Vehicular Micro Cloud in Action: On Gateway Selection and Gateway Handovers

Florian Hagenauer[a], Christoph Sommer[a], Takamasa Higuchi[b], Onur Altintas[b], Falko Dressler[*,a]

[a] *Heinz Nixdorf Institute and Dept. of Computer Science, University of Paderborn, Germany*
[b] *Toyota InfoTechnology Center, U.S.A., Mountain View, CA*

Abstract

Recently, the concept of the vehicular micro cloud has been established. The core idea is to use cars as a main ICT resource in modern smart cities being able to store, to forward, and to process data. Relying on the communication capabilities of cars, particularly their ability to use short range communication technologies such as IEEE 802.11p, Wi-Fi, or LTE-D2D, virtual micro clouds can be established similar to the concept of the mobile edge in 5G networks, yet, without any infrastructure support. However, there is a key disadvantage: the network will very likely become fragmented due to the mobility of the cars within the city. Furthermore, low penetration rates in early deployments may amplify this fragmentation. In recent work, we proposed the use of clusters of parked cars to overcome such limitations, i.e., to provide a virtual infrastructure in form of a virtual Roadside Unit (RSU) being able to fulfill all the mentioned actions. In this paper, we investigate two core problems of this system, namely the selection of appropriate gateway nodes in the virtual cluster as well as seamless handovers among such gateways required due to very limited contact times of moving cars to a single gateway. Our simulation results confirm the strength of the proposed gateway selection and handover mechanisms.

Key words: Vehicular Cloud, Virtual Micro Cloud, Clustering, Virtual Roadside Infrastructure

1. Introduction

In recent years, connected vehicles left behind research labs and field tests and started to become a reality for drivers [1]. Cars equipped with the necessary technologies (mostly based on IEEE 802.11p and LTE) are already in the market in Japan (*ITS Connect* from Toyota) and are coming to U.S. (General Motors) and European (Volkswagen) markets. Using such networking capabilities, cars exchange messages either in a distributed Vehicular Ad Hoc Network (VANET) or supported by infrastructure like Wi-Fi Access Points (APs) or LTE base stations. Such connected cars are not only the next step towards autonomous and cooperative driving, but also provide a vast amount of other potential applications, which are usually are categorized as *safety*, *efficiency*, and *infotainment*.

While safety applications tend to work well without supporting infrastructure, it has been shown that efficiency and infotainment applications greatly benefit from it [2]. This is because each application category has different application requirements. While safety applications require short delays, the others are able to work with larger delays. The same differences can be found looking at the necessary transfer volume, we see that the amount of data to be exchanged is usually rather small for safety applications. On the other hand, if we consider efficiency applications, complex traffic information data can already consist of a few kilobytes. For infotainment applications, when transmitting images or videos, the amount of transmitted data is even higher (easily a few megabytes considering modern cameras).

In order to transmit medium or large quantities of data, the contact time between cars has to be rather long, which, in general, cannot be guaranteed due to the dynamic nature of vehicular networks. In this scope, the coverage of vehicular networks has been investigated by Naboulsi and Fiore [3]. It was found that, even assuming full market penetration, the topology of a vehicular network covering a city is only connected during rush hours and is heavily fragmented during other times of the day. This has different reasons, e.g., buildings blocking radio transmissions or varying traffic conditions in suburban parts of a city.

In general, the network performance can be improved by adding roadside infrastructure support to the vehicular network. Examples include Wi-Fi APs, IEEE 802.11p Roadside Units (RSUs), or LTE/5G base stations [1, 4]. Deployed at strategically optimized locations, such infrastructure helps solving a number of issues, e.g., scheduling of traffic in a crowded area or acting as a relay node for data transmissions. If the APs are also interconnected, it is possible to transmit data over large distances without problems (essentially due to available bandwidth and guaranteed connectivity). On the downside, all types of network infrastructure require initial deployment and operational

*Corresponding author.
Email addresses: `hagenauer@ccs-labs.org` (Florian Hagenauer), `sommer@ccs-labs.org` (Christoph Sommer), `ta-higuchi@jp.toyota-itc.com` (Takamasa Higuchi), `onur@us.toyota-itc.com` (Onur Altintas), `dressler@ccs-labs.org` (Falko Dressler)

Article published in Elsevier Ad Hoc Networks. (available online 10.1016/j.adhoc.2018.05.014)

costs.

The most critical remaining problem is coverage. Even if Wi-Fi APs or LTE networks are in place, there is no guarantee for perfect coverage – even many Western countries like Germany suffer from quite fragmented LTE coverage all over the country. Furthermore, buildings may block signals and large numbers of nodes may not receive the same network performance compared to a small number. When talking about Wi-Fi APs, which are often sparsely deployed in a city, the contact time between the AP and a driving car may be the limiting factor. This is particularly relevant when transferring larger files or using complex communication protocols. In all these scenarios where the infrastructure network is fragmented, the data must be split into multiple messages (either for protocol purposes or fragments of a larger file) which induces longer transmission times.

In earlier work, we proposed the concept of vehicular micro clouds to overcome these limitations [5]. First ideas on establishing and maintaining vehicular cloud solutions have been proposed at the same time by Gerla [6], Dressler et al. [7], Lee et al. [8]. The core idea is to use either clusters of parked cars [9] or even the establishment of clouds of moving cars [10] to extend coverage of the vehicular network without additional infrastructure support. Such a fully distributed solution is provider-independent and builds on the existing capabilities of connected cars. Furthermore, if needed, it can be integrated into existing infrastructure.

In this paper, we concentrate on the parked cars clustering solution, trying to answer selected research questions that have been identified to have a significant impact on the overall system performance [9]. The core idea is to use the communication capabilities of cars not only when they drive, but also when they are parked. Using clustering techniques, cars get organized into groups acting as a single virtual node [11, 12, 13]. This way, we are able to provide additional (virtual) roadside infrastructure. Such a cluster has also a very high chance to provide an uplink to the Internet either via a Wi-Fi AP or if one of the cars has a cellular connection.

We build upon our earlier conference paper [9] and now study the following core components of the vehicular micro cloud in more detail:

- *Cluster formation* is a central part of using such micro clouds. This includes setup and maintenance phases. Such clusters of parked cars bring the advantage of roadside infrastructure without the need of deploying additional Wi-Fi APs or LTE base stations.

- *Gateways selection* helps significantly reducing the channel load. The idea is to always select only a subset of all parked cars as gateways. In our evaluation, we show that this, while not influencing much the success rate of data delivery between the parked car cluster and driving cars, significantly reduces the load on the network channel.

- *Handover mechanisms* help maintaining connectivity between driving cars and the virtual infrastructure. Cars driving by the cluster will be in contact range of a single gateway for only a limited time period. Handovers to better suited gateways allow the driving car to maintain a connection and, therefore, enables the transfer of larger amounts of data and the usage of more complex protocols.

The rest of the paper is organized as follows. Section 2 summarizes the related work in the context of infrastructure support for vehicular networks, gateway selection, and handover techniques. Section 3 describes the developed concepts for clustering, gateway selection, and handover in detail. Section 4 finally explores the performance of all these algorithms and concepts in order to derive insights on the overall system feasibility and performance. Section 5 concludes the paper with a summary and a brief outlook.

2. Related Work

The proposed concept combines various topics ranging from RSUs to clustering to gateway selection, and to handover handling. We start outlining the need for (virtual) roadside infrastructure and the use of parked cars for this role. This is followed by a discussion of clustering protocols for vehicular networks. Finally, we discuss approaches related to the two main improvements in our architecture compared to the conference paper [9], namely gateway selection and handover handling.

