Dwell Time Estimation at Intersections for Improved Vehicular Micro Cloud Operations

Gurjashan Singh Pannu[1], Seyhan Ucar[2], Takamasa Higuchi[2], Onur Altintas[2], Falko Dressler[1*]

[1]*School of Electrical Engineering and Computer Science, TU Berlin, Germany*

[2]*InfoTech Labs, Toyota Motor North America R&D, Mountain View, CA*

Abstract

Edge computing is becoming a major building block of next generation 5G/6G networks. However, infrastructure might not always be available because of slow deployment. At the same time, vehicular networks are becoming a reality now and cars are being equipped with a variety of short-range communication devices. The idea of vehicular micro clouds is to turn cars into (virtual) edge computing infrastructure. One of the challenging questions in this domain is to maintain data within and among such micro clouds. In this paper, we focus on this task and present a novel solution for such data exchange between vehicular micro clouds. For efficient operation, the dwell times of cars in such a micro cloud need to be known or accurately predicted. In an extensive study based on trace data, we investigate the distribution of dwell times of cars at intersections. We make use of this distribution as an input for designing an improved data exchange algorithm. As not all intersections are the same, adding additional variance further benefits the solution. We evaluated our algorithm in different vehicular densities, and we observed that we could maintain data 22–208 % longer within the micro clouds using our new algorithm. Overall, our results show that our algorithm clearly outperforms previous solutions.

Keywords: Mobile Edge Computing, Vehicular Cloud, Vehicular Micro Cloud, Data Management

1. Introduction

There has been a recent shift in focus of vehicular networking research community towards applications supporting cooperative driving [1, 2] and cooperative perception [3]. This is further supported as modern cars are equipped with wide range of sensors, computing, networking, and storage resources. The sensing on-board units sense abundant data from the surrounding, which can also be cooperatively used by nearby vehicles, bicyclists, and even pedestrians. Cooperative sensing can be used to maintain live 3D maps and even make complex maneuvers safely. The powerful configuration of cars makes them an important Information and Communication Systems (ICT) resource, transforming the Intelligent Transportation Systems (ITS) in future smart cities.

Data generated from applications assisting ITS operations needs to be stored and also requires frequent updates. Uploading the data to data centers is beneficial for further analysis using advanced data analytic techniques. However, when downloading data from data centers, vehicles may experience higher end-to-end delays. This may happen not only due to network capacity limitations, but also due to physical communication distance between the vehicles and the data centers.

To solve similar problems in cellular mobile networks, the Mobile Edge Computing (MEC) [4] architecture has been proposed. The underlying idea is to provide computing and storage capabilities at the edge of the cellular network, which is in close proximity to the users. In vehicular networks, Eltoweissy et al. [5], Gerla [6], Dressler et al. [7] proposed the concept of vehicular cloud computing, which later evolved as the vehicular micro cloud architecture [8, 9]. In simple words, cars cooperatively form a small cluster called *vehicular micro cloud* which offers computing and storage services to nearby cars, pedestrians, and bicyclists. These micro clouds act as virtual edge servers which are physically on the road, in very close proximity to the users, thereby extending the concept of MEC in vehicular networks. The micro clouds can further be classified into two types, i.e. stationary and mobile micro clouds. If the location of micro cloud is fixed to a certain geographic location, e.g., an intersection then the micro cloud is stationary micro cloud. Stationary micro clouds are formed at geographic locations with high traffic density [8]. If micro clouds are formed by cars moving in same direction, we call them vehicular micro clouds.

A stationary micro cloud setup at an intersection can serve as a distributed data storage unit [10]. Stored data contents are associated to a unique vehicular micro cloud. We refer to the associated micro cloud as *source micro cloud*. One of the main open research questions in sta-

*Corresponding author.

Email addresses: `pannu@ccs-labs.org` (Gurjashan Singh Pannu[1]), `seyhan.ucar@toyota.com` (Seyhan Ucar[2]), `takamasa.higuchi@toyota.com` (Takamasa Higuchi[2]), `onur.altintas@toyota.com` (Onur Altintas[2]), `dressler@ccs-labs.org` (Falko Dressler[1])

Article published in Elsevier Ad Hoc Networks 122 (2021). (10.1016/j.comcom.2020.05.001)

tionary vehicular micro cloud research is to keep the data available in the source micro cloud. This becomes very challenging due to vehicular mobility. Cars moving towards an intersection join the stationary micro cloud, collect some data belonging to this micro cloud, and after a while, they leave. If the car leaving the micro cloud is the last one to have a copy of a certain data content, then the data becomes non-recoverable in the source micro cloud. The variation in traffic density adds more challenges on top of it. The micro cloud region can get over-crowded by cars at certain times, and within a fraction of a minute, all cars can leave the micro cloud too. First solutions to tackle this problem were proposed in [11]. The idea is to allow cars to request data contents from not just the current micro cloud they are in, but also from the micro clouds which are located along their followed route. In [11], the proposed algorithm adaptively selects the number of micro clouds located along the route, from which the car should request data. This decision depends on the current network channel utilization. Although, this helps to improve the data life time in the micro clouds, there is still more room for improvements. One of the major factors which can be taken into accounts is vehicle's life time in the micro clouds. If each car knows their estimated life time, they can make optimal decisions to request certain data contents such that the overall life time of data content in micro clouds can be improved further.

In this paper, we enhance the protocol proposed in [11] further, to take into account estimated life time of cars for requesting micro cloud data contents. The enhanced protocol is compared against two cases, i.e., 1) when there is no protocol working to maintain the data, and 2) protocol proposed in [11]. We also evaluate the protocol in many traffic densities to find the benefits of our protocol.

Our contributions can be summarized as follows:

- We propose an enhanced communication protocol for connecting distributed virtual edge clouds, so called vehicular micro clouds. This protocol is based on an empirical distribution of dwell times of cars in realistic city environments.
- As a baseline, we investigated the dwell time distribution of cars at intersections using the mobility models for the city of Luxembourg. We identified the underlying statistical distribution, which, in turn, has been used for the improved protocol design.
- In a detailed performance evaluation with varying car densities in realistic urban environments, we show the benefits of making use of realistic dwell time distributions. Our new protocol improves the data lifetime in micro clouds by 22–208 %.

2. Related Work

Gerla [6], Eltoweissy et al. [5], Dressler et al. [7] introduced the concept of vehicular clouds with an aim to bring the cloud computing concept to vehicular networks. Lee et al. [12] discussed architecture and design principles of vehicular clouds from a systems, networking, and service perspective. The core idea is to organize vehicles to cooperatively use their storage, computing and networking resources for service provision [13].

Exploiting unused vehicular resources is also the foundation of Mobile Edge Computing, originating from *cloudlets* [14] and fog computing [15]. The key idea is to use resources at the edge of the network and therefore closer to the end-user. ETSI has standardized a MEC architecture [4, 16] with potential applications like augmented reality and connected cars [16]. In a survey, Mach and Becvar [17] classified proposed MEC applications by outlining three service categories: consumer oriented services, operator and third-party services, and network performance and QoE improvement services.

MEC and vehicular clouds have been combined by Higuchi et al. [8]. By arranging *micro* and *macro* clouds, a large number of services can be offered. On one hand, micro clouds are usually small vehicular clouds [6, 7] offloading resources and providing locally relevant services. On the other hand, macro clouds are envisioned to span whole cities in order to provide services originating farther away or in a data center.

Recently, Hagenauer et al. [9] investigated a data collection application in vehicular micro clouds. They introduced algorithms to form micro clouds based upon geographic locations and a *dedicated* car chosen as a cluster head in the micro cloud is responsible for collecting data from other members and uploading it. We investigated the benefits of using multiple technologies to avoid overloading any specific channel for uploading data from micro clouds to data centers [18].

This paper focuses on vehicular micro clouds, which are usually formed using clustering concepts. Clustering groups cars based on the similarity of a set of parameters, e.g., position, or direction [19].

Data in vehicular clouds is usually associated with specific geographic locations, thus, favoring Information Centric Networking (ICN) [20] and Named Data Networking (NDN) [21] architectures. Quite some work has been done in this context for efficient data delivery and caching. In mobile ad-hoc networks, Bellavista et al. [22] proposed opportunistic resource replication middle-ware which is tolerant to node exit or failure events. For vehicular networks, Lee et al. [23] proposed exploiting vehicular mobility to diffuse sensed data. Amadeo et al. [24] designed a content-centric framework for vehicular networks on top of IEEE 802.11p protocol stack addressing reliable data delivery. Grassi et al. [25] presented an approach using road-topology to determine shortest paths to data residing locations for efficient fetching in vehicular NDN. Rao et al. [26] proposed pre-fetching and caching in NDN for the networks with user mobility when a handover is about to happen. Grewe et al. [27] defined popularity of data as a function of the number of requests and proposed pre-fetching of data according to their popularity.

