Parked Cars as Virtual Network Infrastructure: Enabling Stable V2I Access for Long-Lasting Data Flows

Florian Hagenauer
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
hagenauer@ccs-labs.org

Christoph Sommer
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
sommer@ccs-labs.org

Takamasa Higuchi
Toyota InfoTechnology Center, Co., Ltd.
Tokyo, Japan
ta-higuchi@jp.toyota-itc.com

Onur Altintas
Toyota InfoTechnology Center, USA, Inc.
Mountain View, CA, USA
onur@us.toyota-itc.com

Falko Dressler
Heinz Nixdorf Institute
Dept. of Computer Science
Paderborn University
Paderborn, Germany
dressler@ccs-labs.org

ABSTRACT

Vehicular networks are envisioned to cover various use cases from safety related applications to infotainment. While there exist standardized solutions for exchanging single messages in a geographical area, coping with longer lasting flows of messages is more difficult. Developed solutions may require high node densities for fast delivery or even require pre-installed infrastructure (e.g., road side units). In this paper, we present a concept where we exploit parked cars to form a virtual network infrastructure. In particular, we form clusters representing a virtual road side unit spanning a relatively large geographical area. To reduce the channel load, we select a subset of the parked cars as active gateways to the cluster. This is done in a way to ensure that the connectivity to the cluster is not impaired. We evaluate the proposed algorithm using the Car4ICT architecture, a concept that enables cars in smart cities to discover and use various kinds of services.

KEYWORDS

Vehicular Networking, Virtual Network Infrastructure, Clustering

1 INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) are becoming reality with first deployments in Japan and the US. For these networks, various applications are envisioned ranging from safety to efficiency to infotainment [20]. Broadly speaking, the way data is exchanged by these applications can be split into two categories: single message broadcasts and long-lasting data flows.

CarSys'17, October 20, 2017, Snowbird, UT, USA

ACM ISBN 978-1-4503-5146-1/17/10…$15.00
https://doi.org/10.1145/3131944.3131952

The first category intends to cover the surroundings of the sender or a specific geographical destination. Single messages are used for safety and efficiency purposes, e.g., warning other drivers of incoming cars or road hazards. Currently, there seems to be a consensus to either use broadcast based algorithms (e.g., n-hop broadcast) or georouting. Concepts for delivery in a timely and efficient manner have been proposed and standardized, e.g., CAM and BSM messages in the ETSI ITS-G5 and the IEEE WAVE standards, respectively [20].

The second category consists of data flows that take longer to transmit, e.g., video streaming or transmission of larger files [23]. Because of the dynamic nature of vehicular networks, messages can be already considered large if they consist of a few kilobytes. For such messages, the contact time between a sender and a receiver is often too short; and thus, infrastructure or more complex algorithms are needed to bridge the gap. In this paper, we consider such long-lasting data flows for which we observe two main challenges: On the one hand, if there is a lot of road traffic, the wireless channel gets overloaded. On the other hand, without any vehicles, there might not even be a car to forward the message [3].

Solutions described in the literature are based on using either Store-Carry-Forward (SCF) or Roadside Units (RSUs). For SCF cars cache the message until a suitable forwarder is found. Such algorithms can be very useful on highways. In cities, however, a lot of information is needed to optimize such algorithms (e.g., travel patterns based on historical data which is hard to acquire). Otherwise, the destination might only be reached with a high delay or even not at all. Collecting such information is not trivial and further contributes to the network load. As an alternative, RSUs can be used to provide Access Points (APs) to the Internet or, by being interconnected, allow to cross larger distances between two cars. However, a dense deployment of RSUs might be very expensive, while a sparse deployment might not improve connectivity. Another issue with RSUs is the rather small contact time to the passing car which might not be sufficient to transmit all data – especially at high network loads. Round-trip times induced by higher layer protocols (e.g., DNS or TLS) further reduce the chance of completing the data exchange before the connection is severed.

In this paper, we propose a solution that exploits parked cars to provide a virtual network infrastructure. It can be used to exchange data and to extend the vehicular network for better connectivity. Multiple parked cars are clustered together as a single virtual network node. It provides longer contact times between parked and driving cars which in turn enables longer data transfer and more complex protocols. The cluster is based on a virtual routing protocol that also features Distributed Hash Table (DHT) functionality. Without loss of generality, we used the Virtual Cord Protocol (VCP) [4] for our experiments but any such protocol will be suitable for our concept. We further integrated the basic concept with the Car4ICT architecture [2] as a proof-of-concept example.

In particular, we give an answer to the following research challenges summarized in [13]: First, which cars should form one cluster? We present, based on a DHT, the framework of a clustering algorithm and discuss certain requirements for such a cluster. Second, how to connect to the cluster? We outline the algorithm to let cars connect to the cluster and stay connected while passing by. This includes algorithms to select certain cars as gateways and how to perform handover between them.

2 RELATED WORK

The work presented in this paper unifies a broad range of topics. In the following, we first cover a set of use cases for RSUs and parked cars followed by a discussion of clustering protocols and in-cluster communication. To be able to communicate with a cluster of cars there is the need for a handover procedure; we introduce selected approaches specifically designed for vehicular ad-hoc networks. Finally, we briefly outline Car4ICT, which we used as a basis for evaluating our virtual network infrastructure.