2.1. Towards Virtual RSUs

RSUs are envisioned to improve the stability of vehicular networks and potentially provide an uplink to the Internet or a centralized cloud server. Besides these core networking functionalities, such infrastructure elements also help overcoming low penetration rates of cars equipped with the respective network modules in early deployment stages [2].

A typical example of such an application is the *ROAMER* routing protocol by Mershad et al. [14]. While transmissions to nodes in the vicinity are handled by means of broadcasts, RSUs help bridging longer communication distances. However, the authors also face problems similar to those that are mentioned, for example, by Naboulsi and Fiore [3] and note that additional measures are required in sparse networks.

A less general example has been proposed by Sommer et al. [15], where parked cars take the role of an RSU to warn driving cars at intersections. The cars parked close to intersections relay safety messages from approaching cars to cars on other roads and help substantially improving the communication reliability.

An infotainment application where parked cars assist an RSU to stream data from the Internet has been proposed by Malandrino et al. [16]. In their architecture, the parked cars support the RSU by downloading and distributing videos. Through optimizations the authors conclude that the parked cars greatly improve the overall system performance.

Generally, many applications can benefit from the existence of an RSU (or a parked car mimicking such RSUs as in our proposed architecture). In this paper, we build upon these findings and propose the use of clusters of parked cars as small to medium scale virtual RSUs. We investigate the algorithms necessary to enable a smooth communication between the virtual RSU and cars driving by.

2.2. Clustering and Gateway Selection

Clustering is often named as a core strategy to maintain complex vehicular networks and to organize them in a manageable hierarchy [7, 8]. Sucasas et al. [17] explore general clustering concepts and point out that vehicular networks are one of the most promising application areas. Nevertheless, there is a lot of room for improvement regarding quality of service and high mobility scenarios.

A clustering algorithm for vehicular networks usually takes care of forming and maintaining a group of cars. The core components of this algorithm have been extensively discussed by Cooper et al. [18]. In order to form a cluster, cars are grouped based on similarity metrics including, but not limited to, speed, direction, geographic position, neighborhood, or interest. One of the main parameters in clustering for classical ad-hoc networks, the energy consumption [19], is usually not a concern in the vehicular application domain.

While most of the existing clustering algorithms use a combination of these parameters, there exist various algorithms that try to take another approach. For example, Cheng et al. [20] use evolutionary algorithms to calculate clusters and Zahidi et al. [21] rely on optimization using Integer Linear Programming. However, these approaches often rely on strong assumptions. Examples include a unit-disk communication model or a fixed topology. These constraints make it hard to anticipate their performance under more realistic conditions.

To coordinate a cluster, nodes are often required to select cluster heads [18]. These control nodes are in charge of maintaining the cluster. However, such cluster heads might not be perfect candidates for maintaining communication to outside nodes, i.e., to establish gateway functionalities.

Gateways should be a minimal set of nodes which enable a seamless connectivity to and from the cluster. A frequently used solution is the *k-barrier coverage* that has been developed in the scope of ad-hoc networks [22]. Here, a certain border region is to be covered by least k nodes. Similarly, Dai and Wu [23] select a set of gateways for routing. This is done based on neighborhood information and can be controlled by a parameter k to determine the number of gateways. Our gateway selection approach also uses geographic positions as a main parameter to provide sufficient coverage.

2.3. Handover Management

To transmit larger amounts of data, a driving car needs to connect to multiple RSUs over time as it passes along the street: connection times to a single stationary node are too short. This problem of handover also exists in our virtual roadside infrastructure, where a driving car connects to multiple gateways over time, e.g., when downloading larger amounts of data. Therefore, we need to perform a horizontal handover (changing access points).

Other than vertical handovers (that is, changes in technology), the problem of horizontal handovers in vehicular networks has only marginally explored so far. In fact, Ghosh et al. [24] note that most of the proposed communication approaches do not consider handover. This is also backed by Bali et al. [25]. They even go further and declare the question of handover management to be one of the core research problems for efficient clustering concepts in vehicular networks.

As a starting point, Ghosh et al. [26] discuss an approach for vehicular networks which relies on probabilistic handovers between overlapping transmission regions of RSUs. Focusing on a packet forwarding scheme, Huang et al. [27] also outline a handover scheme for vehicular networks. In their approach, a common ancestor of the handover candidates selects the best candidate to hand over to. This necessitates that all potential handover candidates are interconnected in a hierarchical topology, which might not always be suitable in a dynamic vehicular networking scenario. To the best of our knowledge, the approach has only been investigated in an analytical fashion and the effects of a more realistic scenario are unclear.

Similar to certain clustering approaches, Mouton et al. [28] propose a handover scheme based on a variety of parameters (direction, roads, network deployment). To make use of this approach, multiple modules have to be deployed throughout the network stack, which makes an integration into an existing stack complicated.

2.4. Car4ICT Architecture

Throughout this paper, we make use of our Car4ICT system architecture [29]. At its core, Car4ICT is a concept providing service discovery functions among smart cars in large-scale smart city environments. Any user is able to offer, discover, and use various kinds of services (e.g., data storage, Internet access, processing capabilities). The whole service management and provision process is handled by connected cars. The concept has been evaluated in urban scenarios [29] and on highways [30]. To exploit the virtual roadside infrastructure proposed in this paper, cars use Car4ICT to discover data and download it from the cluster of parked cars. Due to the extended contact time, the download of larger files and streaming becomes possible without modifications to the Car4ICT architecture.

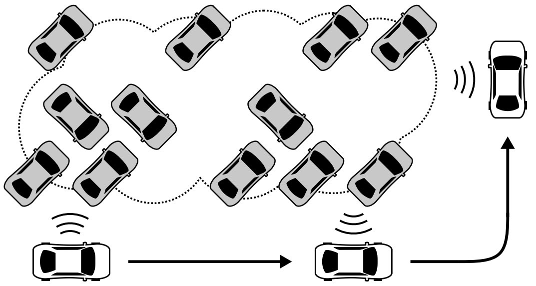

Figure 1: Schematic overview of our concept, where a driving car passes a cluster of parked cars forming virtual network infrastructure. The physical connection is handed over among gateways in the cluster as the driving car passes by.

3. Parked Cars as Virtual Network Infrastructure

Our proposed architecture aims to provide additional virtual networking infrastructure using clusters of parked cars. Driving cars are able to connect to such clusters and use services or applications provided by them. Potential services include the provisioning of storage capacity or connectivity to the Internet via an access point. In Figure 1, we can see a car driving by such a cluster of parked cars. First, in the lower left corner, it requests some content and starts downloading while driving by (lower right, upper right). As the cluster is interconnected, a connection to any of the parked cars is directly possible for accessing provided services.

Besides the basic clustering algorithm, we propose two improvements: In order to reduce the load on the wireless channel, we select a subset of parked cars to communicate with cars driving by, i.e., *gateway selection*. Furthermore, to enable long lasting connections to the cluster, we provide a mechanism for implicit seamless horizontal *handover*. This allows a car to connect to multiple cluster members over time while driving by.

3.1. Prerequisites

We assume a set of features or capabilities being available to all participating cars:

- Networking Technology: To be able to connect to the other cars, being equipped with wireless networking technology is necessary. A dedicated short-range technology is preferable, as it needs no additional central coordinator. Examples include Wi-Fi, LTE-D2D, or, as used in our evaluation, IEEE 802.11p.

