
# VEHICULAR 2015

The Fourth International Conference on Advances in Vehicular Systems, Technologies and Applications

ISBN: 978-1-61208-434-3

October 11 - 16, 2015

St. Julians, Malta

**VEHICULAR 2015 Editors**

Vladimir Sulc, Microrisc, s.r.o., Czech Republic

Josef Noll, University Graduate Center (UNIK), Norway

# VEHICULAR 2015

# Forward

The Fourth International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2015), held between October 11 - 16, 2015 - St. Julians, Malta, continued a series of events considering the state-of-the-art technologies for information dissemination in vehicle-to-vehicle and vehicle-to-infrastructure and focusing on advances in vehicular systems, technologies and applications.

Mobility brought new dimensions to communication and networking systems, making possible new applications and services in vehicular systems. Wireless networking and communication between vehicles and with infrastructure have specific characteristics from other conventional wireless networking systems and applications (rapidly-changing topology, specific road direction of vehicle movements, etc.). These led to specific constraints and optimizations techniques; for example, power efficiency is not as important for vehicle communications as it is for traditional ad hoc networking. Additionally, vehicle applications demand strict communications performance requirements that are not present in conventional wireless networks. Services can range from time-critical safety services, traffic management, to infotainment and local advertising services. They are introducing critical and subliminal information. Subliminally delivered information, unobtrusive techniques for driver's state detection, and mitigation or regulation interfaces enlarge the spectrum of challenges in vehicular systems.

The conference had the following tracks:

- Fundamentals on communication and networking
- Challenges

Similar to the previous edition, this event attracted excellent contributions from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the VEHICULAR 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to VEHICULAR 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the VEHICULAR 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope VEHICULAR 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of vehicular systems, technologies and applications. We also hope that St. Julians, Malta provided

a pleasant environment during the conference and everyone saved some time to enjoy the beauty of the city.

**VEHICULAR 2015 Chairs**

**VEHICULAR Advisory Committee**
Sriram Chellappan, Missouri University of Science and Technology, USA
João Dias, Universidade de Aveiro, Portugal
Carl James Debono, University of Malta - Msida, Malta
Hassan Ghasemzadeh, Washington State University, USA
Johan Lukkien, Eindhoven University of Technology, The Netherlands
Matthias Uwe Pätzold, University of Agder - Grimstad, Norway
Tapani Ristaniemi, University of Jyväskylä, Finland
Hwangjun Song, POSTECH (Pohang Univ of Science and Technology) - Pohang, Korea

**VEHICULAR Industry/Research Chairs**
Alexandre Bouard, BMW Forschung und Technik GmbH, Germany
Daniel Jiang, Mercedes-Benz Research & Development North America, USA
Peter Knapik, Volkswagen AG, Germany
An He, Qualcomm, USA
Khelifa Hettak, Industry Canada / Communications Research Centre - Nepean, Canada
Wenjing Wang, Blue Coat Systems, Inc., USA

**VEHICULAR Publicity Chairs**
Sangmi Moon, Chonnam National University, South Korea

# VEHICULAR 2015

## Committee

### VEHICULAR 2015 Advisory Committee

Sriram Chellappan, Missouri University of Science and Technology, USA
João Dias, Universidade de Aveiro, Portugal
Carl James Debono, University of Malta - Msida, Malta
Hassan Ghasemzadeh, Washington State University, USA
Johan Lukkien, Eindhoven University of Technology, The Netherlands
Matthias Uwe Pätzold, University of Agder - Grimstad, Norway
Tapani Ristaniemi, University of Jyväskylä, Finland
Hwangjun Song, POSTECH (Pohang Univ of Science and Technology) - Pohang, Korea

### VEHICULAR 2015 Industry/Research Chairs

Alexandre Bouard, BMW Forschung und Technik GmbH, Germany
Daniel Jiang, Mercedes-Benz Research & Development North America, USA
Peter Knapik, Volkswagen AG, Germany
An He, Qualcomm, USA
Khelifa Hettak, Industry Canada / Communications Research Centre - Nepean, Canada
Wenjing Wang, Blue Coat Systems, Inc., USA

### VEHICULAR 2015 Publicity Chairs

Sangmi Moon, Chonnam National University, South Korea

### VEHICULAR 2015 Technical Program Committee

Taimoor Abbas, Volvo Car Corporation, Sweden
Aydin Akan, Istanbul University, Turkey
Waleed Alasmary, University of Toronto, Canada
Marica Amadeo, University of Reggio Calabria, Italy
Andrea Baiocchi, SAPIENZA University of Rome, Italy
Irina Balan, Ghent University - IBBT, Belgium
Gaurav Bansal, Toyota InfoTechnology Center, USA
Michel Basset, Université de Haute Alsace, France
Melike Baykal-Gursoy, Rutgers University, USA
Monique Becker, Institut Mines Telecom, France
Luis Bernardo, Universidade Nova of Lisboa, Portugal

Carlo Vallati, University of Pisa, Italy
Rens W. van der Heijden, Institute of Distributed Systems - University of Ulm, Germany
Wantanee Viriyasitavat, Mahidol University, Thailand
Ljubo Vlacic, Griffith University, Australia
Wenjing Wang, Blue Coat Systems, Inc., USA
You-Chiun Wang, National Sun Yat-Sen University, Taiwan
Chih-Yu Wen, National Chung Hsing University - Taichung, Taiwan
Yue Wu, Shanghai Jiaotong University, China
Weidong Xiang, University of Michigan - Dearborn, USA
Jinyao Yan, Communication University of China, China
Zheng Yan, Aalto University - Espoo, Finland / Xidian University Xi'an, China
Wei Yuan, Huazhong University of Science and Technology - Wuhan, China
Peng Zhang, Xi`An University of Posts and Telecommunications (XUPT), China
Wensheng Zhang, Iowa State University, USA
Zhangbing Zhou, China University of Geosciences - Beijing, China & TELECOM SudParis, France
Haojin Zhu, Shanghai Jiao Tong University, China
Yanmin Zhu, Shanghai Jiao Tong University, China

**Copyright Information**

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

**Table of Contents**

# Communication Protocol for a Swarm Based Routing Algorithm Using the IEEE 802.11p Standard

Christian Stolcis, Steve Zakrzowsky and Wilhelm R. Rossak
Chair of Software Technology
Friedrich Schiller University
Jena, Germany
Email: {Christian.Stolcis, Steve.Zakrzowsky, Wilhelm.Rossak}@uni-jena.de

***Abstract*—Within the next few years, cars will be able to communicate with their surrounding infrastructure, with other cars and even drive autonomously. This allows a new generation of applications to improve security and a better use of space and resources. One of these applications is represented by the *Clustered Swarm* algorithm. The *Clustered Swarm* algorithm is a live swarm based algorithm for vehicles which pursues the aim of a global traffic optimisation by performing a massive load balancing of all road participants to improve the individual routes of each user. Hereby it represents a potential solution for the traffic jam problem. This paper introduces the current communication protocol used by the *Clustered Swarm* algorithm which is designed to ensure the integrity of the data, as well as to reduce the amount of transmitted data to enable the use of current and future vehicle-to-vehicle technologies such as the IEEE 802.11p standard.**

***Keywords-IEEE802.11p; data compression; swarm intelligence, traffic optimisation.***

## I. INTRODUCTION

Current technologies in the area of direct vehicle-to-vehicle communication, most notably the IEEE 802.11p [1] standard, have some limitations regarding their bandwidth and their data rate. Especially for applications where a large amount of data needs to be transmitted to many communication partners within a short time window, the current characteristics of the 802.11p standard are insufficient. And so they are for the *Clustered Swarm* algorithm [2], which uses direct vehicle-to-vehicle communication to perform a load balancing of all road participants. For best communication results, regarding the amount of data transmitted, the *Clustered Swarm* algorithm uses its own communication protocol, which will be presented in the present paper.

In Section II, we will give a short overview of the *Clustered Swarm* algorithm for a better understanding, followed by the used communication model in Section III and the communication protocol in Section IV. Finally in Section V, we will discuss the results and in Section VI, we will give a brief outlook for our next steps regarding further improvements of the communication protocol to comply even more with the requirements of the *Clustered Swarm* algorithm.

## II. CLUSTERED SWARM

Looking at traffic on a microscopic level, traffic consists of many individual participants. But during route calculation, only the personal and individual aims of the different drivers are considered, which are mostly represented by reaching the destination as fast as possible. So, current navigation systems perform an individual *local optimisation* during route calculation. Nevertheless the current traffic situation is considered, the resulting route is only optimised for the single user. If too many local optima are too similar in their characteristics, this can have negative effects on the whole traffic, particularly in combination with general traffic influencing measures. If too many drivers take the same diversion because they follow their navigation systems, a new traffic jam can form very fast. Many local optima can therefore counteract a common *global optimum* under some circumstances.

The *Clustered Swarm* algorithm copes with this deficiency by distributing all participating vehicles on the entire road network based on its capacity without the need of a central instance. In this case, the capacity of a road corresponds to the maximum traffic density $K_{max}$ [3], which is the amount of vehicles on the road at the same time, causing a congestion. Depending on the amount of high weight vehicles and the type and construction state of a road, $K_{max}$ is typically about 150 vehicles/km [3]. This relation between the traffic density and the traffic speed is explained in the Fundamental Diagram of Traffic Flow shown in Fig. 1 and expressed in a few words, the more vehicles the less the traffic speed. The *Clustered Swarm* algorithm takes advantage of this dependence between the amount of vehicles and the traffic speed to perform the load balancing.

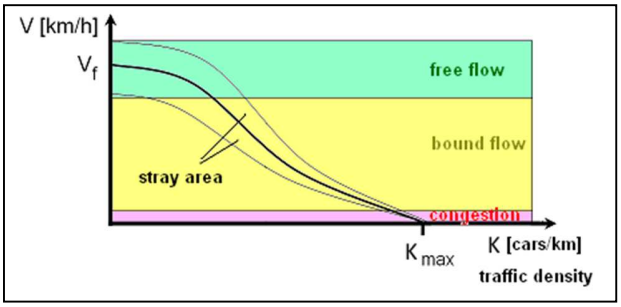


Figure 1. The Fundamental Diagram of Traffic Flow [3]

 ISBN: 978-1-61208-434-3

To accomplish the desired *Emergent Behaviour* [4] following the *Clustered Swarm* algorithm, each vehicle aggregates the estimated traffic density of the road network by exchanging its own route and all other considered routes every time a vehicle is in reach. The first time a vehicle communicates it can only transmit its own route but receives a number of routes from its counterparts. The next time it encounters another vehicle, it transmits its own route and all newly acquired routes. Thereby, it can help spreading knowledge within the swarm, which will be used for route calculation to adapt the routes to the estimated traffic density.

## III. COMMUNICATION MODEL

The key function of the *Clustered Swarm* algorithm is the knowledge and use of the estimated traffic density of the road network. As this information needs to be aggregated, communicated and used in route calculation, a common model is needed to meet the requirements of the three main tasks, especially for communication. As already mentioned, vehicles try to transmit all considered routes. So, the main object is the *Route,* which at the same time is the result of the route calculation. Each *Route* consists of different *PathElements,* which decompose the *Route* into different segments and conform to the edges of the graph used for route calculation. Fig. 2 shows both objects.

The attributes of the *Route* object have the following purpose:

- **VehicleId**: Identifies the vehicle to which the route belongs.
- **VersionNumber**: Current version of the Route. Each time a vehicle recalculates its own route the version number is incremented.
- **NavigationDuration**: Is the total duration the vehicle needs to drive on its route.

The attributes of the *PathElement* object have the following purpose:

- **Id**: Identifies the road within the digital road map used for route calculation.
- **NavigationTimestamp**: Represents the time stamp of the day, when the road will be navigated.
- **DrivingDirection**: Stores the direction in which the vehicle will drive on that road segment.
- **NavigationDuration**: Is the duration the vehicle needs to drive on that road segment.

With the information of the *PathElements,* and therefore the route of a vehicle, each vehicle can estimate how many vehicles will use a road at a given point in time and with this information the vehicle can adapt its own route if necessary due to high traffic density on certain roads [2].

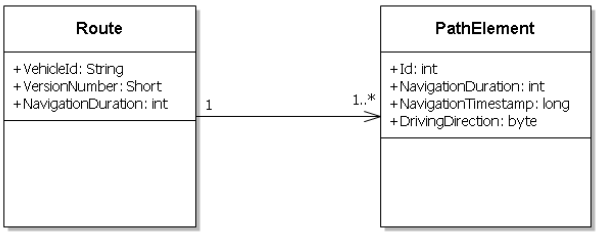


Figure 2. Communication model of the *Clustered Swarm* algorithm

Assuming that only 20% of all vehicles will be able to use the *Clustered Swarm* algorithm, in the near future in Berlin 154,000 vehicles will communicate using a vehicle-to-vehicle technology (in 2012 60.9 % of all registered vehicles in Germany were on the road on working days [5]; Berlin has 1.1 million registered vehicles [6], so, each working day 770.000 vehicles are on the road). If each vehicle communicates with only 50% of all possible vehicles capable of *Clustered Swarm*, each vehicle would save and communicate a maximum of 77,000 routes. A typical route in inner cities has an average amount of 80 road segments (assuming an average length of one trip of 12.3 km [5] and an average length of one road segment of 153.8m for Germany [7]), which corresponds to the *PathElement*. Each *PathElement* has a size of 17 bytes resulting in an average size of 1378 bytes for a Route. In a best case scenario, where the maximum transfer rate of 27 Mbit/s of the IEEE 802.11p standard [8] could be used (which is unlikely), it would take about 30 seconds to transmit the 101 MB of data. In realistic scenarios, where the data rate decreases due to the amount of parallel communication partners, the distances between them, as well as other interferences [8] a transfer of the huge amount of *PathElements* would not be guaranteed. As the vehicles move during communication, the time window for communication could limit the amount of data transmitted even more. This means that there is a big discrepancy between the amount of data to transmit to ensure the functionality of the *Clustered Swarm* algorithm and the transmission rate of the 802.11p standard. To cope with these limitations, we have implemented a communication protocol which reduces the amount of data to be sent while ensuring the integrity of the transmitted data, so that each received data packet contains complete information usable by the *Clustered Swarm* algorithm.

## IV. COMMUNICATION PROTOCOL

### A. Integrity

As the IEEE 802.11p standard is situated in the data link layer of the OSI model [9], the used protocol for transmission has to be defined by the applications using the IEEE 802.11p standard. Given the limitations of the IEEE 802.11p standard compared to the requirements of the *Clustered Swarm* algorithm, a communication protocol is needed, which allows to transmit as much information as possible within a short time window ensuring that received data is usable by the receiver. The integrity of the data is especially important in this case. Since the amount of transmitted data is restricted by the short communication time of two vehicles passing by each other, it is even more necessary that the actual transmitted data can be used by the receiver. That means, the information needs to be sent in a way that each received data packet contains sound data independent of other data packets and thereby usable by the *Clustered Swarm* algorithm.

In common communication protocols (e.g., TCP/IP), the data to be transmitted is distributed among different packages

 ISBN: 978-1-61208-434-3 

TABLE I. SPACE SAVINGS OF DIFFERENT COMPRESSION ALGORITHMS

| | 40 PathElements Avg. Size in Bytes (space savings) | 80 PathElements Avg. Size in Bytes (space savings) | 240 PathElements Avg. Size in Bytes (space savings) | Max. PathElements per Frame |
|---|---|---|---|---|
| Without Compression | 698 (0 %) | 1378 (0 %) | 4098 (0 %) | 83 |
| Deflate | 475.4 (31.8 %) | 858.8 (37.6 %) | 2360.1 (42.4 %) | 141 |
| LZMA | 412.9 (40.8 %) | 753.8 (45.2 %) | 2062.6 (49.6 %) | 163 |
| Gravity MDT | 255.7 (63.3 %) | 475.4 (65.5 %) | 1356.3 (66.9 %) | 255 |
| Deflate + Gravity MDT | 254.7 (63.5 %) | 459.8 (66.6 %) | 1238.0 (69.7 %) | 295 |
| LZMA + Gravity MDT | 259.4 (62.3 %) | 453.8 (67.0 %) | 1195.6 (70.8 %) | 307 |

depending on the Maximum Transfer Unit (MTU) [10]. By specifying the Maximum Transfer Unit size for a network compatible device, including header and protocol-meta-information, the MTU defines the amount of data for one frame [11]. Data packages exceeding the size defined by the MTU are fragmented and distributed over multiple frames. In order to obtain complete and usable data this procedure requires all frames to be received. If one frame is lost, all others have to be resent [10]. This behaviour can lead to major radio transmission interferences, wherefore a fragmentation should be avoided to meet the requirements of the *Clustered Swarm* algorithm.

As described in Section III, the information transmitted by each vehicle is represented by the *Route* object. Since one *Route* represents a complete set of data usable by the receiver, we decided to follow a "one route one frame" approach to guarantee data integrity. This means, one data packet represented by one *Route* should completely fit into the payload of one transmitted frame. To provide enough space to hold a *Route* and at the same time avoid too much data loss in case of a communication failure, we use a size of 1500 bytes as the maximum data packet size. Also, the fact that this size represents the standard for transmitting data over Ethernet since over 30 years, fortified our decision [10]. If a frame is lost, it does not affect other frames and hereby other *Routes*, so that the small time window for communication can be used in the best possible way.

Since the average route size is 1378 bytes (see Section III), longer routes would not fit into the 1500 bytes. To be able to store even long routes, a reduction of the data was needed.

### B. Compression

Thinking about possibilities to reduce the amount of data, a suggestive approach is the compression of the data. Many different algorithms exist (e.g., LZ77 [12], Deflate [13] and LZMA [14]), which are available for various programming languages and allow an easy integration into own applications. But in our special case, the existing algorithms showed low compression ratios (see Table I and Fig. 4), too low for the use in the *Clustered Swarm* algorithm. This mostly relies on the "black box" approach of existing compression algorithms where no semantic information about the data is considered [14]. As the structure of the communication model introduced in Section III offers many possibilities to reduce the data on a logical level, we developed our own compression algorithm called *Gravity MDT Compression*. Basically, the *Gravity MDT Compression* (**G**roup-**V**ar-**I**nt **m**inimal **da**ta compression) combines three approaches to reduce the amount of data without changing its content: **Group-Var-Int-Encoding** [15]**, Delta-Encoding** [14] and **Elimination of redundancy**.

The **Group-Var-Int-Encoding** was developed by Google Inc. on the basis of the *Var-Int-Encoding* [16] which stands for "variable integer" and represents an integer data type that only occupies as many bytes as needed to represent the value. For example, a 32-Bit Integer with the value 1 only needs 1 byte instead of 4 bytes to be stored.

Since the normal *Var-Int-Encoding* stores some extra information to be able to decompress the value, the maximum number of bits to be used for storing a value is 30 [16]. *The Group-Var-Int-Encoding* adds an additional byte to store this extra information which allows to use the full 32-bit-integer value range. The *Gravity MDT Compression* uses the *Group-Var-Int-Encoding* to compress and store the *Ids* and the *NavigationTimestamps* of the *PathElements*.

The **Delta Encoding** or differential encoding is a simple data compression method used to reduce correlating or sequential data [14]. The idea behind the delta encoding is that not the information itself is stored but the difference from an initial state to the current state. Table II shows the *Delta-Encoding* applied to sample *Ids*. So, the Delta Encoding helps to trim the possible big integer values to much smaller values. As the compression ratio of the *Group-Var-Int-Encoding*

TABLE II DELTA ENCODING APPLIED ON SAMPLE PATHELEMENT IDS

| Encoding | Id *PathElement* #1 | Id *PathElement* #2 | Id *PathElement* #3 | Id *PathElement* #4 | Id *PathElement* #5 |
|---|---|---|---|---|---|
| none | 5890234 | 5839494 | 5839274 | 5897947 | 5897366 |
| Delta-Encoding | 5890234 | 50740 | 220 | -58673 | 581 |

 ISBN: 978-1-61208-434-3 

increases with the decreasing size of the values to store, the combination of the *Delta-Encoding* with the *Group-Var-Int-Encoding* allows to reach a very high compression ratio. In the *Gravity MDT Compression* algorithm, it is also used to store the *Id* and *NavigationTimestamp* of the *PathElements*.

**Elimination of redundancy**: Taking a closer look at the communication model, it turns out that the *PathElements* save some redundant information given by the *NavigationDuration* and the *NavigationTimestamp*. Both values are needed by the *Clustered Swarm* algorithm but for communication, one of the values becomes obsolete as both can be calculated considering the other. Since the *PathElements* are stored in the order they are driven during route guidance, the chronological accumulation of the *NavigationDuration* allows the calculation of the *NavigationTimestamp* of all *PathElements* and through the difference of the *NavigationTimestamps* of two following *PathElements* the *NavigationDuration* can be calculated. Also, considering the advantages of the *Group-Var-Int-Encoding* and the *Delta-Encoding*, we decided to store the *NavigationTimestamps* to calculate the *NavigationDuration*. The final structure of the communication protocol considering the three mentioned approaches is visualized in Fig. 3. Each route to be transmitted is converted into this structure and at the same time compressed by applying the three methods.

The first 26 bytes represent a header, which saves information of the *Route* object and some additional meta information used for compression like the *InitialNavigationTimestamp*, the *PathElementCount* and the *PathElementIdsOffset*. The *InitialNavigationTimestamp* is needed by the *Delta-Encoding* as a start value. The *PathElementCount* and the *PathElementIdsOffset* are required since after compressing the *Ids* and the *NavigationTimeStamps* with the *Group-Var-Int-Encoding* the block size is variable. The header is followed by information about the *PathElements,* which are stored in arrays for best compression rate, and save the *DrivingDirections* the *Ids* and the *NavigationTimestamps* of the *PathElements*.

| Block | Field | Size | Σ |
|---|---|---|---|
| Header 1 x Route | VehicleId | 12 Byte | 26 Byte (static) |
| | VersionNr | 2 Byte | |
| | Initial Navigation Timestamp⁺ | 8 Byte | |
| | #PathElement⁺ | 2 Byte | |
| | PathElement IdsSize⁺ | 2 Byte | |
| Driving Direction Array | Driving Direction | n x 1 Byte | n Byte + variable integer size |
| IDs Array | Id | variable | |
| Navigation Timestamps Array | Navigation Timestamp | variable | |

Figure 3. Communication protocol structure

### C. Further compression improvements

A simple way to improve the quality of an algorithm is applying two or more algorithms to the same problem. This is also applicable for the compression of data, by combining the advantages of different approaches to reach a better compression. However, a potential improvement can only be reached in case of the compression ratio since the coding and decoding times increase with the number of the applied algorithms.

Nevertheless, we evaluated the combination of the Deflate and the LZMA with the Gravity MDT algorithm to get the most out of the compression. The results are shown in Table I and in Fig. 4 and Fig. 5, and will be discussed in the next Section.

## V. EVALUATION

The evaluation of the quality of the presented compression algorithms has been determined using JUnit tests. For this purpose we generated a set of *Route* objects with random data. To be able to generate realistic data, the random generator has been restricted. In case of the *NavigationDuration* we used a Gaussian distribution with a mean value of 30 which represents the average duration in seconds to navigate a *RoadElement* retrieved from the used Navteq maps [7]. In addition, the generated number has an upper limit of 3600 (1 hour), so that the range of values for the *NavigationDuration* is [0, 3600].

Fig. 4 shows the determined compression ratios of the different algorithms and combinations of them. The x-axis shows the amount of *PathElements* of a random generated *Route*, and the y-axis shows the arithmetic mean over all generated *Routes* (100,000) with x *PathElements*. The lowest compression ratio of 1.8 has been reached by the Deflate algorithm (blue). This corresponds to a space saving of 45.2% (see Table I). The LZMA algorithm (yellow) reached a compression ratio of 2 which corresponds to a space saving of 53%. The evaluation of the Gravity-MDT algorithm (green) showed a compression ratio of 3.1 and therefore a compression of 68% which is an improvement of 15% compared to the LZMA algorithm.

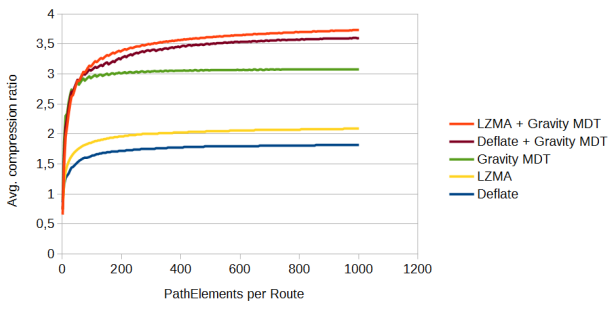


Figure 4: Compression ratios

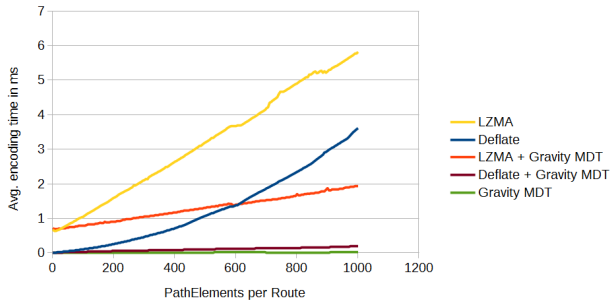


Figure 5: Compression encoding times

As expected, the combined algorithms showed the highest compression ratios, where the LZMA combined with the Gravity MDT algorithm (red) reached the best space saving of 74%.