2.1 Roadside Units

Roadside Units (RSUs) have been considered in vehicular networks for many different use cases. As topologies of vehicular networks usually change rapidly, the addition of RSUs can help improve performance. Mershad et al. [17] propose the *ROAMER* routing protocol. If a message has to be sent to a close-by node, a simple broadcast scheme is used; if the message has to be delivered over a greater distance, RSUs are exploited. Still, a sparse network leads to performance problems and the authors propose to consider using multiple forwarders and data replication.

As the deployment of RSUs is a costly process, more recent works are aiming to provide a similar functionality by using parked cars. Sommer et al. [21] present a system where such parked cars are used to improve intersection safety. These parked cars aim to replace RSUs in the vicinity of intersections and relay messages. Contrary to our approach, their focus lies on safety messages rebroadcasted by a single parked vehicle. Another set of applications for vehicular networks can be grouped under *infotainment*, e.g., video streaming or online gaming. Malandrino et al. [16] investigate a use case where parked cars assist RSUs to download videos. The authors performed optimization studies as well as extensive simulations and conclude that parked cars greatly improve the download performance while assisting the RSU.

Most of these approaches rely on the presence of physical RSUs, deployed and operated for the benefit of each proposed protocol. Conversely, in our approach we want to exploit a cluster of multiple parked cars to act as network infrastructure covering a relatively large area. This allows us to maintain longer connections but induces the need for clustering the parked cars.

2.2 Clustering and Gateway Selection

To create a virtual networking infrastructure, the parked cars have to be clustered. Sucasas et al. [22] discuss the different aspects of clustering in which vehicular networks are identified as one of the most active and promising areas of clustering. However, the questions of how to provide good quality of service as well as how to deal with the high mobility are yet to be solved. Clustering approaches either aim to adhere to a certain application scenario or try forming clusters independent of the scenario but based on a set of available parameters (e.g., speed, direction, position, or interest) [8]. These parameters are then used to select a set of nodes as cluster heads, which, in turn, coordinate all cluster members.

In classical ad-hoc networks, most clustering solutions optimize to reduce the energy consumption as extensively discussed in a survey by Afsar and Tayarani-N [1]. Nevertheless, this is not the main concern for our protocol as the battery of a car often provides enough power to keep the network module running.

As clustering parameters are often situation and/or application dependent, other techniques are being investigated: Cheng et al. [6] propose to use an evolutionary algorithm for calculating the clusters. However one of their assumptions is a fixed unit-disk range model, which is often invalidated in cities by factors such as multi-path propagation and shadow fading. Zahidi et al. [24] employ Integer Linear Programming to determine the topology of the cluster. The clusters are calculated based on a set of constraints. However, some of the constraints (e.g., pre-configured number of cluster heads, star topology of the network) make it hard to deploy the algorithm in arbitrary scenarios.

Awad et al. [4] propose Virtual Cord Protocol (VCP), a scheme for clustering nodes in ad-hoc networks creating a virtual linear network topology. This scheme, together with exploiting shortcuts afforded by the real topology, allows efficient routing of messages between the nodes. As every node is aligned on a virtual cord, it can provide DHT services to store various information.

When forming a cluster, usually cluster heads are selected as distinguished control nodes. As the selection of cluster heads is algorithm-dependent, they might not be the optimal choice for other tasks, e.g., as gateways for communicating with the cluster. In our case, the gateways should be a minimal set of cars which allow for a seamless connection from outside of the cluster. This problem is similar to the *k-barrier coverage* problem from the domain of wireless sensor networks where a certain border region is covered at least k times [15]. As an alternative, Dai and Wu [9] propose to select a dominating set for routing based on local information and use a parameter k to determine the number of neighboring gateways. Similar to these approaches, our algorithm uses the geographical position as a parameter to determine the coverage.

2.3 Handover

When a car passing by wants to transfer data to/from the cluster it may need to connect to multiple parked cars during the transmission. It thus needs a mechanism to perform a horizontal handover. Ghosh et al. [11] study this problem focusing on communication using RSUs. By looking at different metrics, they show that handover in vehicular networks has not been investigated sufficiently as protocols usually focus only on Inter-Vehicle Communication (IVC) or do not consider handover problems. The same point is made by Bali et al. [5] in their survey of different clustering protocols for vehicular networks. One of the core conclusions is that the question of how to perform vertical (i.e., between different networking technologies) and horizontal (i.e., between the same technology) handover with clusters is a very important question for future research.

More recently, Ghosh et al. [10] propose to employ probabilistic handover. Based on studies of overlapping transmission regions, the authors define a probability function. This function is then used to make handover decisions. So far they only discussed their approach in an analytical manner, thus, applicability in more realistic scenarios remains unclear.

A common problem in handover situations is that it is not clear to which endpoint a data packet has to be sent (i.e., which endpoint is currently the best one to reach the destination). Huang et al. [14] try to solve this problem by probabilistically selecting a common ancestor which takes care of such decisions. Their approach slightly increases the delay from this common ancestor to the destination, but decreases the delay occurring when sending data from one RSU to the next due to handovers. Mouton et al. [18] discuss a handover scheme for vehicular networks where the handover decision is based on the current context, e.g., the vehicle's direction, road topology, and network deployment information. One module for route prediction, one for probing APs, and one for maintaining an overview of the road network combined with APs interacting with each other to predict the best possible AP. Their approach needs additional modules throughout the network stack on the driving car which makes it rather complex to realize and integrate. Furthermore, the handover in our approach is triggered by the virtual network infrastructure and not by the passing car.