- Connectivity: All parked cars should be able to reach all other cluster members using the above-mentioned networking technology either directly or using multi-hop routing. As the parked cars are not moving, standard ad-hoc routing techniques could be used. We rely on the Virtual Cord Protocol (VCP) [31], which, beside efficient routing, supports storing data in a Distributed Hash Table (DHT).

- Geographic Position: All parked and driving cars should be able to determine their geographic position. This can be done by using satellite navigation systems like GPS, GLONASS, or Galileo. This information will be used for effective handover, gateway selection, and the clustering process itself.

- Information Storage: Parked cars need to be able to store gateway and handover information. We use VCP for this purpose as well. This also provides an opportunity for applications running on top to use the distributed storage.

3.2. Clustering Algorithm

Our clustering algorithm follows the established concepts for clustering and consists of three core parts: *Setup*, *Maintenance*, and *Departure*. During the *Setup* phase, a car either joins an existing cluster or creates a new one if no cluster exists yet. Afterwards, the *Maintenance* phase periodically exchanges information in order to keep the cluster operational. Finally, if a car is about to leave the cluster, the *Departure* phase is triggered.

Setup phase: When a car stops and eventually parks, it starts listening for *cluster beacons* that are transmitted periodically by potential clusters in the surroundings. If no such beacons were received after a certain time, the car initiates a new cluster and starts periodically sending beacons itself to indicate the presence of the new cluster. If, on the other hand, the car received cluster beacons from an existing cluster, it notifies the cluster of its presence and joins the cluster. In particular, it sends its current control information (necessary clustering information as well as its geographic position) to the node in charge for gateway selection. To ease the selection of gateways, we have cars not join clusters across streets. When creating such clusters in urban environments, we will end up with two kinds of clusters: (1) clusters along a street covering a one-dimensional curve and (2) clusters in a parking lot covering a two-dimensional area.

Maintenance phase: This phase mostly consists of periodic cluster beacons sent by all cluster members. These beacons are used to detect broken network connections within the established cluster and to notify newly arriving cars of the existing cluster. If a car was selected as a gateway, it furthermore informs moving cars about the cluster by sending *access beacons*. Moving cars reacting to such a notification reply with their current control information to be able to connect to the cluster and use its services and applications.

Departure phase: If a car is about to leave the parking lot, e.g., by detecting the driver's intention to start the engine, the departure phase is initiated. The car now informs its neighbors that it is leaving the cluster. The leaving car sends all locally stored data (control information and application data) to other cluster nodes, which are now in charge of maintaining this data (this might not be necessary if the distributed storage provides explicit data replication techniques). Finally, outdated control information is now purged to ensure a properly working cluster. The departure phase operates on a time-scale of tens of seconds, which is sufficient for synchronizing even larger data sets.

In our evaluation we make use of VCP for clustering [31]. The algorithm fulfills most of the listed prerequisites by organizing all parked cars in the form of a virtual cord with coordinates (i.e., cluster positions) between 0.0 and 1.0. Messages can be sent to all members on this cord and the fully distributed algorithm supports joining and leaving of nodes. Furthermore, the cord in VCP allows to store data in a DHT where every car is in charge of a certain cord section.

3.3. Gateway Selection

If all members of the cluster (our virtual RSU) announce the presence of the cluster to moving cars (i.e., if all members act as possible gateways to the cluster), a number of problems arise:

- In a dense parking area, cars are often very close to each other. If all of them would send access beacons to announce the cluster, this would produce redundant messages.
- At the same time, having all cars periodically send beacons would put an unnecessary load on the network.
- In a parking lot, cars can be surrounded by other parked cars taking part in the cluster. Therefore, if a user connects to such a gateway, he would get a worse connection compared to a car at the border of the parking lot.

To avoid sending redundant beacons, announcing badly positioned cars, and reducing the load on the wireless channel, we propose to select only a subset of cars as *gateways*. Only these gateways are then supposed to send beacons announcing the presence of the cluster. As a rule of thumb, only as many cars should be selected as gateway as absolutely necessary and these gateways should be on the perimeter of the cluster.

More formally, we want to cover a certain region by a set of gateways $\mathbb{G}$ so that it reduces the n-covered regions ($n \geq 2$) to a minimum. An optimal solution to this problem would need detailed coverage information from all parked cars. This could easily be achieved when assuming a fixed transmission distance, i.e., a unit-disk model. In reality, however, it is close to impossible to estimate all coverage regions as they change both spatially and temporally. Therefore, we calculate an approximation of $\mathbb{G}$ which only needs information that is readily available in the VCP network: (1) a 1-hop neighbor list of all parked cars and (2) the geographic positions of all nodes. Based on this information a coordinator node can calculate all required gateways. Such a coordinator is a dedicated node

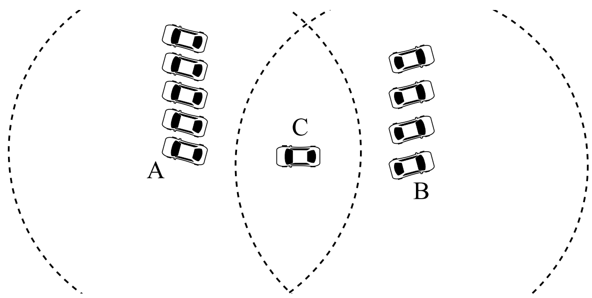

Figure 2: The stopping criteria for the gateway selection. If cars A and B are already a gateway and C is a candidate, C will not be added.

Algorithm 1 Gateway selection along a curve

Input: $\mathbb{N}$, a set of nodes
Output: $\mathbb{G}$, the set of nodes selected as gateways

$n \leftarrow \operatorname{argmax}_{n \in \mathbb{N}} \sum_{n_x \in \mathbb{N}} \text{distance}(n_x, n)$
$\mathbb{G} \leftarrow \{n\}$
$\mathbb{N} \leftarrow \mathbb{N} \setminus \{n\}$
while $\mathbb{N} \neq \varnothing$ **do**
    $n \leftarrow \operatorname{argmax}_{n \in \mathbb{N}} \sum_{n_g \in \mathbb{G}} \text{distance}(n_g, n)$
    if $|\text{neighbors}(n) \cap \mathbb{G}| \geq 2$ **then**
        break
    end if
    $\mathbb{G} \leftarrow \mathbb{G} \cup \{n\}$
    $\mathbb{N} \leftarrow \mathbb{N} \setminus \{n\}$
end while
return $\mathbb{G}$

in the cluster which is known to all nodes (in case of VCPs, we select VCP position 0.0). All necessary information for gateway selection is periodically sent by all parked cars to the coordinator. Our gateway selection now consists of two parts: (1) gateway selection along a curve (Algorithm 1) and (2) gateway selection in an area (Algorithm 2).

Algorithm 1 takes a set of nodes $\mathbb{N}$ as input and returns the set of selected gateways $\mathbb{G}$. If only this algorithm is run, $\mathbb{N}$ equals the set of all cars in the cluster, but generally it can be any subset of the nodes. First, the algorithm takes the node farthest away from all others and adds it to $\mathbb{G}$. Second, in a loop the node farthest away from all nodes in $\mathbb{G}$ is selected. Third, the number of neighbors which are already part of $\mathbb{G}$ is counted. If the number is 2 or higher, the position is already covered by at least two gateways and the gateway selection is done. Otherwise, the node is added to $\mathbb{G}$ and the next node farthest away is selected. This concept is depicted in Figure 2, where cars A and B (with their communication range shown as a simplified unit-disk) are selected as gateways. If now car C becomes a candidate for a gateway, it is rejected as two of C's neighbors are already a gateway.

