Content Replication in Vehicular Micro Cloud-based Data Storage: A Mobility-Aware Approach

Takamasa Higuchi*, Gurjashan Singh Pannu†, Falko Dressler† and Onur Altintas*
*Toyota InfoTechnology Center U.S.A., U.S.A.
†Paderborn University, Germany

***Abstract*—This paper presents an intelligent mechanism for a vehicular micro cloud (*i.e.*, a cluster of connected vehicles) to form a regional distributed data storage over V2V networks. Each member vehicle keeps a subset of data contents in their local data storage device, and hands over the contents to other member(s) when leaving the vehicular micro cloud. In order to cope with potential data loss due to failure of content handovers, the vehicular micro cloud creates replica(s) of each data content, and keeps them at different member vehicles. However, even if the system creates multiple replicas, there still remains the risk of losing all of them at once, as vehicles hosting the replicas could move away from the vehicular micro cloud in a short period of time. Such simultaneous failures could frequently occur, since vehicles in the vicinity often have strong correlation in their mobility. To address this issue, our algorithm assigns data replicas to a small subset of member vehicles, whose future mobility is not expected to be strongly correlated with each other. Simulation results show that our mobility-aware replica assignment algorithm can significantly improve the reliability of data management without increasing the number of replicas in a vehicular micro cloud.**

I. Introduction

The wide-scale deployment of cellular on-board data communication modules is becoming a reality, allowing vehicles to offer enriched information services through interactions with remote data centers. It opens up a new horizon of automotive applications, ranging from provisioning of up-to-date high-definition road maps to cloud-assisted intelligent driving [1]. These emerging services are expected to generate a huge volume of data traffic between vehicles and data centers, possibly with severe constraints on end-to-end latency. Efficient handing of such massive network traffic is a key to sustain evolution of connected vehicle services.

The potential of edge computing [1], [2] has been actively investigated to tackle this challenge. The key idea is to deploy small-scale data centers around the edge of the backbone networks (*e.g.*, cellular base stations and roadside units). Data caching and processing at the edge servers reduce the need for vehicles to access distant data centers by way of backbone networks, mitigating communication latency and network load.

The recent literature has further extended the notion of edge computing by utilizing on-board computer units of vehicles as virtual edge servers [3], [4], [5]. The idea is usually referred to as *cloudification of vehicles*, as vehicles themselves can be part of a distributed cloud computing platform (as shown in Fig. 1). Although individual vehicles may have limited amount of computational resources, their capabilities as virtual

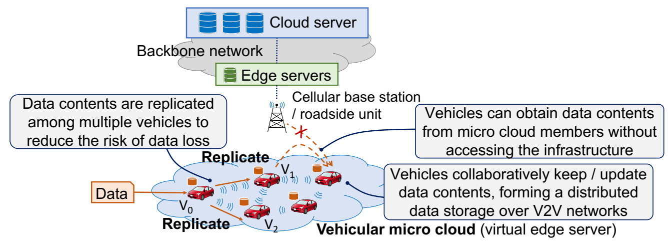

Fig. 1. Architecture of vehicular micro cloud-based data storage

edge servers can be enriched by collaboration among multiple vehicles. The vehicles in the vicinity form a cluster, called *vehicular micro cloud*, in which member vehicles collaboratively perform data processing, data storage, communication and sensing tasks over vehicle-to-vehicle (V2V) networks. Gerla [3], Dressler et al. [4], and Altintas et al. [5] were among the first to establish the fundamental concept of vehicle cloudification, while the authors recently investigated its feasibility by analyzing a realistic vehicle probe data set [6].

A typical use case of vehicular micro clouds is a regional distributed data storage service, where member vehicles collaboratively keep data contents in their on-board data storage devices. It allows vehicles in and around the vehicular micro cloud to request the contents from the micro cloud member(s) over V2V networks, or even update the data on the spot.