2.4 Car4ICT

To study the performance of our approach, we select the existing Car4ICT infrastructure [2], which is able to benefit from such virtual network infrastructure. This architecture envisions cars as hubs in future smart cities, where they act as data/service providers for users. Cars accomplish this by providing mechanisms for service discovery and assist users in utilizing these services. It has been shown that this architecture is very robust for creating scalable network connections, e.g., in smart cities or in cases where network infrastructure is not available after a natural or man-made disaster. Integrating parked cars into the Car4ICT architecture increases the availability of services, as the parked cars can also offer services. In addition, the network becomes more stable as a result of having location-wise stable nodes in the network.

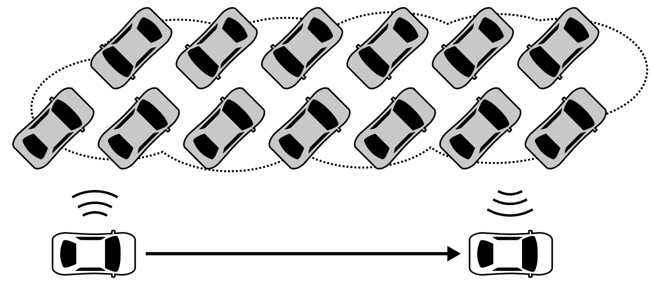

Figure 1: A schematic overview of our approach where a driving car passes a cluster of parked cars forming virtual network infrastructure. The physical connection is automatically handed over among cars in the cluster as the driving car passes by.

3 PARKED CARS AS VIRTUAL NETWORK INFRASTRUCTURE

Our approach is to exploit clusters of parked cars to create virtual network infrastructure. Cars driving by are able to connect to a parked car and, via this car, are connected to all applications/services inside the cluster. Figure 1 depicts a set of parked cars interconnected by a virtual backbone. A single car driving past the parked cars is connected to the virtual network infrastructure and is able to use services provided by any car in the cluster. To maintain a stable connection, it is necessary to hand over the connection to another car at a later point in time. This handover procedure is to be steered by the cars in the cluster to avoid further complexity in driving cars.

For our approach, we do not rely on any specific clustering algorithm, but require certain features to be available:

- To be able to form such a virtual network infrastructure and allow passing vehicles to access it, parked cars need to be equipped with a short-range networking technology. We consider this to be IEEE 802.11p, though any other technology (e.g., Wi-Fi, LTE Direct) can be used.
- For clustering and the selection of gateways, all cars need to be equipped with a system to determine their current geographical position (e.g., a basic GPS receiver).
- Every car in the cluster is able to reach every other cluster member. This does not necessarily mean that there should be 1-hop connectivity but that an address-based routing scheme should be in place. Such a functionality can be seen as a function `send(id, message)` which sends the data `message` to the node with identifier `id`. As topology changes in the cluster are rather small, most of the routing protocols for wireless ad-hoc networks are suitable.
- To store and retrieve information for our protocol we require that there is some kind of distributed storage, preferably a DHT, running in the cluster. This storage should provide the functions `get(hash)->value` and `set(hash, value)` to retrieve and store arbitrary data. Without loss of generality, we assume that the routing address space matches the key space of the DHT, a range of 0 to 1.

- Services and applications offered by parked cars have to be mapped to a hash of the DHT. This allows to clearly identify and map service information.

Based on these features, we first outline how to set up a cluster, followed by the selection of the gateways. The next step in the process is to provide measures for performing handover. Our evaluation is based on two concrete protocols, namely VCP for the underlying DHT and the routing inside the cluster [4] and the Car4ICT architecture on top of the cluster [2]. Note that these are considered examples. Any DHT based clustering/routing solution will do and the system can be used for other VANET applications as well.

3.1 Cluster Setup

The first step in setting up a cluster is creating or joining one. We assume this process begins when a car is parked and locked. Afterwards, the networking module starts listening for existing clusters in its vicinity and periodically starts sending its own messages (i.e., hello messages or beacons). Due to these periodic beacons, which include GPS information, after some time, the car becomes aware of existing clusters and is able to join them. If not, the car starts forming a new cluster and cars that subsequently park in its vicinity are able to join this newly created cluster. VCP supports exactly this kind of clustering and also provides an integrated DHT for distributed data storage and retrieval to/from cluster members. To ease the selection of gateways, we require clusters to not cross any street boundaries. This can be achieved by checking a digital map if there is a street between the position of the potential cluster nodes and the node itself.

If a car, which joined the cluster, offers a certain service or application, it stores this information in the DHT along with its own identifier. This allows gateways to map data to member cars providing the service. In addition, gateways need only one `get()` operation to determine if a service is provided within the cluster.

Whenever a parked car leaves the network (detected, e.g., as a person boarding the car), the underlying clustering scheme is able to take care of it. For data in the DHT, a node would move all data to other cluster members. Note that leaving (and joining) events do not happen on a timescale of single seconds but rather in the order of tens of seconds to minutes; therefore re-organization can be performed with minimal impact on system operation.

