Towards a Vehicular Cloud – Using Parked Vehicles as a Temporary Network and Storage Infrastructure

Falko Dressler*, Philipp Handle§ and Christoph Sommer*

* Dept. of Computer Science, University of Paderborn, Germany
§ Institute of Computer Science, University of Innsbruck, Austria
{dressler,philipp.handle,sommer}@ccs-labs.org

ABSTRACT

We investigate the potentials of a vehicular cloud based on parked vehicles as a spatio-temporal network and storage infrastructure. Vehicular networking solutions have been investigated for more than a decade but recent standardization efforts just enable a broad use of this technology to build large scale Intelligent Transportation Systems (ITS). One of the key questions is whether some pre-deployed infrastructure is needed to enable and to boost vehicular networks. We see many benefits in such infrastructure to store information and to provide connectivity among the vehicles. Yet, instead of using Roadside Units (RSUs), we envision to rely on parked vehicles to provide such vehicular cloud services. In this paper, we demonstrate the feasibility of this idea using the Virtual Cord Protocol (VCP) protocol to enable dynamic cloud services. Our simulation results indicate that this concept is sound and scales even with a highly dynamic number of parkers on the streets.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

vehicular networking, virtual cord protocol, vehicular cloud, parked vehicles

1. INTRODUCTION

After more than a decade of research, vehicular networks are becoming reality currently relying on 3G and 4G networks but already looking at the use of Dedicated Short Range Communication (DSRC) on a large scale [9]. One of the first set of applications is focusing on cooperative awareness, a safety related application based beaconing, i.e., on one-hop broadcasting, using IEEE 802.11p [1] as a base technology.

WiMobCity'14, August 11, 2014, Philadelphia, PA, USA.

ACM 978-1-4503-3036-7/14/08 ...$15.00.
http://dx.doi.org/10.1145/2633661.2633671.

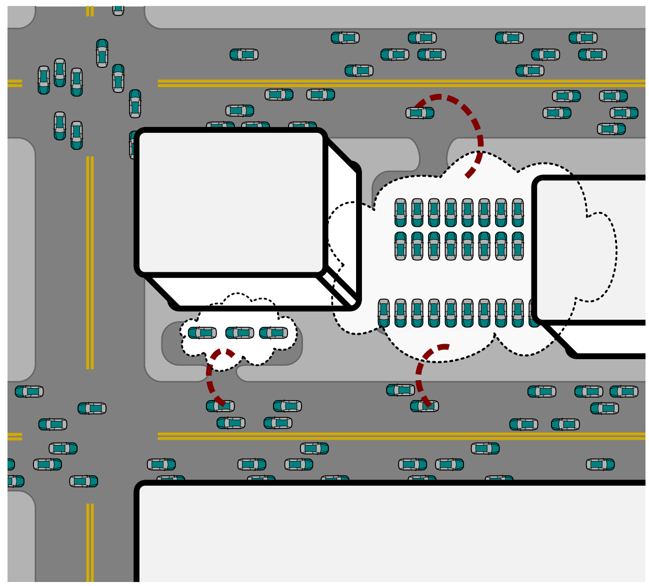


Figure 1: Concept of storing and retrieving information from clouds of parked vehicles

In order to bridge communication gaps, especially during the initial bootstrapping of this technology, the use of Roadside Units (RSUs) is considered deployed at strategic positions [5,15]. This infrastructure enables new applications based on the provided storage capabilities of the backbone network. On the other hand, deploying RSUs densely all over the road network is economically infeasible.

As an alternative, the use of parked cars has been proposed. This concept has been shown to successfully reduce the need for deployed RSUs for a wide range of applications from cooperative awareness [10,16] to content downloading [13,14]. Parked cars have the advantage of being almost ubiquitously available in city and urban environments, i.e., at exactly the locations where RSUs would have to be deployed. Furthermore, the energy management of parked cars has been evaluated already for a specific content downloading application clearly indicating the capabilities of this concept [6].

We aim for another application domain named the vehicular cloud [11, 19]. This idea targets the ability to store and to retrieve location-based data on a large scale. This can be simple information like the position of the next free parking space or road traffic information. Such vehicular cloud services can be organized by parked cars in a city into independent networks or clusters. The formed network will be able to store data and to provide connectivity between the clusters. The key challenge of this approach is the creation and maintenance of such clusters of parked vehicles, especially with regards to the dynamics due to cars leaving and joining.

Our concept is shown in Figure 1. We aim at establishing connected clouds of parked vehicles that are internally maintaining network connectivity and provide storage capabilities. Moving cars passing by can connect to these clouds to upload and download information.