If cars are parked in a parking lot, we cannot directly apply Algorithm 1 as it would select undesired gateways

Algorithm 2 Gateway selection in an area

Input: $\mathbb{N}$, the set of all nodes in the cluster
Input: Δ, digging parameter
Output: $\mathbb{G}$, the set of nodes selected as gateways

$\mathbb{C} \leftarrow$ edges of the convex hull of $\mathbb{N}$
for all $c \in \mathbb{C}$ **do**
  $c_0 \leftarrow$ start of c
  $c_1 \leftarrow$ end of c
  find $p \in \mathbb{N} \setminus \{\text{points on } \mathbb{C}\}$ closest to edge c
  $\varsigma = \dfrac{\text{distance}(c_0, c_1)}{\min\{\text{distance}(c_0, p), \text{distance}(p, c_1)\}}$
  if $\varsigma > \Delta$ **then**
    remove c from $\mathbb{C}$
    add edges (c_0, p) and (p, c_1) to $\mathbb{C}$
  end if
end for
$\mathbb{G} \leftarrow$ Algorithm 1 with $\{\text{points on } \mathbb{C}\}$ as input
return $\mathbb{G}$

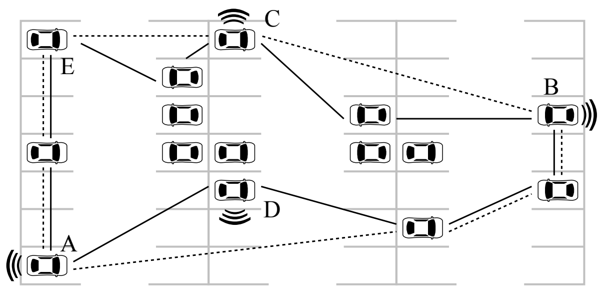

Figure 3: Outline of the proposed algorithm for gateway selection. First, the convex hull is computed (dashed line). Second, additional nodes are added based on the concave hull (solid line). Third, based on Algorithm 1, the gateway selection is performed.

inside the cluster. Therefore, we first perform Algorithm 2, which does not directly select gateways but rather explores the boundaries of the cluster. Selecting only cars on the convex hull of the cluster would potentially lead to large parts of the perimeter without a gateway and therefore insufficient coverage. To circumvent this, we not only calculate the convex hull, but also the concave hull. We use the algorithm presented by Park and Oh [32], which provides a digging parameter Δ. Using this parameter, the number of selected cars can be tuned. In the end, Algorithm 2 yields a set of cars on the perimeter of the cluster, which form a curve. This set is then used as input for Algorithm 1, which yields the final set of gateways.

Figure 3 shows an example of the gateway selection process in a parking lot. In the first step, the convex hull is calculated to select the perimeter of the cluster (dotted line). The second step selects the concave hull by adding additional cars and increasing the perimeter (solid line). This result of Algorithm 2 is then used as input of Algorithm 1, which selects the actual gateways. First, the car which is on average farthest away from all other cars is selected as a gateway (car A). Then, one after another, the cars farthest away from all gateways are selected: B, C, D. Finally, car is E is selected, however, in the example it is able to reach cars A and C and therefore the exit condition is fulfilled and the algorithm is stopped. The selected gateways are A, B, C, and D (marked with wireless signals to indicate that they are still communicating with cars driving by the cluster).

3.4. Handover

One of the reasons for creating virtual roadside infrastructure is the longer contact time between moving cars and the cluster. This allows to transfer more data and use more sophisticated services. A key enabler is to connect to multiple different gateways over time. Our handover mechanism helps reducing the overhead of these connections to the minimum and providing a near-seamless handover experience.

Generally speaking, we talk about two kinds of connections: (1) downloading data from the cluster to the driving car and (2) uploading data from the driving car to the cluster. For the download process, we require the moving cars to reply to the beacons sent by the gateways. Based on these replies the gateways store the following information in DHT:

- The receiving gateway identifier to later identify the best gateway.
- The destination identifier sent by the driving car. This should uniquely identify the driving car to the cluster, but can be different for other clusters.
- The time of contact, i.e., the time the reply to the beacon was received by the gateway.
- The distance between car and gateway as a metric for distance.
- The Signal to Interference and Noise Ratio (SINR) as a connectivity metric.

If a cluster member is about to send data to a driving car c, it starts with gathering all gateway entries for it from the distributed storage. Based on this data, it assigns a weight w_i to all potential gateways i. We calculate w as

$$w_i = \alpha \times d + \beta \times t - \gamma \times s \;, \tag{1}$$

where d is the distance in meters, t the time difference since the last contact in seconds, and s the SINR. The three coefficients α, β, and γ are used to weight the parameters, assuming $\alpha + \beta + \gamma = 1$. These weights are necessary as the range of potential values for the parameters is very different (e.g., the distance can be roughly between 0 m and 500 m, the contact time is between 0 s and 5 s). Finally, the gateway with the smallest weight is selected for data transmissions to the moving car. This procedure is performed for every new packet or every new set of packets sent and constitutes a passive handover. Thus, the gateways themselves do not need to perform any active handover.

The process of uploading is similar. As the car driving by receives access beacons from the cluster it is able to store the access information. If there is now the need to upload data, it can choose an arbitrary gateway, which it received such information from, and send the data to it. The gateway then looks up the actual receiving node and sends the data to it. As the cluster is interconnected, this works for all gateways.

4. Evaluation

In the first part of the evaluation, we look into the effects gateway selection and handover have on the performance of the virtual network infrastructure. In the second part, we investigate the proposed infrastructure in two real-world scenarios showing its advantages and limits. All simulations were performed using the Veins [33] simulation framework together with the Veins LTE extension [34]. This simulator couples (vehicular) network simulation with road traffic micro simulation, which allows the use of detailed networking models together with realistic road scenarios.

We focused on the following metrics in the performance evaluation:

- Success rate (fraction of frames sent that were received): This metric helps studying various aspects related to the transfer of data between a moving car and the vehicular micro cloud. First, it can be used to see if and to what degree the transfer of a single large file is successful. Second, it can be used to investigate streaming.

- MAC busy-fraction (fraction of time that a MAC layer would consider the channel busy): We use this metric to evaluate how congested the wireless channel is in our network. This is not only an indicator how well the protocol works, but also shows if there is capacity left for other applications (e.g., safety applications).

- Number of handovers: When a car downloads data from the cluster it needs to potentially perform multiple handovers. A high number of handovers indicates too many potential gateways which in turn suggests unfitting gateway selection parameters.

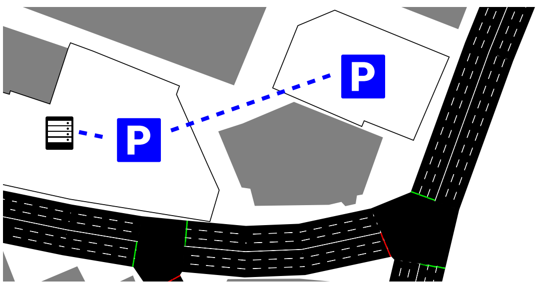

Figure 4: Scenario 1: First real-world simulation scenario with two parking lots along a busy intersection close to Luxembourg main station.

