VEHICULAR 2014

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

ISBN: 978-1-61208-348-3

June 22 - 26, 2014

Seville, Spain

VEHICULAR 2014 Editors

Steffen Fries, Siemens, Germany

Taimoor Abbas, Lund University, Sweden

VEHICULAR 2014

Foreword

The Third International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2014), held between June 22-26, 2014 - Seville, Spain, continued the inaugural event 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.

We take here the opportunity to warmly thank all the members of the VEHICULAR 2014 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 who dedicated much of their time and efforts to contribute to VEHICULAR 2014. 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 are grateful to the members of the VEHICULAR 2014 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that VEHICULAR 2014 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of vehicular systems, technologies and applications.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the charm of Seville, Spain.

VEHICULAR 2012 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
Andre Weimerskirch, ESCRYPT Inc., USA
Alexandre Bouard, BMW Forschung und Technik GmbH, Germany
Daniel Jiang, Mercedes-Benz Research & Development North America, USA
Peter Knapik, Volkswagen AG, Germany
Norbert Bissmeyer, Fraunhofer SIT, 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 2014

Committee

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

Andre Weimerskirch, ESCRYPT Inc., USA
Alexandre Bouard, BMW Forschung und Technik GmbH, Germany
Daniel Jiang, Mercedes-Benz Research & Development North America, USA
Peter Knapik, Volkswagen AG, Germany
Norbert Bissmeyer, Fraunhofer SIT, 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 2014 Technical Program Committee

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

Yuanguo Bi, Northeastern University, China
Norbert Bissmeyer, Fraunhofer SIT, Germany
Pascal Bodin, Orange Labs, France
Ghaleb Bolos, ESIGELEC, France
Mélanie Bouroche, Trinity College Dublin, Ireland
Robert Budde, TU Dortmund University, Germany
Darcy M. Bullock, Purdue University, USA
Chiara Buratti, DEIS, University of Bologna, Italy
Maria Calderon, University Carlos III of Madrid, Spain
Claudia Campolo, University of Reggio Calabria, Italy
Jean Pierre Cances, University of Limoges, France
Lien-Wu Chen, Feng Chia University - Taichung, Taiwan
Ray-Guang Cheng, National Taiwan University of Science and Technology - Taipei, Taiwan, R.O.C.
Yonggang Chi, Harbin Institute of Technology, China
Dong Ho Cho, Korea Advanced Institute of Science and Technology - Daejeon, Republic of Korea
Juan Antonio Cordero Fuertes, INRIA, France
Naim Dahnoun, University of Bristol, UK
Carl James Debono, University of Malta - Msida, Malta
Stefan Dietzel, University of Ulm, Germany
David Hung-Chang Du, University of Minnesota, USA
Trung Q. Duong, Blekinge Institute of Technology, Sweden
Weiwei Fang (方维维), Beijing Jiaotong University (BJTU) - Beijing, China
Michel Ferreira, University of Porto and Instituto de Telecomunicações, Portugal
Alois Ferscha, Institut für Pervasive Computing, Johannes Kepler Universität Linz, Austria
Serge Fdida, UPMC Sorbonne University, France
Malgorzata Gajewska, Gdansk University of Technology, Poland
Slawomir Gajewski, Gdansk University of Technology, Poland
Hassan Ghasemzadeh, Washington State University, USA
Athanasios Gkelias, Imperial College London, UK
Benjamin Glas, ETAS GmbH, Germany
Javier Gozalvez, UWICORE Laboratory, University Miguel Hernandez of Elche, Spain
An He, Qualcomm, USA
Khelifa Hettak, Industry Canada / Communications Research Centre - Nepean, Canada
Daesik Hong, Yonsei University - Seoul, Korea
Javier Ibanez-Guzman, Renault S.A., France
Satish Chandra Jha, Intel Corporation, USA
Daniel Jiang, Mercedes-Benz Research & Development North America, USA
Felipe Jimenez, Technical University of Madrid (UPM), Spain
Georgios Karagiannis, University of Twente, The Netherlands
Gunes Karabulut Kurt, Istanbul Technical University - Istanbul, Turkey
Frank Kargl, University of Ulm, Germany
Wolfgang Kiess, DOCOMO Euro-Labs, Germany
Jungwoo Lee, Seoul National University, Korea