A protocol able to handle such dynamic situations well in the scope of Mobile Ad Hoc Networks (MANETs) is our Virtual Cord Protocol (VCP) [2, 3]. The key feature of this protocol is that it combines overlay Distributed Hash Table (DHT) services with integrated routing. We extended this protocol to fully support this vehicular networking application relying on inter-domain routing capabilities introduced to VCP in [8], which help bridging the gap between otherwise isolated VCP clouds.

Our key contributions can be summarized as follows:

- We developed a complete networking architecture for enabling vehicular cloud services using clusters of parked vehicles.

- One of the key elements is to extend VCP and its inter-domain routing for application in very dynamic vehicular networks.

- We performed extensive simulation experiments to evaluate the capabilities of our framework. Our results clearly indicate the benefits of our solution.

2. RELATED WORK

The idea to use parked cars supporting Inter-Vehicle Communication (IVC) is around since a few years and several applications domains have been identified. We briefly review the state of the art in the following.

One of the first approaches using parked vehicles was our concept to improve cooperative awareness applications using parked vehicles to relay beacon messages at intersections [10, 16]. A similar use of parked cars has been presented in [4]. Parked cars are used to relay all sort of messages to increase the communication range of moving cars.

Content downloading using parked vehicles establishes a second application domain. Parked Vehicle Assistance (PVA) helps increasing the connectivity in Vehicular Ad Hoc Networks (VANETs) [13]. In [14], this idea is further extended using parked cars to establish a backbone network to extend the range of RSUs. Using the freshness of the content as well as the radio efficiency to retrieve the content to measure the quality of the approach, it has been shown that parked cars can substantially improve content downloading performance.

The most similar approach to our work is ParkCast [12]. Cars parking at the road side are organized in line clusters, which are extended as far as possible. An established cluster is used for sequential file transfer to cars that drive along that cluster. This way, content download performance is further improved. GPS and electronic map information are used to identify the parked cars closest at intersections. These cars are assigned a specially functionality, which is replicated to other cars as a warm backup. It has been shown that ParkCast can significantly reduce the delay for downloading large files in comparison to other content downloading solutions.

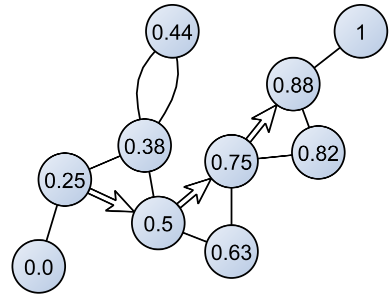


Figure 2: VCP cord and greedy routing

Our approach presented in this paper goes one step further. We specifically allow the use of parked vehicles not only to provide networking capabilities but also to fully enable storage functionality in these very dynamic structures. We follow the idea of vehicular clouds presented in [11] and enable, for the first time, integrated networking and storage capabilities in dynamic vehicular networks using on a large-scale using parked vehicles.

3. FUNDAMENTALS

In this section, we briefly review the main concepts of VCP and its inter-domain routing extension.

3.1 Virtual Cord Protocol

Instead of using two orthogonal layers providing connectivity on the network and data management on an overlay, VCP integrates both layers making use of so-called virtual coordinates [2]. This way, greedy routing can be used without having to face routing voids as typical for geographic routing. The key motivation in the development of VCP was to find an efficient, failure tolerant routing technique with little overhead. In combination with the DHT-like overlay network that is offered by VCP, deterministic data-storage is provided.

For setting up the virtual cord connecting all nodes of the network, each node periodically broadcasts a `hello` message. The `hello` message includes the position of the sender as well as its predecessor and its successor in the cord. Upon reception of a `hello`, the node updates the corresponding entry in its neighborhood table.

To join a cord, a node has to receive at least one `hello` message from a node with a position in the cord. The node then evaluates its neighbor table and starts a join procedure depending on the entries. The successor and predecessor relationship is adjusted so that the joining node becomes the successor of one node and the predecessor of the other node.

The resulting VCP cord is illustrated in Figure 2. The cord connects all nodes that are numbered from a starting position $S = 0.0$ monotonically increasing to a end position $E = 1.0$. The join process ensures that all other nodes will be included in the cord accordingly. The advantage is that the join process affects nodes in a small area only. Thus, it is independent of the total network size and scales well even in big networks.

Data is forwarded using greedy routing based on the structure of the cord in combination with neighborhood information. Figure 2 also illustrates this process: node 0.25 sends a packet to target node 0.88. In addition to supporting greedy routing, the cord structure is also used to store and retrieve data items. The association to a specific node is made using an application dependent hash function.