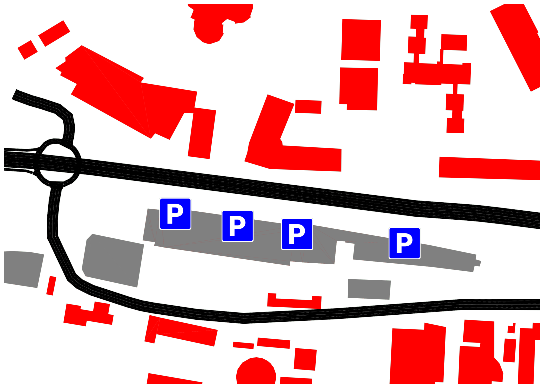

Figure 5: Scenario 2: Second real-world simulation scenario with a large parking lot along a highway.

To evaluate our algorithm, we use three scenarios, a handcrafted artificial one and two real-world scenarios. Both real-world scenarios have been extracted from the Luxembourg SUMO Traffic (LuST) scenario [35]. In the artificial scenario, 50 parked cars were placed along a street using a 30 m spacing between every two cars. Using this scenario, it was possible to investigate gateway selection

Table 1: Simulation Parameters

Parameter	Value
IVC technology	IEEE 802.11p
Channel	5.89 GHz
Transmission power	20 mW
Bandwidth	10 MHz
DHT protocol	Virtual Cord Protocol
Routing	greedy VCP routing
Gateway digging parameter Δ	1
Artificial Scenario	
Parked cars	50
Driving cars	on average 1 every 10 s
Amount of requested data	128 kByte
Simulation duration	360 s of traffic
Repetitions	32
Real-world Scenario 1	
Parked cars	11 west, 9 east
Car4ICT service producer	one at west parking lot
Fraction of equipped vehicles	0.25, 0.5, 0.75, 0.1
Simulated time	600 s during rush hour
Real-world Scenario 2	
Parked cars	40 cars
Car4ICT service producer	on at the parking lot
Fraction of equipped vehicles	0.25, 0.5, 0.75, 0.1
Simulated time	600 s during rush hour

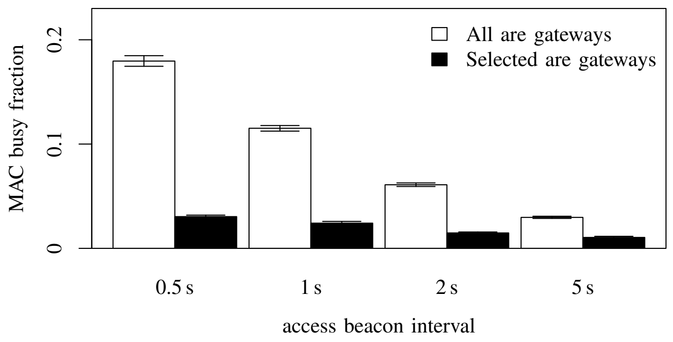

Figure 6: Fraction of time the MAC is observed busy in the artificial scenario. Shown are different access beacon intervals for s active gateway selection as well as for the baseline option where all parked cars act as gateways.

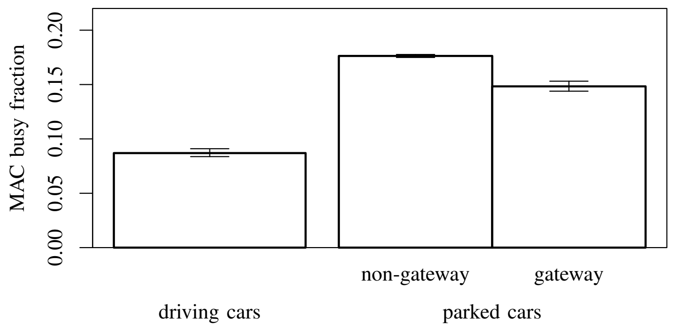

Figure 7: Showing the MAC busy fraction comparing driving and parked cars. Furthermore, there is a distinction between gateways and CMs.

and the handover mechanism.

The two realistic scenarios cover different areas of Luxembourg and help understanding our architecture in different situations. Being close to the train station at an intersection, the first real-world scenario (cf. Figure 4) has two parking lots that are connected to each other and act as a single cluster. As there is an intersection close-by, cars drive past the cluster and are potentially able to have longer connections to the cluster while waiting at the traffic light. Inside the cluster, a single node provides data that is requested and downloaded by the passing cars. The second real-world scenario is a parking lot alongside a freeway ending in the city (cf. Figure 5). Commuters parking their car here and continue by public transportation. As the highway in the north allows for higher speeds, cars do not perceive long contact times with gateways. This makes the scenario particularly interesting for studying rapid and frequent handovers in a realistic environment.

For all simulations, we used the Car4ICT framework (cf. Section 2.4), which helps assigning service provider and consumer roles to the cars and also provides means for data exchange between them. All relevant simulation parameters are summarized in Table 1.

4.1. Gateway Selection Performance

The first evaluated concept is gateway selection. For this, we used the artificial scenario with 50 parked cars. In this scenario, we were particularly interested in the overhead by exchanged control messages and beacons. Thus, we explored the system performance with and without enabled gateway selection.

In Figure 6, we can see that gateway selection reduces the load on the channel significantly. If there is no active gateway selection and every car sends an access beacon every 0.5 s, the load on the channel is already close to 20 %. With gateway selection, the load is substantially reduced to less than 5 %. While the absolute values are not so high for the other broadcast intervals, the difference is still clearly observable.

In Figure 7, we show the channel load as observed by three different types of cars (please note that the application traffic has been enabled for this experiment). Driving cars have the lowest channel load, which is because they are not always in range of the cluster. The key takeaway is that gateways observe a lower channel load compared to other parked cars, which have not been selected as gateways. This is an effect of the gateway selection algorithm. In fact, gateways are selected to be as far away from other cars as possible, particularly in the one-dimensional case. Therefore, gateways receive less transmissions compared to other parked cars.

As can be seen, our proposed handover mechanism works better, the more up-to-date the information is. In order not to impair the decision process, while still keeping a low channel load, we selected a broadcast interval of 1 s for all further simulation experiments.

4.2. Handover Performance

To investigate the handover performance, we set up the artificial scenario to support streaming. The car offering data sends 1 kByte every 0.05 s via the virtual infrastructure to the driving car. The simulations were performed for two configurations: (1) all parked cars being gateways and (2) for a subset of cars selected by our proposed gateway selection algorithm.

We investigate two metrics, the number of handovers and the amount of received data. We plot both metrics in Figure 8 for different weights of α, β, and, implicitly, γ (as $\alpha + \beta + \gamma = 1$). Generally, it holds that the higher a bar, the more handovers happened and the darker a bar, the more data has been received while the moving car was driving by.