Most recently, Grewe et al. [28] formulated cache capacities and probabilities to retrieve the data from the cache successfully. They also presented different caching strategies to improve cache utilization and increase the efficiency of data delivery in information-centric vehicular applications. To avoid excessive cache utilization, Hu et al. [29] also proposed controlled redundancy of data using erasure coding techniques. Higuchi et al. [10] proposed an algorithm that relies upon mobility of the vehicles to find an appropriate and small set of cars which should keep data copies so that the data can stay within its source micro cloud for the maximum amount of time. Their idea is to select those set of vehicles whose mobility is not correlated, i.e., they are not moving in such a way that all of them leave the micro cloud in very short span of time.

While these works aim for efficient data diffusion and cache utilization when the data is present in the network, our designed protocol complements the existing works by proposing a novel protocol which relies on the knowledge of times spent by vehicles in micro clouds to prioritize data transmissions with an aim to recover the data at certain geographic locations which was lost due to vehicular mobility.

3. Protocol Details

In this paper, we focus on maintaining data contents belonging to a stationary micro cloud set up at an intersection. Intersections serve as an appropriate location for setting up micro clouds as vehicles traveling on multiple road segments can meet at one place [8]. We assume that each data content is associated with only one micro cloud, as most of the data contents relevant to micro cloud applications are associated to a certain geographic location, and hence, can be associated to the corresponding stationary micro cloud. We refer to this micro cloud as *source micro cloud* for the data content.

When a car leaves a micro cloud, it needs to transfer the data to some other car which is still a member of the micro cloud, or which may join the micro cloud in near future, such that the data can be retained within the source micro cloud. Data contents can be backed up to a remote cloud when the last car in leaves a micro cloud, but it incurs an additional communication overhead. Our goal is to maximize data availability time within its source micro cloud, i.e., despite cars which hold certain data contents leave the source micro cloud, the data should be available at the micro cloud location.

In this section, we present the details of the protocol which extends our previous work in [11]. The main enhancement is to use an estimated dwell time of a car in the micro cloud in the protocol decision making process. We define dwell time as the time a car spends in a micro cloud. In brief, if an estimated dwell time of a car is known, it can prioritize the data transfer to other cars, depending on data lifetime. In addition, knowing the dwell time of other cars helps in finding a suitable recipient car for data transfer before leaving the micro cloud.

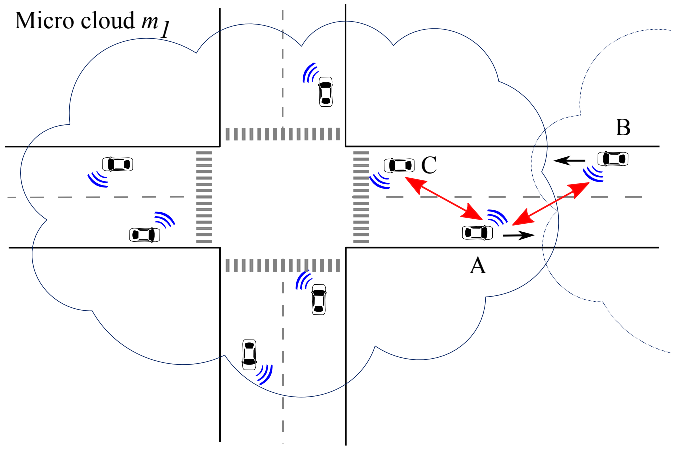


Figure 1: Typical micro cloud scenario: Car A leaves the micro cloud m_1 and car B is approaching m_1. A transfers the data contents of m_1 to B, thus, B brings back the data copies within the source micro cloud m_1.

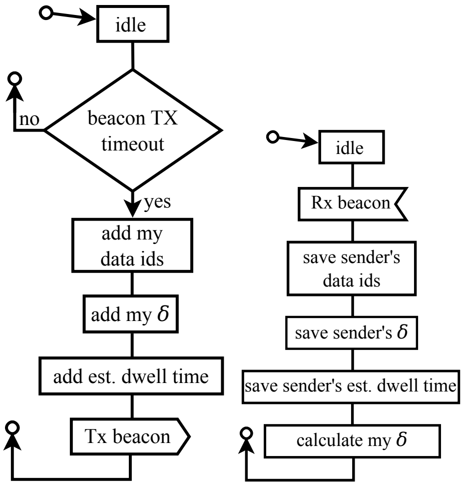
3.1. Prerequisites

For the protocol to work properly, the following assumptions are made.

- *Networking capabilities:* We assume that all cars are equipped with wireless networking radios. In the scope of this paper, we rely upon IEEE 802.11p for inter vehicle communications. However, this could also be replaced by other technologies as the protocol does not make any technology specific assumptions.
- *Geographic position and route:* We assume that all cars are equipped with a GPS device to know their current location and the drivers use a navigation service using which the route followed by a car is known in advance.
- *Micro cloud information:* All of the cars need to know the location of micro clouds and their unique ids. This is an important information for a car as it needs to broadcast data interests for a specific micro cloud. We assume that each car can download information about micro clouds (location, size, and id) along its followed route once before starting the trip.
- *Storage capabilities:* All cars are ready to offer their storage resources to micro cloud services.

3.2. Example Scenario

In Figure 1, we can see a typical micro cloud scenario. The micro cloud m_1 is established at an intersection. Car A, which is currently a member of m_1, is about the leave m_1. If A has the last copy of some data contents associated



(a) transmission (b) reception

Figure 2: Flowchart showing steps taken while sending and receiving a micro cloud control beacon. δ refers to the set of data content ids which a car is missing.

to m_1, then those data contents will be lost from m_1 as soon as A leaves. At this point, A has two options: 1) transfer data to some other car, e.g., car C which is staying in m_1 for longer time, or 2) transfer data to Car B which is currently not a member of m_1, but would soon join m_1. The details about the protocol are explained in next Section 3.3.

3.3. Enhanced Inter Micro Cloud Coordination Protocol

To keep data contents available in their source micro cloud, cars need to know which data contents are present in the micro cloud and where they are present. To get this information, the cars exchange control beacons, which are used to carry advertisements of available data contents. Based on the information received in these advertisements, the cars request the data contents, with an aim to keep them in their source micro cloud for longer time.

3.3.1. Control Beacons

Each car periodically broadcasts control beacons. These beacons include (1) the sender car's id, (2) the current micro cloud id, (3) a set of data contents it has, (4) a set of missing data content ids, which it desires to have access to, and (5) an estimated dwell time of itself. A flowchart describing sending of a control beacon is shown in Figure 2a. We refer to the set of missing data content ids as δ.

Steps taken on receiving a control beacon are shown in Figure 2b. The control beacons are used by receiving cars to maintain a micro cloud metadata table. The information received from the latest control beacon of each car is saved in the metadata table. This includes a list of data content ids which the sender has, a list of data content ids which the sender requests, and the sender's estimated dwell time. Based on the saved information, the receiver car calculates his own δ (cf. Equation (1)). The calculated δ is subsequently included in the next transmitted control beacons.

An example micro cloud metadata table for a car c_1 is shown in Table 1. The first row of the table is reserved for the car itself. We can see that c_1 is a member of micro cloud m_1 and it has two data items with ids d_1 and d_2, both of which are associated to micro cloud m_1. The dwell time for c_1 is t_1, i.e., after t_1 time, c_1 is expected to leave the micro cloud m_1. Car c_2 is also a member of m_1. It has only d_1 data content and is interested in d_2 and d_3. There is an interesting entry for car c_4. It is a member of micro cloud m_2 and has data content d_8 which is associated to micro cloud m_2. Note that it is interested in data contents d_1, d_2 and d_3 which belong to micro cloud m_1. One possible interpretation for this is that c_4 is soon going to leave m_2 and join m_1, thus it requests for known data contents of m_1 early enough, so that even if some cars leave the micro cloud m_1, c_4 can still bring back the associated data to m_1. This is possible when it receives control beacons from c_1.

To generalize, let M be a set of next n micro clouds along the route of a car. It has micro cloud metadata table T with r rows. $T[i]$ represents an i^{th} row $\forall i \in (0, r-1)$ in T. Let there be k_i data elements in the data set in $T[i]$ and j^{th} data element in $T[i]$ is given by $\{d_{T[i],j}, m_j\}$. The 0^{th} row in metadata table has the information about the car itself. So, a set of micro cloud metadata δ for the car is calculated as

$$\delta_{T[0]} = \{\bigcup_{i=1}^{r-1} \bigcup_{j=0}^{k_i} \{d_{T[i],j}, m_j\} \forall m_j \in M\} \setminus \bigcup_{j=0}^{k_0} \{d_{T[0],j}, m_j\}, \tag{1}$$

where $\bigcup_{j=0}^{k_0} \{d_{T[0],j}, m_j\}$ represents data contents owned by the car itself and $\bigcup_{j=0}^{k_i} \{d_{T[i],j}, m_j\} \forall m_j \in M$ represents the latest data contents associated to micro cloud $m_j \in M$ which are owned by another car whose entry is stored in i^{th} row in table T.