Unlike traditional distributed data storage systems, storage nodes in a vehicular micro cloud (*i.e.*, vehicles) can frequently join and leave the cluster due to their mobility. In order to keep data contents in a vehicular micro cloud, the vehicles need to hand over the data contents to other vehicle(s) before they leave the micro cloud. However, the data handover could possibly result in failure due to (i) packet collisions in wireless channels and/or (ii) insufficient network connectivity between vehicles. If a content handover fails, it directly leads to the risk of losing the data content from the vehicular micro cloud.

In this paper, we propose a novel approach to reliably keep data contents in a vehicular micro cloud. The basic idea is to generate multiple replicas of each data content and keep them at different micro cloud members *whose future mobility is not expected to be strongly correlated with each other*. Even if the system creates multiple replicas to cope with potential failures of content handovers, they can be lost at the same time if vehicles hosting the replicas move away from the vehicular micro cloud in a short period of time.

978-1-5386-9428-2/18/$31.00 ©2018 IEEE

Our mobility-aware replica assignment algorithm avoids such simultaneous failures by intelligently selecting an appropriate set of vehicles to assign replicas based on their expected future mobility. The results from simulation experiments using the Luxembourg SUMO traffic scenario [7] show that the proposed algorithm improves reliability of the data storage service without increasing the number of data replicas.

II. Related Work

The existing approaches for vehicle-based distributed data storage services can be classified into two categories: (i) dissemination-based and (ii) handover-based solutions.

Dissemination-based solutions repeatedly disseminate data contents in a designated geographical region, so that all the vehicles in the region can keep a copy of every content. Abiding geocast [8] distributes contents by periodically performing geographically-scoped flooding, or by epidemic data dissemination among vehicles in a designated region. Floating Content [9] dynamically prioritizes data contents such that contents are replicated less aggressively as distance from the center of the designated region increases. A key advantage of dissemination-based solutions is the lower risk of losing data from a cluster of vehicles. However, it potentially leads to inefficient utilization of data storage and communication resources due to excessive redundancy of data caches.

Handover-based solutions [10], [11] assume that data contents are handed over to another alternative vehicle on a unicast basis when a vehicle is leaving the designated region. A typical example is the ad-hoc persistence protocol [10], where data contents are handed over to the vehicles that are expected to travel toward the target area. This approach enables efficient utilization of data storage resources since only a single or a small number of micro cloud members keep the same data content. However, the data contents can be easily lost from the vehicular micro cloud if a data handover between micro cloud members fails. Hou et al. [11] recently tackled this issue by employing erasure coding. An original data content is encoded into N coded blocks, each of which is kept by different vehicles. The original data can be recovered as far as M out of N $(M < N)$ coded blocks remain in the designated region. Although it helps mitigate the risk of data loss, both read and write operations require access to at least M vehicles, potentially causing non-negligible latency and overhead.

A key contribution of our work is to tailor the handover-based approach to achieve the better trade-off between data storage / communication overhead and the risk of data loss. To the best of our knowledge, we are the first to investigate the idea of making intelligent data replication decisions based on the expected mobility of vehicular storage nodes.

III. Preliminaries

A. Overview of Vehicular Micro Clouds

Vehicular micro clouds can be grouped into two categories depending on their mobility [6]. A *stationary vehicular micro cloud* is tied to a certain geographical region such as a road intersection. Each vehicle joins the vehicular micro cloud when entering the designated region, and contributes to the collaborative task execution until it leaves this region. In contrast, a *mobile vehicular micro cloud* moves as member vehicles travel on a road. A vehicle that wants to execute a computational task becomes a so-called cloud leader, and invites neighboring vehicles as members of its micro cloud. The members leave the micro cloud when they move away from the cloud leader. Both types of vehicular micro clouds require data handovers when a member vehicle leaves the group. Our mobility-aware data replication mechanism can be applied to both types of micro clouds.

B. Distributed Data Management Protocol

There have been several well-established protocols for reliable data management in distributed data storage systems. One of the common approaches is *primary-based protocols*, where each data content is managed by a *primary* storage node that is responsible for that particular content [12]. Note that different primary nodes can be selected for individual data contents. In order to cope with the potential failure of the primary node, the node can create replica(s) of the data content and assign them to multiple other nodes as backups. When the primary node is down, another node keeping the replica takes over the primary role. The similar mechanism can be applied to vehicular micro clouds as well by selecting one of micro cloud members as a primary node for each data content. For simplicity of discussion, we assume the use of primary-based protocols in the following sections. Note, however, that our data replication mechanism is also applicable to other common data management schemes (*e.g.,* quorum-based protocols [12]).