Taking a look at the encoding times of the different algorithms in Fig. 5, with the amount of *PathElements* on the x-axis and the coding time in milliseconds on the y-axis, the LZMA (yellow) followed by the Deflate (blue) algorithm achieved the slowest encoding times and showed the fastest increase depending on the amount of *PathElements*. The fastest algorithm regarding the encoding times reached the Gravity-MDT algorithm (green), followed by the Deflate combined with the Gravity-MDT algorithm (brown). Since in the combined approaches the Gravity-MDT algorithm is executed first, the reduced amount of bytes to compress lead to very good encoding times for the Deflate-Gravity-MDT combination.

Considering these results for the compression ratio and the encoding times, we decided to use the combination of the Deflate with the Gravity-MDT algorithm in the communication protocol. Despite the slightly inferior results compared to the best algorithms in the two categories, the approach represents the best choice considering both characteristics.

## VI. Conclusion And Future Work

Current mobile communication technologies, especially the IEEE 802.11p standard enable a broad field of new applications and services to improve different aspects of the current road traffic. However, the IEEE 802.11p standard with its current characteristics is not completely suitable for all future applications, and so it is not for the *Clustered Swarm* algorithm. As the quality of the load balancing performed by the *Clustered Swarm* algorithm is highly depending on the amount and the actuality of the transmitted *PathElements,* it is crucial that within the short communication window as much information as possible is transmitted and that the integrity of the data is guaranteed. To achieve this goal, we build a communication protocol that compresses the routes to be transmitted in a way that one route fits into one 1500 bytes sized frame. Since every received frame contains complete and usable data, the communication is failsafe without the need of additional integrity checks, even if a package is lost.

Beside these advantages, the current implementation of the protocol offers some possibilities for improvement. The longer the route, the more *PathElements* and so the more bytes to transmit, regardless the compression ratio. If a route exceeds the size of 1500 bytes, at the moment the route would be truncated and thereby incomplete. However, not all *PathElements* are really necessary for all vehicles in range. A vehicle driving in the opposite direction does not need the information where all other vehicles intend to go. It only needs information about a smaller area regarding its position and its target. To adapt the transmitted information to the receivers' needs and at the same time reduce the amount of *PathElements*, we started elaborating a filtering mechanism. Based on the position of the receiver and a defined radius, the sender transmits only *PathElements* within the given perimeter. Because this directly affects the quality of the calculated routes and hereby the load balancing, the filter will be evaluated to guarantee a maximum reduction for each route to fit into the 1500 bytes while still maintaining the impact on the traffic. Nevertheless this approach also implies that most routes will not need all the 1500 available bytes so that the capacity of one frame is not fully used. Basically, this corresponds to a knapsack problem where a subset of routes needs to be chosen without exceeding the size of 1500 bytes. Since the communication represents the key mechanism of the *Clustered Swarm* algorithm, a suitable solution for the knapsack problem is already in development to get the most out of it.

## References

[1] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe", Communications Magazine, IEEE (Volume:48, Issue: 1), pp. 62-70, January 2010.

[2] C. Stolcis and E. Pfannerstill, "Clustered Swarm – A live swarm based traffic load balancing algorithm against traffic jams", "in press", ITS World Congress, Bordeaux 2015.

[3] Leutzbach, W., "Introduction into the theory of traffic flow", Springer Berlin, 1972.

[4] J. Kennedy and R. C. Eberhart, „Swarm Intelligence", Morgan Kaufmann, March 1997.

[5] „Motor vehicle traffic in Germany 2010 (KiD 2010)", Federal Ministry of Transport, April 2012.

[6] „Vehicle registrations, inventory of motor vehicles and vehicle trailers by town", Federal Ministry of Transport, January 2014.

[7] „GDF 3.0 Reference Manual v42.0", Nokia, April 2012.

[8] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments", Vehicular Technology Conference, May 2008, pp. 2036-2040.

[9] "IEEE Std. 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", Institute of Electrical and Electronics Engineers Inc., 2007.

[10] K. R. Fall and W. R. Stevens, "TCP/IP Illustrated", Volume 1, The Protocols, Second Edition, Addison Wesley 2012.

[11] D. Murray, T. Konziniec, K. Lee, and M. Dixon, "Large MTUs and Internet Performance", IEEE 13th International Conference, June 2012, pp. 82-87.

 ISBN: 978-1-61208-434-3 

[12] S. Kreft and G. Navarro, “LZ77-like Compression with Fast Random Access”, Data Compression Conference (DCC), 2010, pp.239-248.

[13] P. Deutsch, “DEFLATE Compressed Data Format Specification”, version 1.3 in RFC 1951, May 1996.

[14] E. Leavline and D. Singh, “Hardware Implementation of LZMA Data Compression Algorithm”, International Journal of Applied Information Systems (IJAIS), March 2013, pp. 51-56.

[15] J. Dean, “Challenges in building large-scale information retrieval systems: invited talk.”, Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM), 2009.

[16] A. Stepanov, A. Gangolli, D. Rose, R. Ernst, and P. Oberoi, “SIMD-Based Decoding of Posting Lists in CIKM”, International Conference on Information and knowledge management, 2011, pp. 317-326.

 ISBN: 978-1-61208-434-3 

# Advertising Roadside Services using Vehicular Ad hoc Network (VANET) Opportunistic Capabilities

*Kifayat Ullah*[*], *Luz M. S. Jaimes*[*†]
[*]ICMC, University of São Paulo (USP)
São Carlos Campus, Brazil
[†]University of Pamplona, Colombia
e-mails: kifayat@icmc.usp.br, lsantosj@icmc.usp.br

*Roberto S. Yokoyama*[*], *Edson dos Santos Moreira*[†]
[†]ICMC, University of São Paulo (USP)
São Carlos Campus, Brazil
[*]Instituto de Computação, UNICAMP, Campinas, Brazil
e-mails: sadao@lrc.ic.unicamp.br, edson@icmc.usp.br

***Abstract*—Vehicular Ad hoc Networks (VANETs) support a large number of Intelligent Transportation System (ITS) applications, ranging from safety to non-safety purposes. Non-safety applications encompass comfort, infotainment and marketing of services. In order to promote business, alongside roads and highways, and reach to maximum number of customers, the Business Managers (for instance: gas stations, restaurants, hotels, parking lots, coffee shops, supermarkets, etc.) would need a mechanism to advertise their services using a third party broker. Moreover, travellers (drivers, passengers) would need an efficient and cost effective way to discover these services, during their trips. However, the unique characteristics of VANETs (e.g., dynamic topology, high speed and densities, short inter-contact time, etc.) make the deployment of such applications a challenging task. This paper describes the use of the Opportunistic Service Discovery Protocol (OSDP) –a beaconing based protocol for roadside services discovery. We performed extensive simulation experiments to evaluate the performance of different phases of OSDP, under different traffic densities. The results concluded that: a) the number of advertisement packets, in a given region by different Road Side Units (RSUs), effect the success rate of receptions by moving vehicles; b) short inter-contact times (of around 1s) seem to be sufficient for a fast moving vehicle on a highway to receive at-least 70% of the advertisement packets from RSUs; c) the success rate of receiving response packets is highly affected by the density of neighbour vehicles; d) the model suggests that the system can be used to simulate several business models, including a number of advertisement points, their distances to the business's premises and duration that the packets are stored in the cache, etc.**

***Keywords-Vehicular Ad hoc Networks; Opportunism; Services Advertisement; Roadside Services Discovery.***

## I. INTRODUCTION

Vehicular Ad hoc Networks (VANETs) play an important role in Intelligent Transportation System (ITS), by enabling Vehicles to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications. It has become an emerging field of research by gaining attention from different standardization organizations, government agencies, auto-mobile industry and research communities. The Federal Communications Commission (FCC) has already granted 75 MHz of bandwidth, in the 5.9 GHz band for exclusive use of VANETs. The allocated spectrum is called Dedicated Short Range Communication (DSRC), which enables high speed, short to medium range wireless communications (up to 1000m). The spectrum is divided into seven channels of 10 MHz each. The most important channel is the Control Channel (CCH), which is restricted for the use of safety applications and service announcements. The remaining channels are known as Service Channels (SCHs) [1][2]. VANET consists of many components. The most important components are: On Board Units (OBUs), mounted in vehicles, and Road Side Units (RSUs), installed at fixed locations alongside roads and highways [3].

The Institute of Electrical and Electronics Engineers (IEEE) has developed a system architecture for VANETs called Wireless Access in Vehicular Environments (WAVE). WAVE supports two different protocol stacks: Internet Protocol version six (IPv6) and WAVE Short Message Protocol (WSMP). For Physical (PHY) and Medium Access Control (MAC) layers, an amendment has been made to the 802.11a protocol, called IEEE 802.11p. Similarly, IEEE 1609.4 is standardized for coordination among multi-channel operations. Details about these protocols are provided in [4].

Apart from IEEE 802.11p –the de facto standard for VANET– the Long Term Evolution (LTE), developed by 3rd Generation Partnership Project (3GPP), is considered to be a competitor technology for V2I communications only. Its advantages include: high data rate, low latency, support for high mobility, large coverage area and support for Internet based applications (e.g., live streaming, Voice over IP, online gaming and cloud services etc.). However, the LTE does not support V2V communications. A detailed survey about the strength and weakness of LTE and LTE-Advanced (LTE-A), for V2I communications is presented in [5].

In addition to the standardization efforts from IEEE, the European Telecommunications Standards Institute (ETSI) also constitutes a technical committee for VANET communications. The standard developed by the committee is know as ETSI ITS-G5 [6]. Like the WAVE architecture, the lowest layers of ITS-G5 are based on IEEE 802.11p, with its own channel access and usage algorithms [7]. Unlike the WAVE standard, ITS-G5 has no support for WSMP; instead it uses multi-hop geographical routing. Moreover, the standard did not specify support for WAVE Service Advertisement (WSA), which is used for service announcements in WAVE architecture.

These standardization efforts have created a great deal of opportunities for the deployment of a large number of applications. Broadly speaking, these applications are

 ISBN: 978-1-61208-434-3 

divided into two categories: safety and non-safety applications. Safety applications would ensure safety of lives and properties on the roads, by reducing accidents. On the other hand, the non-safety applications would make the travelling experience, for both the drivers and passengers, more enjoyable, by providing access to a large number of services. In this paper, we tackle the challenge of near-by roadside services advertisement and means to discover them by the travelers.

The rest of this paper is organized as follows. Section II gives an overview of the related work. Section III presents our system model. A motivational scenario is provided in Section IV. The different phases of OSDP are briefly discussed in Section V. Experimental setup is explained in Section VI. Results and Discussions are presented in Section VII. Section VIII concludes this paper, with some future directions.

## II. Related Work

In order to promote business and reach a maximum number of customers, the Business Managers (for instance: gas stations, restaurants, hotels, parking lots, bars, supermarkets, etc.) alongside the roads and highways would need a mechanism to advertise their services using a third party broker. Moreover, travellers (drivers, passengers) would need an efficient and cost effective way to discover those services during their trips. To address the issue, different studies regarding service discovery in VANETs have been conducted.

In [8], the authors studied the problem of providing information about the traffic and road conditions to the drivers. To address the problem, they offered an application layer, location based protocol called Vehicular Information Transfer Protocol (VITP). The vehicle requesting the service sends a query towards the target region using multi-hop transmission. Once the query is received by a vehicle inside the targeted area, it is resolved by the peers and the response is sent back towards the source region, where it is broadcast by the underlying network protocol. However, their protocol is not capable of dealing with dense traffic scenarios and large number of requests.

An Address Based Service Resolution Protocol (ABSRP) was proposed in [9]. The authors used IEEE 802.11a wireless interfaces for communication, and IP addresses for RSUs. The RSUs were interconnected using the Internet as a backbone. Vehicles need to be associated with a RSU each time they enter into its range. To locate a near-by service, the vehicle sends a query to its leader RSU (the one with which it is currently associated). If the service is not present at the current RSU, the request is forwarded to the next RSU using its IP addresses by Internet connection. The weakness of their solution is the time spent in the association process between the vehicle and the RSU, followed by granting access to the Internet, which make the communication ineffective.

An Internet-centric architecture for VANETs was suggested in [10]. The architecture consists of three main components: Roadside Gateways (RGs), Roadside Routers (RRs) and Road Vehicles (RVs). The RRs were connected using heterogeneous backbone network. A hybrid (proactive and reactive) service discovery protocol was proposed. There was no direct V2V communication for service discovery. Instead, the RRs collect and store advertisements from advertising vehicles, in the service information table. A vehicle requiring service sends a query to a near-by RR. If the service is present in the information table, the RR replies. Otherwise, the query is forwarded to the appropriate RR, using network layer routing. They also described a proof of concept, in order to verify the accuracy, completeness and correctness of the proposed algorithm. However, they left the simulations to validate the findings, for future work.

In [11], the authors presented two Location-based Vehicular Service Discovery Protocols (LocVSDP): election based-LocVSDP and naive-LocVSDP. Their scheme operates in four phases: advertisement of services, propagation of requests, election for the leader and service response. The driver sends a query to the neighbour Roadside Router (RR), in the Region of Interest (RI). The RR separates the service information from routing information, and sends the request to other RR, if required. A leader, inside each RI, is selected by using an election process, which is responsible for generating service reply and its propagation to the driver. The disadvantage of this approach is that the election phase is executed for each service query, which leads to high overhead and degrades the overall performance.

In a recent study in [12], the authors addressed the problem of providing location based service information to the travellers by proposing an Opportunistic Service Discovery Protocol (OSDP). Their solution is based on layer-2 and makes use of beacons for service advertisement and discovery. However, they did not performed simulation studies to evaluate the performance of their protocol; instead they carried out real experiments using five Access Points (APs), built with IEEE 802.11a interfaces.

The main objectives of this work are: a) to evaluate the performance of OSDP, presented in [12], by performing an extensive set of simulations; b) to extend the service discovery phase of OSDP. This study is different from the above mentioned proposals in numerous ways. First, we implemented our solution on top of WAVE protocol stack (i.e., IEEE 802.11p, 1609.4, and WSMP standards). Second, our solution does not rely on the Internet for resolving queries and responses. Like OSDP, we make use of beacons for service advertisements and discovery. Finally, we present the simulation scenario in a way that the results could be used by users for designing new business models. We performed simulation studies in order to evaluate the performance of OSDP under different traffic scenarios. For V2I communications, we increased the number of services, advertised by RSUs, while for V2V communications, we change the vehicle densities.

## III. System Modelling

In this section, we discuss the main entities and messages of our system.

### A. System Entities

Following are the main entities of our system.

#### 1) Business Manager (BM)

Business Manager (BM) is an entity, e.g., restaurant, hotel, gas station, coffee shop, supermarket, etc., interested

in advertising its services to the travellers. The offered services would include: food menu, price list, special offers, discount on products and available facilities.

*2) The Broker*

The broker is a third party entity, which would be in charge of feeding one or more RSUs in the neighbourhood of the BM premises, accordingly with a Service Level Agreement (SLA). The SLA would include: number of RSUs to broadcast the advertisement packet, frequency of advertisement packet, distance of the premises from which the packets will be advertised, time and day of the week etc. The BM would register their services with the broker, using Internet connections.

*3) Road Side Unit (RSU)*

RSUs are fixed entities, which are installed at fixed locations along the roads and highways. These RSUs would be administered by the brokers, eventually in association with the roads or highways managers. In order to reach a maximum number of customers, the RSU broadcast the registered services –as per SLA details– using push-based strategy.

*4) On Board Unit (OBU)*

The OBU is an important entity, installed in each vehicle, with storage and processing capabilities. When a vehicle enters in the coverage area of RSU, it starts receiving service advertisement packets, using V2I communications. Those packets are cached for a specific amount of time or a certain distance. BM could make special agreement, such as promotions and incentives, in order to stimulate the vehicles to carry their advertisement packets in their caches and, hence, to be able to transmit to other cars opportunistically. Additionally, the OBUs would send and receive query/response messages in V2V communications, using IEEE 802.11p based wireless interface.

*B. System Messages*

In this subsection, we discuss the messages of our proposed system.

*1) Advertisement Packets*

Advertisement is a well known marketing strategy for announcing products and services to the customers. In our system, the BM would register their services with a broker, via Internet for advertising them to the travellers. The registration process is out of the scope of this work. The RSU(s) would then broadcast advertisement packets for the nearby passing vehicles.

*2) Query Packets*

To discover a desired service, the vehicle would broadcast a query message to the neighbour vehicle(s) using a pull-based strategy. The query is said to be successful, if the querying vehicle receives a response from neighbour vehicles. Otherwise, it is unsuccessful. In our experiments, the vehicle repeats sending the query packet every 20s. Additionally, we use single-hop strategy to broadcast the query packet.

*3) Response Packets*

When a vehicle receive a query packet, it will search its local cache for the required service. If one or more services are found, the vehicle would broadcast the response message. For this work, we assumed that all neighbour vehicles –receiving queries– already have the requested services inside their caches.

## IV. MOTIVATIONAL SCENARIO

In this section, we explain our system model by assuming a motivational highway scenario with two lanes on each side. The scenario is depicted in Figure 1. Vehicles enter into the scenario from two different origins. Four RSUs are deployed, on both sides of the roads. RSUs are interconnected using Internet as a backbone. We assume that all RSUs and vehicles are equipped with WAVE devices and IEEE 802.11p based wireless interfaces. Additionally, each vehicle has a built-in Global Positioning System (GPS) device and an OBU, with storage and processing capabilities. Each RSU will broadcast advertisement packets on a regular interval, using beacons, and vehicles would start receiving these advertisements packets, as soon as they enter into the coverage area of an RSU.

Consider a vehicle *v1*, which has received advertisement packets from RSU1 (gas station, coffee shop), RSU2 (hotel, restaurant, shopping mall), RSU3 (hotel, restaurant, shopping mall) and RSU4 (coffee shop, gas station) respectively. *v1* will store and carry all these services information and will opportunistically forward them to other vehicles, upon receiving a query packet. Now suppose that vehicle *v9* is interested in finding information about a near-by gas station. First, *v9* will check its own local cache. If no such information exists, it will broadcast a query packet towards its neighbour vehicles. In this case, only vehicle *v1* will receive the query packet.

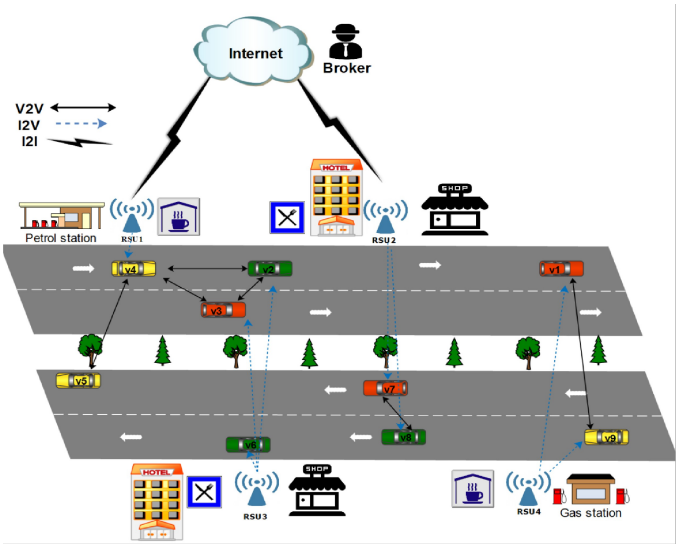


Figure 1. Motivational scenario: services advertisement and discovery.

 ISBN: 978-1-61208-434-3

To resolve this query, vehicle *v1* will search its own cache for gas station information. Since *v1* has information about a single gas station, it will broadcast this information in a V2V communications, using response packet. However, for some reasons, if *v9* does not receive any response packet, it will re-send the query packet, after some time or distance. OSDP protocol uses the unicast approach for sending back responses to the vehicles. We modified the unicast approach, by allowing vehicles to broadcast the response packet. This mechanism would allow other neighbour vehicles to receive advertisement packets without making queries for them.

## V. OSDP PHASES

In this section, we emphasize the main points of OSDP phases. For a detailed description about phases, its operations and message formats, we refer the readers to [12]. It is important to note, that we use single hop communication mechanism in this work for the implementation of all the phases.

### A. Advertisement Phase

This phase refers to the periodical broadcast of advertisements by RSUs, at regular intervals, in a V2I communication scenario. These advertisements contain necessary information about the services (for example: location, name of the service and its descriptions etc.). In our work, we utilize the SCH (Channel No. 174) of IEEE 802.11p protocol, for broadcasting our packets.

### B. Caching Phase

In this phase, the vehicle stores the received advertisements packets in its cache. However, if the service is already present in the cache, it is simply ignored by the vehicle. The main objective behind caching approach is to forward advertisements information, opportunistically, to the querying vehicles, upon requests. We used store-carry-forward mechanism for this phase.

### C. Querying Phase

The purpose of this phase is to discover the services by querying the near-by vehicles in a V2V communication scenario. The vehicle interested in finding a service sends a query message towards the neighbour vehicles. Upon receiving the query, the vehicles check their local caches and if at-least one service is found, it will send towards the requesting vehicle, using broadcast.

## VI. SIMULATION SETUP

The aim of this section is to present the simulation tools and parameters selected for this work.

### A. Simulation Tools

The main purpose of this work is to evaluate the performance of OSDP and elaborate its capabilities, under different traffic densities. To achieve this goal, we carried out extensive experiments with different set of parameters. Nowadays, a large number of simulation tools –ranging from open source to commercial products– are available, therefore, an important aspect of performing VANET simulations is to be cautious in the selection of the appropriate simulator tool. A comprehensive survey about current simulators, their capabilities and approaches is provided in [13].

The first step required to perform VANET simulations, is to use a realistic mobility simulator. Mobility simulator is responsible for defining road networks and generate traffic flows. We used Simulation of Urban Mobility (SUMO) [14], a well known microscopic and open source mobility simulator. After defining road network and traffic flows, the next important step is to enable the vehicles to talk to each other and road side infrastructure. This is achieved by using a network simulator. We used an object-oriented modular discrete event network simulation framework called Objective Modular Network Testbed in C++ (OMNET++) [15]. OMNET++ represents each vehicle as a node inside the network and allows communication among these nodes. Finally, in order to bridge the gap between the two worlds (SUMO and OMENT++), we used an open source bi-directional simulation framework called Vehicles in Network Simulation (Veins) [16]. Veins couples SUMO with OMNET++ using Traffic Control Interface (TraCI) [17]. Veins already implements WAVE protocol stacks. It is mostly noticeable for IEEE 802.11p, IEEE 1609.4 multi-channel operation and comprehensive MAC and PHY layers models. We implemented the OSDP protocol on top of WSMP and IEEE 802.11p.

### B. Simulation Parameters

We defined the traffic parameters inside SUMO. One of the most important parameter is vehicle velocity. In our experiments, we used constant velocity of 120km/h for all vehicles. We performed our experiments using a highway of 140km, with multiple lanes on both sides. Additionally, we deployed a maximum of 7 RSUs (Section-VII A for details) along roadsides, for advertising the services. Vehicles were injected into the scenario from two different origins, with a fixed interval of 1.6s for the first experiment and 40s for the second experiment.

TABLE I. SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Wireless medium | IEEE 802.11p |
| Transmission rate | 6 Mbps |
| Transmission power | 33 dBm |
| Message type | WSMP data |
| Channel bandwidth | 10 MHz |
| Beacon generation rate | 1 Hz |
| Frequency band | 5.9 GHz |
| Radio propagation model | Simple path loss model [18] |
| Simulation Time | Experiment 1: 250 s<br>Experiment 2: 4200 s |

 ISBN: 978-1-61208-434-3

The parameters we used for our network simulator, are summarized in the Table 1.

## VII. Results and Discussions

This section discusses the results we obtained from our simulations. Based on the OSDP phases (Section-V), we divided our experiments into two parts.

### A. *Evaluation of Advertisement Packets*

In this experiment, we tested the first two phases (advertisement and caching) of OSDP in a V2I communication scenario (see Figure 2). We measured the success rate of advertisements packets received by vehicles, in a given inter-contact time. Success rate refers to the total number of received advertisement packets by the total number of delivered packets. An inter-contact time refers to the duration in which a vehicle continues to receive advertisement packets from an RSU, inside its coverage area. As soon as the inter-contact time expires, the vehicle stops receiving advertisements. We considered the same inter-contact times (ranging from 0.2s to 2.0s) used by OSDP. We control the inter-contact time by simTime() function of OMNET++. Similar to OSDP, we increased the density of RSUs, offering the services, from 1 RSU to 7 RSUs.

Each RSU broadcasts a single advertisement packet per second. In OSDP, all APs (which emulate RSUs) were deployed in the same location, therefore, we followed the same setup for the deployment of RSUs in our experiment. To achieve more reliable results, we calculated the average of samples collected from 100 vehicles, for each inter-contact time.