XiangYang Li, Illinois Institute of Technology - Chicago, USA
Qilian Liang, University of Texas at Arlington, USA
Kuang-Hao Lin, National Chin-Yi University of Technology, Taiwan
Thomas Little, Boston University, USA
Rongxing Lu, University of Waterloo, Canada
Johan Lukkien, Eindhoven University of Technology, The Netherlands
Barbara M. Masini, CNR - IEIIT, University of Bologna, Italy
João Mendes-Moreira, Universidade do Porto and LIAAD-INESC TEC L.A., Portugal
Ingrid Moerman, Ghent University - IBBT, Belgium
John Morris, Mahasarakham University, Thailand
Hidekazu Murata, Kyoto University, Japan
Jose Eugenio Naranjo Hernandez, Universidad Politecnica de Madrid, Spain
Arnaldo Oliveira, Universidade de Aveiro, Portugal
Shumao Ou, Oxford Brookes University, UK
Mohammad Patwary, Staffordshire University, UK
Matthias Uwe Pätzold, University of Agder - Grimstad, Norway
Marco Picone, University of Parma, Italy
Adrian Popescu, Blekinge Institute of Technology - Karlskrona, Sweden
Ravi Prakash, University of Texas at Dallas, USA
M. Elena Renda, IIT - CNR - Pisa, Italy
Tapani Ristaniemi, University of Jyväskylä, Finland
Marco Roccetti, University of Bologna, Italy
Vitor Santos, University of Aveiro, Portugal
Susana Sargento, University of Aveiro, Portugal
Miguel Sepulcre, University Miguel Hernandez of Elche, Spain
Won-Yong Shin, Harvard University, USA
Marcin Sokól, Gdansk University of Technology, Poland
Hwangjun Song, POSTECH (Pohang Univ of Science and Technology) - Pohang, Korea
Kemal Ertugrul Tepe, University of Windsor, Canada
Necmi Taspinar, Erciyes University - Kayseri, Turkey
Olav Tirkkonen, Aalto University, Finland
Theodoros A. Tsiftsis, Technological Educational Institute of Lamia, Greece
Carlo Vallati, University of Pisa, Italy
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
Andre Weimerskirch, University of Michigan Transportation Research Institute (UMTRI), USA
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

Table of Contents

Rotated Constellations for a Satellite Communication Link in a DVB-T2 context

Benjamin Ros, Frederic Lacoste
Satellite telecommunications systems department
CNES (French Space Agency)
Toulouse, France
Benjamin.Ros@cnes.fr, Frederic.Lacoste@cnes.fr

Kevin Burgi, Charbel Abdel Nour
Electronics department
Institut Mines-Télécom / Télécom Bretagne
CNRS Lab-STICC, UMR 6285, Brest, France
firstname.lastname@telecom-bretagne.eu

***Abstract*— In this paper, the focus is on DVB-T2-like rotated constellations techniques applied to a satellite link for mobile services. Associated with signal space component interleaving, a QPSK constellation with rotation is evaluated over S-band satellite mobile channel with payload impairments. Channel capacity is first assessed showing an important potential for improvement. Largest gains are obtained for high spectral efficiency and for the cases with high level of impairments. Coded simulations show that rotated constellation can bring up to 1.5 dB gain for medium to high coding rates over a satellite to vehicle channel, confirming channel capacity predictions.**

Keywords*— *rotated constellations, DVB-T2, DVB-NGH, signal interleaving, satellite payload impairments, satellite mobile vehicular link