The results lead to some interesting observations:

- First, when only performing handover based on time (i.e., $\beta = 1$), the most handovers are performed. Whenever a car is in range of multiple gateways, it will regularly switch between those gateways. This is due to the fact that every response to access beacons triggers the selection of a new gateway. Furthermore,

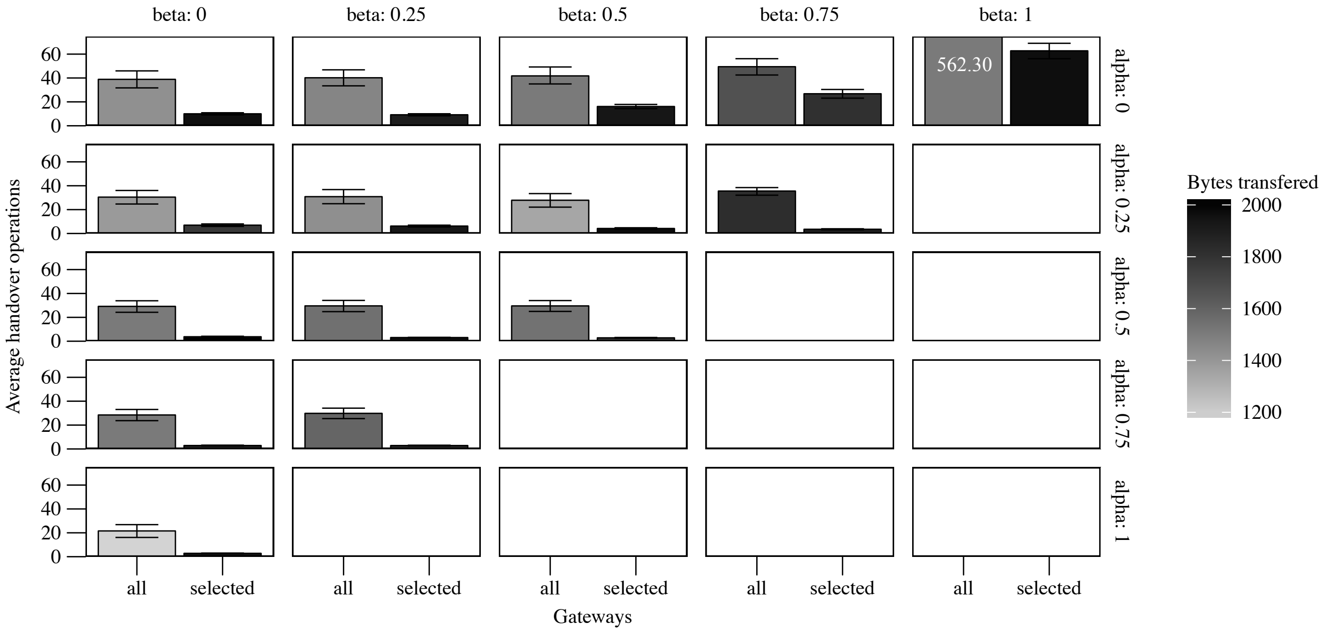

Figure 8: Number of handovers needed for different values of α and β. The coloring indicates how many bytes have been received when streaming data.

by looking at the same configurations regarding gateway selection (i.e., all parked cars are gateways or gateway selection is enabled), it can be seen that the number of handovers does not influence the number of bytes received. This is because the best gateway is selected every time a cluster member intends to send data to a driving car.

- Second, if distance is not included in the weight formula (i.e., $\alpha = 0$), the number of handovers is higher compared to other cases. This indicates that SINR does play a different role compared to distance, as otherwise the first row (except for $\beta = 1$) would be similar to the diagonal (except for $\beta = 1$ and $\alpha = 1$). The fact that the number of handovers differs, shows that SINR covers additional corner cases where a car is close to the gateway while having nevertheless a poor connectivity.

- Third, when all cars are potential gateways, as expected, the number of handovers is significantly higher. Also, the number of potentially bad gateways increases, which leads to more lost packets and in turn less received data when a bad gateway is chosen. This happens particularly when putting emphasize on a single parameter (e.g., $\beta = 0$ or $\alpha = 1$).

While various combinations work well (e.g., even the ones which only rely on a single parameter), we have chosen not to rely on the ones considering just a single parameter. All of those have downsides when covering corner cases as they enable outliers to be selected as gateways (e.g., for β a gateway being far away but having received the latest message). If those are left out, there are two well working

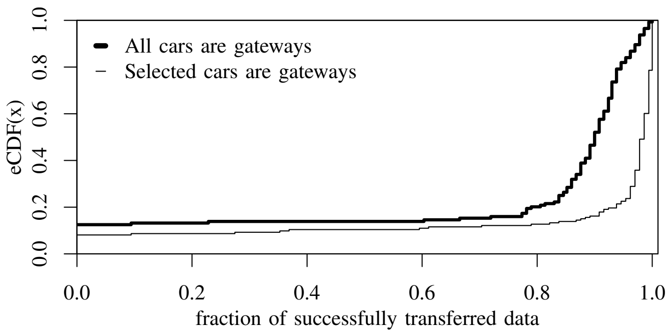

Figure 9: Fraction of data successfully transferred data between the sender (parked car) and the receiver (moving car requesting the data). We also distinguish between all cars act as gateways and the gateway selection algorithm.

solutions: ($\alpha = 0.5$, $\beta = 0$, $\gamma = 0.5$) and ($\alpha = 0.25$, $\beta = 0.5$, $\gamma = 0.25$). Both have a low number of handovers as well as a high number of received fragments during the streaming process.

We further investigated what happens when a car wants to transmit a fixed amount of data and how successful this would be. Instead of sending a fragment every 0.05 s, all fragments of the 128 kByte payload are now sent immediately. Furthermore, we now have a steady stream of cars passing by instead of a single car. This lets us investigate the success when we enable handover and gateway selection compared to having no gateway selection and all cars are gateways. The results shown in Figure 9 indicate that gateway selection comes at some cost. Even though the impact is not negligible, we suggest that the small trade-off in performance is acceptable compared to the huge gains

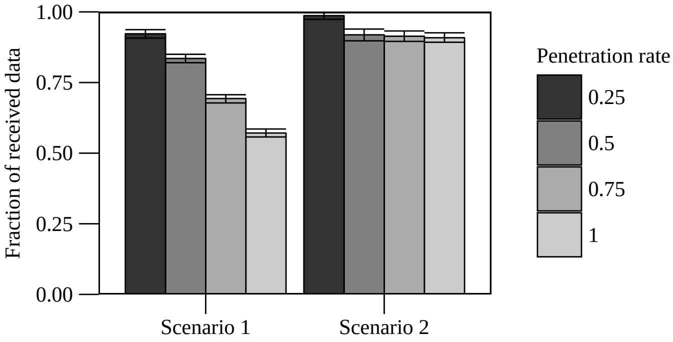

Figure 10: Fraction of successfully transmitted data (512 kByte) in both scenarios and different penetration rates including 0.95 % confidence intervals.

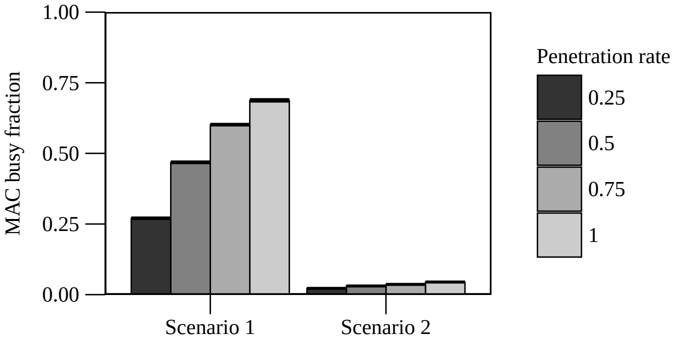

Figure 11: MAC busy fraction when transmitting 512 kByte in both scenarios for various penetration rates including 0.95 % confidence intervals.

on reduced channel load.

4.3. Real-World System Performance

After establishing the system performance in the artificial scenario, our next step is to assess the performance in some two real-world scenarios. For this, we investigated two real-world scenarios (cf. Figures 4 and 5) using two different kinds of applications: (1) cars downloading 512 kByte and the server sending the data immediately and (2) cars receiving a constant stream of data while the server sends 0.25 kByte every 0.2 s.