Figure 3 shows the success rate of advertisement packets received by vehicle in relation with inter-contact time. In general, the success rate increases as the inter-contact time increases. In case of a single service offered by an RSU, inter-contact time of 1s seems to be sufficient, for a vehicle to successfully receive the advertisement packets. It could also be observed that inter-contact time of 0.8s is enough for a vehicle to receive at-least 50% of the offered services, by at-most 7 RSUs. Moreover, the success rate decrease as the number of offered services increase. The reason of this performance degradation is likely due to packet collision at the MAC layer.

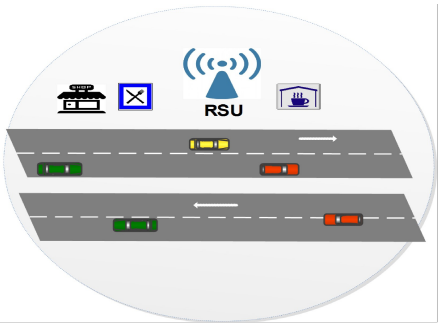


Figure 2. RSU broadcasting service advertisements .

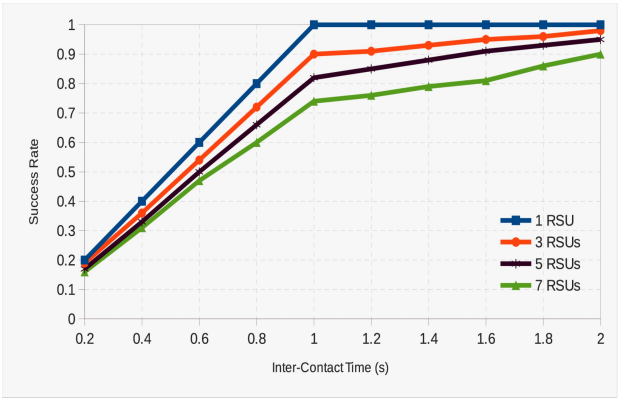


Figure 3. Success rate of service advertisements received by vehicles from RSUs (ranging from 1 to 7), under different inter-contact times.

### B. *Evaluation of Query and Response Packets*

In this experiment, we tested the third phase (service discovery) of OSDP, using V2V communications among vehicles (see Figure 4). We divided our experiment into two sub-categories. In the first experiment, we evaluated the performance of service discovery by changing the number of vehicles responding to the query packet. While in the second experiment, we changed the number of vehicles querying for services.

#### *i) Evaluation of Response Packets*

In this experiment, a single vehicle would query its neighbour vehicles for a required service. We increased the density of neighbour vehicles responding to the query packet, from 1 vehicle to 21 vehicles. Like OSDP, we assumed that all neighbour vehicles have already received and stored the required service in their local caches. Success rate in this case, refers to the total number of received response packets by the total number of neighbour vehicles in that region. Vehicle density means, total number of neighbour vehicles inside the coverage range of querying vehicle.

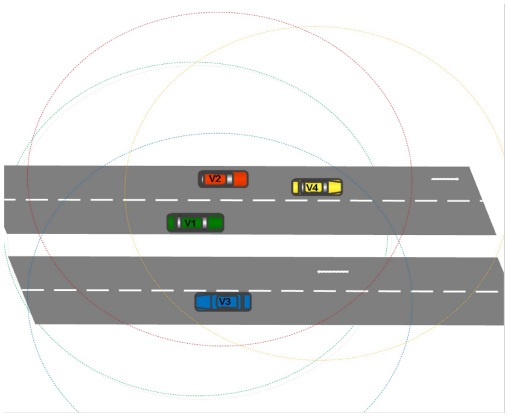


Figure 4. Vehicles querying for/responding to services.

 ISBN: 978-1-61208-434-3

To achieve more reliable results, we calculated the average of samples collected from 100 vehicles for each density. In OSDP, neighbour vehicle send back the response message in a unicast manner. We amended the unicast mode and permitted vehicles to broadcast the response packets.

An important conclusion of OSDP was that inter-contact time does not effect the success rate. Therefore, we consider constant inter-contact time of 0.2s, for all vehicle densities.

Figure 5 depicts that the success rate of response packets is highly dependent on the vehicle density, in a V2V communication scenario. It could be observed that, the success rate is 100%, for traffic densities of up-to 9 vehicles. However, the performance degrades to 50%, as soon as the traffic density reaches 21 vehicles. The reason behind this degradation in performance is due to the attempt made by vehicles to access the channel at the same time.

*ii) Evaluation of Query Packets*

In this experiment, we extend the OSDP by increasing the traffic density, querying for services from 1 vehicle to 21 vehicles. We consider only a single neighbour vehicle to respond to those query packets. We made the assumption that, all the vehicles are interested to search for the same service. The performance evaluation is depicted in Figure 6. It could be observed that, it is difficult to achieve 100% success rate, even in case of small traffic density (e.g., 3 vehicles). The success rate is 50% up-to traffic density of 7 vehicles. Performance further decreases, as the number of vehicles asking for services increases, and it reaches 10% for traffic density of 21 vehicles. We observed a speedy performance degradation in this experiment (as compared to the previous one), because in this case, only a single vehicle is trying to resolve the query packets of different vehicles.

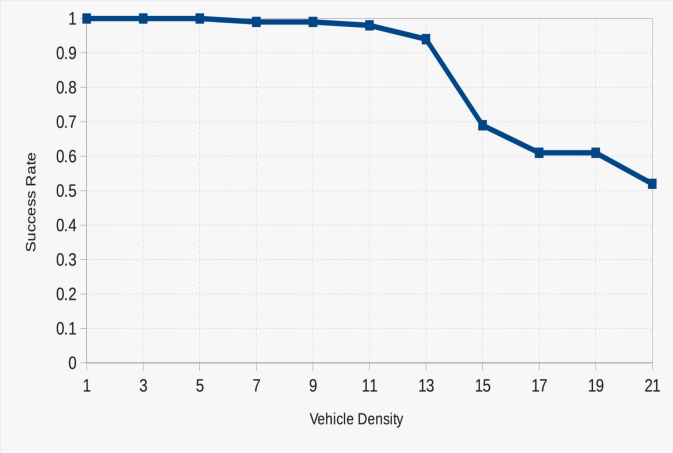


Figure 5. Success rate of received messages with varying number of responding vehicles.

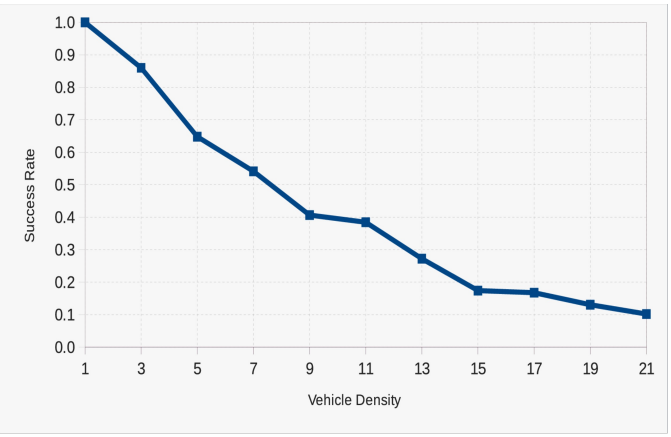


Figure 6. Success rate of received messages with varying number of querying vehicles.

## VIII. Conclusion and Future Work

In this paper, we addressed the problem of roadside services advertisement and discovery in VANET, by implementing OSDP –a recent beaconing based opportunistic service discovery protocol– on top of IEEE 802.11p and WSMP standards, and performed extensive simulations using SUMO, OMNET++ and Veins simulators. To evaluate the performance, we conducted two different set of experiments, under different traffic densities and inter-contact time. In the first experiment, we evaluated the performance of advertisement packets by RSUs in V2I communication scenario. In the second set of experiments, we evaluated the performance of service discovery in V2V communication scenario. We also extended OSDP, by changing the number of vehicles, querying for services. The results conclude that the success rate of service advertisements decreases as the number of offered services, by an RSU, increases. Moreover, success rate of response packets is highly dependent on vehicle densities. The model suggests that the system can be used to simulate several business models, including number of advertisement points, their distances to the business's premises, duration that the packets are stored in the cache, etc.

Presently, we are evaluating the performance of OSDP using more sophisticated path loss models (e.g., Obstacle Shadowing, Two Ray Interference) and Quality of service (QoS) parameters, for more realistic and congested traffic scenarios. As a future work, we would study the security and privacy aspects of OSDP. We also intend to implement OSDP using Named Data Networking (NDN, aka Content-Centric Networking - CCN), for interest based service discovery. We would like to extend OSDP operations to Vehicular Social Network (VSN).

## Acknowledgement

The authors are thankful to the world academy of sciences (TWAS), Italy and to the national council for scientific and technological development (CNPq), Brazil for

providing financial support under the grant number 190275/2011-1.

## REFERENCES

[1] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," in Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE , May 2008 , pp. 2036–2040.

[2] Y. Li, "An Overview of the DSRC/WAVE Technology," in Quality, Reliability, Security and Robustness in Heterogeneous Networks (X. Zhang and D. Qiao, eds.), vol. 74 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, 2012, pp. 544–558.

[3] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A Comprehensive Survey on Vehicular Ad Hoc Network," Journal of Network and Computer Applications, vol. 37, no. 0, 2014, pp. 380 – 392.

[4] R. Uzcategui and G. Acosta-Marum, "WAVE: A Tutorial," Communications Magazine, IEEE, vol. 47, May 2009, pp. 126–133.

[5] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for Vehicular Networking: A Survey," Communications Magazine, IEEE, vol. 51, May 2013, pp. 148–157.

[6] ETSI ES 202 663, "Intelligent Transport Systems; European profile standard on the physical and medium access layer of 5 GHz ITS", Draft Version 0.0.6, October 2009.

[7] D. Eckhoff, N. Sofra, and R. German, "A Performance Study of Cooperative Awareness in ETSI ITS G5 and IEEE WAVE," in Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on, March 2013, pp. 196–200.

[8] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware Services over Vehicular Ad-Hoc Networks using car-to-car Communication," Selected Areas in Communications, IEEE Journal on, vol. 25, Oct 2007, pp. 1590–1602.

[9] B. Mohandas, A. Nayak, K. Naik, and N. Goel, "ABSRP- A Service Discovery Approach for Vehicular Ad Hoc Networks," in Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE, Dec 2008, pp. 1590–1594.

[10] A. Boukerche and K. Abrougui, "A Service Discovery Protocol for Vehicular Ad Hoc Networks: A Proof of Correctness," in Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on , May 2009, pp. 1–8.

[11] K. Abrougui, R. Pazzi, and A. Boukerche, "Performance Evaluation of Location-Based Service Discovery Protocols for Vehicular Networks," in Communications (ICC), 2010 IEEE International Conference on , May 2010, pp. 1–5.

[12] R. Yokoyama, B. Kimura, L. Jaimes, and E. Moreira, "A Beaconing-Based Opportunistic Service Discovery Protocol for Vehicular Networks," in Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on , May 2014 , pp. 498–503.

[13] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A Survey and Comparative Study of Simulators for Vehicular Ad Hoc Networks (VANETs)," Wireless Communications and Mobile Computing, vol. 11, no. 7, 2011, pp. 813–828.

[14] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO-Simulation of Urban MObility-an Overview," in SIMUL 2011, The Third International Conference on Advances in System Simulation , 2011, pp. 55–60.

[15] A. Varga and R. Hornig, "An Overview of the OMNET++ Simulation Environment," in Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Simutools '08, (ICST, Brussels, Belgium), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 60:1–60:10.

[16] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," Mobile Computing, IEEE Transactions on , vol. 10, Jan 2011, pp. 3–15.

[17] A. Wegener, M. Pi orkowski, M. Raya, H. Hellbr uck, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators," in Proceedings of the 11th Communications and Networking Simulation Symposium , CNS '08, (New York, NY, USA), ACM, 2008, pp. 155–163.

[18] K. Wessel, M. Swigulski, A. Kopke, and D. Willkomm, "MiXiM: The Physical Layer An Architecture Overview," in Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools '09, (ICST, Brussels, Belgium), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 78:1–78:8.

 ISBN: 978-1-61208-434-3 

# Public Key Infrastructure and Crypto Agility Concept for Intelligent Transportation Systems

Markus Ullmann* †, Christian Wieschebrink* and Dennis Kügler*
* Federal Office for Information Security
D-53133 Bonn, Germany
Email: {markus.ullmann christian.wieschebrink dennis.kuegler}@bsi.bund.de
† University of Applied Sciences Bonn-Rhine-Sieg
Institute for Security Research
D-53757 Sankt Augustin, Germany
Email: markus.ullmann@h-brs.de

***Abstract*—Secure vehicular communication has been discussed over a long period of time. Now,- this technology is implemented in different Intelligent Transportation System (ITS) projects in europe. In most of these projects a suitable Public Key Infrastructure (PKI) for a secure communication between involved entities in a Vehicular Ad hoc Network (VANET) is needed. A first proposal for a PKI architecture for Intelligent Vehicular Systems (IVS PKI) is given by the car2car communication consortium. This architecture however mainly deals with inter vehicular communication and is less focused on the needs of Road Side Units. Here, we propose a multi-domain PKI architecture for Intelligent Transportation Systems, which considers the necessities of road infrastructure authorities and vehicle manufacturers, today. The PKI domains are cryptographically linked based on local trust lists. In addition, a crypto agility concept is suggested, which takes adaptation of key length and cryptographic algorithms during PKI operation into account.**

***Keywords–Vehicular Ad hoc Networks (VANETs), Vehicle-to-Vehicle Communication (V2V), Vehicle-to-Infrastructure Communication (V2I), Secure Intelligent Transport Systems, Public Key Infrastructures***

## I. INTRODUCTION

Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communication (V2I) (consolidated V2X) has been discussed intensively in recent years. To specify use cases and prepare all the necessary standardizations for V2V and V2I communication, the Car2Car Communication Consortium was initiated by European vehicle manufacturers and supported by equipment suppliers, research organisations and other partners [1]. The results of the technical discussions are a collection of ETSI (European Telecommunications Standard Institute) standards. The first milestone in applying this technology in a realistic setting was the SimTD project with more than 100 vehicles equipped with V2V communication technology in the Frankfurt area in Germany in 2012 and 2013, see [2]. In a next step, the V2X technology will be deployed in large scale intelligent mobility infrastructure projects, for example SCOOP@F [3] in France and the ITS corridor, a joint Intelligent Transportation System (C-ITS) cooperation between Austria, Germany and the Netherlands [4]. In the C-ITS project Roads Work Warning Trailers are equipped with a digital Road Works Warning Gateway (RWWG) to communicate with the bypassing vehicles. This projects mark only the very beginning of ITS technology deployment in Europe. Further plans are already mentioned: the integration of V2X gateways in roadside emergency telephones, sign gantries etc.

The wireless communication technology for cooperative V2V and V2X communication is based on the IEEE 802.11p standard. For this, a frequency spectrum in the 5.9 GHz range has been allocated on a harmonized basis in Europe in line with similar allocations in US. The neccessary specification and standardization is sone by the ETSI. This includes the security standardization as well [5].

According to these standards messages transmitted by vehicles or RSUs shall be digitally signed to guarantee integrity and authenticity. In order to authenticate the corresponding keys a suitable PKI has to be established. A number of practical considerations has to be taken into account when designing such a PKI.

- Many different stakeholders like vehicle manufacturers, transportation infrastructure authorities etc. participate in ITS, especially in multi-national (e.g. European) systems. The PKI should provide flexibility to support different operators managing the vehicles and RSUs in their respective responsibilities.
- Requirements on cryptographic algorithms, domain parameters, key lengths etc. may change over time due to new weaknesses, new attacks or the increase of computer performance. In general, this means that a PKI needs a concept to switch to a new cryptographic setting during its (possibly long) lifetime.
- Revocation of certificates may turn out to be challenging in complex ITS scenarios. A simple mechanism for revoking signing rights should be used.

In this paper, we introduce a multi-domain PKI for ITS based on Local Trust Lists (LTL). This concept considers a IVS PKI domain and different ITS PKI domains. A ITS PKI domain is slightly different to the IVS PKI proposed by the Car2Car Communication Consortium [6]. First, our approach guarantees that the infrastructure components (Road Side Units (RSU)) remain under control of the particular infrastructure authority. Second, the ITS PKI is interoperable with the IVS PKI for the vehicles. This ITS PKI consists of two parts: a Long Term Certification Authority (LT-CA) for

 ISBN: 978-1-61208-434-3 

the identification of RSU gateways and a credential CA (C-CA) for issuing credential certificates to RSU gateways. With the C-CA we take the hostile environment of RSU gateways into account. We assume that attackers are able to manipulate the RSUs like roadside emergency telephone gateways, sign gantry gateways etc. physically. A PKI can not prevent such kind of attacks, but mitigate their effects to a certain degree.

Our ITS PKI proposal supports cryptographic agility in the sense that modifications of cryptographic keys and algorithms during lifetime of the PKI are possible.

Finally, we derive necessary modifications of the existing ETSI certificate format [5] to be compatible to our concept because concepts for the delegation of rights and a crypto agility approach are missing to date. Here, we address only modifications to the ETSI certificate format, which are motivated from an infrastructure perspective. Within this paper we do not analyze the pseudonym concept in depth, which is proposed to assure sender anonymity and message unlinkability for vehicles. (We briefly present this concept in chapter III-B.)

The following sections of this paper are organized as follows: Section II is a description of related work. Section III provides a brief overview of the secure V2V communication specified in the according ETSI standards. Also, the suggested PKI architecture for Intelligent Vehicle Systems, specified in [6], is described. Here, we state the problems if this IVS PKI is used for issuing certificates for ITS RSU gateways, too. In the next Section IV, the multi-domain PKI and ITS PKI concept for RSU gateways and the crypto agility proposal are introduced. Section V briefly addresses security requirements for RSU gateways. Finally, in Section VI we summarize our results.

## II. Related Work

Security and privacy issues in Vehicular Ad hoc Networks (VANETs) are addressed in a lot of research papers. A detailed overview of attacks in VANETs is given by Ghassan Samara et al. in [7]. Di Ma and Gene Tsusik give an overview about security and privacy in emerging wireless networks including VANETs. Overall, a good overview concerning security and privacy in V2X communication can be found in [8]. A detailed analysis of privacy requirements and a comparison with the security requirements in VANETs is given in [9]. Beside that, further security and privacy concepts are presented [10], [11], [12], [13], and [14].

Different trust models for multi-domain PKIs are described in general in [15], [16]. Here, we will follow the naming convention of [16]. It distinguishes between End Entities (EE), that are subject of a certificate (vehicle or RSU gateway), Certification Authorities (CAs), that issue certificates, and root CAs, which are on top of a hierarchy of CAs. In [6] Norbert Bissmeyer et al. suggest a generic PKI for securing V2X communication. The car2car communication consortium adopted this proposal. We outline this IVS PKI in Section III.

## III. Brief Overview Secure V2X Communication

### A. Communication

In the ETSI ITS architecture [17] two different message types are defined. Cooperative Awareness Messages (CAMs) are broadcasted periodically with a maximum packet generation rate of 10 Hz. Based on received CAM messages, vehicles can calculate a local dynamic map of their environment. It is not planned to forward CAM messages hop-to-hop. In contrast, the second message type, Dezentralized Environmental Notification Messages (DENMs) , are event-driven and indicate a specific safety situation, e.g., road works warning (from a RSU gateway) or a damaged vehicle warning (from an IVS gateway). DENM messages can be transmitted hop-by-hop. RWWGs in the C-ITS project transmit DENM messages. Figure 1 illustrates the structure of the DENM message format. For road sign and traffic light gateways etc. new message formats have still to be specified in future.

| Complete Message | | | |
|---|---|---|---|
| | Header | Protocol version | |
| | | Message type | 1 = DENM |
| | | Identifier of the sender | Certificate Identifier |
| | DENM Information | Management Container | Last Vehicle Position (GPS) |
| | | | Event Identifier |
| | | | Time of Detection |
| | | | Time of Message Transmission |
| | | | Event Position (GPS) |
| | | | Validity Period |
| | | | Station Type (Motor Cycle, Vehicle, Truck) |
| | | | Message Update / Removal |
| | | | Relevant Local Message Area (geographic) |
| | | | Traffic Direction (forward, backwards, both) |
| | | | Transmission Interval |
| | | | .... |
| | | Situation Container | Information Quality (low -high, tbd) |
| | | | Event Type (Number) |
| | | | Linked Events |
| | | | Event Route (geographical) |
| | | Location Container | Event Path |
| | | | Event Speed |
| | | | Event Direction |
| | | | Road Type |
| | | A la carte Container | Road Works (Speed Limit, Lane Blockage....) |
| | | | .... |
| | Signature | ECDSA Signature of this message | |
| | Certificate | According Certificate for Signature Verification | |

Figure 1. Examplary message format of DENM. The DENM consists of a header, different data containers, e.g., the management container, a signature and the appropriate certificate.

### B. Security and Privacy Architecture for Secure V2X Communication

To guarantee message integrity and authenticity, all CAM and DENM messages are signed with the cryptographic signature algorithm ECDSA by the sender. Due to privacy requirements (sender anonymity and message unlinkability), the messages are signed using pseudonymous certificates where the used keys and certificates are changed periodically. Therefore, a vehicular gateway has a set of $N$ valid pseudonymous certificates for a period of time. The set size $N$ and the pseudonym change frequency are not specified and can be chosen by the vehicle manufacturer. A Pseudonymous Certification Authority (PCA) is responsible for the issuing of pseudonymous certificates $P_{cert_1} \ldots P_{cert_N}$ to the vehicles. Vehicular pseudonymous certificates $P_{cert}$ can not be revoked. Pseudonymous certificates will only be issued to authenticated vehicles.

To identify a valid vehicle, each vehicular gateway is

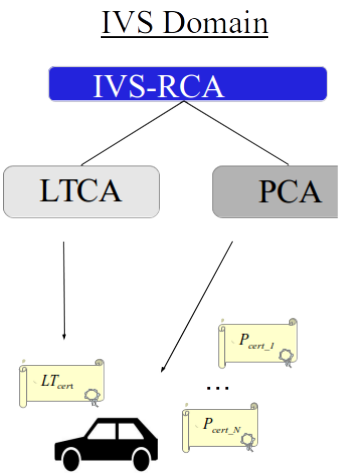


Figure 2. IVS PKI architecture promoted by the car2car communication consortium for Intelligent Vehicular Systems. This PKI consists of the Root Certification Authority (IVS-RCA), the Long Term Certification Authority (LTCA) and the Pseudonym Certification Authority (PCA)

equipped with a long term key pair and a corresponding vehicular long term certificate $LT_{cert}$ for authentication purposes. A key pair and the according long term certificate $LT_{cert}$ are issued to a vehicle at the beginning of the vehicle's lifetime. The issuing process of long term certificates is performed by the Long Term Certification Authority (LTCA). Validity periods of the $LT_{cert}$ and the $P_{cert}$ are not specified to date.

PCA and LTCA operate under a root CA called Intelligent Vehicle System root CA (IVS-RCA). To date, following revocation operations are provided: revocation of a LTCA- and PCA certification authority certificate and revocation of vehicular long term certificates $LT_{cert}$. The architecture of the IVS PKI domain is shown in Figure 2.

Long term certificates and pseudonymous certificates are implemented based on the new ETSI certificate format [5]. This certificate format was designed for the automotive domain and is still not widely applied yet. Primary design principle is shortness of the certificate format due to the necessary transmission over the wireless IEEE 802.11p channel.

### C. Using the IVS PKI for Road Side Units

The IVS PKI domain shown in Figure 2 is proposed by the car2car consortium for issuing certificates to RSU gateways as well. However, security and privacy requirements for vehicles and infrastructure components are not necessarily identical. In contrast to vehicles, RSUs (road work warning, traffic light, ...) do not involve persons during operation comparable to a motorist. Usually they operate without any human supervision. That is the reason that from our point of view, RSU gateways do not have to regard any active privacy concerns. As consequence, RSU gateways do not really need a set of valid pseudonymous certificates at each time. Instead, we propose that RSU gateways need only one Credential Certificate with a specific subject name adressing the RSU for each time frame. Due to security considerations for RSU gateways, see Section V, the validity period of credential certificates should be rather short.

Moreover, arising security weaknesses of the used security technology may be asessed differently by vehicle manufacturers on the one side and an infrastructure authority on the other side. However, the rules of operation for a PKI domain are defined in a single PKI policy, which will be specified by the root certification authority. For this reason, we propose a multi-domain PKI architecture: individual ITS PKIs under control of infrastructure autorities and an IVS PKI under control of the vehicle manufacturers, which are cryptographically linked to each other based on LTLs. So, each individual PKI domain can specify its own PKI policy for their specific needs. In addition, this multi-domain PKI architecture ensures that RSU unit gateways remain under control of the particular infrastructure authority.

The concept of a multi-domain PKI architecture without any superior root CA is not new and already mentioned in [16]. It has been applied globally for electronic passports for many years. Here, any country operates its own root certification authority and has its own local trust list. The different national root certification authorities are cryptographically linked based on local trust lists. This concept works quite well and seems to be a good architecture approach for intelligent transportation systems, too. The benefit of this approach is the possibility to configure PKI domains as needed. A drawback of the multi-domain PKI concept based on local trust list is that each PKI domain has to securely mange is own LTL. More details concerning this issue can be found in Section IV-C.