3.2 Gateway Selection

If there are many parked cars in close proximity, e.g., in a parking lot or along the street side, we only select a subset of them as gateways. These are the nodes that are accessible from the outside and can be used to connect to the cluster. By doing this, we are able to reduce the load on the channel while still maintaining the connectivity from moving cars to our virtual network infrastructure.

In a more formal way, we select the set of gateways $\mathbb{G}$ such that it reduces the amount of n-covered regions (with $n > 2$) to a minimum. The optimal selection of such gateways can be calculated if coverage information is known from all nodes. In the literature, this is commonly handled by assuming a fixed transmission distance, i.e., a unit disk radio propagation model. In reality, however, the calculation of this optimal set is very hard as the covered area is highly irregular and time varying due to reflections and obstacles. Therefore, we need an approximation for calculating $\mathbb{G}$. The information considered by our algorithms is the position of the cars and their 1-hop neighbor information (i.e., cars in the cluster they can communicate with). To optimize the gateway selection, we rely on a coordinator node (e.g., a cluster member with a specific virtual address) to be in charge of the calculation.

Algorithm 1 Gateway selection along a curve

Input: $\mathbb{N}$, the set of all nodes in the cluster
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 $\mathbb{N} \neq \emptyset$ **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 $\mathbb{G}$

Whenever a node joins a cluster, it informs the coordinator node of its position and its neighbors. This information allows to keep a record of the whole cluster containing $\mathbb{N}$, the set of all nodes, and sets of neighbors of all these nodes. We define the neighbors of a node n as the set of all nodes that received beacons from n and vice-versa (i.e., we particularly exclude all unidirectional connections).

Furthermore, we consider two main cases how cars can be parked: along the street and in an (open space) parking lot: First, the goal for cars aligned along the street is to allow a car driving by to be connected to an RSU at all times. Second, if the cars are in a parking lot, we want to expose the cluster to the outside area of the parked cars. Therefore, our approximation consists of two algorithms, one for cars along a curve, i.e., arbitrarily shaped streets, and one for cars in an open parking area.

3.2.1 *Gateway selection along a curve.*

Algorithm 1 selects gateways based on their geographical position and terminates if a minimum coverage has been achieved. The selection process starts with a seed node that is farthest away from all others, i.e., one end of the curve. The algorithm then iteratively selects the node that is farthest away from all previously selected gateways (i.e., the other end, then a node in the middle, and so on). Before adding the node to the set of selected gateways $\mathbb{G}$, it checks if two neighbors of the node under consideration are already gateways. This check ensures that all parts of the cluster are covered by at least one car. Furthermore, it makes sure that areas are not over-covered. By selecting the farthest neighbor in each step, we equally distribute the gateways and not select them from one side only.

3.2.2 *Gateway selection in an area.*

The goal of Algorithm 2 is to expose the virtual network infrastructure to the outside of the parking lot. To do this, the algorithm consists of two parts

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 $p \in \mathbb{N} \setminus$ {points on $\mathbb{C}$} closest to edge c
  $\varsigma = \frac{\text{distance}(c_0, c_1)}{\min\{\text{distance}(c_0, p), \text{distance}(p, c_1)\}}$
  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 {points on $\mathbb{C}$} as input
return $\mathbb{G}$

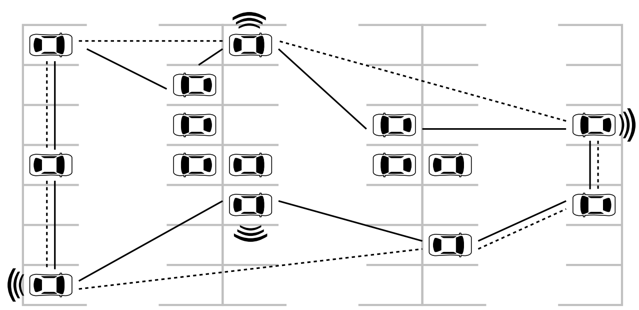

Figure 2: Outline of the proposed algorithm for gateway selection. First, the convex hull is computed (dashed line). Second, additional nodes are added based on the concave hull (solid line). Third, based on Algorithm 1, the gateway selection is performed.

(cf. Figure 2). First, to ensure connectivity to the cluster, nodes on the perimeter of the network are selected using a concave hull algorithm. As a baseline, we selected the algorithm proposed by Park and Oh [19]. In their approach, based on the convex hull, every edge is investigated and it is determined if any points inside the hull are of interest. Based on a *digging parameter* Δ, it is decided if these points should be added to the hull. Second, the nodes on the hull (which thus form a curve) are used as an input for the algorithm for gateway selection along a curve, that is, Algorithm 1.

3.3 Handover

If a car is driving past a cluster of parked cars, it might need to connect to different cars over time in order to maintain connectivity to the cluster. All the time, it could be in the range of multiple gateways at the same time. In both cases, it has to be determined through which gateway to connect to and how to provide seamless connectivity.