3.3.2. Micro Cloud Data

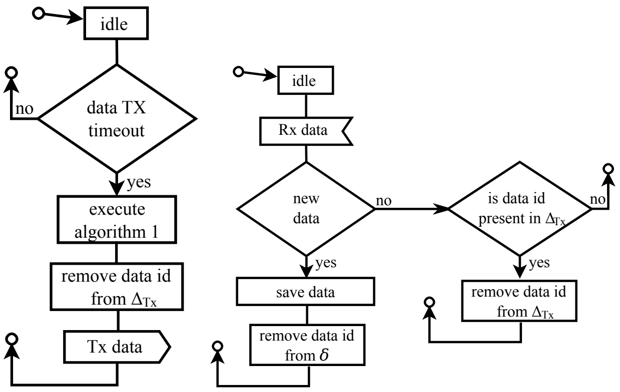
In the micro cloud metadata table, each car maintains the knowledge about dwell times and data contents accessible and requested by itself and other cars. In order to transfer the micro cloud data, the car calculates a set of data content ids which it has but others miss (referred to as Δ_{Tx}).

$$\Delta_{\text{Tx}} = \{\bigcup_{j=0}^{k_0} \{d_{T[0],j}, m_j\}\} \bigcap \{\bigcup_{i=1}^{r-1} \{\delta_{T[i]}\}\}, \tag{2}$$

where $\bigcup_{j=0}^{k_0} \{d_{T[0],j}, m_j\}$ represents the data contents accessible by the car, and $\bigcup_{i=i}^{r-1} \{\delta_{T[i]}\}$ represents all the data contents which are missed by other cars. One data content is selected from the calculated Δ_{Tx} for transmission. Figure 3 shows the steps taken by a car while sending and receiving a micro cloud data content. As shown in Figure 3a, a car executes data transmission scheduling algorithm (cf. Algorithm 1) for selecting the data content

Car	Micro Cloud	Data	Missing Data (δ)	Dwell time
c_1	m_1	$\{\{d_1, m_1\}, \{d_2, m_1\}\}$	$\{\{d_3, m_1\}\}$	t_1
c_2	m_1	$\{\{d_1, m_1\}\}$	$\{\{d_2, m_1\}, \{d_3, m_1\}\}$	t_2
c_3	m_1	$\{\{d_2, m_1\}, \{d_3, m_1\}, \{d_1, m_1\}\}$	ϕ	t_3
c_4	m_2	$\{\{d_8, m_2\}\}$	$\{\{d_1, m_1\}, \{d_2, m_1\}, \{d_3, m_1\}\}$	t_4

Table 1: An example micro cloud metadata table for car c_1.



(a) transmission (b) reception

Figure 3: Flowchart showing steps taken while sending and receiving a micro cloud data content.

that should be transmitted first to serve the data requests of other cars obtained via their control beacons. Subsequently, after the transmission of data content, an entry corresponding to the data content transmitted is removed from Δ_{Tx} (cf. Equation (4)).

On receiving a data content, a car checks whether it already has a copy of this data content or not. If it is received for the first time, the car saves the data content, and removes the data id from the list of requested data ids. If a copy of this data content was already present at the car, then the car removes the data id from list of data contents to be transmitted. This is done in order to avoid repeated transmission of same data contents, as one of the neighboring cars has just transmitted this data content. The process is showing in Figure 3b.

3.3.3. Scheduling Micro Cloud Data Transfer

Δ_{Tx} gives a set of data contents which the car can transfer to other cars. Each car schedules transmission of one data content at a time. Many factors can be taken into account for this selection, e.g., remaining lifetime of data before it gets outdated, last transmission of the data content, dwell time of the cars which requesting data, etc.

In this paper, we consider two deciding factors for scheduling data transmissions, (1) remaining lifetime of data, and (2) dwell time of cars requesting data.

The data content with maximum remaining lifetime is given by

$$d_{\text{Tx}} = \underset{d \in \Delta_{\text{Tx}}}{\text{argmax}}\, \texttt{LIFETIME}(\texttt{d}), \tag{3}$$

where d_{Tx} is the data in Δ_{Tx} who has maximum remaining lifetime and $\texttt{LIFETIME}(d)$ gives the remaining lifetime of the data content d.

Algorithm 1 describes scheduling of data contents for transmission based on the dwell time of cars. In lines 1 and 2, we calculate set of cars which are present in the same micro cloud, and outside the micro cloud based on the contents of the metadata table. The data requests are served alternately to cars which are within the micro cloud and the cars which are approaching (outside) the micro clouds. This helps in completing the data requests from cars present in the micro cloud, as well as to allow other cars approaching the micro cloud to have data in advance, so that if there is a potential data loss, it can be recovered. Lines 5-7 are executed to find car c_{max} within the micro

Algorithm 1 Scheduling data for transmission based on dwell time of cars

Input: $T[i], i \in (0, r-1)$: Micro cloud metadata table
Input: M: List of next future micro clouds
Input: b_{inside}: Boolean to serve request from car in current micro cloud or outside micro cloud
Input: τ: Threshold time

1: $\mathbb{C}_{in} \leftarrow \bigcup_{i=1}^{i=r-1} \{c_i \forall m_i \in M \text{ where } m_i = m_0)\}$ ▷ cars in micro cloud
2: $\mathbb{C}_{out} \leftarrow \bigcup_{i=1}^{i=r-1} \{c_i \forall (m_i \in M \text{ where } m_i \neq m_0)\}$ ▷ cars outside micro cloud
3: $\mathbb{D} \leftarrow \phi$
4: **if** b_{inside} == TRUE **then**
5: $c_{\max} \leftarrow \text{argmax}_{c \in \mathbb{C}_{\text{in}}} \texttt{DWELLTIME\_IN}(\texttt{c}, \tau)$ ▷ car with max remaining time in micro cloud, and greater than τ
6: $\mathbb{D} \leftarrow \Delta_{\text{Tx}} \bigcap \delta_{c_{\max}}$
7: $b_{\text{inside}} \leftarrow$ FALSE
8: **else if** b_{inside} == FALSE **then**
9: $c_{\min} \leftarrow \text{argmin}_{c \in \mathbb{C}_{\text{out}}} \texttt{DWELLTIME\_OUT}(\texttt{c}, \tau)$ ▷ car with minimum remaining lifetime and lesser than τ
10: $\mathbb{D} \leftarrow \Delta_{\text{Tx}} \bigcap \delta_{c_{\min}}$
11: $b_{\text{inside}} \leftarrow$ TRUE
12: **end if**
13: **if** $\mathbb{D} \neq \phi$ **then**
14: $d_{\text{Tx}} \leftarrow \text{argmax}_{d \in \mathbb{D}} \texttt{LIFETIME}(\texttt{d})$
15: **else**
16: $d_{\text{Tx}} \leftarrow \text{argmax}_{d \in \Delta_{\text{Tx}}} \texttt{LIFETIME}(\texttt{d})$
17: **end if**
18: **return** d_{Tx}

cloud which has maximum remaining time in the micro cloud. $\mathbb{D}$ contains list of data contents requested by the c_{max} which can be transmitted by the car executing this algorithm. On the other hand, lines 9-11 help finding car c_{min} outside the micro cloud with minimum dwell time and list of data contents requested by it which can be transmitted. In lines 13-16, the car finally selects one data content from the list $\mathbb{D}$ based on the maximum remaining lifetime of data in the micro cloud, and initiates the data transmission.

When data request is served to the car inside the micro cloud, the remaining dwell time of the car needs to be greater than a certain threshold τ. This ensures, that the car receiving the data is going to stay within the micro cloud for at least τ time. On the other hand, in the case that a vehicle requesting a data content currently belongs to a different micro cloud, the request will not be fulfilled until the vehicle's dwell time in its current micro cloud becomes less than a threshold. This ensures that the car receiving data content is joining the micro cloud soon.

Whenever a car receives a data d_{Tx}, it saves the data, i.e. $\bigcup_{j=0}^{k_0}\{d_{\mathrm{T}[0],j}, m_j\} \bigcup d_{\mathrm{Tx}}$.

To avoid saturating the channel by transmitting the same data contents by many cars, receiving d_{Tx} also triggers removal of the data content from Δ_{Tx}, i.e.,

$$\Delta_{\mathrm{Tx}} = \Delta_{\mathrm{Tx}} \setminus d_{\mathrm{Tx}}. \tag{4}$$

Allowing cars to request data contents not just from their current micro cloud but also the next micro clouds along the route helps to recover the lost data back in micro clouds. However, there is a limit on the number of upcoming micro clouds for which it can effectively request data. This is because increasing data requests eventually leads to frequent data exchange, which can quickly overload the wireless channel.