3.2 Inter-Domain Routing

VCP has been extended to also support inter-domain routing between multiple VCP cords [7, 8]. This is provided by selecting nodes that are able to communicate with other domains as gateways, and storing information about these gateways in the local VCP cord. If a gateway node no longer receives `hello` messages from a foreign domain, it removes the gateway information from the local cord.

If a packet needs to be routed to another domain, gateway information is looked up in the local domain using the destination domains ID. Once a gateway is identified, the node sends the packet to the gateway using greedy routing. The gateway then bridges the inter domain gap by sending it to a neighbor in the destination domain. Within the destination domain, again greedy routing is performed. Updates of data throughout the different domains are performed by a publish-subscribe mechanism. In order to also bridge multiple domains, swarm intelligence concepts have been explored in [8], yet, we focus on single-hop inter-domain routing for this work only.

4. VEHICULAR CLOUD USING VCP

We have already introduced the concept of using parked vehicles to establish a vehicular cloud in Figure 1. Parked cars are considered to establish a dynamic vehicular cloud providing storage and network capabilities. Whenever a car parks, i.e., it stops for a longer time period (we assume that the parking state can be determined according to the use of the keys of the car), it either joins an existing cloud or creates a new one. Cars leaving the parked cloud have to be removed accordingly. On the other hand, moving cars may set up a connection to the parking clouds at any time to upload or download information.

Essentially, we have to distinguish between two basic operations:

- Setup and management of VCP networks covering a group of connected parked vehicles, and
- temporary integration of moving vehicles into the vehicular cloud to upload and download information.

For this, each vehicle needs to first distinguish between the two modes of operation, driving and parking. If driving, it creates a single node VCP domain relying on inter-domain routing for data exchange whenever connected to another VCP domain. If parking, the vehicle starts joining an existing cord or, if not successful, initiates a new cord. In the following, we study the dynamic cord creation and the data management procedure of our vehicular cloud approach in more detail.

4.1 Dynamic Cord Creation

One of the preliminaries of VCP is that at least one node is preconfigured as a start node, and in combination with inter-domain routing all nodes joining a network need to have the same domain ID. However, this is not feasible in very dynamic scenarios such as vehicular networks. We therefore modified the original VCP algorithm to dynamically select a domain and to initiate new VCP cords if necessary.

The selection of available domains follows a simple timer approach. Every new node is initialized with an `UNKOWN_DOMAIN` value. A timer t_{listen} is started as soon as the node is initialized for both parking and arriving cars.

The node remains silent after its creation, since all data that is included in `hello` messages is yet undetermined, hence `hello` messages would not add any benefit to surrounding nodes. Received `hello` messages are processed as usual and entries are added to the neighborhood table. With the first timer event the node evaluates the entries in the neighborhood table and generates a domain set. The generated set includes all reachable entries with a valid position, regardless of the domain. If the domain set is empty, i.e., no `hello` message was received, the timer is started again. Otherwise, the number of neighbor table entries per domain are counted and the domain with the highest amount of neighbors is chosen. The node then starts the normal join process to become a member of the respective domain and also starts sending `hello` messages.

The creation of a new domain is also triggered via a timer event. To avoid simultaneous creation of many cords in the beginning, or if a great amount of cars park in a short time in a previously empty area, a random factor is added to the timer value. In the initialization process a node N calculates its own timeout as:

$$t_{\text{convert}}^{N} = t_{\text{convert}} + \text{uniform}(0, t_{\text{variance}}) \tag{1}$$

This timer t_{convert} completely controls the domain creation process. If no `hello` was received in this period, the node creates a completely new VCP cord. Thus, the node selects a unique domain ID, sets its cord position to S and modifies its predecessor and successor to point to itself. Being a fully self-organizing approach, it might lead to an unecessary high number of small domains. We investigate the performance of this solution in Section 5. In most cases, we can report a negligible amount of small, i.e., single node, cords.

4.2 Data Storage

Data management is based on the provided DHT functionality. The storage application running on the parking cars is quite straightforward and reduced to the absolutely needed functionality. The application only provides `PUBLISH` and `LOOKUP` operations. VCP's greedy routing is used to forward the messages to the destination domain and node.

Upon reception of a `LOOKUP` message, the application hashes the identifier of the requested data and loads the data from its local storage. If no data item is present matching the request, the lookup request is ignored. Otherwise the data is wrapped in a `DATA` message and addressed to the requesting node. The generated packet is handed to the network layer for delivery.

Upon reception of a `PUBLISH` message, the application also maps the data to the local store and saves the pointers, i.e., hash values, accordingly. If the data store is already filled to its maximum capacity, the request is dropped. Otherwise, an `ACKNOWLEDGE` message is sent to the originating node.