Let us first consider the direction from the car to the cluster, i.e., uploading data. The establishment of an upload link is handled by the gateways and their periodic beacons. These beacons are received by cars driving by the cluster. Whenever a car intends to send data to the virtual network infrastructure, it can send it to any gateway it recently received a beacon from. The choice of the gateway is arbitrary as the cluster represents a virtual backbone. On receiving such data, the gateway uses the DHT to perform a destination lookup. Based on this lookup, the gateway sends the message to the destination inside the cluster.

Second, we need to consider the direction from the RSU towards the car passing by, i.e., downloading data. Here, the case is more difficult as it involves a possible handover between two gateways during an ongoing data communication. Again, we make use of the beaconing information and require cars to reply to these beacons with an identifier. This is either fixed or, considering privacy preserving techniques, a temporary pseudonym. However, we require the identifier being constant during an ongoing data communication. When a gateway receives a reply from a car, it stores this information in the DHT together with a timestamp and the distance between the gateway and the car. This way, every cluster node can obtain information to identify the best suitable gateway via which the moving car can be reached. Best suitable refers to the gateway that received the most recent updates from the moving car. If multiple gateways have a similar timestamp, we also factor in the geographical distance from the gateways to the car to break the tie.

3.4 Case study: VCP & Car4ICT

For our case study, we need a clustering/DHT algorithm as well as an application layer protocol running on top of the virtual network infrastructure. For the DHT, we selected VCP [4]. VCP arranges all nodes along a virtual cord by assigning virtual addresses in the range $[0, 1]$. VCP provides routing functionality based on these addresses. The cord as well as neighbor information is used for greedy routing, where nodes are able to choose shortcuts to physical neighbors in their vicinity. The cord guarantees reachability from all positions in the network.

Furthermore, VCP provides DHT services, storing and retrieving key/value data at nodes defined by mapping each key to an address where data is stored. Determining this address is straightforward: each address uniquely maps to one node and each node knows its successor and predecessor on the virtual cord; it is thus easy to calculate which range of keys is served by each node. For gateway selection, a well-known node is needed, which we fixed to address 0. This was done because VCP ensures that node 0 always exists.

As an example application we used the Car4ICT architecture [2]. Car4ICT enables users to discover and utilize services offered by other users. The cars assist in the service discovery process as well in the data exchange after a matching service was found. Adding parked cars to the system opens up further possibilities for the Car4ICT architecture as such cars at static locations make discovery more robust.

Car4ICT makes use of the DHT provided by the virtual network infrastructure by storing offered services in it. This comes naturally as services are distinguished by *identifiers* consisting of a hash and

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 Efficiency Study	
Parked cars	50
Driving cars	on average 1 every 10 s
Amount of requested data	128 kB
Simulation duration	360 s of traffic
Repetitions	32
Car4ICT Case Study	
Gateway digging parameter Δ	1
Car4ICT parked cars	11 west, 9 east
Car4ICT service producer	one at west parking lot
Amount of requested data	128 kB, 1024 kB
Fraction of equipped vehicles	0.25, 0.5, 0.75, 0.1
Simulated time	600 s during rush hour

metadata. The hash can be used to store/retrieve a service in the DHT. Whenever a gateway receives a request it is able to utilize the DHT to search for already known services matching the request. By providing a distributed service table for Car4ICT, we are able to increase the number of available services known to the cluster without proactively sharing service tables.

4 EVALUATION

Our evaluation consists of two parts. We start with an efficiency study to show how selected gateways perform compared to all cars acting as gateways. In the second step, we integrate the proposed concept of the virtual network infrastructure into the Car4ICT architecture in a realistic scenario and evaluate the performance.

For all evaluations, we used the Veins LTE vehicular network simulator [12] as well as real road topologies, building and parking lot locations based on the Luxembourg traffic scenario of Codeca et al. [7]. Table 1 shows the main simulation parameters.

4.1 Gateway efficiency study

The first part of our evaluation investigates gateways and their effect on the cluster. We argue that we select specific gateways to reduce the load on the channel. In the following, we show that gateways actually reduce the load while maintaining the same success rate as without them. To do this, we created an artificial scenario with 50 cars parked along the road. These cars use our clustering algorithms to form a virtual infrastructure and to select gateways. We investigate two scenarios: one where all cars act as gateways and one where only a subset was selected. We investigate the load caused by control traffic with no data plane traffic.

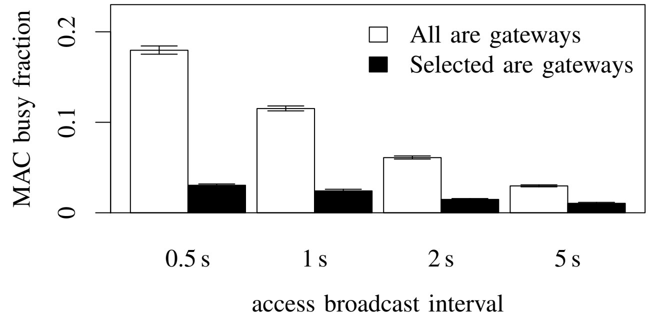

Figure 3: MAC busy fraction including 95 % confidence intervals for the single street scenario.