## IV. ITS PKI Concept

### A. Role of Credential Certificates for Road Side Units

The primary use case for RSU gateways is the transmission of information, e.g., as DENM message to the vehicles using the wireless IEEE 802.11p channel. Due to integrity and authenticity reasons, these messages have to be signed. Therefore, the RSU gateways need specific keys and according certificates. RSUs do not have to regard any privacy concerns, as explained in Section III-C. Technically, this means that RSU gateways do not have to have pseudonymous keys and certificates. Instead, we propose that RSU gateways have only one valid credential key pair and one corresponding credential certificate at each time. Only in the transition phase between two certificate validity periods a RSU gateway has two valid credential certificates $C_{cert_{N-1}}$ and $C_{cert_N}$.

The RSU gateway should be implemented in such a way that it acts in his designated role and transmits DENM messages only if it owns a valid credential certificate. By this a possible misuse of RSU gateways is made more difficult.

### B. ITS PKI Architecture

As mentioned above, we propose that RSUs have only one credential key pair and one corresponding credential certificate $C_{cert}$ at each time. The secret key corresponding to such a $C_{cert}$ is used for signing RSU gateway messages, e.g., DENM messages. For this reason, these certificates have to be implemented according to the ETSI certificate format. Since it is technically challenging to distribute certificate revocation lists (CRLs) to vehicles in time, credential certificates should have a short validity period, for example one day. Thereby implicit revocation of $C_{cert}$ becomes possible by not issuing new credential certificates to RSU gateways. The exact validity period of credential certificates have to be specified according

 ISBN: 978-1-61208-434-3

to a detailed risk assessment concerning the addressed RSU type. For example RWWG are deployed for road works sites which usually are established for one or two days. It may be good practice then to issue a credential certificate with a validity period of a few days to a RWWG shortly before it is deployed.

For authentication purposes, e.g., to obtain credential certificates (for example on a daily basis) an infrastructure component requires a long term identification certificate $LT_{cert}$. These long term certificates $LT_{cert}$ are issued by Long Term Certificate Authority (LT-CAs) during the enrolment of the RSU gateway. A long term certificate $LT_{cert}$ is used within a certificate request for credential certificates towards the C-CA. We suggest that the credential key pair is generated within the secure element of the RSU gateway and the credential certificate is only issued after mutual authentication of RSU gateway and C-CA and only if the $LT_{cert}$ of the RSU gateway is not revoked. Therefore, the LT-CA has to provide a CRL for revoked long term certificates $LT_{cert}$.

A $LT_{cert}$ is only visible inside the ITS PKI and is not transmitted to vehicles. In particular, it is not communicated over the IEEE 802.11p channel. For this reason, we suggest to implement the ITS $LT_{cert}$ according to the X.509 v3 certificate profile. This profile is widely applied and provides all necessary certificate services like time stamping, issuing CRLs etc. The validity period of a $LT_{cert}$ should be at the order of years, e.g., five to six years for RSU gateways like RWWGs. As a rule, 5 to 6 years seems to be reasonable concerning useable cryptography or hardware security vulnerabilities. Due to different certificate issuing policies and certificate formats the LT-CAs and the C-CAs are attached to different root certification authorities, which are termed LT-RCA and C-RCA respectively.

Due to the long validity periods of long term certificates, certificate revocation, implemented as CRL according X.509 v3, is suggested. Once a long term certificate is revoked, no credential certificates are issued to the RSU gateway any more.

Due to the short validity period of credential certificates of RSU gateways, the RSU gateways require an online communication channel, e.g., via GSM to receive new credential certificates.

### C. Crypto Agility

Figure 4 shows how the validity periods of the certificates within the ITS PKI domain relate to each other. The validity periods follow the shell model, i.e. the validity periods of certificates are enclosed in the validity periods of superior certificates.

1) A certificate of a CA is in one of three states: *active*, *passive* or *expired*. After generation of a key pair the according certificate is in state *active*. Over time the certificate state changes from *active* to *passive* to *expired*.
2) A certificate in state *active* is used for issuing certificates to subordinate CAs or RSU gateways.
   - Assume that a LT-RCA root key pair (secret key: ${}^{RCA}LT_{SK\_1}$, public key: ${}^{RCA}LT_{PK\_1}$) is generated at time 0 of Figure 4. The secret key ${}^{RCA}LT_{SK\_1}$ is used to sign and issue a self-certified LT-RCA certificate ${}^{RCA}LT_{cert\_1}$, first. The certificate ${}^{RCA}LT_{cert\_1}$ is in state *active*.
   - The secret key ${}^{RCA}LT_{SK\_1}$ is used to sign CA certificates: ${}^{CA}LT_{cert\_1}$ and ${}^{CA}LT_{cert\_2}$.
   - The certificate ${}^{RCA}LT_{cert\_1}$ switches to state $passive$ at time point I when the next root key pair (secret key: ${}^{RCA}LT_{SK\_2}$, public key: ${}^{RCA}LT_{PK\_2}$) and according certificate ${}^{RCA}LT_{cert\_2}$ are issued. Now, the certificate ${}^{RCA}LT_{cert\_2}$ is in state $active$. A certificate in state $passive$ is not used to issue certificates any longer. However it is still needed to verify already issued subordinate certificates.

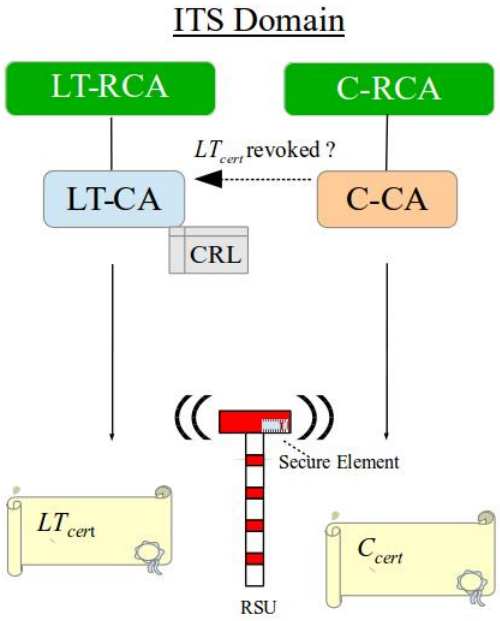


Figure 3. ITS PKI domain architecture. An ITS PKI domain consists of a LT-CA for issuing long term certificates $LT_{cert}$ and a C-CA for issuing credential certificates $C_{cert}$.

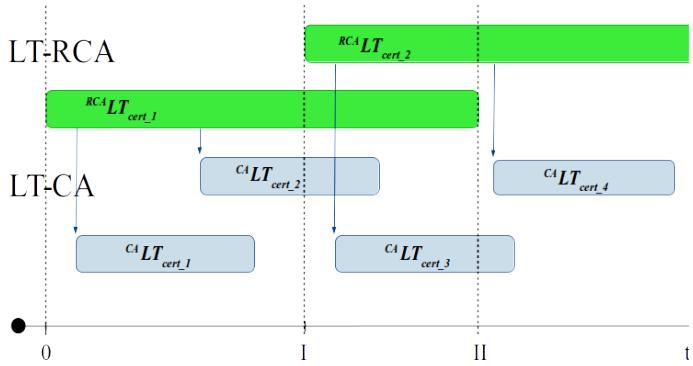


Figure 4. Certificate shell model. The validity period of a certificate is within the validity period of the issuing Certification Authority. E.g., the validity period of ${}^{CA}LT_{cert\_1}$ is within the validity period of ${}^{RCA}LT_{cert\_1}$

At time point II certificate ${}^{RCA}LT_{cert\_1}$ expires.

3) Certificate ${}^{RCA}LT_{cert\_2}$ is termed *Link Certificate* because it is signed with the former LT-RCA secret key ${}^{RCA}LT_{SK\_1}$.

Over long lifetimes the requirements for cryptographic mechanisms are changing. This has implications for the cryptographic mechanisms applied within the PKI domain, too. The cryptographic setting of the PKI has to be adapted according to current cryptographic requirements. All CAs in a PKI have to follow the rules and instructions of the root CA. Therefore, changes of a cryptographic setting for a whole ITS PKI are prescribed by the root certification authority LT-RCA or C-RCA.

Changes to the following components are conceivable:

1) Elliptic Curve Domain Parameter (e.g., because longer key lengths are necessary)
2) Hash algorithm (e.g., due to new hash collision problems)
3) Signature algorithm (e.g., due to weaknesses in the used signature algorithm).

We suggest to implement a new PKI crypto setting by means of a link certificate, assuming that the certificate format allows the specification of cryptographic parameters. Obviously, modifications can only be applied if the infrastructure components are technically able to perform the new algorithms.

The validity period of a $LT_{cert}$ and a $C_{cert}$ differ a lot. A $LT_{cert}$ has a validity period of several years, whereas a $C_{cert}$ has a validity period of few days at most. If the issuing PKIs C-CA and ITS-C-RCA have similar short validity periods with respect to the shell model, the cryptographic settings between $LT_{cert}$ and $C_{cert}$ can differ. In particular, shorter keys can be used for signing $C_{cert}$ towards signing a $LT_{cert}$. Today, the ETSI certificate format only provides the NIST Elliptic Curve Domain Parameter P-256 [18] with 256 bits long secret keys. This key length is sufficient for the very near future. It is however highly probable that longer key length have to be used for long term certificates $LT_{cert}$ in future.

### D. Secure Trust Establishment between PKI domains

An examplary architecture of a multi-domain PKI with three PKI domains (ITS_I, IVS and ITS_II) is shown in Figure 5. In our example there is only one IVS domain with the IVS-RCA to issue certificates for vehicles managed by the vehicle manufacturers and two separate ITS domains ITS_I and ITS_II with the root CAs C-RCA_I and C-RCA_II managed by different infrastructure authorities. These two ITS domains issue credential certificates to RSU gateways in their respective domain. Now trust relations between the different PKI domains have to be established somehow. This can be accomplished by securely exchanging self-signed certificates of the respective root CAs of the PKI domains. Each root CA maintains a LTL containing the certificates of the root CAs of the other domains it trusts. The LTL of a PKI domain is signed (for authentication reasons) and issued to all members of the domain by the root CA, e.g., C-RCA_I manages the LTL for the ITS_I domain. Each PKI domain can individually define the needed rules that are sufficient to trust a separate PKI domain.

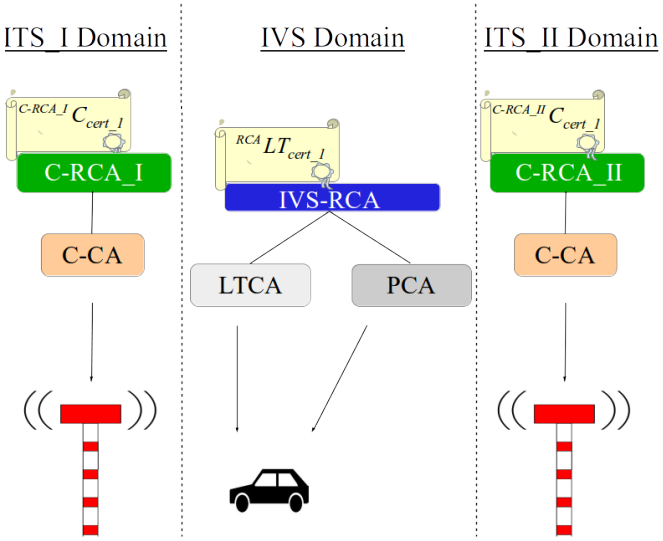


Figure 5. Examplary multi-domain PKI architecture with one IVS domain and two ITS domains: ITS_I and ITS_II.

To verify the authenticity of RSU gateway DENM messages in our examplary architecture, the vehicles have to know the root PKI certificates of the PKI domains ITS_I and ITS_II: ${}^{C-RCA\_I}C_{cert\_1}$ and ${}^{C-RCA\_II}C_{cert\_1}$. If the IVS PKI domain trusts in the ITS_I and ITS_II PKI domains the certificates ${}^{C-RCA\_I}C_{cert\_1}$ and ${}^{C-RCA\_II}C_{cert\_1}$ are elements of the LTL of the IVS PKI domain. If a LTL of a PKI domain is changed all entities of the PKI domain (subordinate CAs and EEs) have to know this information. A time-critical situation arises when one specific PKI domain, e.g., the ITS_I PKI domain loses trust and has to be removed from the LTL of the IVS PKI domain. In this case all affected entities in the IVS PKI domain have to update their LTL as soon as possible.

Based on the currently discussed ITS applications, trust relations between the different ITS domains, here ITS_I and ITS_II, are not really required since no messages are exchanged between these domains. In our example the LTL of the two ITS domains just contain the ${}^{RCA}LT_{cert\_1}$.

### E. Necessary ETSI Certificate Format Adaptations

In our paper, a multi-domain PKI based on LTLs and an according crypto agility concept is presented. The described mechanisms require some adaptation of the current ETSI certificate format.

*a) Elliptic curve cryptography:* The ETSI certificate format regards only Elliptic Curve Cryptography (ECC) performed on NIST domain parameters P-256. These domain parameters have a specific structure to perform ECC calculations very fast. But this structure opens specific side channel attacks. For example, even effective countermeasurements like point blinding and scalar blinding of ECC implementations are not sufficient to resist side channel attacks on NIST ECC implementations, see [19]. Therefore, further cryptographic ECC domain parameters (e.g., brainpool curves) should be added [20].

*b) Rights management:* Fire trucks and police vehicles need specific rights during action. These rights have to be

 ISBN: 978-1-61208-434-3

coded within certificates, too. But only qualified CAs may issue these kind of certificates. The ETSI rights management concept should be enhanced in a sense that a subordinate CA can only assign restricted rights to issued certificates.

*c) Link certificate:* The ETSI certificate has to support link certificates to support crypto agility.

To date, ECDSA is specified as signature algorithm. ECDSA is an appendix signature. Because all entities (vehicles and RSU gateways) share only one wireless communication channel (IEEE 802.11p) it is important to restrict the length of CAMs and DENMs to avoid message collisions on the wireless IEEE 802.p channel at best. An alternative to appendix signatures are signatures with message recovery. For elliptic curve cryptography, e.g., Abes signature scheme with message recovery is applicable, see [21].

## V. Security Requirements for RSU Gateways

RSU gateways operate in a potentially hostile environment. Attackers are able to physically manipulate these RSUs including the electronic gateway components. Also misuse of RSU gateways can not be excluded. First, these RSU gateways need a specific security functionality to resist active attacks and against removal of RSU gateways. But secondly, the PKI architecture has to appropriately regard this attack scenario as well. The idea is that a RSU gateway only acts in its designated role, e.g., as RWWG station, if it owns a valid credential certificate $C_{cert}$.

Moreover, security requirements for RSU gateways should be carefully analyzed and specified, e.g., in form of a Protection Profile (PP) according the Common Criteria.

The RSU gateways have to be satisfy following exemplary security requirements:

1) RSU gateways need a secure storage for cryptographic keys and have to be equipped with side channel resistant implementations of cryptographic algorithms.
2) RSU gateways are resistant against active attacks and removal from the RSU.
3) A RSU gateway is only able to act in his designated role if it owns a valid credential certificate.

If RSU gateways have specific resistance against active attacks they can play an import role as separate trust anchors in a cooperative ITS system, e.g., for implementing secure time synchronization, distribution of CRLs etc.

## VI. Conclusion

The proposed PKI of the Car2Car Communication Consortium for Intelligent Vehicular Systems (IVS PKI) does not regard all needs of RSUs. For this reason we suggest a multi-domain PKI to adequately address the requirements of vehicle manufacturers and infrastructure authorities. The PKI domains are cryptographically linked based on LTLs. In this paper the PKI architecture is only briefly described. Details have to be specified within the PKI policy documents of the different PKI domains. An open issue is the discussion of our multi-domain PKI proposal with stakeholders.

## VII. Acknowledgement

The authors would like to thank Sandro Berndt and Arno Spinner from the Federal Highway Research Institute (BASt), our colleague Hans-Peter Wagner and the anonymous referees for valuable remarks.

## References

[1] Car 2 Car Communication Consortium, "Mission, News, Documents," 2015, https://www.car-2-car.org.

[2] SimTD, "Secure intelligent mobility," 2008-2013, http://www.simtd.de/index.dhtml/deDE/index.html.

[3] European Commission, "SCOOP@F," 2013, http://inea.ec.europa.eu/en/ten-t.

[4] BMVI, "Cooperative its corridor rotterdam-franfurt-vienna joint deployment," 2014, http://www.bmvi.de.

[5] ETSI, "Intelligent Transport Systems (ITS);Security; Security header and certificate formats, ETSI TS 103 097 V1.1.1," 2013, http://www.etsi.org.

[6] N. Bissmeyer, H. Stubing, E. Schoch, S. Gotz, J. P. Stotz, and B. Lonc, "A generic public key infrastructure for securing car-to-x communication," in 18th ITS World Congress, 2011.

[7] G. Samara, W. A. Al-Salihy, and R. Sures, "Security analysis of vehicular ad hoc nerworks (vanet)," in Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on. IEEE, 2010, pp. 55–60.

[8] Hagen Stübing, Multilayered Security and Privacy Protection in Car-to-X Networks - Solutions from Application down to Physical Layer. Springer Vieweg, 2013.

[9] F. Schaub, Z. Ma, and F. Kargl, "Privacy requirements in vehicular communication systems," in Computational Science and Engineering, 2009. CSE'09. International Conference on, vol. 3. IEEE, 2009, pp. 139–145.

[10] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," Journal of Computer Security, vol. 15, no. 1, 2007, pp. 39–68.

[11] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, "Batch verification of short signatures," in Advances in Cryptology-EUROCRYPT 2007. Springer, 2007, pp. 246–263.

[12] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, 2008.

[13] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference. VDE, 2007, pp. 1–12.

[14] K. Plößl and H. Federrath, "A privacy aware and efficient security infrastructure for vehicular ad hoc networks," Computer Standards & Interfaces, vol. 30, no. 6, 2008, pp. 390–397.

[15] J. Linn, "Trust models and management in public-key infrastructures," RSA laboratories, vol. 12, 2000.

[16] R. Nielsen, "Memorandum for multi-domain public key infrastructure interoperability, rfc 5217," Tech. Rep., 2008.

[17] ETSI, "ETSI EN 302 665 V1.1.1: Intelligent Transport Systems (ITS) - Communications Architecture," 2010, http://www.etsi.org.

[18] Recommended Elliptic Curves For Federal Government Use, National Institute of Standards and Technology, July 1999. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[19] B. Feix, M. Roussellet, and A. Venelli, "Side-channel analysis on blinded regular scalar multiplications," in Progress in Cryptology–INDOCRYPT 2014. Springer, 2014, pp. 3–20.

[20] Brainpool, "ECC Brainpool Standard Curves and Curve Generation, Version 1.0, available online at http://www.ecc-brainpool.org/ecc-standard.htm," 2005.

[21] M. Abe and T. Okamoto, "A signature scheme with message recovery as secure as discrete logarithm," in Advances in Cryptology-ASIACRYPT99. Springer, 1999, pp. 378–389.

 ISBN: 978-1-61208-434-3 

# Schedule Rating Method based on a Fragmentation Criterion

## Schedule Optimization in Corporate Carsharing of Electric Vehicles

Falko Koetter, Julien Ostermann, Daniel Valerian Jecan
Fraunhofer IAO, Germany
email: firstname.lastname@iao.fraunhofer.de

***Abstract*— Sharing electric vehicles among companies and users can increase efficiency of corporate car fleets. Due to charging times and limited ranges, high quality scheduling is necessary to achieve a high degree of utilization and in turn economic efficiency. Schedule optimization can help improving a schedule regarding utilization and cost, but does not take into account possible future bookings. If a schedule is fragmented, no future bookings can fit in. In this work, we introduce a measurement for the fragmentation of schedules and use it to minimize fragmentation and in turn maximize future booking potential. We evaluate the approach using synthetic and real-life trials, showing that fragmentation reduction can lead to increased utilization in electric vehicle fleets.**

***Keywords-corporate carsharing; fragmentation; schedule optimization; schedule management; electric vehicle fleets.***

## I. INTRODUCTION

Even though electric vehicles (EVs) have been available for some time, recent developments like the rising costs of fossil fuels, technological progress in vehicle technology and availability of regenerative energy make the economic use of EVs in company car fleets more and more feasible. The market potential is noticeable, estimated at about 1 million EVs until 2020 just for Germany and in particular for company car fleets at 30% of newly bought cars [5]. The drawbacks of EVs, such as the high fixed cost for vehicles and charging infrastructures, have to be overcome by employing high utilization. The Shared E-Fleet project [15] researches the economic operation of shared car fleets, making it possible even for small and medium enterprises, which could not economically operate a fleet on their own.

Charging times and limited range are special challenges in the context of using EVs for corporate car sharing, which need to be taken into consideration when scheduling vehicles and business trips. When operating at capacity, the consequences of minor disruptions like delays or lost battery level can affect the future schedule, as trips might not be started on time or without sufficient charge to reach the destination.

To reach high utilization of car fleets while minimizing cost or ecological impact and compensating these disruptions, continuous optimization of the schedule is used.

While regular optimization techniques can be used to optimize a fleet schedule, e.g., regarding minimization of emissions, possible future states are not taken into account. These include the potential for future bookings, for which suitable timeslots in the schedule need to be available. If the schedule is fragmented, i.e., trips are distributed uniformly among vehicles, there may be no timeslot for a future booking, even though in aggregate, enough unused time on vehicles is available.

In this work, we describe a rating method for vehicle schedules based on a fragmentation criterion and use it to provide optimized schedules with minimum fragmentation, thus ensuring maximum opportunities for future bookings.

The contribution of this work is as follows: We introduce a fragmentation ratio, a measurement for the fragmentation of schedules and show how it is used as part of a closed loop optimization system.

The remainder of this work is structured as follows: Section II describes related work. Section III describes the schedule optimization problem. Section IV defines schedule fragmentation in relation to concepts from memory management. Section V presents the fragmentation rating and the algorithm for computing it. Section VI describes an evaluation scenario from the Shared E-Fleet project. Finally, Section VII gives the conclusion and outlines future work.

## II. RELATED WORK

Optimization of vehicle fleets encompasses multiple domains like routing, charging and scheduling. Different schedule optimization algorithms exist, but do not match the corporate car sharing scenario [13].

In our case, trips have a fixed start and end time and they need to be distributed among a set of vehicles with the purpose of maximizing utilization and minimizing costs. The solution for the optimization of scheduling and charging presented in [3] proves that the routing problem is NP-complete. While this approach provides a solution for charging and schedule optimization, the reaction to disruptions and schedule fragmentation are not covered.

Several methods for a feasible solution to NP-hard schedule optimization problems can be found in [4], but they do not refer specifically to the shared fleet scenario.

Schedule defragmentation is a problem in other fields of application as well. For example, [8] introduces an algorithm for scheduling lectures to classrooms by moving chunks from the least to the most occupied rooms. The implementation is described in [11], defragmenting a classroom schedule. However, compared to our work, defragmentation is the primary and not a secondary optimization goal. The schedule of healthcare professionals

 ISBN: 978-1-61208-434-3 

can be defragmented to provide maximum potential for additional appointments by preserving large chunks of free time, similar to how this work aims at maximizing potential for additional bookings [9]. In a similar manner, [10] provides a defragmentation algorithm to minimize patient waiting times. In comparison, our scenario has additional constraints such as vehicle range.

While the general concept can be applied to schedule defragmentation, different consistency conditions specific to EV use are not covered. Additionally, these static scheduling problems do not cover frequent re-optimization, as needs to be applied in a car fleet schedule.

## III. SCHEDULE OPTIMIZATION

In a shared car fleet, users book trips for a predefined time and destination, starting and ending at a car fleet station, where the vehicle can be charged. A schedule decides which trips are performed by which vehicles. As vehicles differ in range, cost per kilometer and emissions, there is potential for optimization. Another goal of optimization is enabling a high degree of utilization, which includes leaving a maximum potential for future trips. A user books a trip in the fleet, a specific vehicle is only assigned shortly before the trip starts, enabling trips to be moved by optimization beforehand.

Two different algorithms are used for schedule optimization. An alternative search algorithm searches a feasible fit for a trip in the schedule. This algorithm is used to check availability during booking and needs to provide immediate answers. The other algorithm, periodic optimization, performs a full optimization of the schedule, potentially redistributing any future trips [12].

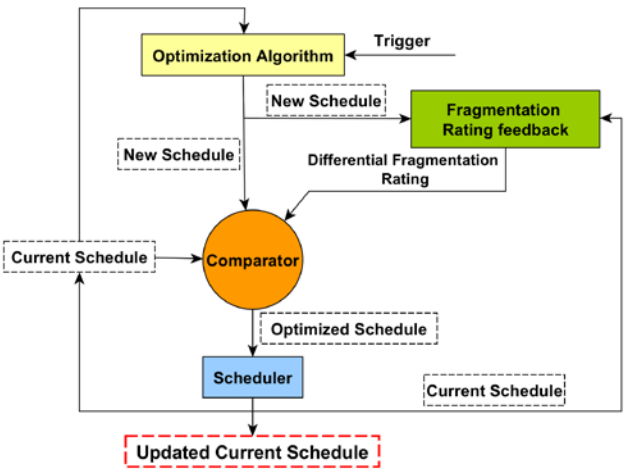


Figure 1 Closed loop schedule optimization scheme.