To overcome this problem, the protocol takes current channel utilization into account. Each car periodically measures the channel utilization relying on the carrier sensing mechanism. The channel utilization is given by

$$\gamma = \frac{t_{\mathrm{busy}}}{t_{\mathrm{busy}} + t_{\mathrm{idle}}} \ , \tag{5}$$

where t_{busy} is the time for which channel was sensed busy and t_{idle} is the time for which channel was sensed idle.

Let N be the maximum number of micro clouds (excluding the current micro cloud) for which a car is allowed to request data. Then, the number of micro clouds (excluding the current micro cloud) for which the car actually requests data based upon current channel utilization is given by

$$n = \begin{cases} N - \lfloor \gamma \times \omega \rfloor & \text{if } N > \lfloor \gamma \times \omega \rfloor \\ 1 & \text{if } N \leq \lfloor \gamma \times \omega \rfloor \end{cases} \ , \tag{6}$$

where ω is a scaling factor to optimize the n selection. In our evaluations, we assume ω to be 10 for the simplicity of converting $\lfloor \gamma \times \omega \rfloor$ to integers ranging in $[0, 10]$. It can be observed that as channel utilization γ increases, the car reduces the number of micro clouds for which it requests data. In case of very high channel utilization, where $N \leq \lfloor \gamma \times \omega \rfloor$, the car will request data for the current micro cloud and the next one.

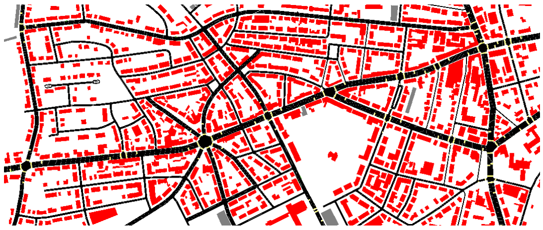


Figure 4: Screenshot from SUMO running LuST scenario showing an arterial road which connects inner city of Luxembourg to freeway.

Furthermore, the data transmission interval is also adaptable. We introduced a data transmission window which is given by

$$\mathrm{Tx}_{\mathrm{window}} = [0, \mathtt{max}(1, \lfloor \gamma \times \omega \rfloor)] \ . \tag{7}$$

A random value within this $\mathrm{Tx}_{\mathrm{window}}$ is selected as the next transmission interval. As the channel utilization increases, the window size increases as well, thus increasing the average data transmission interval which helps avoiding channel overloading.

In general, the protocol is capable to function in a distributed fashion without any additional infrastructure support. We believe, deploying roadside units running the protocol can further improve the performance. However, within the scope of this paper, we assume that there is no infrastructure available in the region.

4. Dwell Time Analysis

To evaluate the performance of the designed protocol, we selected Luxembourg city scenario [30]. The scenario features 24 hours of realistic traffic mobility in a typical European city. We selected an arterial road connecting city center of Luxembourg city to a freeway. Several roads from the city connect to this arterial road. The area is shown in Figure 4.

To maintain data within a micro cloud setup at an intersection, it becomes important to understand the dwell time of cars in micro cloud. The dwell time for a car can be understood as amount of time a car spends in a micro cloud. This includes time spent moving towards an intersection, waiting for the traffic light to turn green, and the time spent moving away from the intersection until it leaves the micro cloud.

The main idea behind studying dwell times is that, if data request made by cars relies on their estimated dwell times (or their remaining time in the micro cloud), we could benefit from several perspectives: (1) Cars can decide to not request data contents, if they will no longer be in a micro cloud for considerably long time. (2) Cars which

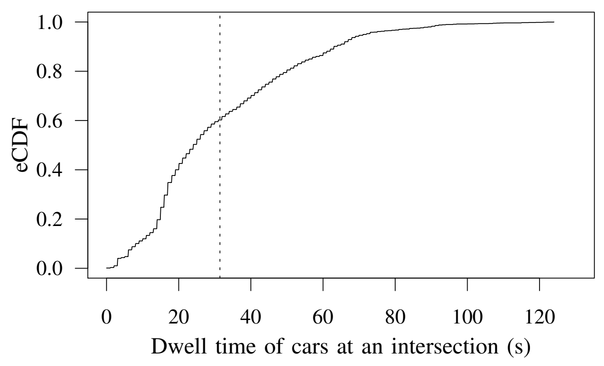


Figure 5: eCDF of dwell time of cars at an intersection on an arterial road.

may stay in a micro cloud for longer time can choose to request for more data contents and help in maintaining the data for longer times.

With this aim, we recorded the dwell time of cars at each intersection in the selected scenario. In total, there were 219 intersections in the region. Intersections have usually been considered a favorable location for the micro clouds since they provide connectivity to cars in many directions.

At each intersection, we recorded the times for which a car was within a 100 m radius distance. For a typical intersection, an eCDF of dwell times in the morning hours is shown in Figure 5.

As we can see, there are some cars, which cross the intersection very fast, because at the time when they approached intersection, the traffic light is already green. On the other hand, there are many cars which stay more than a minute, and even up to 2 minutes. The vertical dotted line shows the average dwell time of cars at the intersection.

In practice, the dwell times can be learned with high levels of accuracy using machine learning techniques. However, thorough analysis of the dwell times is not the main focus of our research. To proceed with the performance evaluation of the proposed protocol for maintaining data contents within a micro cloud, we use Maximum Likelihood Estimation technique to find the fitting distribution.

In general, Maximum Likelihood Estimation is a technique to estimate the parameters of a given distribution using an available data. To define formally, if $x_1, x_2, x_3, ..., x_n$ are the observations from n independent and identically distributed random variables drawn from a probability distribution f_0, where f_0 is known to be from a family of distributions f that depend on some parameters θ, the goal of maximum likelihood estimation is to maximize the likelihood function

$$L = f(x_1, x_2, x_3, ..., x_n|\theta), \tag{8}$$

or

$$L = f(x_1|\theta) \times f(x_2|\theta) \times f(x_3|\theta) \times ... \times f(x_n|\theta). \tag{9}$$

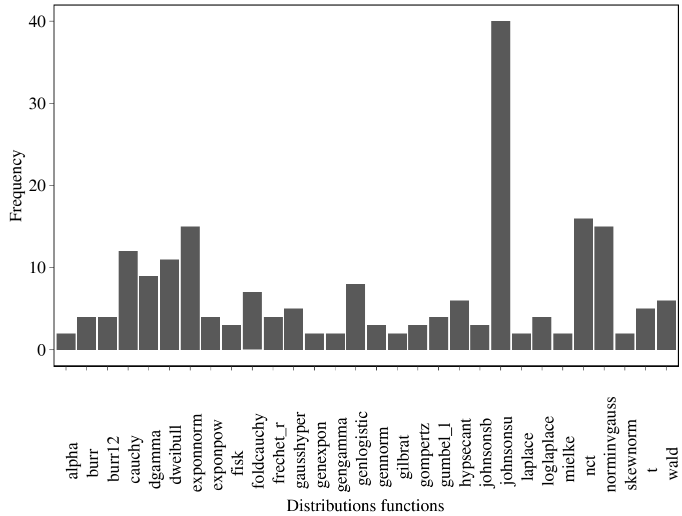


Figure 6: The plot shows the frequency of different distributions which show highest p-value for the fitting to recording intersection time of cars at different intersections.

We estimated parameters for several known distributions, and for each of the estimations, we tested goodness of fit using Kolmogorov-Smirnov test [31]. Table 2 shows a list of distributions which were fit to recorded dwell times, and their corresponding p-values in decreasing order for one of the main intersections which is also used in protocol performance evaluation in Section 5. It can be seen that p-value corresponding to Johnson's S_U distribution [32] is the largest. Except for Johnson's S_U and Generalized Normal distributions, other distributions can be rejected to fit the given set of recorded data as their p-value is quite low.

We tried to fit several distributions on dwell time recorded data for each of the intersections in the region of interest as shown in Figure 4, and Figure 6 shows the frequency of distributions exhibiting the highest p-values among all of the intersections. In order to maintain the readability of the plot, we show the distributions which fit the dwell time data more than once. It can be seen that Johnson's S_U distribution shows the highest p-value for maximum number of intersections, even in different times of the day.

Figure 7 shows a histogram of normalized dwell time

Distribution	p-value
Johnson's S_U	0.7421
Generalized Normal	0.5648
Log Normal	0.2447
Normal Inverse Gaussian	0.1193
...	...
Half-Cauchy	8.119×10^{-49}

Table 2: Calculated p-values for the distribution fitting to recorded dwell-times sorted in decreasing order.