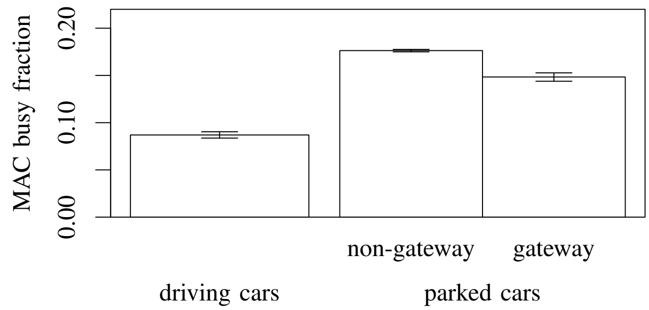

Figure 4: MAC busy fraction including 95 % confidence intervals for the single street scenario for the different types of cars.

In Figure 3, we show the average MAC busy fraction in this single street scenario. We first look at the configuration in which all parked cars act as possible gateways (white bars). If access broadcast messages are sent every 0.5 s, the load on the channel is already close to 20 %. This is due to all cars driving by answering to these access messages with their current data to enable seamless handover. By reducing the broadcast interval, the load on the channel gets smaller, however, the large interval may reduce the quality of the handover process. If we only select a subset of gateways, the MAC busy fraction is much smaller (black bars). Here, even if the interval is 0.5 s, the channel is used less than 5 %. This indicates that selecting gateways both improves the performance and reduces the channel footprint of the control messages.

In Figure 4, we investigate the MAC busy fraction for the different car roles in the scenario. We distinguish driving cars, cars being selected as gateways, and parked cars which have not been selected as a gateway. The most interesting takeaway here is that gateways have a lower network load compared to the other parked cars. This is due to the gateway selection algorithm. It selects cars which are farthest away from the others and in our scenario this means the cars at the ends of the line. These cars have much less neighbors compared to cars in the middle of the cluster and therefore less DHT traffic from VCP.

As encouraging as these results are, they do not yet reveal the influence of the gateway selection on the applications running on

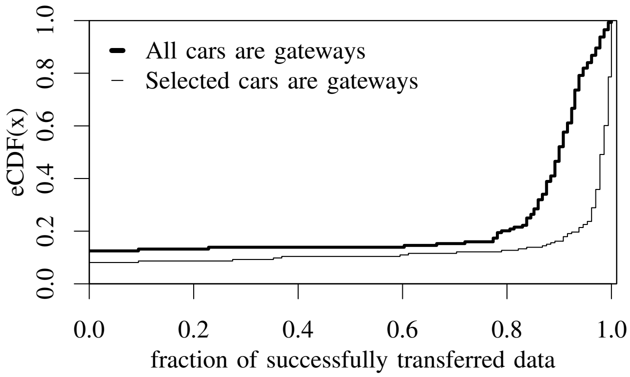

Figure 5: Success rate of transferring 128 kB to a passing by car in the single street scenario.

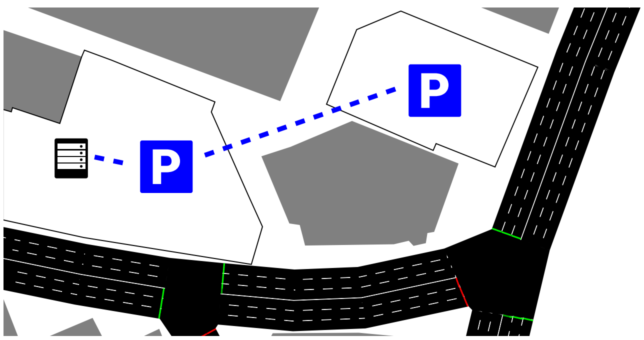

Figure 6: More realistic scenario used for studying handover decisions. Two parking lots near Luxembourg main station forming one large cluster to provide virtual network infrastructure.

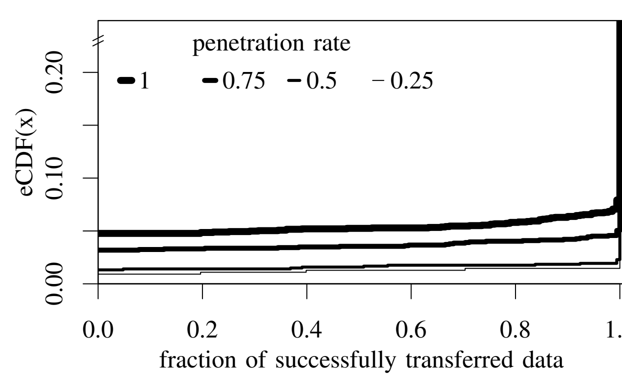

Figure 7: Success rate of transferring 128 kB to a passing car during morning rush hour in the realistic scenario.

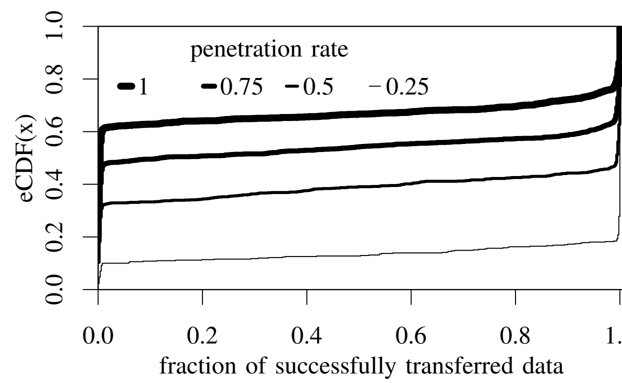

Figure 8: Success rate of transferring 1024 kB to a passing car during morning rush hour.

top of the virtual infrastructure. To investigate this further, we use Car4ICT and stored 128 kB data at the last car on the street. Cars driving by request this data and download it via the cluster.