For the periodic optimization, we take into account two main components of schedule optimization, namely the algorithm used for optimizing the scheduling procedure and the feedback mechanism that has the purpose to indicate the degree of optimization or in other words the optimization rating of a schedule. As in the case of a typical closed loop control system [6], the optimization algorithm would be used as an actuator, changing the state of the schedule and the optimization rating as the feedback responsible for providing the necessary insight for improving the scheduling algorithm. This article focuses on one proposed feedback method, namely the fragmentation of a schedule. Fragmentation reduces the possible utilization of vehicles, as fragmented schedules offer shorter slots for future trips during alternative search. Therefore low fragmentation is a prerequisite for high utilization, so it is an important aspect to be evaluated and optimized.

Figure 1 shows the closed loop system used in the optimization process. In a closed loop configuration, the components involved are connected in a cyclic manner, influencing each other. Therefore, a holistic approach towards the analysis of the system is necessary [2].

The closed loop has four main components:
***Optimization algorithm*** – used for scheduling based on an algorithm that is meant to optimize specific goals
***Scheduler*** – deploys the current schedule
***Fragmentation Rating*** – the feedback mechanism used for comparing the old and the new schedule
***Comparator*** – based on the feedback received from the Fragmentation Rating component, delivers the optimum schedule

The optimization process starts with the current schedule, containing all current and future trips assigned temporally or permanently to vehicles. This schedule is given to the optimization algorithm, which after being triggered (periodically, by an event, e.g., a delay, or by an administrator), delivers a new schedule, using a greedy optimization algorithm, which is described in [12]. The new schedule and the current schedule are next given to the fragmentation rating component, where the computation of the rating takes place. The fragmentation rating is implemented as a maximization function (the higher the rating, the lower the fragmentation ratio) and the output is given in the form of a differential rating between the new and the current schedule. Using the provided rating, the comparator decides between the current and the new schedule and passes on the optimal schedule to the scheduler. At this point, the current schedule is replaced with the new optimized schedule (which might be the same one, if the new schedule does not receive a better rating, as the optimization algorithm is not fragmentation aware at the stage covered by the article). The optimization process is repeated every time the optimization algorithm is triggered.

## IV. FRAGMENTATION OF A SCHEDULE

While the initial schedule optimization only took into account a goal function (for either cost or emission minimization), during model trials we noticed that high utilization was hard to be achieved, as end-users tended to book trips which could not be scheduled to any vehicle, though globally enough free capacity was available. We noticed the problem occurred when trips were evenly

 ISBN: 978-1-61208-434-3 

distributed between vehicles, providing many opportunities for a future trip to overlap existing trips, therefore becoming impossible to book without re-optimization.

Thus, we added a soft goal to our approach in the form of the fragmentation of a schedule. Borrowing from operating systems memory fragmentation, we applied this concept to the area of scheduling trips among cars. In the context of computer memory, fragmentation usually refers to storage space that is not used efficiently, meaning we are dealing with reduced capacity and/or performance [7]. This can lead to different undesirable situations, one of which is not being able to allocate memory in certain areas of the storage space.

We will now introduce the basic principles related to memory management and fragmentation in the area of computer systems and we will transfer them to our proposed concept of the fragmentation of a schedule.

The dynamic memory is designed as a buffer between the physical big storage devices (e.g., hard-disks) and the small size but high speed memory of the processor, i.e., the cache. Basically any application needs to allocate memory from dynamic memory in order to run. Blocks of memory are allocated in chunks and whenever the application does not need such a chunk anymore, that particular space can be freed. However, because the size of these chunks is variable, after a while, depending on the actual memory usage of the application, the number and the size of long continuous regions of memory space could reduce significantly [1].

As an analogy, in order for a trip to take place, it needs to be booked in the schedule, which means we need to allocate that trip a time slot on a single vehicle inside the schedule (the schedule corresponds to the dynamic memory).

In the case of memory management, we work with allocating space chunks, while in the case of schedule management we are dealing with allocating time chunks. However, there is an important difference to be mentioned. The time to be allocated in the schedule is replicated among the vehicles available in the fleet, i.e., every vehicle has its own timeline, parallel to the others, so a specific time interval can be booked to any available vehicle. This is not true in the case of memory management, where every chunk of storage space is unique and it cannot be allocated to two or more processes at the same time. When a trip is cancelled for whatever reason, that specific time interval can be freed, as in the case of memory allocation. To continue the analogy, the time chunks allocated for trips are variable in size and when trips get cancelled, some variably sized slots are left unused, which can make booking longer trips harder. On the other hand, memory management does not depend directly on time, so the state of the memory could stay the same even if time passes. That is not the case with schedule management, because we are actually dealing with allocating time and the mere passing of it determines the state of the schedule to change along with it. For example, currently running trips cannot be reallocated and late trips may extend their allocated chunks, necessitating future changes.

The two main types of fragmentation related to memory are internal and external. Internal fragmentation usually occurs when the allocated memory (addressed in fixed size partitions) does not match the requested memory and the remaining unused part is wasted, as it cannot be allocated to other processes. External fragmentation however is generated when variably sized partitions are used, but as soon as some segments of memory are freed, some unusable small gaps can appear between the occupied blocks of memory. If we analyze the schedule management situation, we can only have external fragmentation, because when we allocate time for a trip, we can allocate the exact interval needed, if the vehicle is available for booking, so we are not bound to some fixed sized intervals, as in the case of memory fragmentation.

Therefore, the fragmentation for a schedule is computed using the chunks of time which are not used and their associated properties (duration and time interval of the day).

## V. FRAGMENTATION COMPUTATION

The initial configuration is that every vehicle is fully charged and ready to be booked.

There are a couple of terms to be defined:

***Window*** – The time interval (in the future) the fragmentation is calculated for (a default one day window starts at 7 AM and ends the next day at 6:59AM)

***Fragment*** – a chunk of time between two bookings which is not used (as seen in Figure 2)

***Maximum Fragmentation Ratio*** – the maximum rating you could get within a given window when the vehicles are fully charged without any booking.

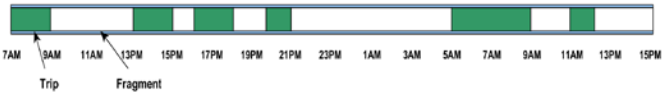


Figure 2 Example of a schedule.

The computation of fragmentation is meant to be used for optimizing the fragmentation goal, which means reducing the actual fragmentation of the schedule. By taking a snapshot of the schedule at different moments in time, we can analyze its structure and extract a list of fragments for each vehicle. The span of a fragment is between the end of the previous trip and the start of the next one, all within the given window for analysis. After getting the list of all fragments in the schedule, we compute the fragmentation ratio using two general, relevant properties of the fragments: duration and the time interval in which the trip is located in.

We incorporated these properties in the fragmentation ratio formula (1), represented by weights, as follows:

$$FR = \sum_{1}^{n}(\text{duration}(\text{frag}_i) * w1(\text{frag}_i) * w2(\text{frag}_i)) \quad (1)$$

where n is the number of fragments, w1 is the duration weight ( computed in (2) ) and w2 is the time interval weight ( computed using (3) ).

The fragments which have a longer duration have a higher weight w1, because longer and shorter trips can all fit in an extended time interval (see Figure 4). However, shorter

 ISBN: 978-1-61208-434-3 

fragments have a lower usability (trips of less than 20 min are less likely). Regarding w2, the weight for the time interval, trips during regular business hours (7AM till 17PM), have a higher weight than the trips outside of this interval (see Figure 5), taking into consideration that the main target of the Shared-E-Fleet project is the business sector [3].

The functions used for the weight computation are synthetic functions, but using the real data coming from statistics regarding usage of the electronic fleet, new and more relevant weighting functions can be computed. The initial formula (with x as the duration of the trip) used is:

$$\text{DurationWeight} = \frac{0.79}{\text{fw}*60}\text{x} + 0.01 - \frac{0.79}{\text{fw}*60} \qquad (2)$$

The *fw* parameter stands for fragmentation window, which is always a factor of 1440, i.e., how many minutes there are in one day. A window starts at 7AM in the morning (0) and it ends at 6.59AM the next day (1440). If there is one trip today at 1PM and 1 trip tomorrow at 10AM, the fragmentation window is 2880 minutes (48H) (see Figure 3).

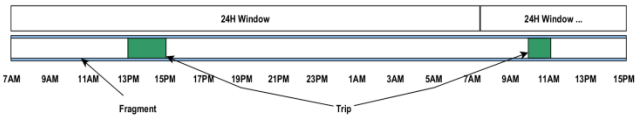


Figure 3 Fragmentation window.

An arbitrary maximum weight of 0.79 is given to a fragment containing the whole analyzed window. We intend to adjust the weight functions according to the real life usage profile obtained from the model trials. The subtraction in the formula is used for keeping the weight normalized.

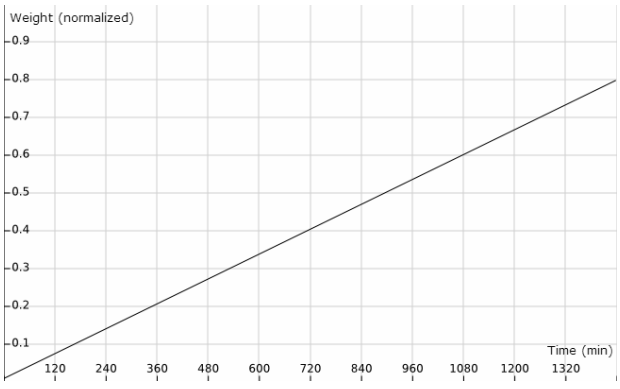


Figure 4 Weight function for the duration of the fragment.

$$\text{TimeIntervalWeight} = \left|\frac{\text{w}_{\text{end}} - \text{w}_{\text{start}}}{2}\right| \qquad (3)$$

$$\text{w}_{\text{start}} = \frac{-0.8}{\text{fw}*60}\text{startInterval} + 0.9 \qquad (4)$$

$$\text{w}_{\text{end}} = \frac{-0.8}{\text{fw}*60}\text{endInterval} + 0.9 \qquad (5)$$

The startInterval and endInterval variables are normalized for a one day duration, so values are between 0 and 1440 by using the fragmentation window defined previously. The parameters $w_{start}$ and $w_{end}$ are computed using formulae (4) and (5) and used for determining the time interval weight.

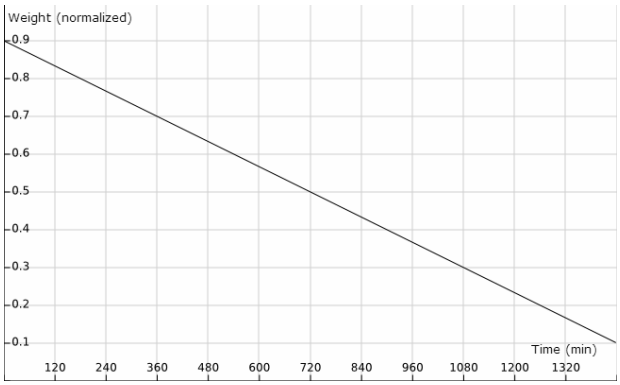


Figure 5 Weight function for the time interval of the fragment.

We plan to further adjust these synthetic functions by using statistical data from model trials.

Using those weights, the fragmentation ratio function is computed as a value between zero (meaning there is no fragment in the schedule, so no free slots) and maximum fragmentation ratio (meaning all vehicle schedules are empty and ready to be booked). Note that due to the duration weight, longer continuous fragments provide a higher fragmentation rating, and due to the time interval weight, fragments during business hours provide a higher fragmentation rating. Thus, schedules which provide large fragments of free time slots during business hours for future bookings are selected after optimization.

## VI. EVALUATION

Within the Shared E-Fleet project, this method is integrated into an optimization system, containing algorithms for booking, alternative search, partial and full optimization implemented as a Java prototype [12]. This optimization system is part of the larger Shared E-Fleet architecture, providing a EV fleet management solution[14].

We evaluated the fragmentation ratio approach, both using synthetic tests and by application during three long-term model trials in German industrial parks.

The partial and full optimizations are implemented in the optimizer component. The state of an EV is updated if real-time notifications such as delays, malfunctions or returns are received. The optimization algorithm implemented is a greedy algorithm using backtracking [12], minimizing total emissions and compensating disruptions like delays, which was run as is during the initial phase model trials. As a whole, the optimization scales linearly with the number of trips, allowing use of large schedules.

Before deployment in the model trials, we used synthetic test data with randomized bookings to determine the suitability of the approach.

In the following, we show a simplified example using synthetic test data.

 ISBN: 978-1-61208-434-3 

We used the fragmentation ratio computation in order to check the improvement of the fragmentation goal between two snapshots of a schedule, one with three trips, spanning over a one day window and the second one being the optimized version of the first one, using the optimization algorithm.

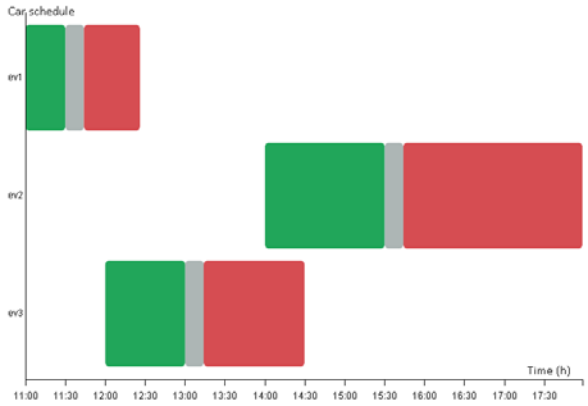


Figure 6 – Original version of the schedule.

A fleet of three vehicles is used in the example. As seen in Figure 6, in the original schedule we have a 30 min trip using ev1, starting at 11.00, a 1h 30min trip using ev2, starting at 14.00 and a 1h trip using ev3, starting at 12.00.

Figure 7 shows the optimized schedule (optimization was carried out at 10.38am). It can be observed that instead of sparsely using three vehicles, two of them were completely freed and all three trips are booked on one vehicle, therefore increasing utilization and decreasing fragmentation. Note that charging can be deferred as long as the remaining charge is sufficient for the next trip.

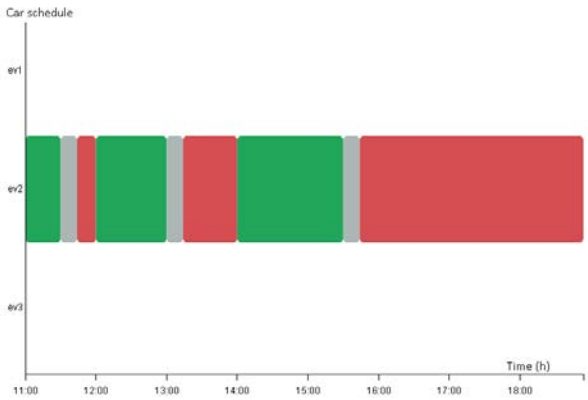


Figure 7 – Optimized version of the schedule.

The gray area after each trip represents a buffer which is intended to account for user behavior (minor delays, unloading the vehicle, connect to charge station, etc.). The red areas indicate charging times.

After running the analysis, we found 6 fragments for the old schedule (Figure 8) and 4 fragments for the new schedule (Figure 9) after optimization. The value of the fragmentation rating for the old schedule was approximately 794 and for the new one is around 1051, so the fragmentation was successfully reduced. The new schedule has two fragments of 1440 minutes, which means that two vehicles are completely free, as seen in Figure 7.

```
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 11:00:00 CEST 2015
fragment duration: 21
-----------------------------------------------
fragment start: Wed Jul 15 12:26:03 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1332
-----------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 14:00:00 CEST 2015
fragment duration: 201
-----------------------------------------------
fragment start: Wed Jul 15 17:58:36 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1000
-----------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 12:00:00 CEST 2015
fragment duration: 81
-----------------------------------------------
fragment start: Wed Jul 15 14:29:43 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1209
-----------------------------------------------
Fragmentation of the schedule: 794.6474138932936
```

Figure 8 – Fragmentation rating of the old schedule.

```
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1440
-----------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 11:00:00 CEST 2015
fragment duration: 21
-----------------------------------------------
fragment start: Wed Jul 15 18:54:00 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 944
-----------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1440
-----------------------------------------------
Fragmentation of the schedule: 1051.625424654092
```

Figure 9 – Fragmentation rating of the new schedule.

If the new schedule has a higher fragmentation rating, the new schedule actually has lower fragmentation, so it is better than the old one and the optimization was successful.

The model trials were implemented in three technology parks over a time period of a year, in the context of real small and medium sized enterprises. While the cars are providing a range of over 100 kilometers, the average booking contained a trip with less than 50 kilometers and 3 hours in length, facilitating optimization. Figure 10 shows the distribution of trips throughout the day, indicating predominant use during business hours, which we aimed to accommodate using the fragmentation ratio. Figure 11 shows the fleet utilization in one model trial, as well as the success percentage, indicating how many booking requests could be fulfilled. Due to novelty value, the demand reached

 ISBN: 978-1-61208-434-3 

its peaks in the first month. However, initial hardware problems in addition to suboptimal fragmentation created a perception of low availability. Bookings were rejected even though vehicles could be seen as available. To increase utilization, the fragmentation rating comparison was added to optimization at the end of August. The improvement in scheduling allowed on average 21.6% increase in utilization, with a higher success ratio. In December, utilization was lower due to weather conditions as well as holidays.

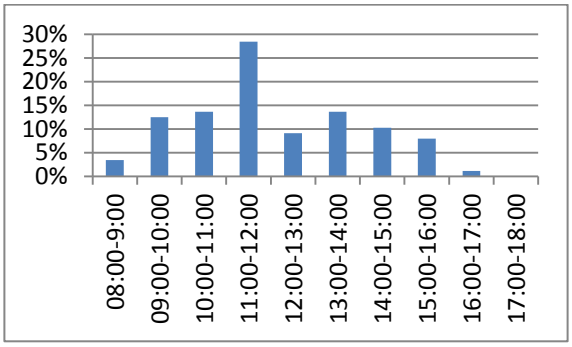


Figure 10 – Distribution of trips during the day.

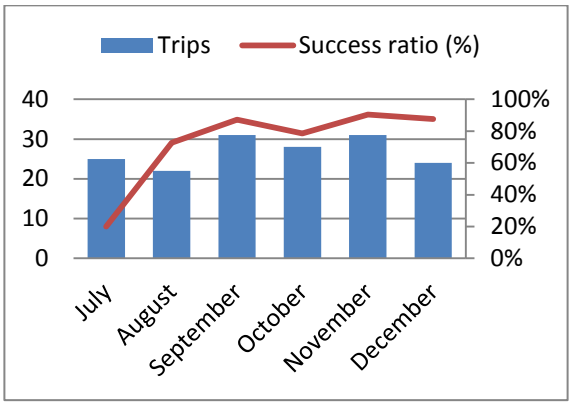


Figure 11 – Trips and success ratio.

Other model trials lack comparison values as they were started with fragmentation rating in place.

## VII. CONCLUSION

Optimizing vehicle schedules is a necessity for economic operation of shared fleets. In this work, we have introduced the fragmentation ratio, a rating for fragmentation of vehicle schedules, based on concepts from memory management in operating systems. Fragmentation rating complements schedule optimization to further increase utilization by prioritizing large time slots for future booking.

We evaluated the concepts by using synthetic data, as well as introducing them in a running model trial, showing notable improvements in utilization.

In future work, we will integrate fragmentation rating with other soft goals related to charging times and energy management (e.g., optimal utilization of photovoltaics for charging). Currently, the rating is applied after optimization, stopping fragmented schedules from replacing less fragmented schedules. In future work, we would like to adapt the optimization algorithm with fragmentation awareness in addition to other optimization goals. Additionally, we would like to evaluate weighing individual vehicle utilization using the fragmentation ratio, which could provide benefits in larger fleets.

## ACKNOWLEDGEMENT

This research has been supported by the IKT II program in the Shared E-Fleet project. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01ME12105. The responsibility for this publication lies with the authors.

## REFERENCES

[1] A. S. Tanenbaum, “Modern Operating Systems (3rd Edition),” Prentice Hal, ISBN-13: 978-0136006633, 2007.

[2] K. J. Åström and R. M. Murray, “Feedback Systems: An Introduction for Scientists and Engineers,” Princeton University Press, ISBN 0-691-13576-2, 2008.

[3] O. Sassi and A. Oulamara, “Joint scheduling and optimal charging of electric vehicles problem,” Computation Science and its Applications-ICCSA 2014, Springer, 2014, pp. 76-91.

[4] V. Pillac, M. Gendreau, C. Gueret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” European Journal of Operational Research 225(1), 2013, pp. 1-11.

[5] P. Plötz, T. Gnann, A. Kühn, and M. Wietschel, “Market introduction scenarios for e-vehicles,“ Karlsruhe: Fraunhofer-Institut für System- und Innovationsforschung ISI, 2013, (in German).

[6] R. C. Dorf and R. H. Bishop, "Modern control systems," Pearson (Addison-Wesley), 1998.

[7] M. S. Johnstone and P. R. Wilson, "The memory fragmentation problem: solved?," ACM SIGPLAN Notices, Vol. 34, No. 3, ACM, 1998, pp. 26-36.

[8] E. Humo and Z. Vejzović, "A Heuristic Approach To Classroom-Period Schedule Optimization," Advanced Engineering, 1st year (Vol.2), 2007, pp. 165-172.

[9] J. Lian, et al., "Clinical appointment process: improvement through schedule defragmentation," Engineering in Medicine and Biology Magazine, IEEE 29.2 (2010), pp. 127-134.

[10] C. E. Nelson, et al., "Visual scheduling to improve workflow and throughput in the University of Virginia Health System Pediatric Specialty Clinic," Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2011.

[11] Z. Vejzovic and E. Humo, "A software solution for a mathematical model of classroom-period schedule defragmentation," EUROCON, 2007, The International Conference on: Computer as a Tool, IEEE, 2007.

[12] F. Koetter, „Dynamic schedule optimization in shared EV fleets,“ http://www.shared-e-fleet.de/images/Dynamische_Einsatzoptimierung_von_gemeinsam_genutzten_ Elektrofahrzeugflotten.pdf, 2015, (In German).

[13] M. Bielli, A. Bielli, and R. Rossi, "Trends in models and algorithms for fleet management," Procedia-Social and Behavioral Sciences 20, 2011, pp. 4-18.

[14] J. Ostermann, T. Renner, F. Koetter, and S. Hudert, "Leveraging Electric Cross-Company Car Fleets through Cloud Service Chains: The Shared E-Fleet Architecture," Global Conference (SRII), 2014 Annual SRII, IEEE, 2014.

[15] Shared E-Fleet. [Online]. Available from: http://www.shared-e-fleet.de/ 2015.07.14

 ISBN: 978-1-61208-434-3 

# Checking and Verifying Security Requirements With the Security Engineering System Model Core

Hendrik Decke
Volkswagen AG Group Research
Email: hendrik.decke@volkswagen.de

Jean-Pierre Seifert
Technical University of Berlin
Email: jpseifert@sec.t-labs.tu-berlin.de

***Abstract*—As the need for security engineering methodologies for embedded and/or distributed systems rises several different approaches have been proposed. Especially the automotive sector is pursuing the development of new ways to better consider security in the design process. Nevertheless, most of these approaches are custom-tailored for specific use-cases or application domains and are not applicable for other domains. We propose a security requirements engineering process with a generic system model core, which can be customized with application domain specific extensions. This allows the instantiation of application domain adjusted security requirements engineering methodologies without much effort. Additionally, the generalisation of the system model allows the exchange of checking or verification methods with only a small need for adaptation to new application domains. We present our system model core and demonstrate its extensibility on the example of vehicular systems. We then show two methods for formal inspection of the system model. First, we show how the security engineer can be assisted by consistency checking of the system model, then we show how to verify the sum of generated security requirements to ascertain the correctness of the security concept.**

***Keywords–Security engineering; requirements engineering; requirements verification; system model core.***

## I. INTRODUCTION

How to design secure distributed and/or embedded systems is a repeatedly discussed question and it is recognized in most industries that requirements engineering is critical to the success of any development project. How to extend requirements engineering into the realm of security has been proposed with a multitude of approaches. Being able to learn from existing methods in security requirements engineering for software systems, most of these approaches focus on particular aspects of the security requirements engineering problem while additionally focusing on a very specific use case or application domain. Given these results, the question why these approaches did not lead to a broader industry adoption was examined by [1] and [2].

For example, the work in [2] presents three interesting factors for missing adoption of methodologies.

1) The business case for employing security best practices is missing.
2) Developers lack security expertise, which is currently required to employ security best practices.
3) The risk of committing to a particular security approach is too high.

Furthermore, [1] gives a broad range of properties and criteria for methodologies to lead to broader industry adoption. Without repeating all of them we saw the feasibility to improve in three major fields.