I. Introduction

The second generation of the Digital Video Broadcasting Terrestrial standard (DVB-T2) [1], shows excellent results over terrestrial channels, with limited mobility. To cope with higher mobility requirements with the possibility of covering a larger interval of signal to noise ratios and spectral efficiencies, DVB-NGH specifications have been drafted. They introduce advanced Multiple Input Multiple Output (MIMO) antennas schemes, low coding rates and a complementary satellite component for a larger coverage. Indeed, satellite transmission seems to be an attractive and efficient way to broadcast TV or radio over large areas. More recently, ITU studies [10] or 5GPP reflexion groups have promoted satellite as a service or a coverage complement of terrestrial systems. But for mass market considerations, use of very-well designed waveforms for satellite link [9] is not to consider, in the idea of having the same chipset for receiving terrestrial and satellite link. Non optimized transmission using a terrestrial standard with well choosen settings may be preferred. This thought may explain why in this paper, DVB-T2 transmission is studied on a satellite link. Thus, as rotated constellations are considered to improve the quality of transmission on terrestrial channels, purpose of this work is to assess if such technique may bring such benefits over satellite vehicular channel.

In a conventional Quadrature Amplitude Modulation (QAM), half of the information bits are carried by the I component and the remaining half by the Q component. When constellation rotation is considered, every constellation point in signal space has its own projection on the I and Q components separately. Therefore, information regarding all bits to be transmitted is carried over both I and Q components. Component interleaving is performed afterwards such that the I and Q signals carrying the information regarding one symbol are sent in a different time and over a different subcarrier of the DVB-T2 Orthogonal Frequency Division Multiplexing module. Now, if deep fading occurs at a particular time and frequency, it is highly unlikely that it would affect both I and Q components of the same symbol. Thanks to increased robustness, rotated constellations were adopted in the DVB-T2 and DVB-NGH standards [1] [2] [3].

Nevertheless, despite this adoption, no particular study on its potential benefits over a mobile satellite channel has been performed. In this paper, transmission from a geostationary satellite over S-band mobile propagation channel is studied. Roof top vehicular antenna affected by intermediate tree shadowing is considered at receiver side. The underlying channel aggregates propagation effects and payload impairments such as non-linearity effects, phase noise, and filtering.

The paper is organized as follows: first, the principle of constellation rotation is recalled in Section II. Then, the Land Mobile Satellite (LMS) three-state channel model over S-band is defined in Section III. It is followed by channel capacity computations for rotated and classical QPSK constellation in Section IV. Coded simulation results for a DVB-T2-like waveform with constellation rotation are provided in Section V. These simulations equally include satellite payload impairments. Section VI concludes the paper.

II. On the rotation of constellation principle

Assuming that a Rayleigh-like fading can occur independently on the phase and quadrature component of a constellation symbol, the situation depicted in Fig. 1 represents a typical study case for constellation rotation. Bit information is carried by both I and Q and therefore is not totally lost. Half of the total amount of transmitted bits would have been affected by deep fading in case of non-rotated constellation. Nonetheless, independent fading is assumed possible thanks to frequency/time I/Q interleaving.

Indeed, real propagation channel show correlation both in time and in frequency.

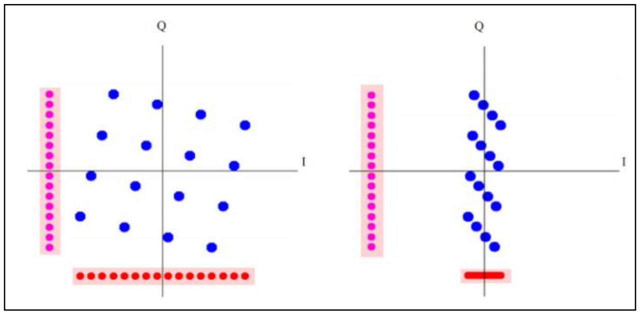


Figure 1. On the left hand side: rotated constellation without fading; on the rigth hand side: rotated constellation with fading on the I component.