In Figure 5, we show the success rate of transferring these 128 kB from the producer to the consumer. We can see that no matter if gateways are selected or not, a small number of cars (about 15 %) do not receive any packets at all. We were able to confirm this was due to lost packets requesting the data between the applications. Such issues could probably be solved by using a flow control mechanism as used, for example, in TCP. Otherwise, in both scenarios most of the data could be transferred. The outcome for the scenario where all cars act as gateways is obviously worse compared to the selected gateways. This is because the MAC busy fraction was larger (roughly 50 % compared to 15 %) and, therefore, less channel capacity was available for the data transmission.

4.2 Realistic scenario

As the first set of evaluation was performed using an artificial setup, our next step was to investigate a more realistic scenario. For this, we extracted a busy intersection from the Luxembourg scenario [7] as can be seen in Figure 6. Besides road geometry and road traffic, the scenario includes buildings and the locations of parking lots. We used 19 randomly placed parked cars which form a cluster spanning two parking lots. One of the cars in the western parking lot offers data to download. We simulated 10 min of traffic during the morning rush hour and used different fractions of cars being equipped with Car4ICT technology and requesting the offered data.

The success rate of downloading 128 kB can be seen in Figure 7. A small number of requests is not successful (less than 5 %), but most of the time all data is transferred. For this amount of data, the load on the network is low and it is possible to transmit the data to nearly all cars requesting it, independent of the penetration rate. It can already be observed, that with higher penetration rate, the number of successful transmissions is slightly lower.

This trend is more clearly visible in Figure 8, where we show the success rate for requesting 1024 kB. While, for a penetration rate of 0.25, nearly 80 % are successful, if all cars are equipped with Car4ICT and request the data, the success is less than 40 %.

The reason for the small success rate can be seen in Figure 9. For the higher penetration rates, the MAC is busy more than 60 % of the time, while for a penetration rate of 0.25 it is slightly above 35 %. Because of this, the load on the network is too high and data transfer is not always possible.

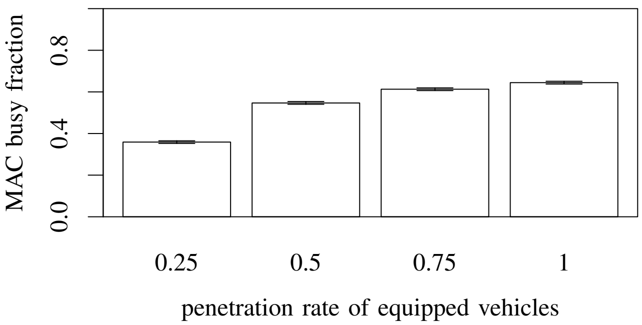

Figure 9: The MAC busy fraction for the realistic scenario while downloading 1024 kB.

In these simulations, we also investigated the application layer packet delay, i.e., the time it takes from the creation of the data until it is received by the downloading application. Note that, there is potential delay included for acquiring the correct gateway for the packet. With such a high load on the network, packets had to stay longer in queues and/or needed retries to be transmitted. This was especially visible for a data size of 1024 kB, where only 11 % of packets arrived in less than 1 s. However, only 3 % of the packets took longer than 10 s. This delay can still be acceptable for non-safety application scenarios.

5 CONCLUSION

In this paper, we outlined a concept for virtual network infrastructure based on parked cars. By clustering such cars, we are able to create a stable backbone network that can be used for data transmissions from and to passing cars as well as to store data. In particular, we presented a handover scheme that supports continuous connection to such a cluster. In our investigation, we observed that the load on the network quickly increases if all parked cars act as gateway nodes between cluster and road. To prevent this, we developed an algorithm to select cars as gateways while still providing sufficient coverage for successful connections to moving cars. In our evaluation, we have shown that this approach greatly reduces the load on the network while barely decreasing the success rate for transferred data. We investigated the problem in a realistic scenario and discovered that the performance decreases with an increase in penetration rate of cars download data from the cluster – which points to potential future work regarding load balancing mechanisms.

REFERENCES

[1] M. Mehdi Afsar and Mohammad-H. Tayarani-N. 2014. Clustering in sensor networks: A literature survey. *Elsevier Journal of Network and Computer Applications* 46 (Nov. 2014), 198–226. https://doi.org/10.1016/j.jnca.2014.09.005

[2] Onur Altintas, Falko Dressler, Florian Hagenauer, Makiko Matsumoto, Miguel Sepulcre, and Christoph Sommer. 2015. 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)*. IEEE, Hong Kong, China, 654–659. https://doi.org/10.1109/INFCOMW.2015.7179448

[3] R. Atallah, M. Khabbaz, and C. Assi. 2016. Multi-hop Vehicle-to-Infrastructure Communications: A Feasibility Study, Modelling and Performance Analysis. *IEEE Transactions on Vehicular Technology* 66, 3 (June 2016), 2801–2810. https://doi.org/10.1109/TVT.2016.2586758