**Notation** By describing the security engineering system model core (*SESMC*) as a generic core, we hope to allow the easy exchange and reuse of varying approaches to different parts of security requirements engineering. This reduces the risk of committing to one particular security requirements engineering approach since additional and conceivably needed features can be implemented and added easily. By adding application domain specific extensions we can demonstrate how a security requirements engineering methodology can be instantiated for the vehicular domain.

**Tool Support** By designing *SESMC* and the corresponding security requirements engineering process to be implementable as a software tool for the security engineer, we improve the ability to master the complexity of a large embedded and/or distributed system, which directly increases the business case for employing a security requirements engineering methodology. Furthermore, the software tool may assist engineers with a lower security expertise.

**Formal Methods** The use of formal methods and verification leads to increased confidence in the result and can ideally point out outstanding problems in the system model.

### A. Contributions

We propose a generic system model core to be used for a security requirements engineering process to create methodologies for different application domains. For this system model core, we present methods to perform consistency checks and to allow the verification of a broad range of properties of the system model, including the new concept of non-traceability, with the use of the ProVerif tool [3]. When appropriate, we will relate our elucidations to possible uses in vehicular systems.

### B. Structure

Section 2 presents the background and related work. In Section 3, we present the generic system model core and the security requirements engineering process. Section 4 describes the consistency checking of our system model core. Section 5 presents the steps that lead to the verification of a set of security requirements contained in our system model core. Section 6 presents the conclusions.

## II. RELATED WORK

As already mentioned, several examples of security requirements engineering methodologies have been evaluated to create

 ISBN: 978-1-61208-434-3 

our generic system model core. Likewise, methodologies in the field of access policy engineering and model-based software design have been considered as to discover basic approaches to recurring problems in engineering to lay the groundwork for our approach.

The most influential paper for our work is [4]. They present their **Wor**kbench for Model-based **S**ecurity **E**ngineering (WorSE), which supports the modelling and verification of access control policies. Their main contribution is a domain-independent abstraction for security policy design and verification, which they call the security (model) core. This core may be extended for domain-specific analysis, just as subclasses can be derived from super classes in object-oriented software design. As the formal semantics of this security core are clearly defined any extension can still rely on all properties of the parent core, which yields an interesting way to allow domain-specific analysis in a flexible workbench.

Again in [5], our concept of a generic but extensible system model core is motivated. They argue that in the field of (pattern-driven) security methodologies for distributed systems over a dozen approaches have been presented, but with respect to system applicability these approaches are either highly specific or generic. Likewise, they motivate to position a security methodology in the early development phases (analysis and design), because *"This is where all security countermeasures are planned [6], as well as where, according to [7], approximately half of all major security flaws can be prevented. [8]"*

In [9], [10], [11], a methodology for the security engineering and partitioning of hardware/software systems is presented, which is the most similar methodology to our own. It is named AVATAR, extending on the Object Management Group's Systems Modeling Language (SysML) [12], and enriched with artefacts for security engineering and implemented in the workbench TTool. The authors position the process in parallel to the hardware software partitioning of the y-chart approach [13] so that the asset, threat and security requirement identification can be done in the early design steps of the systems development life cycle (SDLC).

AVATAR is requirement-driven. Given some security requirements, which are designed in separate diagrams and kept in parallel to the application model of the target of evaluation (TOE), the threats and corresponding attack trees can be subsequently defined. Possible risks are then manually annotated to the security requirements, so that the decision for chosen security mechanisms can be documented. After the system model has been finished, the non-security relevant properties can be model-checked using UPPAAL [14] and the security-requirements (confidentiality and authenticity) can be proven by the ProVerif toolkit [3]. ProVerif [3] is a security protocol analyser. It is able to proof security properties over given cryptographic protocols assuming the Dolev-Yao attacker model [15] with very small resources.

Our own approach, which will be presented in Section III, however is goal-oriented and focuses on keeping all information inside one cohesive system model. With the addition of protection groups and threat hierarchies we hope to enable the design of a whole system inside one model without an unproportional increase in size and complexity. It should be noted that, although we do not present methods to check non-security relevant properties like liveness or computational intensity, we deem our approach to be applicable to these properties and that model-checking, i.e., by using UPPAAL, should be feasible. We increase the amount of usable protection goals in comparison to AVATAR in our version of a ProVerif export. The different possible queries will be presented later in Section V.

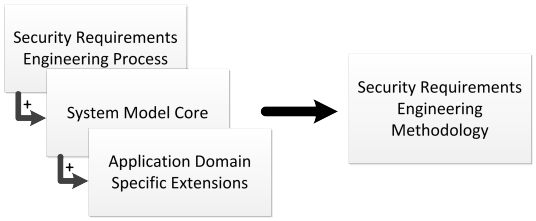


Figure 1. Creating a methodology from the process and system model core.

## III. A GENERIC ONTOLOGY FOR SECURITY REQUIREMENTS ENGINEERING

For our approaches of consistency checking and verifying security concepts to be adaptable to different security requirements engineering methodologies, we define a generic **se**curity **s**ystem **m**odel **c**ore (*SESMC*) with a corresponding ontology. To implement our methods for consistency checking and verification for an existing security engineering methodology the required artefacts have to be identified and matched to *SESMC*. With providing *SESMC* we hope to present a useful mechanism to exchange property checking, property verifying and other algorithms between different security engineering methodologies of all kinds.

Additionally, *SESMC* can be used to instantiate new security requirements engineering methodologies from our proposed process by providing application domain specific extensions. We see our core as adaptable to different application domains, since we only define the recurring artefacts needed in all security engineering domains. By adding the application domain specific characteristics, i.e., communication medium restrictions, cryptography costs or specific attacker models, it is possible to construct an applicable methodology from our core. This is shown in Figure 1. An example of an instantiation for the vehicular domain will be described at the end of this section.

First, we will define the artefacts of our system model core, then we will show how these artefacts are used in our requirements engineering process.

### A. System model core

Our system model core (see Figure 3) consists of model artefacts, risk analysis artefacts and security requirements artefacts.

At the centre of the security requirements engineering process are the data instances. They are the assets that should be protected and which are used in the dataflows and processes of the TOE. We allow to define several types of data, while we expect to at least differentiate between data instances representing cryptographic keys (secret and public) and common payload like sensor readings or calculation results. We propose to add a special type for person-related data, like address data

and names, so that privacy goals and issues can be easier identified.

*1) Model artefacts:* The model artefacts consist mainly of executing units, which represent the different nodes, roles or tasks of the system, and communication mediums, which connect the executing units to allow dataflows. Both can be provided with different application domain specific properties (e.g., tamper protection or cryptography acceleration) to allow for more precise analysis. The connections between executing units and communication mediums are consequently named connections and they are used by the dataflows to transfer entities of data. Lifecycle phases represent different stages in the lifetime of a system, which build on one another. They allow to model and analyse a system before or after particular points in its lifetime, e.g., production, operation mode 1, operation mode 2 and decommissioning. Each executing unit has one knowledge set for each lifecycle phase defining the initial knowledge before every process and dataflow of this lifecycle phase.

The dataflows and processes themselves are different for each lifecycle phase to allow modelling varied behaviour. The dataflows are executed concurrently by the executing units but they consist of ordered dataflow steps to structure the communication in time for each dataflow. Each dataflow step describes a message consisting of data instances, which is communicated over connections between two executing units.

To represent the tasks implemented by the executing units we define one or many processes on the executing units, which are likewise executed concurrently. The processes consist of actions, which we will assume to adhere to certain assumptions. The actions describe the processing of data instances inside the executing units. Typical examples for actions are *send, receive, create, save and load*. We allow different notations for the processes and actions (i.e., pi-calculus [16], kripke structures [9] or UML activity diagrams [17]) as long as they satisfy certain properties; e.g., the notation can be translated into a directed acyclic graph and some recurring patterns can be identified. We will discuss the assumptions and properties in more detail in Section V. There is at least one process on a executing unit for each dataflow it is part of to implement the send and receive actions and to synchronize dataflows and process. More processes per executing unit are possible to allow asynchronous processing. To allow for a more precise reflection of reality executing units and communication mediums can be grouped to define hierarchies and boundaries (organisational and/or physical) in the system.

For executing units and model groups we allow to define a multiplicity value to model the replication of nodes, which can be useful to model sets of identical nodes (executing units) or reoccurring patterns of nodes (model groups). The possible values for multiplicity are up to the designer of the methodology to regard the needs of the application domain.

*2) Risk analysis artefacts:* The model artefacts are annotated with the risk analysis artefacts. The protection goals define which protection goal categories (properties) should be ascertained for which data instance. To be exact we recommend to define distinct protection goals for one data instance in different parts of the system model. We call these parts *Protection Groups* and they cluster data flow steps and actions and allow to propagate the protection goals to all influenced model artefacts (e.g., the sender or executor of a dataflow step or an action). This remedies the need to define confidentiality for each single dataflow step and all executing units of a large dataflow, as all the individual parts of the dataflow can be encapsulated inside one protection group for which one protection goal is set. For each type of protection goal category we define the possibility to define options to further elaborate the meaning of the goal. We see our categorization and options as an example of how protection goals could be formulated, so that we can show the translation to verification queries (see Section V). It is up to the designer of the methodology to define the fitting application domain specific goals.

**integrity** is the property of protection against modification. It defines which node may create or send a specific data instance. As we outline in Section III-C, we recommend to add an access control matrix to the system model to assign write permissions to nodes per data instance. This can be used for access control checking and to further refine authentication permission. For example, a node may only be allowed to obtain asymmetric keys to create digital signatures, if it is allowed to write the specific data instance.

**confidentiality** is the property of secrecy. It defines which node may gain knowledge of a data instance. Again, we recommend to add an additional knowledge permission matrix to the system model to allow for easy modelling of knowledge permissions.

**availability** of a data instance may be important for systems to fulfil their function. Therefore, it should be modelled as a protection goal. This property is difficult to ensure and verify formally but should nevertheless be regarded because denial-of-service attacks are a serious threat.

**authenticity** is a property with many facets. It may be needed for authorization purposes or to ascertain the origin or integrity of a data instance. We recommend to define separately for the data instance, the sender, the receiver and/or executor if authentication is needed. Additionally, we allow to model the need for non-repudiation and the property of freshness as options, which is needed in many situations.

**privacy** is the property of confidentiality and self-determination of person-related data. This property implies confidentiality but may also add the requirement to be non-traceable. It is stronger because not only the knowledge of a data instance shall be prevented, but also the plain existence of one particular dataflow or process with one specific data instance shall be indistinguishable from a dataflow or process with another data instance. It is meant to prevent the tracking of users or entities by identifying them by the ciphertexts of their personal data. We differentiate between non-interference like it is described by [18], which decides if the attacker is able to notice if a data instance changes in between sessions and non-traceable, which defines that the attacker is not able to recognize if the same data instance is used in different sessions.

The protection goals each possess one damage potential assessment and they are endangered by different threats. The damage potential assessment quantifies the estimated expected


Figure 2. Structure of dataflows and processes

damage when the protection goal is violated by a threat. For each of the threats a risk assessment has to be carried out. We do not dictate any form in which the threats shall be modelled or on how to perform the risk assessment, since the field of threat modelling already offers very elaborate solutions. Merely it must be possible to differentiate between the threats that do not constitute a risk for the protection goals and the threats that do, which will then realise the damage potential.

*3) Security requirements artefacts:* The security requirements artefacts are then mitigating the threats with relevant risks. We differentiate between security mechanisms and security requirements, which are grouped by a security concept. The security engineer can define multiple security concepts for one system model to allow for easy comparison between competing security solutions. Although most literature subsumes security measures and security requirements under the term *security requirements*, we need to distinguish because of verification precision.

Security mechanisms are direct measures refining dataflows and processes to represent implementations of cryptographic, physical or organisational means to increase security, whereas security requirements are textual requirements that dictate the mitigation of a risk or threat. Only cryptographic security mechanisms can be verified with ProVerif, because organisational or physical mechanisms and textual requirements do not provide the necessary details for the analysis. These types of requirements are checked informally in the consistency check. Textual requirements can be seen as a fallback or placeholder in situations when the designer of the system does not want to provide all the details for a security mechanism, so he can temporarily mitigate a threat to analyse the remainder of the system. Textual requirements can then be replaced by more detailed mechanisms later.

Security mechanisms must describe which changes they apply to the messages of dataflows steps and the actions of processes. For example, a deterministic symmetric encryption must describe how it is applied to a message (the message $x$ becomes $symmEnc(x, key)$) and this encryption action has to be inserted before the send action of the corresponding dataflow step (the action $send(x)$ is extended to $x2 = symmEnc(x, key); send(x2)$).

In addition, security mechanisms can instantiate specific attribute or node requirements for the system model. For example, a cryptographic measure may require a secure key storage in an executing unit, or the existence of a PKI may be needed when using asymmetric cryptography, which has to be modelled by an application domain specific extension. These attribute or node requirements have to be regarded as they lead to (more) complete systems. These requirements can be checked and not fulfilled requirements can be presented to the security engineer in the consistency check.

How these system model artefacts are generated is up to the methodology. In Section IV and V, we assume that the methodology has been carried out and the complete system model can be used for checking and verifying.

### *B. Security requirements engineering process*

Our proposal for a security requirements engineering process consists of the following steps.

1. **Initial architecture** Here, the TOE is designed. First, the executing units, communication mediums and groups are placed and connected to form the topology design of the system. Then, dataflows and processes are added, with their corresponding dataflow steps and actions, to define the communication and process inside the system. Lifecycle phases may be used to define multiple succeeding behaviours.
2. **Protection goal definition** Now the protection groups can be defined for which the security engineer can then define the desired protection goals.
3. **Threat definition** Given the protection goals it is possible to define the threats against these goals. For each goal a damage assessment defines the amount of estimated potential damage if the protection goal is violated.
4. **Risk assessment** For each threat tree (since they may be hierarchically ordered) the security engineer may choose not to assess the risk of all the leafs of the tree(with the most detailed threats). He may choose to assess on a higher abstraction level to increase efficiency. The assessment is then executed with an application domain specific risk system.
5. **Security concept design** Given the risks of the system the security engineer decides which security mechanisms to add. After adding all chosen mechanisms the requirements for the operation of the system can be defined. Node or attribute requirements may be needed to implement security mechanisms. As outlined earlier textual requirements may be used to mitigate threats, if the definition of actual mechanisms is out of scope for the current investigation. However textual requirements decrease the confidence in verification results, as they lack important implementation details.
6. **Verification** When the system model is finished the defined protection goals can be verified. It may be necessary to add further implementation details to decrease inaccuracies, so that the verification model can be built. This includes defining the exact format of messages (payload, order of cryptographic primitives, etc.), the knowledge of the nodes before the communication and process of the current lifecycle phase and the replication details of the processes (how often a node executes a process), if these details have been ignored or not modelled in the preceding steps. It is important to add as many details as possible, as we use the security protocol analyser ProVerif. ProVerif uses the closed-world assumption, which dictates that only facts that have explicitly been modelled as true are true. Everything else is false and non-existent and can not be used to attack the system.

 ISBN: 978-1-61208-434-3 

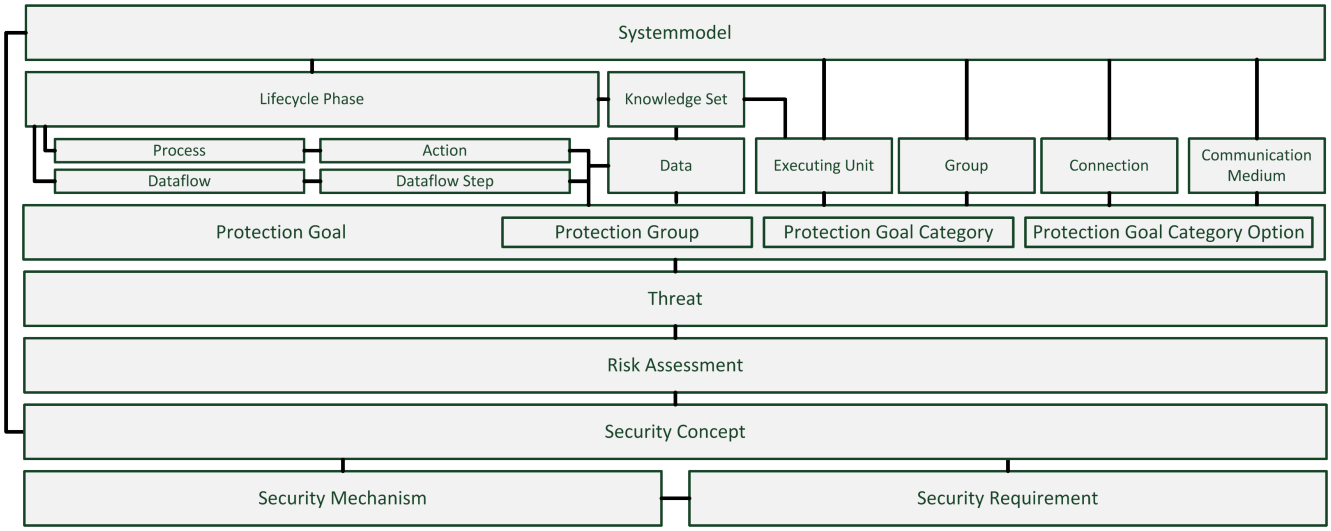


Figure 3. Main artefacts of the system model core.

**Additional steps** Between the different steps of the process consistency checks may be performed to find contradictions and misconceptions. We describe our approach to consistency checking in Section IV. After the creation of the security concept further analysis and optimization steps may be performed.

**Refinement** Since several assumptions about the initial architecture of the TOE may be changed throughout the security engineering process the methodology must consider to change parts of the TOE without nullifying all results. Only the artefacts associated with the changes should be marked as invalid and should be analysed again.

### C. Matching existing methodologies to the system model core

We formalize our system model core and its extensions with sets, relations and functions, which are representing the artefacts mentioned above. To match an methodology to our system model core these elements have to be identified so that a system model conforming to our definitions can be extracted to use our proposed consistency-checking and verification methods. The following examples illustrates how the system model core is structured and provides an understanding for the following sections.

$\mathbb{E}$ is the set of Executing Units.
$\mathbb{M}$ is the set of Communication Mediums.
$\mathcal{T} \subseteq \mathbb{M} \times (\mathbb{M} \vee \mathbb{E})$ is the relation connecting the executing units and communication mediums, creating the topology graph and representing the connections.
$\mathbb{P}$ is the set of processes.
$\mathbb{A}$ is the set of actions.
$\mathcal{P} \subseteq \mathbb{P} \times \mathbb{A}$ is the relation connecting processes and actions.
$G_{\mathcal{A}} = (\mathbb{A}, \mathcal{A})$ is the directed acyclic graph defining the order of actions (successor relation) with the set of actions $\mathbb{A}$ as nodes and the relation $\mathcal{A}$ as edges.

It can be seen that all artefacts (see Figure 3) are contained in sets. Relationships between these sets are modelled with relations and functions depending on the shape of the relationship. The only exception from the rule is the directed acyclic graph $G_{\mathcal{A}}$, which defines the order of the actions. We omit the remaining sets, relations and functions as they can directly be inferred from the description of the ontology above.

As we outlined in the description of the protection goal categories we additionally recommend to add two access control matrices to the system model to allow for more fine-grained control over write and knowledge permissions. Without the access control matrices the write and knowledge permissions can only be inferred from the dataflows and processes, so that every node that receives or sends a data instance may write or know this data instance. This could be imprecise as some nodes may only need to forward or relay an encrypted message, which could lead to contradictions. Further in the verification steps we can use this matrix to generate more precise secrecy queries.

### D. Using the system model core for the vehicular domain

The matching of real world entities in the vehicular domain to the system model artefacts is not as straightforward as one might expect. Especially the application domain specific extensions allow to model an arbitrary amount of detail if desired by the security engineer designing the security engineering methodology. Even the level of detail the security engineer tries to represent in one system model is very dependent on the TOE. Therefore, the extensions have to be chosen very carefully to allow modelling of all possible TOEs and to match the desired level of accuracy in the early design phases. We identified two main kinds of TOEs in the vehicular domain. Either a new control-unit is developed and it is modelled with all its external interfaces or a function is being developed and all involved control-units with the connected external IT systems have to be modelled.

In the first case the different components of the control-unit will be represented by the executing units and all external and internal communication uses appropriate communication mediums. The control-unit itself is represented by a group, which defines the physical border - the plastic shell. In the majority of cases the counterparts at the other ends of the external interfaces (communication mediums) can be represented by a single executing unit. Possible extensions are confined to properties relevant for dimensioning of the

 ISBN: 978-1-61208-434-3 

hardware capabilities. Examples are the bandwidth or computational capabilities of the communication mediums and executing units and the arising confronting costs by security mechanisms. If standardised communication mediums are used (e.g., IEEE 802.11) several default security mechanisms (read WPA2) should automatically be considered in the security concept as these mechanisms should come at low additional cost. Any special security features of executing units (i.e., tamper protection or hardware acceleration of cryptographic primitives) should also be modelled to enhance the accuracy of the risk assessment.

In the case of a distributed system to implement a new function the abstraction level has to be chosen higher to reduce the system model complexity. It is not feasible to model each control-unit with its internal components. Therefore, in the majority of cases each control-unit is represented by one executing unit. To allow modelling sensors or human-machine interfaces (HMIs) we propose an extension that adds sensors and HMIs as a decorator to executing units. They only need a name and a visual representation in the system model to be regarded in the analysis steps. For example, the modification of a sensor can then be modelled as a threat. These sensors and HMIs can then be referenced in actions and be used to create new instances of data. Hence it is possible to model the information flow of data from the source to the sink. Furthermore, we define a special car group that can only be instantiated once in each system model. It clearly defines the executing units belonging to the car and therefore allows fine-tuning the risk assessment in the later steps of the process as well as defining useful default values for properties like bandwidth capabilities of not yet detailed communication mediums. The executing units outside of this car group may represent IT systems, backends, diagnostic tools, users, other cars (abstracted as a single executing unit) or one of many other possibilities.

In each case we propose a special action to model the impact on external assets. If this action is executed it has an impact on one or many external assets like the quality of the driver assistance systems, the locks or the driving status of the car. It is meant to represent the physical interaction of the system model with its environment. As the system model core focusses only on digital IT systems this extension allows to better represent actuators or mechanics. Consequently the damage and risk assessment can be influenced by the existence of such external asset actions.

## IV. Consistency checking a system model

As outlined earlier not only the completed system has to be verified but also individual steps, along the way to completion, should be checked so that contradictions, discrepancies and inconsistencies can be corrected immediately and not later on in the design phase. We will call them consistency issues. These consistency issues are mostly not harmful to the security of the system directly, but they do not allow a precise analysis and verification because the resulting model of the system will not be implementable or at least the needed changes will be more expensive later.

There are several categories of possible consistency issues for which we define rules to find them in the system model. There are topology issues, dataflow and process issues, goal and risk issues, security concept issues and general warnings. Conditions for the instantiation of a consistency issue should be defined, which can be presented to the security engineer. These conditions could be checked regularly after each change to the system model or only when the security engineer wishes to check the design. This is up to the methodology, especially because the possible size of the system model can be very application domain specific and the duration to check the rules can not be estimated beforehand. We recommend to allow the security engineer to ignore or delete consistency issue, as they may be false positives. For example, an unused connection may be unused because the missing dataflow will be modelled later.

*1) Topology and process issues:* Consistency issues in the topology or the placement of processes are usually remnants of old versions of the model, which were not aligned with changes. They lead to an unconnected system model where single connections and processes are existing without being used or being relevant. Furthermore, the consistency between the dataflows and processes should be checked as for each dataflow step there should be corresponding send and receive actions in the correct order. These dependencies should be automatically resolved when the methodology is implemented as a software tool.

*2) Dataflow step and action issues:* Here, we aim for inaccuracies and blunders, which may origin from different causes. Perhaps a group of security engineers worked on the same system model and a misconception happened or changes in the system model due to external feedback led to flawed alterations. Examples include the violation of the knowledge and usage access control matrices, if these were implemented, or a data instance originates from multiple sources.

*3) Goal and risk issues:* These issues try to hint at misconceptions regarding the protection goals and risk assessments of the system model. They draw the attention to different parts of the system model where the comprehension of the goals or risks may potentially be contradictory or even conflicting. This allows to find misconceptions but in addition it allows to check the consistency of different analysis results, that may be produced by different people. In case of a very large system model it may be needed to divide the workload of modelling the security analysis artefacts between several security engineers. After each one has contributed his goals and derived risks these may be checked for consistency to create a consolidated solution. Depending on the application domain different conflicts for protection goals or damage potentials are possible. For example, if confidentiality is required for one dataflow step over a communication medium representing the internet, it should be required for all transmissions over this communication medium. Likewise the damage potential for all these transmissions should be similar.

*4) Security concept issues:* After the security concept has been built it is helpful to check for the most common blunders. Aside from unfulfilled attribute or node requirements the list of possible issues has to be compiled for each application domain and may include one or many of the following: *sign-then-encrypt vs. encrypt-then-sign, using sensor data as a cryptographic key, missing seed for hash, et cetera.*

 ISBN: 978-1-61208-434-3

*5) General warnings:* General warnings describe situations, which clearly show potential to be harmful. This includes warnings when a step of the security requirements engineering process has not been conducted yet or unfulfilled node or attribute requirements.