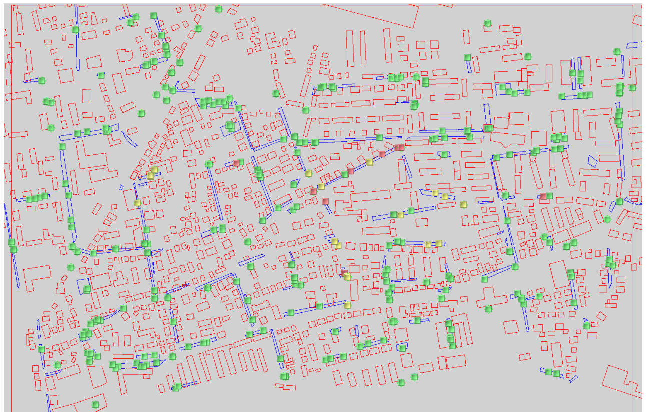


Figure 3: Screenshot of the ROI with 300 nodes

5. PERFORMANCE RESULTS

The performance evaluation is performed in two steps. First, we investigate the efficiency of the domain generation process in the very dynamic vehicular networking environment. The results are used in the second part in which we evaluate the application layer performance, i.e., the storage and retrieval of data within the clouds of parking vehicles.

5.1 Models and Parameters

All the simulations have been performed using the Veins simulation framework [18]. This tool couples the road traffic microsimulation tool SUMO with the network simulator OMNeT++ and its MiXiM framework that is providing models for wireless communications. Thus, very realistic simulation of the vehicles' mobility is provided in combination with accurate network simulation.

For the calculation of the signal attenuation the simple path loss model has been used. Focusing on city scenarios, we also cover radio signal shadowing caused by buildings [17].

We rely on the same scenario used in [10], which has been made publicly available. A screenshot of the selected Region of Interest (ROI) is shown in Figure 3. Using this scenario provides the huge benefit to rely on a validated configuration that already provides realistic traffic flows as well as potential parking spaces for the city of Ingolstadt, Germany.

The red polygons represent buildings, the blue polygons indicate parking areas. The lanes, where cars are also allowed to park, are not drawn to avoid overloading the screen shot, but their course can be guessed by looking at the gaps between the buildings.

We also indicate the operation of VCP in this figure. Each filled square represents a parking car. This has been used for visual validation purposes. Green cars are fully set up meaning they have a valid position in the cord and all neighbors in the same domain also have a valid position. Yellow cars have a valid position, but there is at least one car in the neighborhood, with the same domain ID, who has not calculated a valid position yet. Red cars simply do not have a valid position.

To determine the best parameters for the dynamic creation of new domains, we performed a parameter study including a total of 300 simulations. In particular, we were mainly interested in the protocol behavior for different configurations of the following parameters:

Table 1: Simulation parameters

	Parameter	Value
General	ROI	$1.5\,\mathrm{km}^2$
	repetitions	10
	ratio	0.7
NIC	transmission power	10 mW
	bit rate	18 Mbit/s
	queue size	14
	Channel	5.89 GHz (control channel)
VCP	hello inaccuracy	0.05
	approve delay	5 s
	propose delay	5 s
	position delay	2 s

- t_{convert} is the time a node waits for a `hello` message from a neighbor with a valid position.
- t_{variance} is the maximum value of a uniform distribution from which a value is drawn and added to t_{convert}.
- To analyze the influence of the amount of nodes on the creation of new cords, each set of parameters was replicated with 100, 300 and 500 nodes. The nodes were all placed randomly in the ROI at the start of the simulation.

Table 1 summarizes the most important parameters used in our simulation study.

5.2 Domain Creation Process

We start investigating the performance of the domain creation process. Figure 4 depicts the number of nodes with a valid cord position over time for different timeout values t_{convert} and a total number of 100 parking cars. The 10 % and 90 % quantiles are indicated by the error bars and (for better readability) only included for the value of $t_{\mathrm{covnert}} = 100\,\mathrm{s}$, which has finally been chosen for the second set of experiments.