IV. Mobility-Aware Content Replication

A. Reliability Metrics

Assume that a data content c is stored at a micro cloud member V_0 at time t_0. The vehicle V_0 then becomes a primary node for this content and keeps it in its on-board data storage device until the vehicle leaves the micro cloud at time t'_0. We define *time-to-live (TTL)* of this data content in the vehicular micro cloud by:

$$TTL(c; V_0) = t'_0 - t_0. \tag{1}$$

The data content has to be handed over to another micro cloud member before TTL expires (*i.e.*, before V_0 leaves the micro cloud). A vehicle that offers longer TTL can be considered more reliable, since it can keep the content longer in a vehicular micro cloud without performing data handover, which may occasionally result in failure. Therefore, we employ the TTL of a data content as a reliability metric in this paper.

In order to improve reliability against failure of data handovers, the primary node can create k more replicas and assign them to other micro cloud members $V_1, \ldots, V_k$. When the primary node fails to handover the content, one of the micro cloud members that keeps the replica takes over the primary role. Thus, the data content remains in the vehicular micro cloud as long as at least one replica is kept by any micro

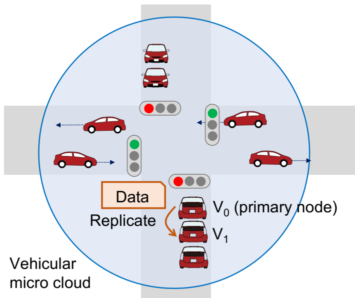

Fig. 2. Example of mobility correlation issue

Algorithm 1 Mobility-aware data replication

become a primary node for a data content c
$\mathcal{S} \leftarrow$ all the micro cloud members
$\mathcal{R} \leftarrow$ vehicles in $\mathcal{S}$ that already keep a replica of c
while $|\mathcal{R}| < k + 1$ and $|\mathcal{S}| > 0$ **do**
  $\mathcal{S} \leftarrow \mathcal{S} \ \setminus \{v \in \mathcal{S} \mid \exists v' \in \mathcal{R},\ \text{CORR}(v, v') = True\}$
  if $|\mathcal{S}| > 0$ **then**
    $v \leftarrow \text{SELECT\_VEHICLE}(\mathcal{S})$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{v\}$
  end if
end while

cloud members. With the k additional replicas, the TTL in Eq. (1) can be extended as:

$$TTL(c; V_0, \ldots, V_k) = \max_{0 \le i \le k} (t'_i - t_0). \tag{2}$$

The TTL tends to become longer by increasing the number of replicas, while excessive replication leads to inefficient use of data storage resources. In order to achieve reasonable reliability with a small number of replicas, we need an intelligent replica assignment algorithm, which identifies the vehicles that can effectively improve the TTL.

B. Vehicle Mobility Information

We assume that vehicles in a vehicular micro cloud are aware of the unique ID, position, and speed of other micro cloud members. It can be accomplished by periodically exchanging beacon messages (*e.g.*, Basic Safety Messages) over V2V networks. Our algorithm employs the collected vehicle mobility information for replica assignment.

C. Vehicle Mobility Correlation Issue

The existing data handover mechanisms have employed the following heuristic criteria to identify the vehicles that are likely to stay in a cluster for a long period of time:

- *Vehicle velocity* [13]: select the slowest vehicle.
- *Vehicle orientation* [10], [13]: select a vehicle moving toward the center of a micro cloud region.
- *Distance to the current data holder* [11], [13]: select a vehicle that is the closest to an existing replica holder (for better link quality of V2V communications).