The results of downloading 512 kByte can be seen in Figure 10. As can be seen, the amount of received data decreases with increasing penetration rate. The effect is much stronger in the first scenario compared to the second scenario where there is only a slight decrease. The drop observed from a penetration rate of 25 % to 50 % in the second scenario is due to an increased amount of received erroneous frames due to collisions. We did not observe such a rise of erroneous frames from 50 % to 75 % and from 75 % to 100 %, respectively. The big drops in the first scenario indicate an overloaded channel for higher penetration rates. We also observed no significant difference when disabling handover (data not shown). This is because the download of 512 kByte is most of the time handled by a single gateway.

The MAC busy fraction can be seen in Figure 11. It confirms that the load on the channel in the first scenario becomes too high for higher penetration rates. While the busy fraction for a penetration rate of 25 % is still well below 50 %, it already is going close to 50 % for a penetration rate of 50 %. For the two higher penetration rates, it increases even further. This is due to the high number of cars in the simulation requesting data and in turn putting too much load on the channel.

While there was not much difference related to the handover mechanism when downloading 512 kByte, there is a significant difference when data is streamed to the driving cars. This can be seen in Figure 12, particularly when looking at the results for the second scenario. Having handover enabled increases the amount of data received by 15–30 % depending on the penetration rate.

These observations lead to two key takeaways: (1) the amount of received data in the first scenario is much lower compared to the second one and (2) handovers improve the performance of the system. Both results can be explained with the scenario itself, more specifically with the contact time between the cluster and the driving cars. While the length of the parking lot is roughly 150 m in the first scenario, it covers roughly 500 m in the second one. This means the cars are able to receive more data in the second scenario where potential contact time is longer (e.g., it takes 40 s to transmit 500 kByte while it takes 80 s to transmit 100 kByte). Due to the shorter contact time, there is also barely any effect of handover in the first scenario.

Finally, we can see in Figure 13 why the amount of received data is decreasing much stronger in the first scenario compared to the second one. Even though the MAC busy fraction in the second scenario grows slightly, it generally stays very low. Due to the higher number of cars in the scenario, the load on the channel grows rapidly in the first scenario and becomes too high for useful data transfers. This leaves room for future work regarding load balancing in rush hour scenarios.

5. Conclusion

In this paper, we investigated the concept of virtual vehicular network infrastructure. In particular, we propose clusters of parked cars to act as virtual Roadside Units (RSUs). This is especially useful as vehicular networks may be severely fragmented – especially if the penetration rate of equipped vehicles is low during initial deployment. In this scope, we focused on two core research questions: (1) the selection of gateway nodes to connect moving cars to the cluster and (2) a mechanism for seamless handover between gateways of the same cluster. We investigated the performance of the system in an artificial as well as in two close-to-real-world scenarios. Our results clearly show the benefits of using such gateway nodes to reduce the channel load. Furthermore, the handover mechanism allows to maintain a longer lasting connection to the cluster and, in turn, to exchange more data. For future work, we aim to investigate load balancing mechanisms to better cope with situations with an unusually high network load.

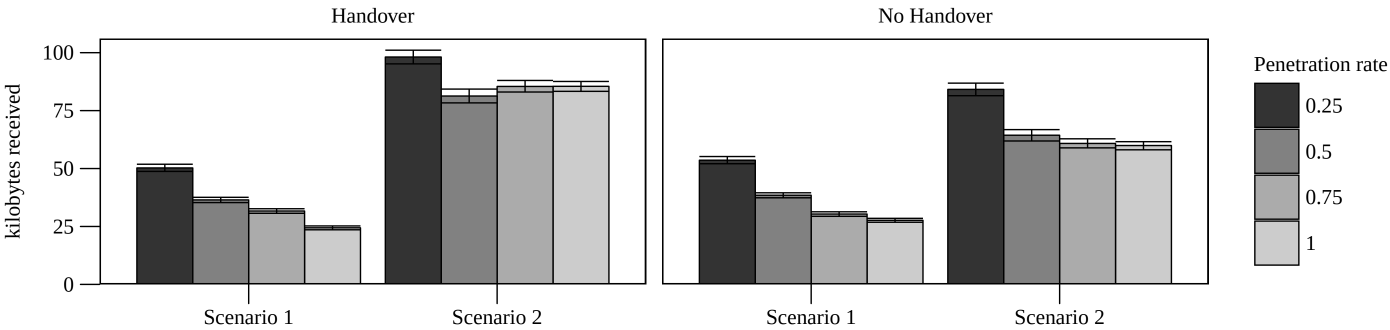

Figure 12: Amount of received streaming data (0.25 kByte every 0.2 s) in both scenarios and different penetration rates including 0.95 % confidence intervals.

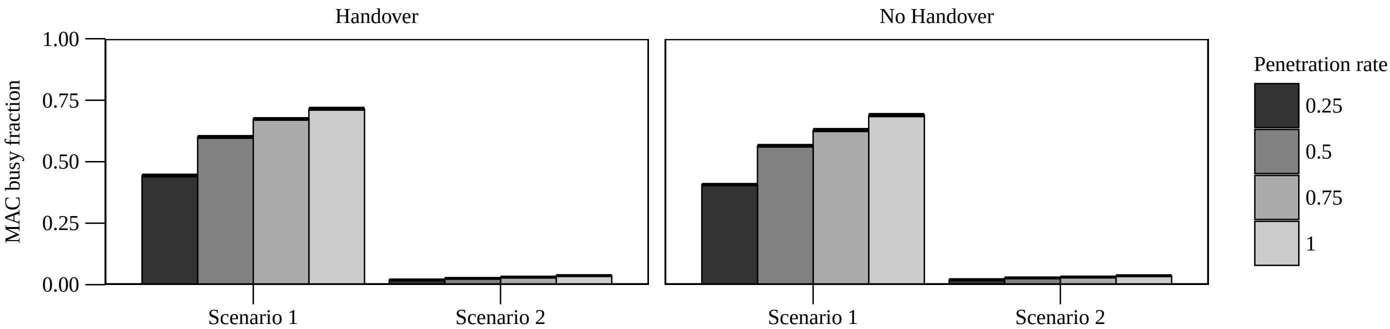

Figure 13: MAC busy fraction when streaming 1 kByte every 0.05 s in both scenarios for various penetration rates including 95 % confidence intervals.

References

[1] Dressler, F., Hartenstein, H., Altintas, O., Tonguz, O.K.. Inter-Vehicle Communication – Quo Vadis. IEEE Communications Magazine 2014;52(6):170–177. doi:10.1109/MCOM.2014.6829960.

[2] Sommer, C., Dressler, F.. Vehicular Networking. Cambridge University Press; 2014. ISBN 978-1-107-04671-9. doi:10.1017/CBO9781107110649.

[3] Naboulsi, D., Fiore, M.. Characterizing the Instantaneous Connectivity of Large-Scale Urban Vehicular Networks. IEEE Transactions on Mobile Computing 2017;16(5):1272–1286. doi:10.1109/TMC.2016.2591527.

[4] Zheng, K., Zheng, Q., Chatzimisios, P., Xiang, W., Zhou, Y.. Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions. IEEE Communications Surveys and Tutorials 2015;17(4):2377–2396. doi:10.1109/COMST.2015.2440103.

[5] Higuchi, T., Joy, J., Dressler, F., Gerla, M., Altintas, O.. On the Feasibility of Vehicular Micro Clouds. In: 9th IEEE Vehicular Networking Conference (VNC 2017). Torino, Italy: IEEE; 2017, p. 179–182.

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

[7] Dressler, F., Handle, P., Sommer, C.. 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). Philadelphia, PA: ACM; 2014, p. 11–18. doi:10.1145/2633661.2633671.