In the scenario with 100 parkers, trends are clearly visible. The higher the value of t_{convert}, the longer it takes for the network to reach a high percentage of valid nodes. Timeouts in the range of 20 s . . . 120 s eventually lead to more or less

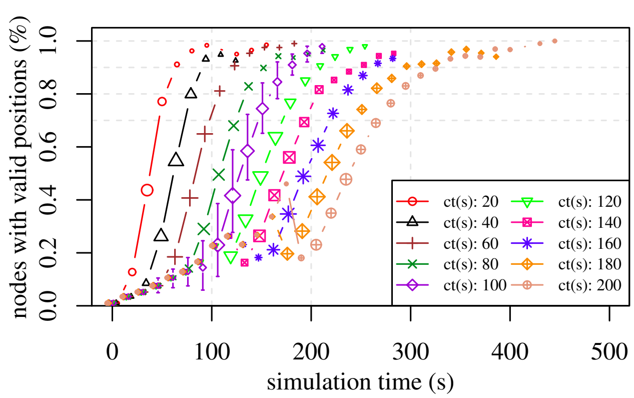
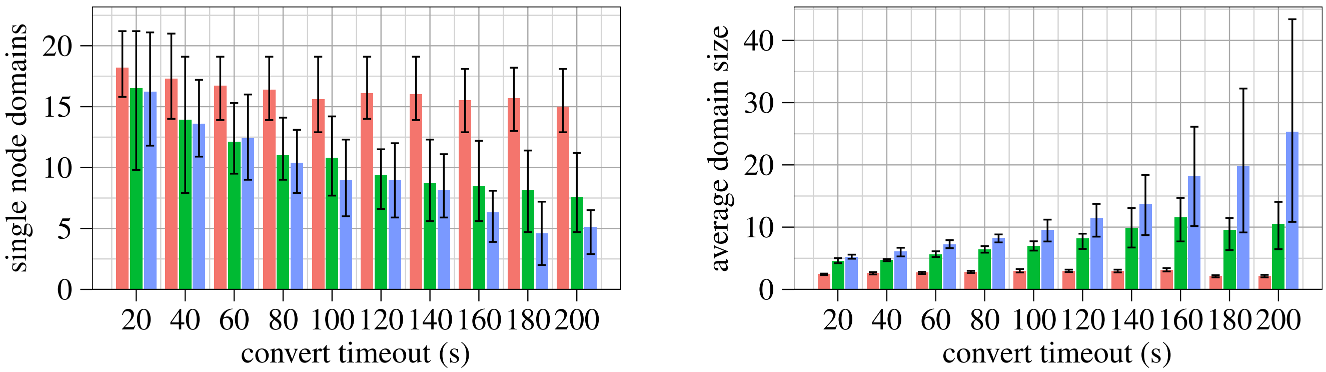


Figure 4: Percentage of nodes with valid positions using 100 parking cars



(a) Remaining Single Node Domains

(b) Average Domain Size

Figure 5: Resulting domain size for different numbers of parking cars: 100 (red), 300 (green), and 500 (blue)

the same amount of nodes with a valid position. For values higher than 120 s, the cord creation process becomes slightly unstable.

In our simulation experiments, we experimented with up to 500 parkers (data not shown). Using a timeout of 100 s, our dynamic cloud solution is able to integrate at least 80 % of the parkers into the network.

In a second set of plots (Figure 5), we evaluate the size of the resulting VCP cords. We use bar plots showing the mean value with error bars indicating the 10 % and 90 % quantiles. It makes little sense to have a huge number of domains consisting of only one node. This would lead to problems related to data replication and to reduced storage per node. The number of domains that remain in that state is a good indicator for the quality of our approach.

Figure 5a shows the number of remaining single node domains. We observe a positive trend with increasing values for the timeout t_{convert} although the amount of single node domains is not converging to a value close to zero. The main reasons can be seen in the limited communication range, the shadowing of obstacles and the random placement of the nodes, resulting in some nodes being unable to communicate with other nodes. This can be seen especially for low node numbers, i.e., very sparse distributions of parking cars. For higher numbers of parkers, a significant influence of t_{convert} can be observed, again motivating rather large values for this timeout.

The second indicator used to evaluate the different convert delays is the average size of domains. We recorded this value at the end of each simulation run.

Figure 5b summarizes the results. As can be seen, the domains remain small for sparse distributions of parking cars even with higher convert delays as it was expected, because of the already explained physical separation. The relatively small increase of the average domain size with 300 nodes can be explained with the amount of single node domains lowering the average.

The interesting aspect, however, is that only with 500 parking cars a clear increase in the average domain size is visible. The extension of the cords is significantly faster due to the higher node density. This results in more nodes being able to join an adjacent cord in a short time and therefore a faster propagation of the cord.