The general architecture for a constellation rotation transmitter adopts a Bit Interleaved Coded Modulation scheme (BICM) [4] as depicted in Fig. 2. It is composed of the association of an outer error correcting code and a QAM mapper applying a constellation rotation and I/Q interleaving separated by a bit interleaver. The code represents a combination of a Bose-Chaudury-Hocquenghem (BCH) code and a Low Density Parity Check (LDPC) code. I/Q interleaving is performed in time an frequency via a set of cell, time and frequency interleaving as defined in the DVB-T2 standard [1]. Interleaved components are afterwards mapped to OFDM subcarriers according to the DVB-T2 frame builder.

Rotation angle Φ was chosen following a set of predefined optimization criteria related to the L-product distance [5] and signal space characteristics in case of deep fades [3]. The choice is assessed under a DVB-T2 context, considering that fading over I and Q component is uncorrelated. This condition is reached thanks to the long frequency and time interleavers adopted in the DVB-T2 standard. Adopted rotation angle values in this article, the same as in DVB-T2 thanks to the interleavers, are summarized in TABLE I.

TABLE I. ROTATION ANGLE FOR A DVB-T2 CONTEXT

Modulation	**QPSK**	**16-QAM**	**64-QAM**	**256-QAM**
Φ (degrees)	***29,0***	***16,8***	***8,6***	***3,6***

At the receiver side (see Fig. 3), one main difference exists when compared with a conventional bit interleaved coded modulation with an OFDM receiver. Log-likelihood ratios (LLRs) provided by the demapper are now computed over rotated symbols. This induces the use of a 2-component (in-phase and quadrature) demapper. Indeed, the Euclidean distance of received symbol now should be computed over the two projections over the I and Q axes as follows:

$$LLR(b_i) = ln \frac{\sum_{x \in C_i^1} exp\left(-\frac{\left|I-\rho_I I_x\right|^2+\left|Q-\rho_Q Q_x\right|^2}{2\sigma^2}\right)}{\sum_{x \in C_i^0} exp\left(-\frac{\left|I-\rho_I I_x\right|^2+\left|Q-\rho_Q Q_x\right|^2}{2\sigma^2}\right)} \quad (1)$$

where x is a symbol of the QAM constellation, C_i^j represent the symbols of the constellation carrying the bit b_i when b_i is equal to j, I and Q are the received in phase and quadrature components, $\rho_{I/Q}$ is the fading on the I or Q component, $2\sigma^2$ is the Additive White Gaussian (AWGN) noise variance, I_x and Q_x denote the reference symbols of the rotated QAM constellation.

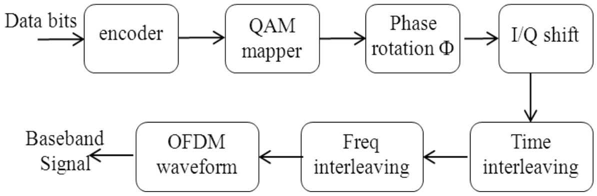


Figure 2. General architecture for an OFDM transmitter with a rotation of constellation

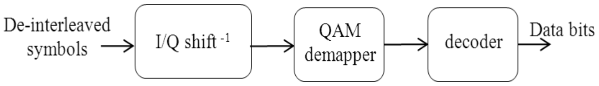


Figure 3. Processing at the receiver for rotated constellations.

III. S-BAND 3 STATES MODEL

This model detailed in [6] has been widely used since the 90's. It is fully empirical (or equivalently statistical) and relies on a specific S-band measurements dataset including various environments (open, intermediate tree shadow, heavy tree shadow, suburban, urban). Basically, it divides the LMS propagation channel into 3 shadowing states:

- State 1 : "LOS" -Line of Sight-
- State 2 : "Shadowing", corresponding typically to isolated trees in suburban areas
- State 3 : "Heavy Shadowing/Blockage" corresponding typically to houses in suburban areas.