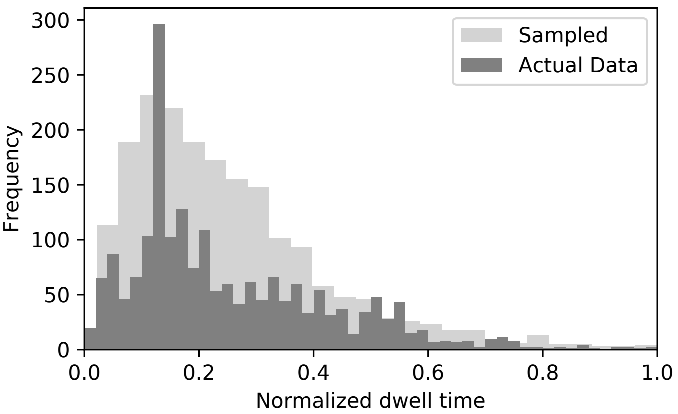


Figure 7: Histogram showing normalized dwell times as observed in data recorded from SUMO, and the samples drawn from Johnson's S_U distribution.

data as observed in SUMO at a specific intersection (light gray) and the samples drawn from Johnson's S_U distribution with estimated parameters. The probability distribution function for Johnson S_U distribution is defined as:

$$f(x, a, b) = \frac{b}{\sqrt{x^2+1}}\phi(a + b\log(x + \sqrt{x^2+1})) \quad (10)$$

for all $x, a, b > 0$, and ϕ is the normal pdf. Here, a and b are the shape parameters.

Obviously, the overall mapping of Johnson's S_U to all intersections is not mapping perfectly. This is because some intersections deviate strongly from this distribution (cf. Figure 6). As the main goal of this paper is not to find a perfectly fitting distribution, we consider this mapping as a good starting point to cover intersections for which the real distribution is unknown with a very high probability.

5. Performance Evaluation

In order to evaluate the performance of the designed protocol, we used Veins [33] which couples road traffic simulator SUMO[1] with network simulator OMNeT++.[2] The simulation scenario is the same which has been used for studying dwell time of cars, as shown in Figure 4. We setup 7 micro clouds at the main intersections of the arterial road. We ran simulations during three different times of the day in order to study the impact of different traffic densities. The different times and the average number of cars in the micro cloud are shown in Table 3.

The estimated dwell time of the cars were drawn from the distributions with their known parameters from the maximum likelihood estimations. In each of the micro clouds, a new data content was generated every 15 s. The lifetime of each data content is configured to 150 s. Control beacon interval in the protocol is set to 1 s. We also configure the cars to express their interest in only a certain percentage of the total data in micro cloud (25, 50, 75 and 100 %). Here, 25 % data interest means that at any point of time, a car is interested to request for only 25 % of the total micro cloud data. This parameter has been selected keeping in mind that some of the contents might not be relevant to a particular vehicle depending on a set of applications it runs on the on-board computer unit. From a user's perspective, the vehicle may not want to request data contents which are not relevant to it.

We compared several configurations of our protocol for performance evaluation:

- *Baseline*: In this configuration, the cars do not request for any data which belongs to other micro clouds, i.e., there is no coordination between cars belonging to other micro clouds. All of the cars are interested in data belonging to the current micro cloud only. The data requests are fulfilled once every second. The transmission of data contents is scheduled periodically every second. This configuration serves as the baseline.

- *Dynamic*: Here, the cars are interested in requesting data belonging to their current micro cloud, as well as next n micro clouds that they will be joining in near future. In this configuration, the dwell time of cars is not considered. This configuration of protocol is same as our previous work [11]. In this case, we use $N = 6$, i.e., a car can request for data contents of current and at most next 6 micro clouds along its route.

- *Ideal*: In this configuration, the cars have precise knowledge of their dwell times in the current micro cloud, i.e., the dwell times are 100% accurately known. This configuration helps us get insights about the best possible performance that we can get from the designed protocol.

- *Dwell*: In this configuration, dwell time of cars is drawn from the distribution with known parameters. That is, each time a random number is chosen from the fitted Johnson's S_U distribution.

- *Err*: In this configuration, we add ±10% error to the actual dwell time of cars as used in ideal configuration. This configuration helps us understand the performance of protocol when the dwell times are estimated within a certain error range.

Time	Traffic density	Avg. cars in micro cloud
08:00	235 $1/\text{km}^2$ (High)	24
11:15	58 $1/\text{km}^2$ (Low)	6
13:00	157 $1/\text{km}^2$ (Medium)	16

Table 3: Average number of cars in a micro cloud for different times of the day

[1]http://sumo.dlr.de
[2]http://www.omnetpp.org

For the baseline configuration, $n = 0$ because there is no coordination between neighboring micro clouds and cars request for data belonging to their current micro cloud only. For the remaining configurations, the value of n is selected adaptively based on the current channel utilization (cf. Equation (6)). The data transmission window is also adaptable as defined in Equation (7). We rely upon IEEE 802.11p at a data rate of 6 Mbit/s in our simulations. To avoid any boundary effects in data collection, we recorded data in the central micro cloud which is surrounded by 3 other micro clouds in both directions. Each simulation configuration was repeated 15 times with different seeds to improve statistical evidence. All relevant simulation parameters are summarized in Table 4.

5.1. Channel Utilization

First of all, we look into the channel utilization in different configurations. We measure channel utilization as the fraction of time, for which the channel was sensed busy by the MAC layer. It is given by Equation (5). Each car measures the channel utilization every second.

Figure 8 shows the average channel utilization in all the configurations with a 0.95 confidence interval. In all configurations, except for 10 kByte data size in high traffic scenario, channel utilization for the baseline configuration is much less than other configurations. This is because the cars in baseline configuration request data belonging to their current micro cloud only. However, in other configurations, the cars are able to adaptively increase or decrease the number of micro clouds for which they request the data contents. As the network utilization is not very high, the cars request data belonging to their future micro clouds, by adaptively increasing the n value. Hence, we see a higher channel utilization.

For 10 kByte data size in high traffic scenario, when the channel utilization in baseline is sufficiently high, the cars try to reduce the channel utilization by increasing the data transmission window and decreasing the n value (cf. Table 5).

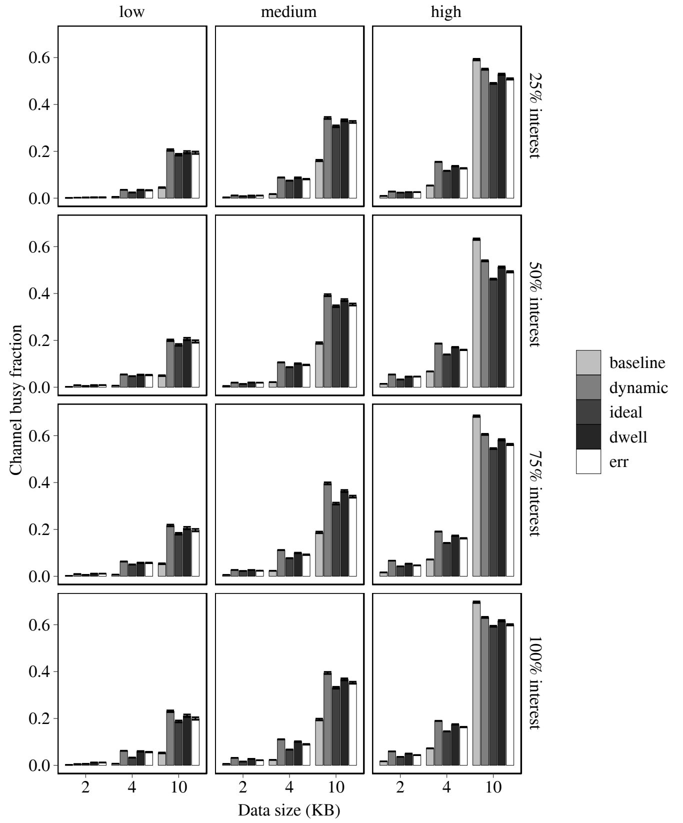


Figure 8: Average channel busy fraction observed in the micro cloud.

Parameter	Value
Channel	5.89 GHz
Transmission power	20 mW
Bandwidth	10 MHz
Data rate	6 Mbit/s
New data generation interval	15 s
Data lifetime	150 s
Data size generated	2, 4 and 10 kByte
Control beacon interval	1 s
N (cf. Equation (6))	6
n (Baseline)	0
n (other configurations)	cf. Equation (6)
Data interest	25, 50, 75 and 100 %
ω in adaptive protocol	10
Repetitions per configuration	15
Simulation duration	600 s

Table 4: Simulation parameters.

