains timestamps of cars reaching the intersection in a table and broadcasts which cars can cross next. This arrival time of cars is received via periodic beacons sent by each car. The priority is given to cars with the earliest arriving times. If a car has crossed the intersection, its entry is removed from the table. When the leader has crossed the intersection, it selects the next leader and hands over the table. In the case that there are no further cars, or the leader handover fails, a new leader is elected. Figure 2 shows the state machine of a car at the intersection.

III. Existence of multiple data versions

We look into the number of versions of same data content which can co-exist in a micro cloud. To do so, we count the number of cars at the intersection participating in the intersection management application and compare their neighbor tables entry by entry. Figure 3 shows the number of table versions in the micro cloud and the car count. Increasing the number of cars in the micro cloud leads to an increase in different versions of data. This is because the channel gets more congested or there are packet losses due to shadowing and interference with other cars. Also, cars are constantly joining and leaving the micro cloud and need at least one beacon interval time period to receive from every surrounding vehicle a beacon message and setup their own neighbor table. Questions that may be raised with the increase of data versions could be: what is the impact on the performance of the application and whether the number of versions may converge fast enough to be negligible.

To reduce the versions of same data co-existing in the micro cloud at different cars, we enhanced the SIM protocol to keep history data about the cars which have already crossed the intersection in the neighbor table. We refer to this enhancement as Advanced-SIM protocol. The history data consists of cars quitting the state `LEADER_CROSSED`, and of cars with state `LEFT` to resolve data version conflicts. This increases the size of the neighbor table which needs to be maintained. However, the extra information helps in conflict resolutions in case the leader handover fails. The history entries are kept with `LEADER_LEFT` or `LEFT` status to improve synchronization during leader handover events and afterwards, in the case that not every car receives the message. Table I summarizes the comparison of SIM and Advanced-SIM protocols.

Table I
Summary of comparison between SIM and Advanced-SIM.

Implementation	SIM	Advanced-SIM
Number of Cars	100	100
Avg. waiting time at intersection(s)	58.3	55.3
Maximum waiting time(s)	230.1	85.8
Data bandwidth used(kB/s)	4.9	4.7
Average messages per car	242.9	220.9
Total messages	867783	807588
Total cars passed intersection	3573	3656

IV. Conclusion

In this paper, we showed the importance of analyzing data versions for typical micro cloud applications. Based on a SIM protocol, we showed that even in simple applications with minor channel congestion and only a few nodes synchronizing data versions to share common information may become a problem. We conclude that data versions depend on many parameters in micro clouds, such as the underlying algorithm, communication performance, and traffic density. We can further argue that some applications can deal with a small percentage of different data versions depending on the specific implementation. However, this often comes at some cost, usually in terms of added delays or an increase in channel usage. If versions start to deviate too much from each other, e.g., through high message losses, micro cloud operations can become sub-optimal. We showed that the effect of having different data versions should be taken into consideration when developing micro cloud applications with distributed data along the nodes.

References

[1] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, 2014.
[2] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, “Virtual Edge Computing Using Vehicular Micro Clouds,” in *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, Honolulu, HI: IEEE, Feb. 2019.
[3] I. Ahmad, R. M. Noor, M. R. Zaba, M. A. Qureshi, M. Imran, and M. Shoaib, “A Cooperative Heterogeneous Vehicular Clustering Mechanism for Road Traffic Management,” *International Journal of Parallel Programming*, Jan. 2019.
[4] A. Touil, A. Sbai, and F. Ghadi, “Cluster-based Data Collection Scheme for Vehicular Ad-hoc Networks,” *Procedia Computer Science*, vol. 148, pp. 62–69, Jan. 2019.
[5] T. Higuchi, S. Ucar, C.-H. Wang, and O. Altintas, “Vehicular Micro Cloud as an Enabler of Intelligent Intersection Management,” in *12th IEEE Vehicular Networking Conference (VNC 2020), Demo Session*, Virtual Conference: IEEE, Dec. 2020.