Our goal in this paper is to maximize the TTL in Eq. (2) and consequently improve the reliability by intelligently assigning k more replicas to an appropriate set of micro cloud members. A straightforward approach would be to assign the replicas to the top k vehicles that are selected based on one of the above-mentioned criteria. However, as we will show in Section V, it often results in limited improvement in reliability. Fig. 2 shows an example of such scenarios. In this example, we assume that a primary node V_0 is assigning a new replica based on the *vehicle velocity* criterion. Even though V_1 stops at a traffic light, it may not be a good idea to assign the replica to V_1, because it is likely to leave the micro cloud shortly after V_0 once the traffic light turns to green. Such correlation of vehicle mobility could significantly affect the TTL and limit the benefit of keeping multiple replicas, as it leads to the risk of all the replicas being lost from the micro cloud within a short period of time.

D. Algorithm Design

Algorithm 1 shows overview of the mobility-aware data replication algorithm that runs on a primary node of each data content. The $\text{CORR}(v, v')$ is a function that returns $True$ if future mobility of a pair of vehicles v and v' is expected to correlate with each other, while the $\text{SELECT\_VEHICLE}(\mathcal{S})$ function selects the best vehicle in $\mathcal{S}$ to assign a new replica based on one of the criteria in Section IV-C. The key idea underlying the algorithm is to avoid correlation of mobility among multiple replica holders by introducing a mobility-aware filtering mechanism. When a primary node creates a new replica of a data content c, it identifies the micro cloud members, whose future mobility is likely to correlate with the primary node itself or any other vehicles already keeping a replica of c. Then it removes these member vehicles from a set of candidates to assign the new replica.

An important research question is how to design the function $\text{CORR}(v, v')$. In this paper, we employ a simple heuristic where mobility of a pair of vehicles v and v' is predicted to correlate if they meet the following two conditions:

- Distance between v and v' is less than a pre-defined threshold (*e.g.*, 50m)
- Both v and v' are traveling in the same direction along any lane of the same road segment

Although the heuristic above allows only rough estimation of mobility correlation, we will show in Section V that even such simple hard-coded decision criteria can help substantially improve reliability of data management. Alternatively, we could also perform short-term prediction of vehicle mobility based on the current and/or historical information on vehicle mobility in each region, or employ driving intention information shared among connected automated driving vehicles. We leave further exploration in these directions for future work.

V. Simulation

We have evaluated performance of the mobility-aware replica assignment algorithm through simulation experiments.

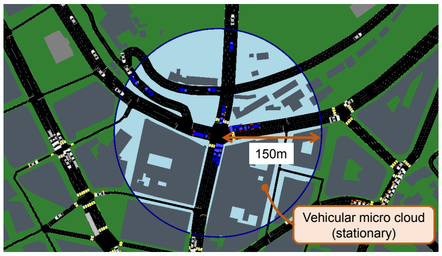

Fig. 3. A snapshot of the simulation area at 7am

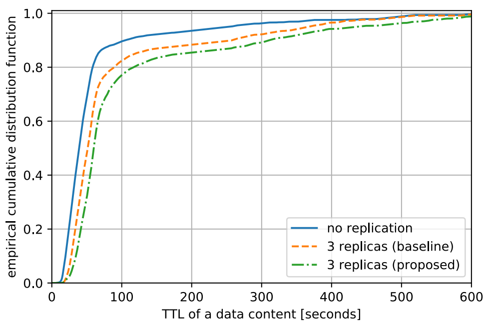

Fig. 4. Simulation results

We employed the traffic simulator SUMO[1] and the Luxembourg SUMO Traffic (LuST) scenario [7] to generate a realistic vehicle probe data of the city of Luxembourg. The LuST scenario covers a real road network spanning over a 11km $\times$ 13km city region, where realistic vehicle mobility is simulated based on public statistics on road traffic. We recorded positions of all the simulated vehicles every second to generate synthetic vehicle trace data from 7am to 8am.

From the data set, we randomly picked up a road intersection in the city center, and then set up a stationary vehicular micro cloud with the radius of 150m as shown in Fig. 3. For simplicity of discussion, we assume all the vehicles are equipped with a V2V communication module, and become a member of the micro cloud when entering this region.