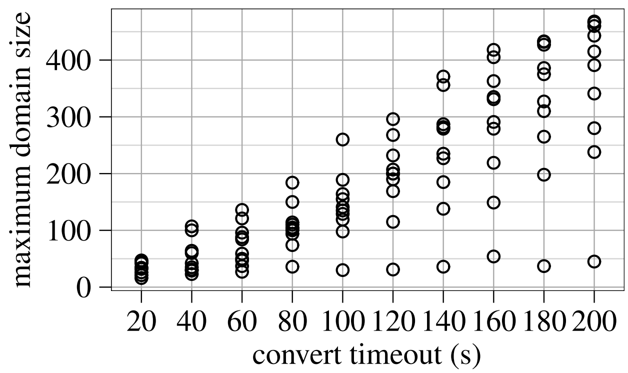


Figure 6: The maximum domain size using 500 parking cars

All the shown evaluations focused on the ability of the implementation to form stable cords with all nodes eventually obtaining a valid position. In combination with the average size of the domains created in the ROI and the overall amount of distinct domains, it was shown that higher values tend to lead to better results. This is especially true for the scenarios with higher node densities.

The picture is however not yet complete. Results from previous experiments demonstrated that VCP can establish cords within a huge amount of nodes [8]. The results also showed that with an increasing cord size especially with high mobility the success rate of routing operations, decreases significantly.

Therefore, we also investigated the maximum domain size in each simulation run. For low densities (e.g., using 100 parkers), large isolated clusters can rarely be observed (data not shown). Yet, for high densities, this seems to be a frequent phenomenon, if also larger timeout values are used.

Figure 6 shows the results for 500 parking cars. Apparently, except for one replication, the nodes build one very large clusters of up to 450 nodes, if timeout values of 140 s and higher are used. With the highest value of 200 s in 50% of the replications more than 80% of the nodes are part of one domain. Such huge domains should be definitely avoided by selecting smaller values.

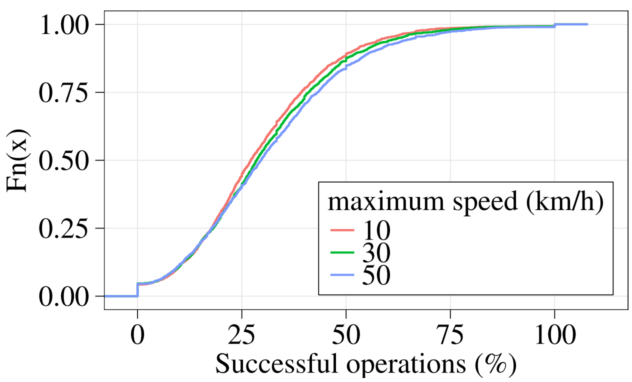


Figure 7: The success rate of lookup operations per car in form of an eCDF

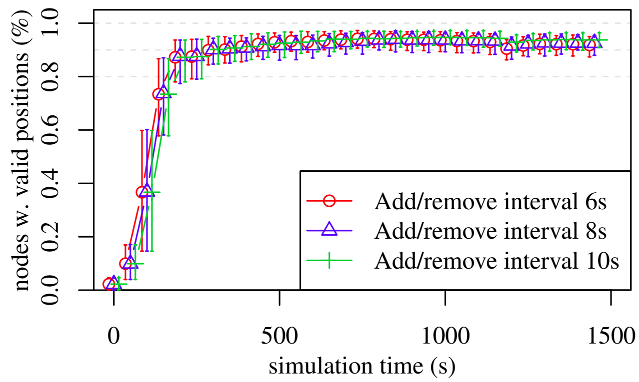


Figure 8: The percentage of nodes with valid position over time, with parking and leaving cars

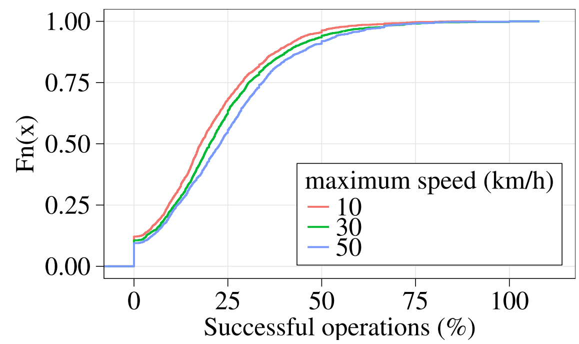


Figure 9: The success rate of lookup operations per car during rush hour in form of an eCDF

In summary, a parameter of $t_{\text{convert}} = 100\,\text{s}$ shows best results. It leads to a high amount of nodes with a valid position, while at the same time limiting the maximum domain size.