In the generative model, the state change (large scale) is synthesised using a 3-state Markov chain whose input parameters are the initial state vector and the transition matrix. Two more input parameters have also been added:

- Each state is assumed to have a minimum length ("state length")
- A transition between two states happens over a given transition distance ("state transition length").

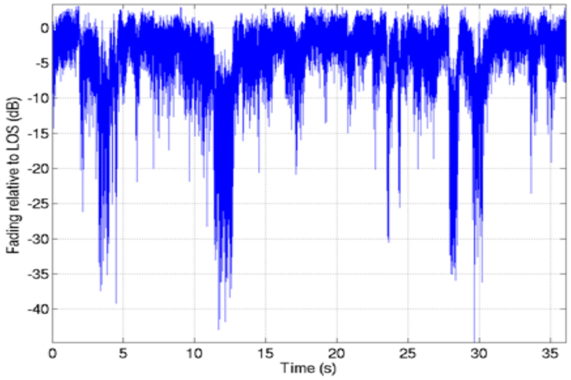


Figure 4. Excerpt from LMS channel for intermediate tree shadow environment

Then, inside each state (mid-scale), the channel fading is assumed to follow a Loo distribution (basically the distribution of a Rice process whose direct path amplitude is log-normally distributed) [6]. The input parameters for each state are the log-normal parameters of the direct path amplitude and the multipath power assumed to be constant. Another input parameter at this intermediate scale is the shadowing correlation length that influences the dynamics of the direct path amplitude change within one state.

At small scale, in our implementation, the fading dynamics are taken into account by using a Zheng-Xiao model (Classical Doppler spectrum). One should note that this model is narrow band. In other words, fading coefficients are varying only along the time axis. For simulations in this paper, Intermediate tree shadow environment is used, with vehicle moving at 50 km/h, seeing satellite with 40 degrees elevation. A channel excerpt is shown in Fig. 4.

IV. CHANNEL CAPACITY

When equiprobable assumption is made at the transmitter, the channel capacity can be assessed by computing mutual information between transmitted and received symbols, respectively X and Y, as it has been done in [8]. In addition, a computation of mutual information using LLR metrics is derived in [7]. This computation is performed via averaging measured mutual information over a sufficiently large number of samples constituting a sufficient statistic. Channel capacity C is therefore defined as:

$$C = I(X,Y) \tag{2}$$

Nowadays, BICM is adopted in most advanced telecommunication standards. Assuming bit interleaving with infinite depth, channel capacity can be written as the sum of *log₂(M)* independent channels, provided that mutual information computation is averaged over a long period.

$$C = \sum_{i=1}^{\log_2 M} E[i(b_i; I, Q)] \tag{3}$$

E is the expectation operator, b_i is the transmitted bit, I and Q are the in phase and quadrature received symbols, i is the mutual information operator, M is the number of states of the modulation. Mutual information between bit b_i and received I and Q components is defined by

$$i(b_i; I, Q) = log_2 \frac{1}{P(b_i)} + log_2 P(b_i \mid I, Q) \tag{4}$$

taking into account that LLR general definition is given by

$$LLR(b_i) = \ln \frac{P(b_i=1 \mid I,Q)}{P(b_i=0 \mid I,Q)} \tag{5}$$

And that :

$$P(b_i = 0 \mid I, Q) + P(b_i = 1 \mid I, Q) = 1 \tag{6}$$

we obtain:

$$P(b_i \mid I, Q) = \frac{1}{1+e^{(-1)^{b_i}.LLR(b_i)}} \tag{7}$$

As a result *i(b$_i$;I,Q)* becomes

$$i(b_i; I, Q) = \log_2 \frac{1}{0.5} + \log_2 \frac{1}{1+e^{(-1)^{b_i}.LLR(b_i)}} \tag{8}$$

Using (3), channel capacity C, in bits/s/Hz, is now equal to

$$C = \log_2 M * E\left[1 - \log\left(1 + e^{(-1)^{b_i}.LLR(b_i)}\right)\right] \tag{9}$$