Every second, a cloud leader (*i.e.*, a special member of the micro cloud) instructs a member vehicle with the lowest velocity to store a new data content. The selected vehicle becomes a primary node for this data content, and then assigns two more replicas in addition to its own replica of the content. In this experiment, we do not perform any data handover between vehicles, because our focus is on designing an algorithm for a primary node to intelligently assign replicas such that the TTL of the content can be effectively improved. As a selection policy for the SELECT_VEHICLE($\mathcal{S}$) function, we employ the vehicle velocity-based criterion, which selects the slowest vehicle in $\mathcal{S}$ after the mobility-aware filtering. To investigate the benefit of the mobility-aware filtering mechanism, we also evaluated a straightforward algorithm (referred to as *baseline*), which assigns replicas to the k slowest micro cloud members.

Fig. 4 shows the cumulative distribution of TTL of data contents. In order to identify the lower-bound of the performance, we also plot the results from a scenario where the primary node does not create any additional replica (referred to as *no-replication*). The average TTL is 63 seconds when the primary node does not create any additional replica. The *baseline* algorithm slightly improves the average TTL to 89 seconds by assigning additional replicas to the two slowest vehicles. The TTL increases if either of these two vehicles stay longer in the vehicular micro cloud than the primary node, while mobility correlation among the replica holders may limit the benefit of data replication. The *proposed* replica assignment algorithm can improve the average TTL by 24% compared to the *baseline* algorithm without increasing the number of replicas, showing the effectiveness of the mobility-aware filtering.

[1]http://sumo.dlr.de/index.html

VI. Conclusion

We have proposed the mobility-aware data replication algorithm for vehicular micro cloud-based data storage, which identifies an appropriate set of vehicles to assign data replicas based on their expected future mobility. The simulation results show that our algorithm improves the reliability of data management by 24% without increasing the data storage overhead. The future work includes investigation of its applicability to mobile vehicular micro clouds, more detailed performance evaluation with different parameter configurations and vehicle mobility scenarios, analysis on communication overhead and identification of the optimal number of data replicas.

References

[1] Automotive Edge Computing Consortium, "General principle and vision, white paper," 2018.
[2] 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.
[3] M. Gerla, "Vehicular cloud computing," in *Proc. Med-Hoc-Net*, 2012, pp. 152–155.
[4] F. Dressler, P. Handle, and C. Sommer, "Towards a vehicular cloud - using parked vehicles as a temporary network and storage infrastructure," in *Proc. of ACM WiMobCity*, 2014, pp. 11–18.
[5] O. Altintas, F. Dressler, F. Hagenauer, M. Matsumoto, M. Sepulcre, and C. Sommer, "Making cars a main ICT resource in smart cities," in *Proc. IEEE INFOCOM Workshop*, 2015, pp. 654–659.
[6] T. Higuchi, J. Joy, F. Dressler, M. Gerla, and O. Altintas, "On the feasibility of vehicular micro clouds," in *Proc. IEEE VNC*, 2017.
[7] L. Codeca, R. Frank, and T. Engel, "Luxembourg SUMO traffic (LuST) scenario: 24 hours of mobility for vehicular networking research," in *Proc. IEEE VNC*, 2015, pp. 1–8.
[8] C. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding geocast: Time–stable geocast for ad hoc networks," in *Proc. ACM VANET*, 2005, pp. 20–29.
[9] J. Ott, E. Hyytia, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *Proc. IEEE PerCom*, 2011, pp. 136–146.
[10] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent content-based information dissemination in hybrid vehicular networks," in *Proc. IEEE PerCom*, 2009, pp. 1–10.
[11] B. Hu, L. Fang, X. Cheng, and L. Yang, "Vehicle-to-vehicle distributed storage in vehicular networks," in *Proc. IEEE ICC*, 2018.
[12] M. van Steen and A. S. Tanenbaum, *Distributed Systems, Third Edition*. CreateSpace Independent Publishing Platform.
[13] C. Wu, T. Yoshinaga, Y. Ji, T. Murase, and Y. Zhang, "A reinforcement learning-based data storage scheme for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6336–6348, 2017.