While comparing dynamic, ideal, dwell, and err configurations, we can observe that the channel utilization is minimum in the ideal configuration. This is because the cars have 100% accurate knowledge of their dwell times. As a result, the cars are able to reduce their data transfer when a car is about to move out of the current micro cloud. For ±10% error configuration, there are a few cases when the car already moves out, but it is assumed by other cars to be in the same micro cloud. For dwell configuration, the estimations are within the acceptable range, but are assigned randomly to the cars, due to which this configuration shows higher channel utilization.

5.2. Data Availability in Micro Cloud

After looking into channel utilization, we study the impact of the protocol on the data availability.

Figure 9 shows the average time for which each of the data content is available in its source micro cloud. The error bars represent 0.95 confidence intervals (Table 6 shows more detailed number for the 10 kByte configuration). From the recordings, we can observe that the performance in baseline configuration is the worst. In baseline configuration, at low vehicular densities, the cars are not able to maintain data in the micro cloud for very long time. The data lifetime increases for higher vehicular densities. However, for higher traffic density, with increasing size of data contents, the average lifetime of data decreases. This is because we observed higher channel utilization for larger data size (cf. Figure 8). Because of higher channel utilization, effective data transfer was less than the other cases. In other configurations, as the cars could request data contents belonging to their future micro clouds, they were able to bring back the lost data contents to their source micro cloud.

For dynamic configuration, the average lifetime of data is relatively less than the ideal, dwell, and err configurations. This shows that relying on dwell time of cars can actually help us maintaining data in micro clouds for a longer time. Interestingly, we also see that the accuracy of dwell times play a very important role. Ideal which has 100% accurate dwell times show the best average lifetime. Err configuration with ±10% error on ideal values has slightly less average lifetime, and the dwell configuration where the estimated lifetimes are drawn from the distribution, but may or may not be accurate, the lifetime is even smaller.

With our proposed protocol, the requests of the cars are not served when they are about to leave the micro cloud. This contributes in reducing the channel utilization to some extent, as seen in Figure 8. However, the cars still have access to many data contents, which they are still able to pass on to the cars which join the micro cloud soon. These cars are able to pass on the data content to other cars approaching the micro cloud to recover data losses. As the effective channel utilization in this case is comparatively better than the dynamic configuration, overall data transfer is more effective and we could maintain data for higher

Interest	Config	Low traffic	Medium traffic	High traffic
25%	Dynamic	355.2%	113.1%	-6.4%
	Ideal	310.5%	90.9%	-17.2%
	Dwell	334.1%	107.1%	-10.1%
	Error	329.0%	102.5%	-13.9%
50%	Dynamic	306.2%	108.3%	-14.5%
	Ideal	266.0%	83.3%	-27.3%
	Dwell	317.0%	97.1%	-19.3%
	Error	295.4%	87.2%	-22.0%
75%	Dynamic	306.7%	111.8%	-10.8%
	Ideal	239.5%	65.7%	-19.8%
	Dwell	284.4%	94.4%	-14.3%
	Error	267.9%	81.5%	-17.2%
100%	Dynamic	336.7%	105.8%	-9.3%
	Ideal	254.4%	72.9%	-14.8%
	Dwell	300.6%	91.6%	-11.4%
	Error	277.9%	84.1%	-13.9%

Table 5: Channel utilization for 10 kByte data for the studied traffic loads compared to the baseline.

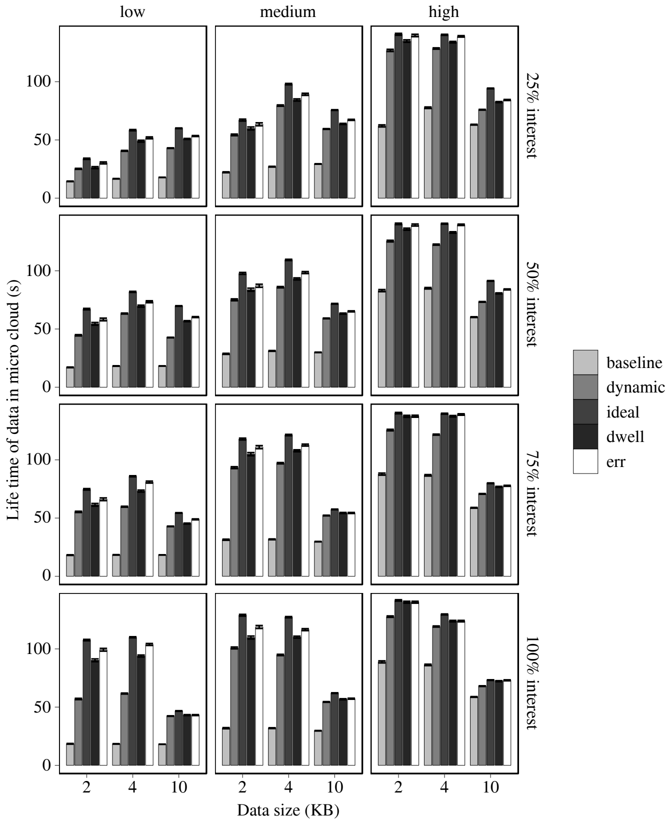


Figure 9: Average time for which each data is available in the micro cloud.

times.

5.3. Fraction of Data Available in Micro Cloud

In addition to the average lifetime of data contents in their source micro cloud, we also study fraction of data contents which are available in the micro cloud to get insights about total data that can be maintained within a source micro cloud. It is calculated as:

$$\mathrm{f}_m = \frac{\text{Unique data contents present in micro cloud } m}{\text{Total data contents expected in micro cloud } m}. \tag{11}$$

To evaluate this metric, we sampled total unique data contents present in a micro cloud and total unique data expected to be present in the micro cloud periodically every second. Figure 10 shows the fraction of data available in different configurations with 0.95 confidence intervals (Table 7 shows more detailed number for the 10 kByte configuration).

From the plots, in the baseline configuration, the amount of data contents which are present in the micro cloud is very low, because the data is lost as soon as the last car containing the data content moves out of the micro cloud. This is the main problem due to vehicular mobility. In general, with increasing vehicular density, the overall fraction of data contents maintained in the micro cloud increases, as there are many other cars, who may have the copy of a data content.

In dynamic configuration, the cars from other micro cloud are able to request data contents for the micro clouds which they join soon. As a result, we see a tremendous increase in the fraction of data contents which are available in the micro cloud. As the size of data contents increase from 4 kByte to 10 kByte, the fraction of data contents available in the micro cloud is reduced. This is because, the larger data contents are transferred in chunks. The data is counted as available in micro cloud only when all chunks of data are present in the micro cloud. Although, when only a few chunks of data content are in the micro cloud, it is also considered to be missing data. If we apply the protocol on per chunk basis, the data fraction would increase.

In configurations which rely on dwell times, we see an improvement in the fraction of data contents available in the micro cloud. This is supported by the lower channel utilization observed in Figure 8 and higher lifetime of data Figure 9.

By maintaining larger amounts of data (cf. Figure 10) for significantly larger time (cf. Figure 9), vehicular micro clouds can be considered as a good platform for applications like cooperative driving, intersection management, etc. even in low and medium traffic densities.

5.4. Redundancy of Data in Micro Cloud

Since our protocol works opportunistically without requiring any extra infrastructure, it is important to study the level of data redundancy in the micro cloud, i.e., on an average, how many data copies are present in a micro cloud. Figure 11 shows the average number of data copies in micro cloud with 0.95 confidence intervals. For low traffic density,

Interest	Config	Low traffic	Medium traffic	High traffic
25%	Dynamic	139.0%	102.6%	20.4%
	Ideal	233.7%	160.6%	49.6%
	Dwell	182.7%	120.3%	30.7%
	Error	197.0%	130.6%	35.0%
50%	Dynamic	131.0%	95.9%	21.8%
	Ideal	281.3%	136.8%	52.9%
	Dwell	208.5%	109.9%	33.8%
	Error	227.3%	114.5%	39.5%
75%	Dynamic	134.5%	76.5%	19.6%
	Ideal	200.6%	93.4%	35.1%
	Dwell	149.2%	82.4%	29.1%
	Error	168.8%	83.9%	30.6%
100%	Dynamic	133.0%	83.0%	16.9%
	Ideal	154.4%	107.2%	25.8%
	Dwell	136.3%	89.7%	22.9%
	Error	135.8%	91.4%	25.7%

Table 6: Data availability improvement for 10 kByte data for the studied traffic loads compared to the baseline.