5.3 Data Management

We now concentrate on the question how well the storage and retrieval of data works using the established clouds of parking vehicles. For this, we started the application to continuously store and retrieve data. The delay between two `PUBLISH` or `LOOKUP` attempts was set to 5 s to keep the network load at a reasonable level and reflecting a moderate data production and consumption rate. Each data operation was only attempted once. This models uncritical data like traffic information or the information about free parking spaces. If the data is still relevant in the next application loop cycle another storage or retrieval attempt could be initiated.

A look at the expected query delay of VCP, evaluated in other publications, showed that the response to a data operation can be expected within a few hundred ms. Therefore the maximum time to wait for a response to the executed data operation has been configured to 1 s. We now consider dynamic parkers only. We used 300 vehicles out of which at least 100 were parking at any time. The scenario was evaluated with the maximum allowed speed of cars set to the common speed limits in European cities (10 km/h, 30 km/h, and 50 km/h). Please note that the set value resembles the maximum speed a car is allowed to drive. A car still breaks at intersections, and accelerates afterwards. Therefore the speed of a car is not constantly at the set value. Each simulation was repeated ten times in order to increase statistical confidence.

5.3.1 *Parking and Leaving Cars*

For the first set of results, we added and removed parked cars uniformly over the simulation time. Therefore, the overall amount of cars remained more or less stable. In this set of simulations, we explored three different add/remove intervals of parked vehicles (6 s, 8 s, and 10 s).

The overall picture of the success rate of lookup operations is depicted in Figure 7 in form of an eCDF (we show data for the 6 s add/remove interval only, the results for the other intervals show a similar behavior). We observe a trend that the success rate increases with a higher mobility of the driving cars. With a speed of 50 km/h the amount of cars with more than 25 % success rate is about 5 % higher than with 10 km/h. Yet, as can be seen in this figure, the overall performance is only marginally dependent on the speed of the vehicles.

To better understand the behavior of the overall system, we further plot the percentage of nodes with a valid position in Figure 8. In this plot the data is grouped by the add/remove interval. It is clearly visible that VCP has no problem dealing with the addition and removal of new nodes. After an initial transient period, most of the nodes are able to obtain a valid position. Different intervals to add or remove parked cars have no critical influence.

5.3.2 *Rush hours*

VCP can obviously deal very well with the appearance and removal of nodes. However, it is unclear whether an increase in the frequency of such events may affect the stability of cords. We therefore evaluate this scenario, which corresponds to rush hour traffic and the related parking events. In this rush hour scenario, a higher amount of cars leave in the morning and return in the evening. This was modeled by a temporary decreased time between parking events for a limited duration. Separate simulations were performed modeling a morning rush hour with a higher leave interval and an evening rush hour with a higher join interval of nodes. Each rush hour started at a simulation time of 500 s. In the following, results for the morning rush hour and a duration of 500 s are presented.

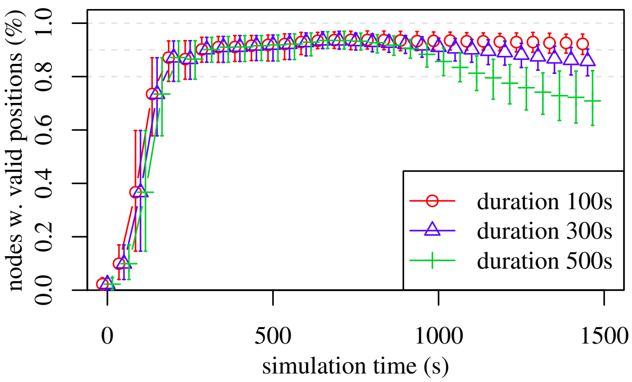


Figure 10: The percentage of nodes with valid position over time during rush hour

We show the results for the highest vehicle density (300 parkers) in Figure 9. About 50 % of the cars have success rate of below 25 %. In this rush hour scenario, the speed of the vehicles has a much larger influence compared to the more stationary setup.

Plots for the percentage of nodes with a valid position make the explained stability changes of the cords more apparent. Figure 10 shows the results for the moring rush hour, i.e., for a decreasing number of parking cars over time. We distinguish three different lengths of the rush hour (100 s, 300 s, and 500 s). The increasing amount of cars parking also increases the overall amount of nodes with a valid position. This can be seen at a simulation time of 100 s, which marks our rush hour.

The negative effect of leaving cars is also very clear in Figure 10. With a duration of 300 s the cords already begin to struggle. This is even worse for 500 s. Interesting is that the decrease in the curve continuous after the rush hour is over. About 100 cars are leaving during the rush hour, which leads to unstable cords. New parked cars need more time to join a cord, since the cord is busy repairing the relationships. This leads to a decrease in the amount of nodes with a valid position even after the end of the rush hour.