[8] Lee, E., Lee, E.K., Gerla, M., Oh, S.. Vehicular Cloud Networking: Architecture and Design Principles. IEEE Communications Magazine 2014;52(2):148–155. doi:10.1109/MCOM.2014.6736756.

[9] Hagenauer, F., Sommer, C., Higuchi, T., Altintas, O., Dressler, F.. Parked Cars as Virtual Network Infrastructure: Enabling Stable V2I Access for Long-Lasting Data Flows. In: 23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017), 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2017). Snowbird, UT: ACM; 2017, p. 57–64. doi:10.1145/3131944.3131952.

[10] Hagenauer, F., Sommer, C., Higuchi, T., Altintas, O., Dressler, F.. Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection. In: 23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017), 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2017). Snowbird, UT: ACM; 2017, p. 31–35. doi:10.1145/3131944.3133937.

[11] Khakpour, S., Pazzi, R.W., El-Khatib, K.. Using clustering for target tracking in vehicular ad hoc networks. Elsevier Vehicular Communications 2017;9:83–96. doi:10.1016/j.vehcom.2017.02.002.

[12] Ucar, S., Ergen, S.C., Ozkasap, O.. Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination. IEEE Transactions on Vehicular Technology 2016;65(4):2621–2636. doi:10.1109/TVT.2015.2421277.

[13] Hagenauer, F., Sommer, C., Higuchi, T., Altintas, O., Dressler, F.. Using Clusters of Parked Cars as Virtual Vehicular Network Infrastructure. In: 8th IEEE Vehicular Networking Conference (VNC 2016), Poster Session. Columbus, OH: IEEE; 2016, p. 126–127. doi:10.1109/VNC.2016.7835943.

[14] Mershad, K., Artail, H., Gerla, M.. ROAMER: Roadside Units as message routers in VANETs. Elsevier Ad Hoc Networks 2012;10(3):479–496. doi:10.1016/j.adhoc.2011.09.001.

[15] Sommer, C., Eckhoff, D., Dressler, F.. IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation. IEEE Transactions on Mobile Computing 2014;13(8):1733–1745. doi:10.1109/TMC.2013.80.

[16] Malandrino, F., Casetti, C., Chiasserini, C.F., Sommer, C., Dressler, F.. The Role of Parked Cars in Content Downloading for Vehicular Networks. IEEE Transactions on Vehicular Technology 2014;63(9):4606–4617. doi:10.1109/TVT.2014.2316645.

[17] Sucasas, V., Radwan, A., Marques, H., Rodriguez, J., Vahid, S., Tafazolli, R.. A survey on clustering techniques for cooperative wireless networks. Elsevier Ad Hoc Networks 2016;47:53–81. doi:10.1016/j.adhoc.2016.04.008.

[18] Cooper, C., Franklin, D., Ros, M., Safaei, F., Abolhasan, M.. A Comparative Survey of VANET Clustering Techniques. IEEE Communications Surveys & Tutorials 2017;19(1):657–681. doi:10.1109/COMST.2016.2611524.

[19] Afsar, M.M., Tayarani-N, M.H.. Clustering in sensor networks: A literature survey. Elsevier Journal of Network and Computer Applications 2014;46:198–226. doi:10.1016/j.jnca.2014.09.005.

[20] Cheng, H., Cao, J., Wang, X., Das, S.K., Yang, S.. Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility. Wiley Wireless Communications and Mobile Computing (WCMC) 2009;9(6):759–771. doi:10.1002/wcm.627.

[21] Zahidi, S.Z.H., Aloul, F., Sagahyroon, A., El-Hajj, W.. Optimizing Complex Cluster Formation in MANETs Using SAT/ILP Techniques. IEEE Sensors Journal 2013;13(6):2400–2412. doi:10.1109/JSEN.2013.2254234.

[22] Kumar, S., Lai, T.H., Arora, A.. Barrier coverage with wireless sensors. ACM/Springer Wireless Networks (WINET) 2007;13(6):817–834. doi:10.1007/s11276-006-9856-0.

[23] Dai, F., Wu, J.. Distributed dominant pruning in ad hoc networks. In: IEEE International Conference on Communications (ICC 2013). Anchorage, AK: IEEE; 2003, p. 353–357. doi:10.1109/ICC.2003.1204198.

[24] Ghosh, A., Vardhan, V., Mapp, G., Gemikonakli, O., Loo, J.. Providing ubiquitous communication using road-side units in VANET systems: Unveiling the challenges. In: 13th International Conference on ITS Telecommunications (ITST 2013). 2013, p. 74–79. doi:10.1109/ITST.2013.6685524.

[25] Bali, R.S., Kumar, N., Rodrigues, J.J.. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. Elsevier Vehicular Communications 2014;1(3):134–152. doi:10.1016/j.vehcom.2014.05.004.

[26] Ghosh, A., Paranthaman, V., Mapp, G., Gemikonakli, O., Loo, J.. Enabling seamless V2I communications: toward developing cooperative automotive applications in VANET systems. IEEE Communications Magazine 2015;53(12):80–86. doi:10.1109/MCOM.2015.7355570.

[27] Huang, C.M., Chiang, M.S., Hsu, T.H.. PFC: A packet forwarding control scheme for vehicle handover over the ITS networks. Elsevier Computer Communications, Special Issue on Mobility Protocols for ITS/VANET 2008;31(12):2815–2826. doi:10.1016/j.comcom.2007.12.010.

[28] Mouton, M., Castignani, G., Frank, R., Engel, T.. Enabling vehicular mobility in city-wide IEEE 802.11 networks through predictive handovers. Elsevier Vehicular Communications 2015;2(2):59–69. doi:10.1016/j.vehcom.2015.02.001.

[29] Altintas, O., Dressler, F., Hagenauer, F., Matsumoto, M., Sepulcre, M., Sommer, C.. Making Cars a Main ICT Resource in Smart Cities. In: 34th IEEE Conference on Computer Communications (INFOCOM 2015), International Workshop on Smart Cities and Urban Informatics (SmartCity 2015). Hong Kong, China: IEEE; 2015, p. 654–659. doi:10.1109/INFCOMW.2015.7179448.

[30] Hagenauer, F., Sommer, C., Onishi, R., Wilhelm, M., Dressler, F., Altintas, O.. Interconnecting Smart Cities by Vehicles: How feasible is it? In: 35th IEEE Conference on Computer Communications (INFOCOM 2016), International Workshop on Smart Cities and Urban Computing (SmartCity 2016). San Francisco, CA: IEEE; 2016, p. 788–793. doi:10.1109/INFCOMW.2016.7562184.

[31] Awad, A., German, R., Dressler, F.. Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks. IEEE Transactions on Mobile Computing 2011;10(9):1214–1226. doi:10.1109/TMC.2010.218.

[32] Park, J.S., Oh, S.J.. A new concave hull algorithm and concaveness measure for n-dimensional datasets. Academia Sinica Journal of Information Science and Engineering 2012;28(3):587–600. doi:10.1688/JISE.2012.28.5.10.

[33] Sommer, C., German, R., Dressler, F.. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing 2011;10(1):3–15. doi:10.1109/TMC.2010.133.

[34] Hagenauer, F., Dressler, F., Sommer, C.. A Simulator for Heterogeneous Vehicular Networks. In: 6th IEEE Vehicular Networking Conference (VNC 2014), Poster Session. Paderborn, Germany: IEEE; 2014, p. 185–186. doi:10.1109/VNC.2014.7013339.

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