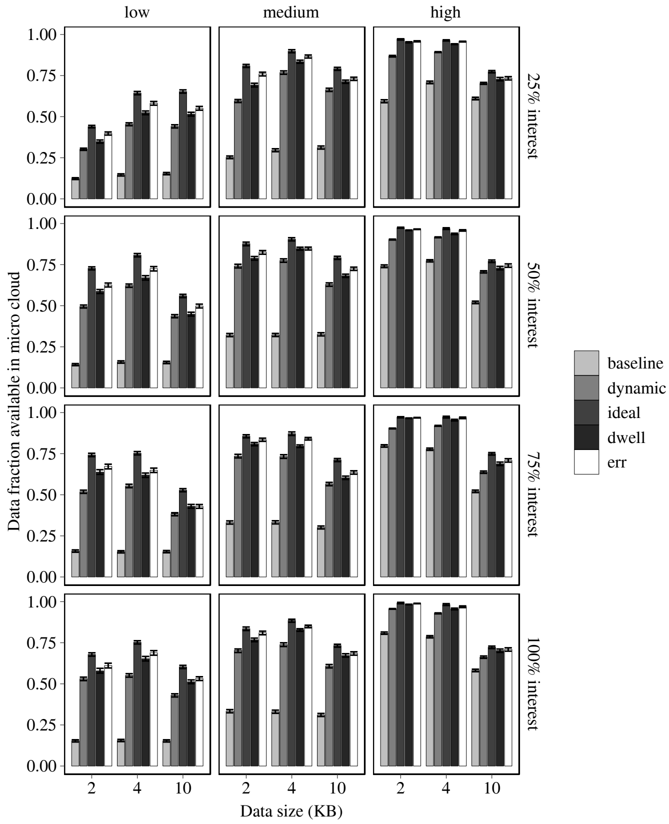


Figure 10: Average fraction of data available in the micro cloud.

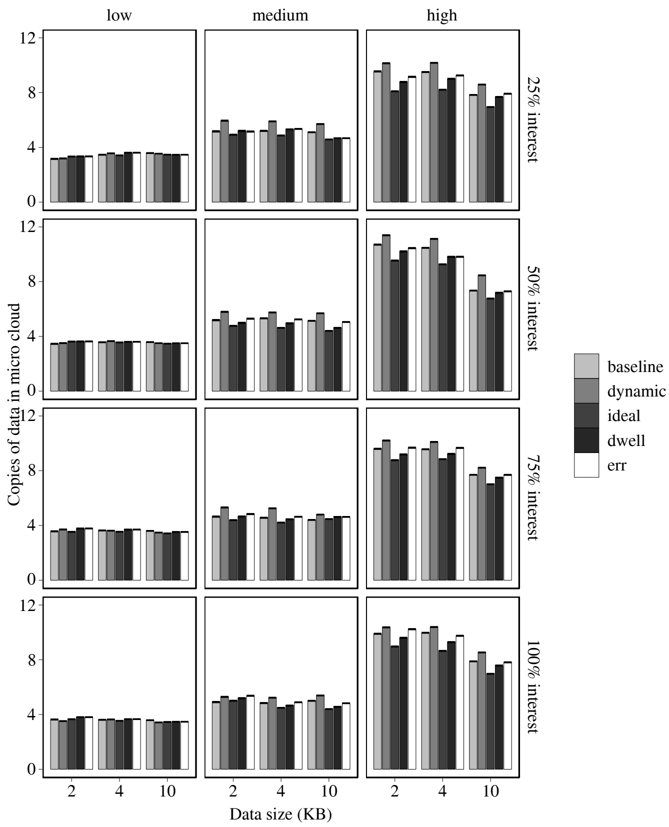


Figure 11: Average number of data copies in micro cloud.

there are approximately 4 copies of the data content on an average. This is also traffic density dependent. As the traffic density increases, the number of data copies in micro cloud also increase.

For higher traffic densities, we notice an interesting trend. The level of redundancy increases slightly for the dynamic configuration. This can be understood as cars bringing a copy of data contents from outside of the micro cloud. However, in configurations which rely on dwell times, the level of redundancy is relatively lower than that of dynamic configuration. This is an artifact of the reduced number of data requests served when the car is about to leave the current micro cloud. However, an interesting observation is that this phenomenon helps us in getting benefits in overall channel utilization, and hence, helps in effective data transfer between the cars to bring back more data contents into the micro cloud with lower levels of redundancy.

5.5. Number of Transmissions per Data Content

In the proposed protocol, cars are interested in the data contents of the current as well as future micro clouds, so we are also interested to see how many total transmissions of a data content take place. We measure the total transmissions for each data inside and outside the source micro cloud. Figure 12 shows the average transmissions per data content with 0.95 confidence intervals. As expected, the number of data transmission in the baseline are minimum among all configurations. This is because the data contents are requested by the cars belonging to the same micro cloud only.

Interest	Config	Low traffic	Medium traffic	High traffic
25%	Dynamic	181.1%	112.6%	15.1%
	Ideal	325.3%	153.4%	26.7%
	Dwell	240.1%	128.5%	19.3%
	Error	258.4%	133.9%	20.2%
50%	Dynamic	181.1%	93.1%	35.5%
	Ideal	260.5%	143.9%	48.0%
	Dwell	189.8%	109.4%	40.0%
	Error	221.6%	122.5%	42.9%
75%	Dynamic	149.3%	92.2%	22.4%
	Ideal	243.1%	143.6%	43.9%
	Dwell	178.9%	102.9%	30.1%
	Error	179.0%	114.0%	34.1%
100%	Dynamic	175.0%	111.9%	14.1%
	Ideal	288.1%	156.6%	24.5%
	Dwell	231.4%	135.1%	20.4%
	Error	240.5%	143.2%	21.7%

Table 7: Data fraction availability for 10 kByte data for the studied traffic loads compared to the baseline.

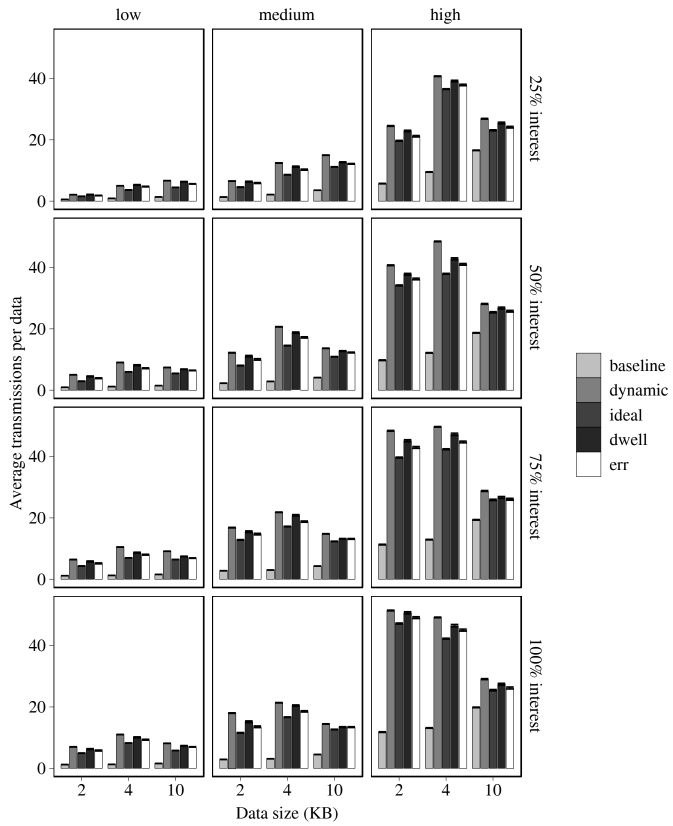


Figure 12: Average number of transmissions for each data in micro cloud.

In dynamic configuration, the cars requests more number of data contents, as they request not only for the data belonging to current micro cloud, but also next micro clouds which it will join in near future. As a result, to fulfill their requests, the number transmissions per data content increase. An interesting observation is for the data content size of 10 kByte. In medium and high traffic density, the average number of transmissions per data content decreases compared to that of smaller data contents. This is because the dynamic configuration adapts the data transmission window based on the current channel utilization. Under high channel utilization, the number of data content requests decrease.

In configurations using dwell times, the average data transmissions are less than that of dynamic configuration. This is because the cars reduce the data requests for the current micro cloud, when they are about to leave the micro cloud.

In dwell configuration, we can see increase in the data transmissions compared to err configuration. This is because of the inaccuracy in the estimated dwell times. When the cars are not aware of the time when they are going to leave the micro cloud, they can continue to request data contents.

6. Conclusions

In this paper, we presented an inter micro cloud coordination protocol extending our previous work from paper [11]. The aim of this protocol is to keep data contents available in a micro cloud setup at intersections in an urban scenario. A micro cloud is a small cluster of cars which acts as a virtual edge server offering caching and computational resources. In this context, it becomes important to keep the current data within its source micro cloud, but it is often challenged by the mobility of cars. Cars join a micro cloud, collect some data, and contribute to the micro cloud services. As soon as they leave, the data may get lost.

The main idea of the protocol is that cars participating in a micro cloud request for data contents not just for the current micro cloud, but also request data contents which belong micro clouds located along their followed route in advance. As an enhancement, the requests from the cars also depend upon the time which the cars will spend in the micro cloud. We refer to this time as dwell time. If the cars are about to leave a micro cloud, they stop requesting data belonging to the current micro cloud. This helps in reducing the overall channel utilization in the network, because of which we saw improvements in over effective data transfer between cars.