[4] Abdalkarim Awad, Reinhard German, and Falko Dressler. 2011. Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks. *IEEE Transactions on Mobile Computing* 10, 9 (Sept. 2011), 1214–1226. https://doi.org/10.1109/TMC.2010.218

[5] Rasmeet S Bali, Neeraj Kumar, and Joel J.P.C. Rodrigues. 2014. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. *Elsevier Vehicular Communications* 1, 3 (July 2014), 134–152. https://doi.org/10.1016/j.vehcom.2014.05.004

[6] Hui Cheng, Jiannong Cao, Xingwei Wang, Sajal K. Das, and Shengxiang Yang. 2009. Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility. *Wiley Wireless Communications and Mobile Computing (WCMC)* 9, 6 (June 2009), 759–771. https://doi.org/10.1002/wcm.627

[7] Lara Codeca, Raphaël Frank, and Thomas Engel. 2015. Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research. In *7th IEEE Vehicular Networking Conference (VNC 2015)*. IEEE, Kyoto, Japan. https://doi.org/10.1109/VNC.2015.7385539

[8] Craig Cooper, Daniel Franklin, Montserrat Ros, Farzad Safaei, and Mehran Abolhasan. 2017. A Comparative Survey of VANET Clustering Techniques. *IEEE Communications Surveys & Tutorials* 19, 1 (Feb. 2017), 657–681. https://doi.org/10.1109/COMST.2016.2611524

[9] Fei Dai and Jie Wu. 2003. Distributed dominant pruning in ad hoc networks. In *International Conference on Communications (ICC '03)*. IEEE, Anchorage, AK, 353–357. https://doi.org/10.1109/ICC.2003.1204198

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

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

[12] Florian Hagenauer, Falko Dressler, and Christoph Sommer. 2014. A Simulator for Heterogeneous Vehicular Networks. In *6th IEEE Vehicular Networking Conference (VNC 2014), Poster Session*. IEEE, Paderborn, Germany, 185–186. https://doi.org/10.1109/VNC.2014.7013339

[13] Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. 2016. Using Clusters of Parked Cars as Virtual Vehicular Network Infrastructure. In *8th IEEE Vehicular Networking Conference (VNC 2016), Poster Session*. IEEE, Columbus, OH, 126–127. https://doi.org/10.1109/VNC.2016.7835943

[14] Chung-Ming Huang, Meng-Shu Chiang, and Tz-Heng Hsu. 2008. PFC: A packet forwarding control scheme for vehicle handover over the ITS networks. *Elsevier Computer Communications, Special Issue on Mobility Protocols for ITS/VANET* 31, 12 (July 2008), 2815–2826. https://doi.org/10.1016/j.comcom.2007.12.010

[15] Santosh Kumar, Ten H. Lai, and Anish Arora. 2007. Barrier coverage with wireless sensors. *ACM/Springer Wireless Networks (WINET)* 13, 6 (Dec. 2007), 817–834.

[16] Francesco Malandrino, Claudio Casetti, Carla-Fabiana Chiasserini, Christoph Sommer, and Falko Dressler. 2014. The Role of Parked Cars in Content Downloading for Vehicular Networks. *IEEE Transactions on Vehicular Technology* 63, 9 (Nov. 2014), 4606–4617. https://doi.org/10.1109/TVT.2014.2316645

[17] Khaleel Mershad, Hassan Artail, and Mario Gerla. 2012. ROAMER: Roadside Units as message routers in VANETs. *Elsevier Ad Hoc Networks* 10, 3 (Sept. 2012), 479–496. https://doi.org/10.1016/j.adhoc.2011.09.001

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

[19] Jin-Seo Park and Se-Jong Oh. 2012. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *Academia Sinica Journal of Information science and engineering* 28, 3 (2012), 587–600.

[20] Christoph Sommer and Falko Dressler. 2014. *Vehicular Networking*. Cambridge University Press. https://doi.org/10.1017/CBO9781107110649

[21] Christoph Sommer, David Eckhoff, and Falko Dressler. 2014. IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation. *IEEE Transactions on Mobile Computing* 13, 8 (Aug. 2014), 1733–1745. https://doi.org/10.1109/TMC.2013.80

[22] Victor Sucasas, Ayman Radwan, Hugo Marques, Jonathan Rodriguez, Seiamak Vahid, and Rahim Tafazolli. 2016. A survey on clustering techniques for cooperative wireless networks. *Elsevier Ad Hoc Networks* 47 (Sept. 2016), 53–81. https://doi.org/10.1016/j.adhoc.2016.04.008

[23] Mehrnaz Tavan, Roy D. Yates, and Dipankar Raychaudhuri. 2016. Connected Vehicles Under Named Object Architectures. In *1st ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2016)*. ACM, New York, NY, 60–61. https://doi.org/10.1145/2980100.2980109

[24] Syed Zohaib Hussain Zahidi, Fadi Aloul, Assim Sagahyroon, and Wassim El-Hajj. 2013. Optimizing Complex Cluster Formation in MANETs Using SAT/ILP Techniques. *IEEE Sensors Journal* 13, 6 (June 2013), 2400–2412. https://doi.org/10.1109/JSEN.2013.2254234