### A. Implementing the consistency check

The consistency check can be implemented in many ways depending on the type of the implementation of the methodology. If the process is handled by handwork and manual annotations on paper then a questionnaire could be helpful. If the methodology has been implemented as a software environment/tool the variety of possible implementations becomes apparent. As the structures of the rules are very disparate they do not tend to reveal an obvious object oriented pattern to allow for easy decomposition. We therefore recommend (and chose for our own implementation) to use anonymous functions like lambda functions in C++11 or Java8, which return a string with the result of the rule check or even a more complex object describing possible fixes for the issue, which could be executed automatically.

## V. VERIFYING A SECURITY CONCEPT

After the security engineer has finished the security concept for the system at hand he may want to verify if his concept is able to formally ensure the chosen protection goals. We propose a method to transform our system model core to one or many ProVerif [3] models so that the given protection goals can be queried. In Section III, we presented our generic system model core. We proposed that the actions which constitute the processes have to adhere to certain assumptions to allow the extraction of ProVerif models.

1) All actions are nodes of a directed acyclic graph ($G_{\mathcal{A}}$), which as a whole represents all processes. This graph must necessarily be disconnected if there is more than one process. The connection between a sending action and the corresponding receiving action is given by one specific dataflow step, which is recorded outside of $G_{\mathcal{A}}$. The usage of a directed acyclic graph excludes the concept of looping (i.e., while-loops) from our syntax. However this does not exclude to define a single action to represent a while-loop.
2) The creation of a new value for a data instance can be associated with a distinct action.
3) The assignment of a new value to a data instance can be associated with a distinct action.
4) There are distinct send and receive actions or the sending and receiving of data can be associated with other distinct actions.
5) All possible successors of conditional actions (if-else-then, etc.) must be determinable.
6) All actions associated with sending and receiving reference the used communication mediums and data instances or these can be inferred from other sources.
7) For each dataflow step the exact structure of the send message can be inferred. This includes the usage of cryptographic primitives on parts of the message. The easiest solution would be to always use the security mechanisms on whole messages, but more precision is possible.
8) The data instances that are assumed to be known to an executing unit, before a process is executed, have to be modelled. These includes the usage of additional lifecycle phases to distribute the data beforehand or to define the executing unit as the origin of the data instance.

Given these assumptions we can verify the protection goal properties by using the security protocol analyser ProVerif. We divide the verification into four steps show in Figure 4.

1) System model partitioning
2) Process extraction
3) Attacker initialisation and execution
4) Attack trace parsing

For the analysis we allow to model executing units as malign. This means that the attacker may have already gained control over this node and is able to control its behaviour. This can be an important function for many application domains and has consequences for the verification steps. Communication mediums are assumed to be under the control of the attacker according to [15].

### A. System model partitioning

In the first step we partition the system model into independent parts. These parts define isolated dataflows and processes, which do not affect any dataflow or process in another part, which can be interpreted as the information flow graph of one specific data instance. This isolation step is important to reduce the size of the verification models. The result of the partitioning step are directed acyclic graphs of actions. The root nodes of one directed acyclic graph are the creation actions for the data instance, or the root actions of processes if the executing unit already knows the data instance and the data instance is used in this particular process.

Our approach takes the directed acyclic graph of one process, which is taken from the relation $\mathcal{A}$ (successor relation), starts with the creation or root action(s) for one arbitrary data instance and then adds additional action nodes, including all the paths that lead to the additional node, when one of the following conditions is met by the path. The next node to check is chosen by arbitrarily selecting one not already checked edge out of the directed acyclic graph of the process under evaluation. The following binding conditions have been defined. Additional processes are checked when the last binding condition is met.

- The action node is a conditional and the condition contains the data instance.
- The action node alters the data instance in any way.
- The action node uses the data instance to create a new data instance.
- The action node uses the data instance for sending or receiving.

In the case of a sending or receiving action an additional edge is added between the two nodes of the directed acyclic

 ISBN: 978-1-61208-434-3 

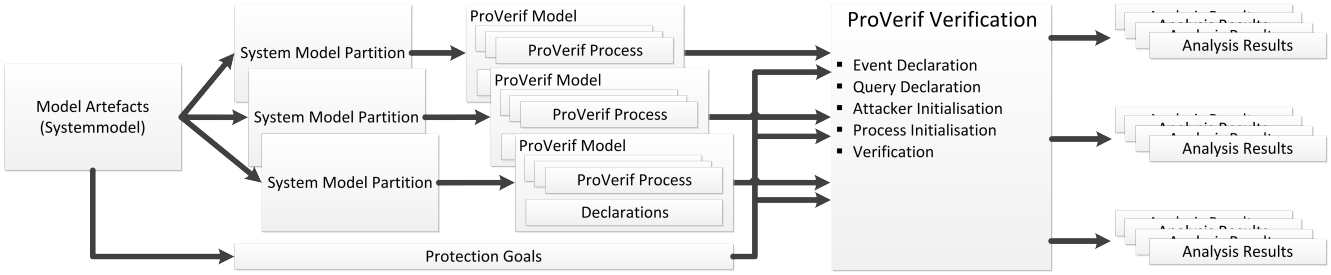


Figure 4. ProVerif Verification

graphs ($G_{\mathcal{A}}$) of the two affected processes. This represents the dataflow step crossing the two processes. The search is then eventually continued in the newly added part of the graph to connect the two processes further. It is important not to discard the information which action belongs to which process (i.e., with using the relation $\mathcal{C}$) as this is needed for the process extraction.

### B. Process extraction

After the directed acyclic graph (system model partition) has been built, it must be transformed to processes for the ProVerif tool. ProVerif supports several input formats from which we chose the typed pi-calculus as it fits our partitions from the last step. A ProVerif model in the typed pi-calculus consists of a declaration part for channels, functions, reductions, equations, events and queries and the process definitions and instantiations.

The channels can be instantiated directly from the communication mediums with *free channelName: channel.*. ProVerif allows to model a channel as private (by adding *[private]* to the declaration), so that a possible outside attacker can not read any messages of the channel. This property could be controlled by an application domain specific extension that allows communication mediums to be marked as *hidden* from outsiders. It allows to limit the capabilities of the attacker, as he may not be able to control and manipulate all communication mediums simultaneously. If an executing unit marked as malign is connected to a *hidden* communication medium this option has to be removed. The malign node effectively bridges non-hidden and hidden communication mediums, as the attacker may not read from or write to the private medium, but can instruct the malign node to do. Without limiting the generality of our approach we will assume that all channels have been defined as non private, but we will indicate where this option could be relevant.

Next the ProVerif processes for the different processes of the system model partition can be built. Because all processes execute concurrently we will create one ProVerif process for each process. Actions from our system model have to be translated to ProVerif constructs. As we do not dictate a specific notation, we have given assumptions at the beginning of this section, which we take for given for the translation. These leads to a straightforward mapping of ProVerif constructs to the actions.

When these constructs (and possibly others) have been assigned to actions in the chosen notation, the translation to ProVerif subprocesses can begin. First, the actions in the directed acyclic graph can be annotated with the proper ProVerif constructs. Then, the ProVerif constructs are written into the subprocesses regarding the action's membership to processes and regarding the branches defined by conditional actions. At the end the main process is written and all subproccesses are instantiated. The replication of subprocesses (processes) could be an attribute of the system model, which has to be regarded, but without limiting the generality we will always use the replication. How and when events are created will be presented in the explanation of the authenticity protection goal query.

The semantics of cryptography have to be encoded into each ProVerif model, either by adding a library to the command line or by adding all definition to the ProVerif model in the declaration part. Adding them individually to each model allows to only add the cryptographic primitives needed and to allow fine-grained control over the capabilities of the attacker, although in most cases the attacker will be able to execute all primitives in accordance to Kerckhoffs's principle [19] and Shannon's maxim [20]. We recommend to add the definitions individually to decrease the complexity for the solver. The needed primitives can be directly inferred from the used security measures in the system and adhere to the recommendations from the ProVerif manual [18], although we add the session id as an additional parameter to the primitives to model the passing of time and to allow to distinguish between ciphertexts from different sessions. This addition is presented in Figure 5 and further described with the privacy query.

The events mark important steps in the execution of a process and are needed to argue about authentication properties. They are part of the queries, which are needed to verify the protection goals of the system model. For the five presented protection goals four are representable by ProVerif queries including the options we described earlier. Availability can not be proven by ProVerif as it can not be guaranteed by cryptography (alone).

*1) Integrity:* can not be checked by ProVerif directly, with the meaning defined earlier. The consistency check would already warn if a node sends data, which it has no permission for. All other relevant cases are falling within the authenticity goal.

*2) Authenticity:* is a very diverse goal. Additionally, to the integrity, which is at the core of this property, it may demand to ascertain the identity of the sender and receiver of a message, to ascertain non-repudiation of sending and receiving, and/or to ascertain freshness of messages. Non-repudiation can be divided into two distinct requirements. It must be ensured that the sender and/or receiver of a message is authenticated and this fact has to be retrievable; i.e., by logging all messages. As

 ISBN: 978-1-61208-434-3 

such organizational measures can not be proven by ProVerif, we only prove the authentication requirement, but the logging requirement can be modelled by adding an attribute or node requirement to the security concept.

The authenticity property is modelled by events in ProVerif. Depending on the chosen combination of authenticated parties we directly create the events and correspondence assertions as described in [18]. Freshness is modelled by using the injective correspondence. They usually take the form of

*query x: key; inj-event(serverTerminates(x)) ==> inj-event(clientAccepts(x)).*

This can be interpreted as *if the server terminates while using key x*, then *client accepts key x* has happened exactly one time before.

*3) Confidentiality:* can directly be translated to a knowledge query: *query attacker(secret).* If knowledge permissions have been defined they can be checked for all nodes without permissions. This can be done by checking if the data instance is send in plaintext over all communication mediums the executing unit under evaluation is connected to. This can be extended to include ciphertexts using keys the executing unit knows.

*4) Privacy:* is interpreted as non-interference (in accordance with [18]) of the processes regarding the data instance expressed by *noninterf dataInstance.*. As outlined earlier we additionally allow to define non-traceability, which is translated to a knowledge query. If the attacker is not able to find a pair of duplicate ciphertexts (created with deterministic encryption) with the same payload and key from different sessions of the processes, then non-traceability is given.

This can be accomplished with the lines in Figure 5. The types of the data instance and the functions have been chosen for clarity and are not needed to implement the functionality. Type declarations have therefore been left out. We define one function to represent the traceability event and one reduction to allow the creation of the traceability event. When two ciphertexts, which were created with the same payload and key, but at two times, are combined, they create a traceability event. We then let ProVerif prove that if the attacker is able to generate a traceability event, then the two times must be equal. If they are not equal, a violation of the traceability goal was found.

### C. Attacker initialisation and execution

After the events and queries have been defined, all that is needed for verification is to initialise the attacker. This consists of building the attacker's knowledge set and capabilities. The knowledge set of the attacker consists of all public names and all the data instances known by the malign nodes. These data instances are sent once over a public channel, before all other processes execute, to allow the attacker to learn them. The capabilities are represented by the functions and reductions known to the attacker. In most cases the attacker will have knowledge of all functions and reductions used in the system. It is conceivable that additional capabilities can be given to the attacker under certain circumstances, i.e., when a cryptographic primitive becomes broken, but this should be a rare case. The ProVerif model can then be analysed and the results from the queries can be retrieved.

```
01    fun symmEnc(identityType , keyType,
02    timeType): identityCiphertextType.
03
04    reduc forall i: identityType,
05    k: keyType, t: timeType,
06    t2: timeType;
07    symmDec(symmEnc(i,k,t),k,t2) = i.
08
08    fun traceabilityEvent(
10    identityType, timeType, timeType):
11    traceabilityEventType.
12
13    reduc forall i: identityType,
14    k: keyType, t: timeType,
15    t2: timeType;
16    noticeDuplicateCiphertext(
17    symmEnc(i,k,t), symmEnc(i,k,t2)) =
18    traceabilityEvent(i, t, t2).
19
20    free identityA:
21    identityType [private].
22
23    query i:sid, i2:sid; attacker(
24    traceabilityEvent(
25    identityA, new t[!1 = i],
26    new t[!1=i2])) ==> i=i2.
27
28    process (
29    !(new t : timeType; (
30    (* instantiate other processes *)
```

Figure 5. Recognizing a tracability violation.

### D. Attack trace parsing

Given that the security engineer won't be interested in results without issues, we focus on the found attack traces if they exist. For each property the amount of possible attack traces is rather large, so the parsing has to be flexible. We recommend to condense several attack trace steps to increase legibility and to transport the relevant information. The different lines where the attacker learns all the different parts to start the attack are the first information that can be preprocessed. How to present further parts of an attack is up to the implementer of the methodology.

## VI. Conclusion and Future Work

We presented our generic system model core *SESMC* including the corresponding ontology. We believe that *SESMC* allows to create useful security requirement engineering methodologies. Furthermore, it allows to share algorithms and methods on the system model core, which can be extended to use the additional application domain specific extensions. We presented two different applicable methods. The consistency checking of the system model allows the security engineer to check whether the current state has none of the modelled contradictions and misconceptions, which should increase productivity and confidence in the system model. The verification with the security protocol analyser ProVerif allows to formally check the defined protection goals. Although ProVerif is limited by the modelled implementation details it allows to verify the system model at different stages in the design phase up to the first stages of implementation.

We will extend on our work to respect more of the criteria

 ISBN: 978-1-61208-434-3

outlined by [1] in the future. To create a complete security requirements engineering process with an accompanying methodology we have to describe the steps of the process and the essential results of each step. In addition, we see possibilities in encapsulation of security knowledge and usage of catalogues to allow for better reuse of engineering results and additional tool-support should be provided for all steps of the security requirements engineering process.

## ACKNOWLEDGEMENTS

The authors would like to thank Bruno Blanchet for answering the questions regarding ProVerif and Jörn Eichler for his useful insights in the state of the art of security engineering.

## REFERENCES

[1] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "Engineering security into distributed systems: A survey of methodologies." J. UCS, vol. 18, no. 20, 2012, pp. 2920–3006.

[2] B. Whyte and J. Harrison, State of Practice in Secure Software: Experts' Views on Best Ways Ahead. IGI Global, 2011, pp. 1–14.

[3] B. Blanchet, V. Cheval, X. Allamigeon, and B. Smyth, "Proverif: Cryptographic protocol verifier in the formal model," 2010, accessed August 30 2015. [Online]. Available: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[4] P. Amthor, W. E. Kühnhauser, and A. Pölck, "Worse: A workbench for model-based security engineering," Computers & Security, vol. 42, 2014, pp. 40–55.

[5] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "A comprehensive pattern-driven security methodology for distributed systems," in Software Engineering Conference (ASWEC), 2014 23rd Australian. IEEE, 2014, pp. 142–151.

[6] C. Bidan and V. Issarny, Security benefits from software architecture. Springer, 1997.

[7] A. Jaquith, "The security of applications: Not all are created equal," At Stake Research., 2002, accessed March 24 2011. [Online]. Available: http://www.atstake.com/research

[8] G. Hoglund and G. McGraw, Exploiting software: how to break code. Pearson Education India, 2004.

[9] G. Pedroza, L. Apvrille, and D. Knorreck, "Avatar: A SysML environment for the formal verification of safety and security properties," in New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on. IEEE, 2011, pp. 1–10.

[10] L. Apvrille and Y. Roudier, "SysML-Sec: A model-driven environment for developing secure embedded systems," Proc. of SARSSI 2013, Mont-de-Marsan, France, 2013.

[11] ——, "Towards the model-driven engineering of secure yet safe embedded systems," arXiv preprint arXiv:1404.1985, 2014.

[12] O. M. Group, "OMG systems modeling language," 2006, accessed August 30 2015. [Online]. Available: http://www.omgsysml.org/

[13] B. Kienhuis, E. F. Deprettere, P. Van Der Wolf, and K. Vissers, "A methodology to design programmable embedded systems," in Embedded processor design challenges. Springer, 2002, pp. 18–37.

[14] K. G. Larsen, P. Pettersson, and W. Yi, "Uppaal in a nutshell," International Journal on Software Tools for Technology Transfer (STTT), vol. 1, no. 1, 1997, pp. 134–152.

[15] D. Dolev and A. C. Yao, "On the security of public key protocols," Information Theory, IEEE Transactions on, vol. 29, no. 2, 1983, pp. 198–208.

[16] D. Sangiorgi and D. Walker, The pi-calculus: a Theory of Mobile Processes. Cambridge university press, 2003.

[17] S. J. Mellor, M. Balcer, and I. Foreword By-Jacoboson, Executable UML: A foundation for model-driven architectures. Addison-Wesley Longman Publishing Co., Inc., 2002.

[18] B. Blanchet and B. Smyth, "Proverif 1.89: Automatic cryptographic protocol verifier, user manual and tutorial," 2014, accessed August 30 2015. [Online]. Available: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[19] A. Kerckhoffs, La cryptographie militaire. University Microfilms, 1978.

[20] C. E. Shannon, "Communication theory of secrecy systems," Bell system technical journal, vol. 28, no. 4, 1949, pp. 656–715.

 ISBN: 978-1-61208-434-3 

# Embedded Network Combining CAN, ZigBee and DC-PLC for Motorhome

Fabienne Nouvel
IETR/INSA
Rennes, France
e-mail : fabienne.nouvel@insa-rennes.fr

Hussein Kdouh
IETR/INSA
Rennes, France
e-mail: hussein.kdouh@insa-rennes.fr

***Abstract*— Today, the number of motorhomes increases in Europe and North America as they offer greater individual freedom. As motorhome users spend the most of their time in their confined area, it seems essential to develop new solutions that make their life easier. In order to meet the new needs of customers, a new centralized architecture of a control system based on ubiquitous wired and wireless solutions is studied in this paper. The objective of this study is to verify the feasibility of ubiquitous technologies in this original environment. Different measurements have been conducted on a motorhome using Controller Area Networks (CAN), ZigBee and direct current power line communications (DC-PLC). Results have shown that these technologies may be used in a future hybrid control system in a motorhome.**

***Keywords*— Motorhome; Controller Area Networks; CAN; ZigBee; Power Line Communications;***

## I. INTRODUCTION

Recent results of motorhomes sales confirm the enthusiasm of people for this type of vehicles, especially in Europe and North America [1]. Motorhomes perfectly meet the needs of freedom, mobility and the wishes of autonomy that users seek. Otherwise, they may be an economic solution with respect to expensive hotel and trains reservation rates in touristic areas.

Motorhomes are similar to homes as they include a kitchen, a bathroom, sleeping facilities and a dining room. One can found several amendies such as refrigerator, gas cylinder, boiler, air-conditioner, folding step, and water (fresh, gray and black) tanks. Furthermore, similar electric installations can be found. Electrical equipments may be powered by different energy sources available in the motorhome, such as main and auxiliary house batteries, engine battery, solar cells and / or external alternative (AC) 230 V available in the camping areas.

Although today's motorhomes have a broad array of choices to be connected to Internet, motorhomes do not use communication network to interconnect the equipments in the home area.

Camper van can be either considered as a vehicle or as a mobile home. If it considered as a vehicle, electronic units are nowadays connected by networks. Many standards are available, the most widespread are control area network (CAN), local interconnect network (LIN) and Flexray [2]. These networks reduce the number of wires while achieving low to medium data rates and better reliability. However, these buses are placed in the motor area and mainly used for driving and automotive controls. Furthermore, they must be separated from the possible camper van "home network", due to security levels.

If we consider it as a "mobile home", many technologies have been developed to make the home smart. The technologies include solution for building automation as well as the domestic control activities [3]. Furthermore, devices may allow remote access.

It seems to be interesting to combine networks used for control driving with solutions used for domestic tasks. Actually, all the amenities in camper van are manually controlled by the users or via separated control systems. Some control systems are proposed by manufacturers but these solutions are proprietary solutions and not easy to modified. In this paper, we propose to replace separated non connected control systems in a camping-car by a centralized control system. Another objective is to replace point to point connection using a specific wire by a communication network that will reduce the weight and number of cables. Proposed new architecture may be based on both on wired and wireless communications technologies that connect a central unit to the different amenities in the motorhome.

Among the possible solutions, CAN seems to be attractive as it is widely used both in automotive [4] and in a variety of industries including building, automation and manufacturing. Moreover, CAN has been already used to connect home appliances for smart home applications [5] [6].

We have also to consider the wireless solution in order to offer easy plug and play applications while achieving low cost. Among the technologies, ZigBee, an IEEE 802.15.4 standard-based solution defined by the ZigBee Alliance [7], was developed specifically to support sensing, monitoring and control applications. The ZigBee solution offers significant benefits, such as low power, robust communication and a self-healing mesh network. Common applications supported by ZigBee include: personal monitoring, security, access control and safety monitoring, home, building and industrial automation [7].

As many devices in the motorhome are powered by AC or DC power, another solution to consider reducing the number of cables is power line communication (PLC). There is a growing in interest for PLC, including smart grids, home networking and control, as well as automotive uses [8].

 ISBN: 978-1-61208-434-3 

Hence, the proposed PLC technologies have been studied in cars and indoor environments, but they have not been tested in motorhome environment. Therefore, we will present in this paper the results obtained from different measurements in order to study the feasibility of these three technologies in motorhomes.

The remainder of this paper is organized as follows. Section 2 describes the functionalities and the architecture of the proposed control system. Section 3 deals with the CAN-based part of the system, including signals and messages definitions, network dimensioning and simulation and measurement results. Section 4 focuses on ZigBee tests conducted in the motorhome and presents the obtained results. Section 5 presents the results of PLC measurements and results. Finally, a conclusion is drawn in Section 6.

## II. System Functionalities and Architecture

The considered embedded control system must carry out several functionalities. We can cite according to the priority:

- Measuring capacity, voltage and current of available energy sources (main and auxiliary batteries, solar cells) and switching between available energy sources
- Measuring the energy consumption of appliances (refrigerator, TV, pumps, lights)
- Measuring weather data
- Generating alarm when detecting an intrusion
- Turning on/off and controlling brightness of lights
- Exchanging with the environment through internet.

We aim to develop a user-friendly control system that can accomplish these functionalities and possible supplementary ones. The system must provide high quality of service in terms of reliability, robustness, delays, and at the same time ensure the convenience and comfort of motorhome users.

To ensure the passengers' convenience and comfort, a unique user interface based on embedded OS (Linux, Android) will then replace the different interfaces used currently for motorhome appliances. Passengers will simply send orders or receive measurements to or from appliances by touching a user-friendly screen.

The second objective is ensuring reliable data communication between the central control unit and equipments. Different communication technologies exist in the market for this kind of applications. We have to select the best adapted solutions to our environment and applications.

The above mentioned functionalities may be categorized into three main categories: high level security, low level security and infotainment functionalities. The first two categories do not need high data rates but they must be secure, the frames are mainly orders or sensor measurements. However, infotainment functionalities, multimedia data, video surveillance or internet, necessitate high data rates.

Therefore, we propose to combine different communication technologies with respect to considered functionalities. High level security functionalities will be accomplished using sensor and actuators connected to corresponding equipments and exchanging via a CAN bus to a central control unit. Low level security functionalities such as measuring weather data and controlling lights will be accomplished by ZigBee nodes. High data rates functionalities will be finally exchanged using PLCs without using additional wires. Figure 1 shows the general architecture of proposed control system and communication technologies used with respect to functionalities.

In order to verify the feasibility of these technologies, we have conducted several measurements in a motorhome provided by the motorhome manufacturer Autostar. The motorhome comprises two main parts. The sleeping compartment contains a fixed bed, a shower cubicle and a wardrobe. The living compartment includes a kitchenette area with a refrigerator, grill and sink. Below the floor of the camper, we can find other amenities such as water tanks and pump, gas cylinders, batteries and boiler. Following sections will describe the performed experiments and present the obtained results. We will begin with CAN measurements.

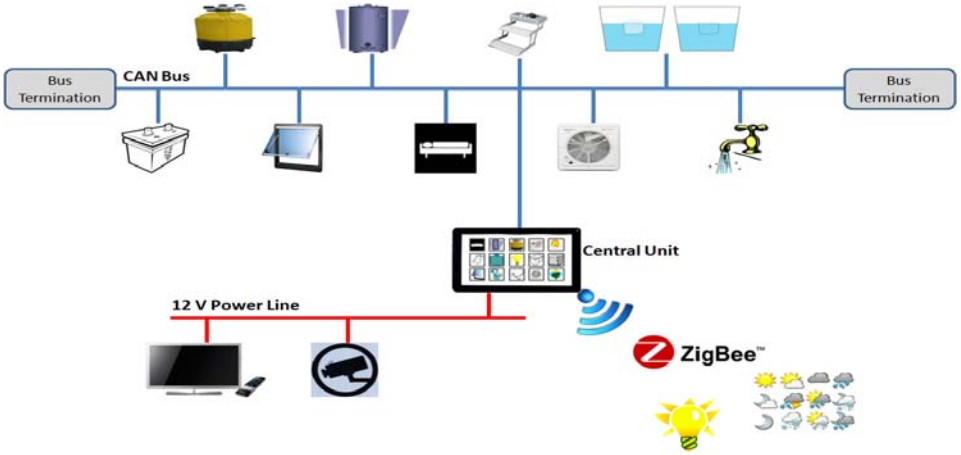


Figure 1. Architecture of the proposed control system based on CAN bus, ZigBee and PLC networks

 ISBN: 978-1-61208-434-3 

## III. CAN MEASUREMENTS

Most of system functionalities will be performed via the CAN bus, like monitoring water tanks, switching between energy sources, controlling air-conditioner and ventilator, etc. Several CAN nodes (each one connected to one or more equipment) will communicate to the central node in the camping car. The user interface will help the user to send orders and analyze measurements sent by sensors given by the other CAN nodes.