We studied the performance of protocol in several configurations, (1) ideal – where cars have 100 % accurate knowledge of their dwell times, (2) an estimation of the dwell times with maximum likelihood estimations, and using the outcome from the distribution as estimated dwell time of cars, and 3) adding 10 % error to the ideal configuration. The three configurations have been compared with the baseline, where the protocol is not running at all, and our previous work [11].

We evaluated our algorithm in different vehicular densities, and we observed that we could maintain data 22–208 % longer within the micro clouds using our new algorithm. So, relying on dwell times can help in improving the data lifetime in micro clouds further. However, the accuracy of the dwell times plays a crucial role.

With good accuracy, the protocol is able to help in reducing overall channel utilization. As a consequence, we are able to maintain more data contents for longer period of time in their source micro cloud.

References

[1] O. Altintas, S. Banerjee, F. Dressler, G. Heijenk, Executive Summary - Inter-Vehicular Communication Towards Cooperative Driving, in: Dagstuhl Seminar 18202 on Inter-Vehicular Communication - Towards Cooperative Driving, Schloss Dagstuhl, Schloss Dagstuhl, Germany, 2018, pp. 31–59. doi:10.4230/DagRep.8.5.31.

[2] T. Higuchi, S. Ucar, C.-H. Wang, O. Altintas, Vehicular Micro Cloud as an Enabler of Intelligent Intersection Management, in: 12th IEEE Vehicular Networking Conference (VNC 2020), Demo Session, IEEE, Virtual Conference, 2020.

[3] S. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, D. Rus, Multivehicle Cooperative Driving Using Cooperative Perception: Design and Experimental Validation, IEEE Transactions on Intelligent Transportation Systems (TITS) 16 (2015) 663–680. doi:10.1109/TITS.2014.2337316.

[4] Mobile Edge Computing (MEC), Framework and Reference Architecture, GS MEC 003 V1.1.1, ETSI, 2016.

[5] M. Eltoweissy, S. Olariu, M. Younis, Towards Autonomous Vehicular Clouds, in: J. Zheng, D. Simplot-Ryl, V. C. M. Leung (Eds.), 2nd International ICST Conference on Ad Hoc Networks (ADHOCNETS 2010), Springer, Victoria, Canada, 2010, pp. 1–16. doi:10.1007/978-3-642-17994-5_1.

[6] M. Gerla, Vehicular Cloud Computing, in: 11th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2012), IEEE, Ayia Napa, Cyprus, 2012, pp. 152–155. doi:10.1109/MedHocNet.2012.6257116.

[7] F. Dressler, P. Handle, C. Sommer, Towards a Vehicular Cloud - Using Parked Vehicles as a Temporary Network and Storage Infrastructure, in: 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2014), ACM International Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity 2014), ACM, Philadelphia, PA, 2014, pp. 11–18. doi:10.1145/2633661.2633671.

[8] T. Higuchi, J. Joy, F. Dressler, M. Gerla, O. Altintas, On the Feasibility of Vehicular Micro Clouds, in: 9th IEEE Vehicular Networking Conference (VNC 2017), IEEE, Turin, Italy, 2017, pp. 179–182. doi:10.1109/VNC.2017.8275621.

[9] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, 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), ACM, Snowbird, UT, 2017, pp. 31–35. doi:10.1145/3131944.3133937.

[10] T. Higuchi, G. S. Pannu, F. Dressler, O. Altintas, Content Replication in Vehicular Micro Cloud-based Data Storage: A Mobility-Aware Approach, in: 10th IEEE Vehicular Networking Conference (VNC 2018), IEEE, Taipei, Taiwan, 2018. doi:10.1109/VNC.2018.8628438.

[11] G. S. Pannu, F. Hagenauer, T. Higuchi, O. Altintas, 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), IEEE, Washington, D.C., 2019. doi:10.1109/WoWMoM.2019.8792973.

[12] E. Lee, E.-K. Lee, M. Gerla, S. Oh, Vehicular Cloud Networking: Architecture and Design Principles, IEEE Communications Magazine (COMMAG) 52 (2014) 148–155. doi:10.1109/MCOM.2014.6736756.

[13] A. Boukerche, R. E. De Grande, Vehicular Cloud Computing: Architectures, Applications, and Mobility, Elsevier Computer Networks (COMNET) 135 (2018) 171–189. doi:10.1016/j.comnet.2018.01.004.

[14] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The Case for VM-Based Cloudlets in Mobile Computing, IEEE Pervasive Computing 8 (2009) 14–23. doi:10.1109/MPRV.2009.82.

[15] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog Computing and Its Role in the Internet of Things, in: 1st Workshop on Mobile Cloud Computing (MCC 2012), ACM, Helsinki, Finland, 2012, pp. 13–16. doi:10.1145/2342509.2342513.

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

[17] P. Mach, Z. Becvar, Mobile Edge Computing: A Survey on Architecture and Computation Offloading, IEEE Communications Surveys & Tutorials 19 (2017) 1628–1656. doi:10.1109/COMST.2017.2682318.

[18] G. S. Pannu, T. Higuchi, O. Altintas, F. Dressler, Efficient Uplink from Vehicular Micro Cloud Solutions to Data Centers, in: 19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2018), IEEE,

Chania, Greece, 2018. doi:10.1109/WoWMoM.2018.8449787.

[19] C. Cooper, D. Franklin, M. Ros, F. Safaei, M. Abolhasan, A Comparative Survey of VANET Clustering Techniques, IEEE Communications Surveys & Tutorials 19 (2017) 657–681. doi:10.1109/COMST.2016.2611524.

[20] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A Survey of Information-Centric Networking, IEEE Communications Magazine (COMMAG) 50 (2012) 26–36. doi:10.1109/MCOM.2012.6231276.

[21] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. C. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named Data Networking, ACM SIGCOMM Computer Communication Review (CCR) 44 (2014) 66–73. doi:10.1145/2656877.2656887.

[22] P. Bellavista, A. Corradi, E. Magistretti, REDMAN: an Optimistic Replication Middleware for Read-only Resources in Dense MANETs, Pervasive and Mobile Computing 1 (2005) 279–310. doi:10.1016/j.pmcj.2005.06.002.

[23] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, A. Corradi, Mobeyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network, IEEE Wireless Communications 13 (2006) 52 – 57. doi:10.1109/WC-M.2006.250358.

[24] M. Amadeo, C. Campolo, A. Molinaro, CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks, IEEE Communications Letters 16 (2012) 1380–1383. doi:10.1109/LCOMM.2012.072012.120282.

[25] G. Grassi, D. Pesavento, G. Pau, L. Zhang, S. Fdida, Navigo: Interest Forwarding by Geolocations in Vehicular Named Data Networking, in: 16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2015), IEEE, Boston, MA, 2015.

[26] Y. Rao, H. Zhou, D. Gao, H. Luo, Y. Liu, Proactive Caching for Enhancing User-Side Mobility Support in Named Data Networking, in: 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2013), IEEE, Taichung, Taiwan, 2013.

[27] D. Grewe, S. Schildt, M. Wagner, H. Frey, ADePt: Adaptive Distributed Content Prefetching for Information-Centric Connected Vehicles, in: 87th IEEE Vehicular Technology Conference (VTC 2018-Spring), IEEE, Porto, Portugal, 2018. doi:10.1109/VTCSpring.2018.8417748.

[28] D. Grewe, M. Wagner, S. Schildt, M. Arumaithurai, H. Frey, Caching-as-a-Service in Virtualized Caches for Information-Centric Connected Vehicle Environments, in: 10th IEEE Vehicular Networking Conference (VNC 2018), IEEE, Taipei, Taiwan, 2018.

[29] B. Hu, L. Fang, X. Cheng, L. Yang, Vehicle-to-Vehicle Distributed Storage in Vehicular Networks, in: IEEE International Conference on Communications (ICC 2018), IEEE, Kansas City, MO, 2018.

[30] L. Codeca, R. Frank, T. Engel, Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research, in: 7th IEEE Vehicular Networking Conference (VNC 2015), IEEE, Kyoto, Japan, 2015. doi:10.1109/VNC.2015.7385539.

[31] F. J. Massey, The Kolmogorov-Smirnov Test for Goodness of Fit, Journal of the American Statistical Association 46 (1951) 68–78. doi:10.1080/01621459.1951.10500769.

[32] N. L. Johnson, Systems of Frequency Curves Generated by Methods of Translation, Biometrika 36 (1949) 149. doi:10.2307/2332539.

[33] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, IEEE Transactions on Mobile Computing (TMC) 10 (2011) 3–15. doi:10.1109/TMC.2010.133.