6. CONCLUSION

We studied the potential of using parked vehicles to establish a vehicular cloud able to store information and to provide network connectivity between moving vehicles and the vehicle storing the requested data. We made use of a virtual coordinate based routing solution, namely Virtual Cord Protocol (VCP), which provides Distributed Hash Table (DHT) services and integrated greedy routing with guaranteed delivery. Our simulation results clearly indicate the feasibility of our approach, which is able to cope with the dynamics of vehicles parking and leaving even at rush hour times. The key challenge was to enable communication between the moving vehicles and the clouds of parked vehicles. This has been realized using inter-domain routing techniques for information exchange between single-node moving networks and larger scale parking clouds. Future work include optimized inter-domain routing techniques to even bridge the gap between several clouds and to enable data replication between these networks.

7. REFERENCES

[1] Wireless Access for Vehicular Environments. Draft Standard P802.11p/D4.0, IEEE, March 2008.

[2] A. Awad, R. German, and F. Dressler. Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks. *IEEE Transactions on Mobile Computing*, 10(9):1214–1226, September 2011.

[3] A. Awad, C. Sommer, R. German, and F. Dressler. Virtual Cord Protocol (VCP): A Flexible DHT-like Routing Service for Sensor Networks. In *5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2008)*, pages 133–142, Atlanta, GA, September 2008. IEEE.

[4] J. Balen, G. Martinovic, K. Paridel, and Y. Berbers. PVCM: Assisting multi-hop communication in vehicular networks using parked vehicles. In *4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2012)*, pages 119–122, Oct 2012.

[5] J. Barrachina, P. Garrido, M. Fogue, and F. J. Martinez. D-RSU: A Density-Based Approach for Road Side Unit Deployment in Urban Scenarios. In *IEEE Intelligent Vehicles Symposium 2012, Workshops*, Alcalá de Henares, Spain, June 2012. IEEE.

[6] R. Crepaldi and R. Kravets. Governing Energy for Parked Cars. In *10th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2013)*, pages 87–94, Banff, AB, Canada, March 2013. IEEE.

[7] F. Dressler, A. Awad, and M. Gerla. Inter-Domain Routing and Data Replication in Virtual Coordinate Based Networks. In *IEEE International Conference on Communications (ICC 2010)*, Cape Town, South Africa, May 2010. IEEE.

[8] F. Dressler and M. Gerla. A Framework for Inter-Domain Routing in Virtual Coordinate based Mobile Networks. *ACM/Springer Wireless Networks (WINET)*, 19(7):1611–1626, October 2013.

[9] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz. Inter-Vehicle Communication - Quo Vadis. *IEEE Communications Magazine*, 2014. to appear.

[10] D. Eckhoff, C. Sommer, R. German, and F. Dressler. Cooperative Awareness At Low Vehicle Densities: How Parked Cars Can Help See Through Buildings. In *IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston, TX, December 2011. IEEE.

[11] M. Gerla. Vehicular Cloud Computing. In *11th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2012)*, pages 152–155, Ayia Napa, Cyprus, June 2012. IEEE.

[12] N. Liu, M. Liu, G. Chen, and J. Cao. The Sharing at Roadside: Vehicular Content Distribution Using Parked Vehicles. In *31st IEEE Conference on Computer Communications (INFOCOM 2012), Mini-Conference*, Orlando, FL, March 2012. IEEE.

[13] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao. PVA in VANETs: Stopped cars are not silent. In *30th IEEE Conference on Computer Communications (INFOCOM 2011), Mini-Conference*, pages 431–435, Shanghai, China, April 2011. IEEE.

[14] F. Malandrino, C. Casetti, C. F. Chiasserini, C. Sommer, and F. Dressler. The Role of Parked Cars

in Content Downloading for Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 2014. to appear.

[15] P. Salvo, F. Cuomo, A. Baiocchi, and A. Bragagnini. Road Side Unit coverage extension for data dissemination in VANETs. In *9th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2012)*, pages 47–50, Courmayeur, Italy, January 2012. IEEE.

[16] C. Sommer, D. Eckhoff, and F. Dressler. IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation. *IEEE Transactions on Mobile Computing*, 2014. to appear.

[17] C. Sommer, D. Eckhoff, R. German, and F. Dressler. A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments. In *8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011)*, pages 84–90, Bardonecchia, Italy, January 2011. IEEE.

[18] C. Sommer, R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, January 2011.

[19] Y.-T. Yu, T. Punihaole, M. Gerla, and M. Sanadidi. Content routing in the Vehicle Cloud. In *IEEE Military Communications Conference (MILCOM 2012)*, pages 1–6, Orlando, FL, October 2012. IEEE.