### *A. CAN Overview*

CAN 2.0A/B is a network protocol developed for connecting the sensors, actuators, and controllers in a vehicle. CAN supports data rates from 5 kbps up to 1 Mbps, which allow the CAN network to be used to share status information and is available for real-time control. The medium access control (MAC) level uses the carrier sense multiple access/collision detection (CSMA/CD) protocol to access the network, using the field "Identifier" of the frame.

CAN allows multiple devices (referred to as "CAN nodes") to connect to each other on a single bus. CAN nodes do not have strict master/slave roles. Instead, each CAN node may operate as a transmitter or receiver at any time. In reception, each node decides if the data is relevant by looking at the message frame's "Identifier", which describes the content of the message.

### *B. System Design*

The CAN bus functionalities are accomplished by sensors or actuators connected to corresponding equipments. A sensor (actuator) will send (receive) a digital signal to (from) the central CAN node via the CAN bus.

The first step in the network design was the definition of signals that will be exchanged through CAN messages and the identifiers between the central node and the appliances. A database of about 100 signals has been defined for the system.

The second step was the definition of CAN messages and the assignment of digital signals. One or several signals have been assigned to each CAN message. A total of 35 CAN messages/identifiers have been finally defined for the whole network. These messages can be categorized into four different categories:

- Order messages are generated by the central node and contain an order sent to one or several actuators, such as opening or closing a gas cylinder. This type of message may be generated manually by the user when touching the screen, or automatically by the central node for security reasons, such as ordering the closure gas cylinders and folding the step when the motorhome starts moving.
- Alert messages are automatically generated by CAN nodes to warn the central node about a special event, such as intrusion detections.
- Sensing messages are periodically generated by CAN nodes and contain sensor measurements such as water tank levels, batteries voltages, and others.
- Life sign messages are periodically generated by CAN nodes to ensure their connectivity to the bus. When a CAN node sends a life sign message, it waits for an acknowledgment from the central node. In absence of this acknowledgment, it repeats the transmission of the life sign message for a user-predefined maximum number of trials. If no acknowledgment is received, a CAN node will be considered disconnected from the bus and must stop sending its alert and sensing messages. However, receiving any type of message (acknowledgment or order message) from the central node will reactivate the CAN node.

As the identifier of a CAN message determines its priority on the bus, we have assigned to each message an identifier based on its priority in the application. Lowest identifiers (highest priority) have been assigned to order messages. The second, third, and fourth levels of priority have been assigned to alert messages, sensing messages and life sign messages respectively.

The third step is the network dimensioning, i.e., the estimation of number of CAN nodes in the network. In order to minimize the number of CAN nodes in the network, we have connected equipments that are nearly placed in the motorhome, to the same CAN node. For example, Gas cylinders and water tanks have been connected to one CAN node.

### *C. CAN Bus Simulation*

To study the performance of our CAN architecture, we have used the development and testing software tool CANoe from Vector GmbH. CANoe is a versatile tool for the development, testing and analysis of entire Electronic Control Unit (ECU) networks as well as individual ECUs. It supports network designers, development and test engineers at equipment manufacturers and suppliers over the entire development process – from planning to the start-up of entire distributed systems or individual ECUs. CANoe supports CAN, LIN, FlexRay and other networks.

At the beginning, CANoe has been used to create simulation models which simulate the behavior of the CAN nodes. Over the further steps of nodes development, these models serve as the basis for analysis, testing and the integration of bus systems and nodes. This makes it possible to detect problems early and correct them. Graphic and text based evaluation windows are provided for evaluating the results [9].

The simulation procedure consists of developing all CAN nodes, exchanged signals and messages, and finally a GUI to simulate the central control node. CAN nodes have been firstly developed through Vector CAPL (CAN Access Programming Language) based on the C programming language. Using CANoe in combination with CAPL makes it possible to create custom tool applications with user defined behavior. Secondly, the database of CAN messages and

 ISBN: 978-1-61208-434-3 

signals is recreated in Vector CANdb++ (which is a data administration program with which communication databases can be created and modified in the form of CAN databases) and added to the simulated bus. Signal generators have been added to generate random or predefined sensing values to the CAN nodes. Figure 2 explains the architecture of our CAN bus simulator using CANoe.

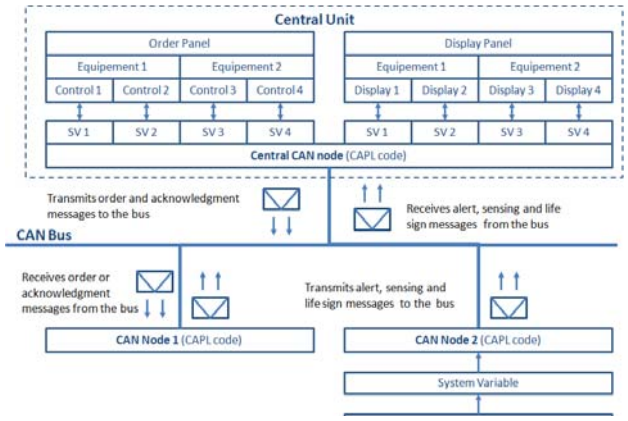


Figure 2. Architecture of the CAN bus simulator using CANoe.

Several simulations scenarios have been carried out to verify the correct behavior of individual CAN nodes (success and periodicity of CAN messages transmission) and the whole network (bus load and end-to-end delays). As stated before, this step helps us to detect problems early to prevent them in the implementation phase. Different data rates between 5 up to 1000 kbps and periods of data transmission, between 0.2 and 10 seconds, have been tested. In all simulation scenarios, all nodes send their CAN messages at the same time to simulate worst case scenarios.

By studying the simulation database and filtering messages by CAN node, we have found that all nodes can send successfully their CAN messages to the central unit for data rates higher than 50 kbps regardless the period of transmission. However, the bus load may reach values from 20 to 100% for rates less than 100 kbps. The maximum end-to-end delay, defined as the time between the generation of a CAN message by a CAN node and its reception by the central node, is less than 100 ms for all data rates higher than 30 kbps.

Using these results, a data rate of 500 kbps and a period of 1 second have been adopted in our application. For this scenario, a bus load of only 0.77% and a maximum end-to-end delay of 3.79 ms have been obtained. These values show that data traffic exchanged in our application is largely lower than the capacity of the CAN network. That proves also that possible adding of future supplementary functionalities, will not affect the performance of existing network. Simulation results motivate us to start the implementation step.

### *D. Implementation and Tests of CAN Nodes*

Each CAN node has been developed using an ARM Cortex-M3 LPC1758 board from NXP Semiconductors which supports two CAN peripherals. A generic embedded C code has been developed for all CAN nodes. A 4 bits local address represented by a 4 bits input to the microcontroller, has been assigned to each CAN node. It determines the part of the generic code to be executed to perform the node functionalities. Based on the CAN node address, different General Purpose Input Output (GPIO), serial ports and/or analog-to-digital converter pins are configured to exchange digital signals with connected sensors and actuators.

CANoe has been also used to test the behavior of the developed CAN nodes. Using a particular interface, CANoe provides the possibility of replacing one, several or all simulated nodes by real ones. The conducted test includes 3 real CAN nodes ( based on the LPC1758-microcontroller) and 4 simulated CAN nodes as well as the simulated central node. The interface CANcardXL allows mixing hardware and software running CANoe (on a laptop). In order to test the behavior of all CAN nodes, the test has been repeated several times. In each test, identifiers of simulated and real CAN nodes have been exchanged.

The CANcardXL interface is a hardware interface between the real bus and the simulated one. This hybrid network has been tested each time for 3.5 hours (12600 seconds) and each CAN node was sending its sensing message every one second and its life sign message every 15 minutes. The data rate was set to 500 kbps for all nodes.

By studying the database of exchanged CAN messages provided by the Trace window of CANoe, we have found that each CAN has sent 12600 sensing messages to the central node as well as 16 life sign messages, which means that all CAN messages have been successfully sent. Moreover, the period between two successive CAN messages was around 1 second (with several tens or hundreds of microseconds corresponding to the arbitration phase between CAN nodes trying to send at the same time).

A last experiment has been carried out without simulation using different hardware CAN nodes and a central node emulated thanks to an USB-MUX-4C2L board. Similarly, the results of this test have shown a frame transmission ratio equal to 100% for the CAN nodes and the periodicity of transmission (one second between two successive messages) has been also respected.

## IV. ZIGBEE MEASUREMENTS

Several low rate and low security level functionalities such as turning on and off lights, measuring external temperature, wind speed, TNT signal levels may be communicated to central node by ZigBee communications. In order to verify the possibility of wireless communications between central and peripheral actuator/sensor nodes inside and outside the motorhome, point-to-point measurements have been performed using two ZigBee nodes in a motorhome to evaluate the radio link quality.

### *A. Measurement Procedure*

Two ZigBee development boards from Silicon Laboratories have been used in these measurements (one transmitter (Tx) and one receiver (Rx)). Each board features a

 ISBN: 978-1-61208-434-3 

a C8051F121 microcontroller and a Chipcon CC2420 2.4 GHz 802.15.4 transceiver. The used frequency was 2.45 GHz and the transmit power was 0 dBm for both nodes. The sensitivity of both ZigBee nodes is -95 dBm. Tx was connected to a laptop via USB connection to configure the measurement parameters and to save results. Tx, which represents the central node, was fixed above the entrance of the motorhome. Rx has been moved between 9 different locations inside and outside the motorhome. For each location, 180 packets have been transmitted by Tx and the Packet Delivery Ratio (PDR) (percentage of packets received successfully) as well as the Received Signal Strength Indication (RSSI) have been measured. A packet is considered as successfully received only when Tx receives an acknowledgment from Rx.

Figure 3 represents the mapping of the motorhome and the locations of Tx and Rx nodes. Locations A, B, C, D, F, G, H and I are all inside the motorhome. However, location E is on the roof of the camper.

### B. *Results*

Table 1 presents obtained measurement results. For each Rx location, we give the minimal, maximal, and mean values of RSSI, as well as the PDR.

Measurement results of locations A, B, C, D, F, G, H and I show an excellent connectivity inside the motorhome with a PDR of 100 % and an average of RSSI above -70 dBm. The wireless node of the central control unit may have a reliable communication with all the ZigBee sensor/actuator nodes placed inside the motorhome. Closing the curtain (represented by a green dashed line in Figure 3) which separates the two compartments and blocking the line-of-sight between Tx and Rx at locations (A, B, C, D and F) has not affected the link quality. Moreover, the communication between Tx and Rx (location D) has not been blocked when closing the door of wardrobe, in spite of the attenuation of 10 dB with respect to other locations in the sleeping compartment. This remarkable link quality may be explained by the short distances separating Tx and Rx (the length of the motorhome is only 7 m) and the absence of severe obstacles that can totally block wireless communication (metallic walls for example).

However, location E (on the roof of the camper) does not show the same link quality. PDR decreased to 93% and the average of RSSI to -77 dBm. There is no direct visibility between Tx and Rx for this location. Radio signals are received by Rx due to reflections on walls and ceiling. It is worth mentioning here that all these measurements were performed when the motorhome was parked in a huge hangar. Therefore, reflections on the walls of the hangar guided radio signals between Tx and Rx. Although the mentioned results are not very bad, they may be worst if the motorhome was moving in an open area (outside reflections). In order to have an efficient connection between the central node and the external node, it will be recommended to place the node inside the motorhome and connect it by short cables to external sensors (temperature, humidity, etc.). In this case, successful data transmission will be more guaranteed and independent of the motorhome location. However, in this system, the security aspect has not been considered. It may be taken into account in future work.

TABLE I. OBTAINED RESULTS FROM ZIGBEE MEASUREMENTS

| Rx Location | RSSI (dBm) | | | PDR (%) |
|---|---|---|---|---|
| | *Minimal* | *Maximal* | *Mean* | |
| A | -67 | -54 | -58.95 | 100 |
| B | -62 | -54 | -57.43 | 100 |
| C | -65 | -49 | -54.82 | 100 |
| D | -76 | -61 | -69.93 | 100 |
| E | -85 | -60 | -77.12 | 93 |
| F | -62 | -54 | -59.15 | 100 |
| G | -58 | -51 | -55.91 | 100 |
| H | -47 | -55 | -49.44 | 100 |
| I | -60 | -50 | -56.95 | 100 |

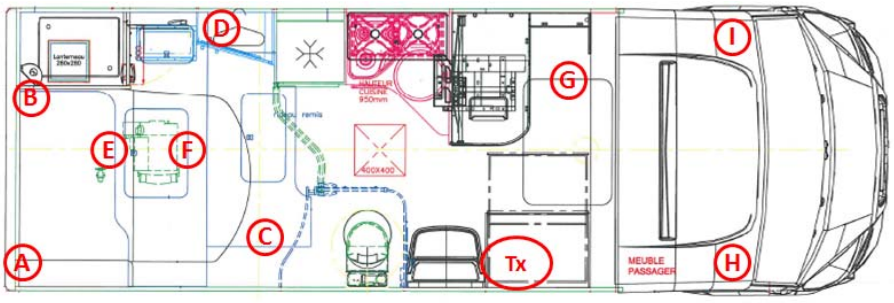


Figure 3. Locations of ZigBee transmitter and receiver in the motorhome.

 ISBN: 978-1-61208-434-3 

## V. PLC MEASUREMENTS

Several functionalities like video monitoring, multimedia exchange or Internet sharing may be added to the system. These functionalities do not need high level of security, but need a relatively high throughput which cannot be provided by CAN or ZigBee networks. We propose to study the feasibility of PLC inside the motorhome, on the DC power lines. This technology does not need new wires and is already used in indoor networks. Two types of measurements have been carried out in the same motorhome. The first one studies the transfer function on the electrical harness and the second one uses indoor PLC modems to determine the possible throughput in a motorhome.

### A. *Transfer Functions*

In order to determine the best frequency band for PLCs as well as the effect of the battery and amenities in a motorhome, we started by measuring the transfer function of the electrical harness of a motorhome. Autostar has provided an electrical harness before its installation in a motorhome as well as a motorhome with installed harness. Using a Vector Network Analyzer (VNA), we have measured the S21 parameter for two different locations (AB and AC) on the harness in three different scenarios: a free harness (without equipment and battery connected), in the motorhome with only connected equipments (without battery) and finally with connected battery and equipments. The 12 V DC battery of the camper is totally independent from the engine one. The measurement point B is located in the living room and corresponds to the television location. Position C is located in the bed room. The performed measurements may simulate multimedia sharing between the two compartments of the motorhome. The measurement parameters are presented in Table 2.

Figure 4 shows the measurements positions in the motorhome. Figure 5 and Figure 6 present the obtained results in the three scenarios for paths AB and AC, respectively. One can observe a remarkable attenuation for frequencies less than 3 MHz may be noticed for the two scenarios. Secondly, the effect of connecting equipments start at frequencies higher than 50 MHz, however the effect of connecting a battery is clear for all frequencies (an average of 15 dB of attenuation). Finally, we can clearly conclude from these measurements that frequency band [3, 30 MHz] is the less affected by the battery and equipments and may be convenient for PLCs in a motorhome. More particularly, the bandwidth between 8 and 14 MHz has a minimum attenuation in this band.

TABLE II. S21 MEASUREMENT PARAMETERS

| | |
|---|---|
| VNA Model | Agilent FieldFox Handheld Analyzer N9918A |
| Frrequency band | [30 kHz – 100 MHz] |
| Resolution | 4001 points (24,986 kHz) |
| IF bandwidth | 30 kHz |
| Coupling | Capacitive |

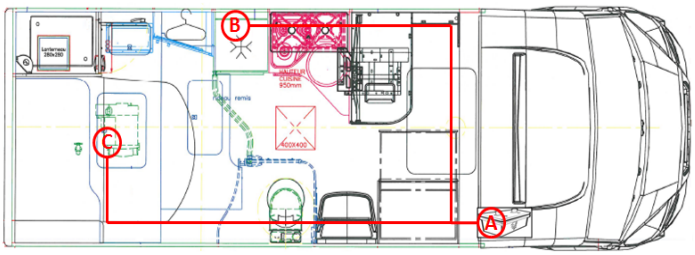


Figure 4. Locations of DC-PLC transmitter and receiver in the motorhome.

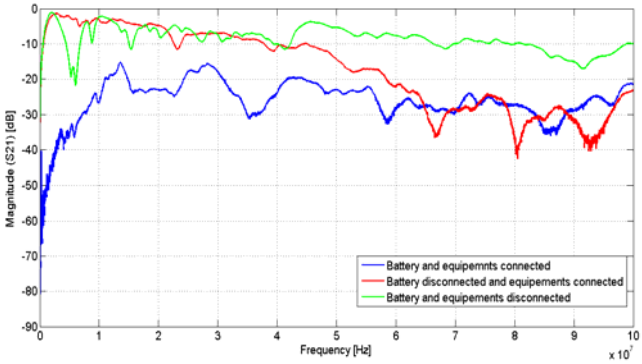


Figure 5. Transfer function (S21) for path AB.

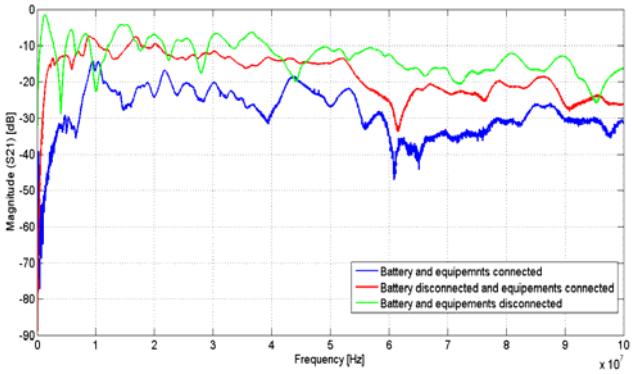


Figure 6. Transfer function (S21) for path AC.

### B. *Throughput Measurements*

The study of PLC throughputs is realized using PLC Devolo 200Av modems [10] based on the HomePlug AV standard [11]. These modems are commonly used in indoor networks and cars [12] and high throughputs have been demonstrated. It is interesting to test them in a motorhome. As the power line network in-motorhome is different from that in a house, the modems have been modified to be used and plug on the DC power lines. The modifications affect mainly the coupling and the power supply.

The HomePlug AV standard is the second generation of PLC systems developed by the HomePlug Powerline Alliance. Now it is suitable for multimedia applications like HDTV or

 ISBN: 978-1-61208-434-3 

VOIP. The PHY layer uses a windowed-OFDM modulation in the [2 – 28 MHz] frequency band over 972 subcarriers. The HPAV can use different modulation order from binary phase-shift keying (BPSK) up to 1024 quadrature amplitude modulation (QAM) for each sub-carrier according to the channel characteristics. To counteract the channel multipath effects and the inter-symbol interferences, the HPAV uses a guard interval (GI). Moreover, several GI (5.56 µs, to 47.12 µs) can be used depending on the channel and so the throughput can be improved. A frequency mask is used to avoid interferences mainly with amateur radio bands. This is the reason why the pulse-shaped OFDM symbols is different than the classic rectangular window. Thanks to this specific window, the out-of-band noise is reduced and the notches are deeper. The central coordinator uses the channel estimation in order to establish a specific QAM modulation for each OFDM sub-carrier. HPAV uses a two-level MAC framing scheme. Indeed, the MAC frames are divided into 512 bytes segments called PHY Block (PB). An uncorrectable Forward Error Correction (FEC) code is used and a header is added with the numbers of the PB.

In the measurement setup, we use a test bed with two PLC modems and two laptops running jperf tool, which is used for network testing by transmitting transmission control protocol (TCP) and user datagram protocol (UDP) streams. These modems have been plugged firstly to positions A and B, and then to positions A and C respectively to measure the throughput of the same scenarios studied in the previous section.

Figure 7 shows the obtained throughput for the two paths with disconnected and connected battery respectively. We can remark from these scenarios that obtained throughputs are quasi-similar for the two paths and are about 20 Mbps, which is higher than throughputs provided by other bus technologies (maximum 10 Mbps for FlexRay). However, we can notice a small decrease of about 1 Mbps for the two paths when the battery is connected. Similar results have been achieved with UDP and so not represented. These results show that HPAV modems can be used for PLCs in a motorhome for multi-media applications.

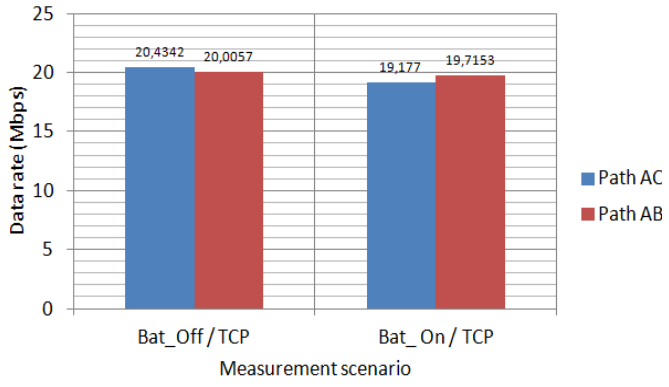


Figure 7. Obtained throughputs for paths AB and AC.

## VI. CONCLUSION

We have presented in this paper centralized hybrid architecture for motorhome control mainly based on CAN bus, in addition to PLC and ZigBee technologies. Different simulations and measurements have been carried out to study the performance of these technologies in the motorhome.

Firstly, CAN bus simulations have been carried out using CANoe to evaluate the performance of the future network. Then, CAN nodes board and an embedded C code have been developed and tested successfully on the network in the motorhome.

Otherwise, ZigBee measurements have shown an excellent link quality between the central node and all locations inside the motorhome. Motorhome environment is not a difficult from a “radio wave propagation” view, as distances between nodes will be small and there is no severe obstacle that can totally block wireless communications. The security may be considered in future work.

Finally, PLC experimentations have shown the possibility of using PLC modems used in indoor applications in a motorhome. Possible data rates of about 20 Mbps in realistic scenarios can be provided by PLCs.

The obtained results confirm the possibility of using ubiquitous communication technologies to achieve a smart, convenient and comfortable motorhome. In future work, electromagnetic measurements will be performed in order to test the robustness of the network in real environment with the camper van in motion.

## REFERENCES

[1] RVBusiness, “SSI: Motorhomes sales up 9.1% during October”, Available from http://www.rvbusiness.com/2014/12/ssi-motorhomes-sales-see-9-1-rise-in-october, [retrieved 12, 2014].

[2] F.Nouvel, W. Gouret, P. Maziero and G. EL Zein, " Automotive Network Architecture for ECUs Communications", in Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange, IGI Global, pp 69-90, 2009.

[3] M. Ortiz, M. Diaz, Manuel, F. Bellido,E. Saez, Edmundo and F. Quiles, "Smart Home Automation Using Controller Area Network", International Symposium on Distributed Computing and Artificial Intelligence, pp 167-174, 2011.

[4] K. H. Johansson, M. Törngren and L. Nielsen, “Vehicle Applications of Controller Area Network,” in Handbook of Networked and Embedded Control Systems Control, 1st ed. Basel, Switzerland: Birkhäuser Basel, sec. 6.1, pp 741-765, 2005.

[5] K. C. Lee and H. H. Lee, “Network-based fire-detection system via controller area network for smart home automation,” IEEE Trans. on Consumer Electronics, vol. 50, no. 4, pp. 1093-1100, 2004.

[6] K. H. The, W. L. Ng, C. K. Ng and N. K. Noordin, “Home Appliances Management System using Controller Area Network (CAN) Communications", APCC, 17th Asia-Pacific Conference on , pp.899-904, 2011.

[7] Website of ZigBee Alliance, ZigBee Alliance. [Online]. Available: http://www.zigbee.org.

[8] X. Lu; Y. Sun; and I. H. Kim, "Reliable power line communication A vehicle to smart home and smart energy," Consumer Electronics (ICCE), 2012 IEEE International Conference on, pp.86-87, 2012.

[9] Website of Vector Informatic GmbH. Vector Informatic GmbH. Available at http://www.vector.com [retrieved 6, 2015].

[10] Website of PLC modem Devolo. Available at http://www.devolo.com

 ISBN: 978-1-61208-434-3 

[11] Official website of the HomePlug Powerline Alliance (HPA). HomePlug Powerline Alliance. Available at http://www.homeplug.org [retrieved 08, 2015].

[12] P. Tanguy, F. Nouvel, P. Maziearo, "Power Line Communication standards for in-vehicule networks," 9th ITST2009, Intenational Conference on Intelligent Transport Systems Telecommunications, pp.533-537, 2009.

 ISBN: 978-1-61208-434-3 