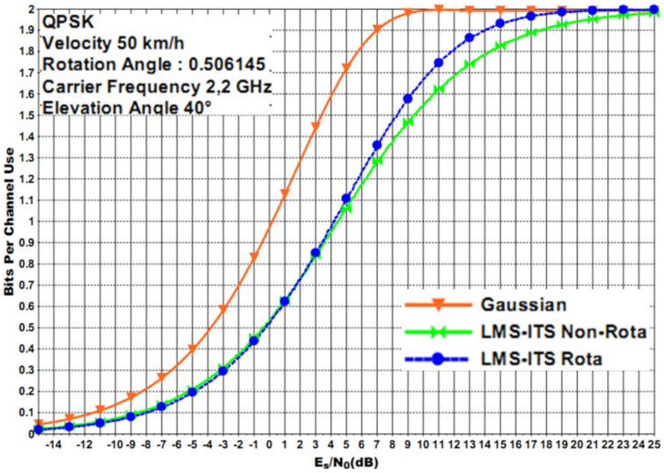


Figure 5. Channel capacity for LMS-Intermediate Tree Shadow Environment (LMS-ITS). 250 ms Time Interleaving

Results in Fig. 5 over LMS-ITS environment show mainly that for high spectral efficiencies (or high signal to noise ratios), rotation of the constellation can show a significant gain in required signal to noise ratio Es/N0, reaching in some cases more than 1.5 dB.

Interleaving depth and type does not affect obtained results despite computation via Monte Carlo averaging of measured mutual information. Indeed, averaging is performed over a large number of channel uses or transmissions far superior to interleaving depth. The expectation being between the transmitted bit and its LLR, the order of bit transmission does not alter the average value over the sequence.

V. CODED SIMULATIONS

A. *Parameters and simulation chain*

From channel capacity results, suitable working points where rotated constellations bring an important gain seem to be those with high spectral efficiencies. To verify this assertion, the DVB-T2-like transmission chain [1] shown in Fig. 6 was chosen considering ideal receiver as a first step. Error rates are computed at the output of the LDPC decoder. Indeed, the assumption made in the DVB-T2 specification that a frame error rate of 10^{-4} at the output of the LDPC decoder corresponds to a frame error rate of 10^{-7} at the output of the BCH decoder is also assumed. Simulation parameters are summarized in table II.

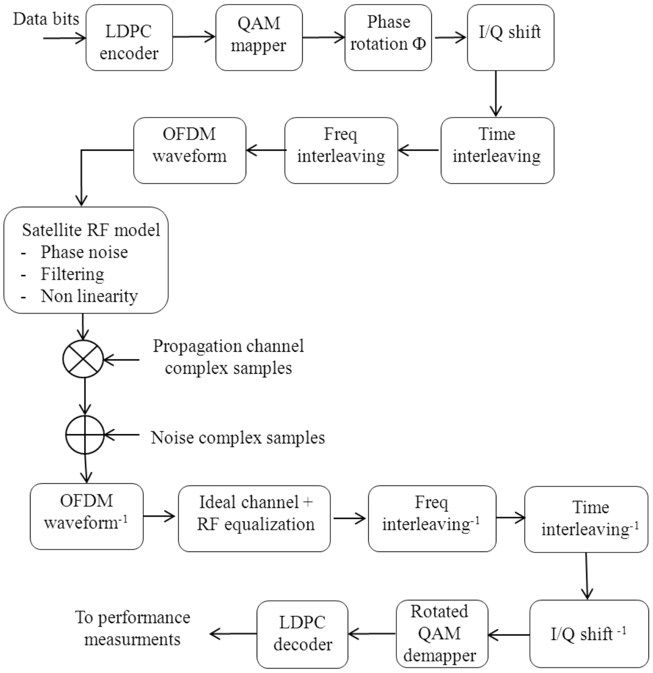


Figure 6. DVB-T2 like Transmission chain

Considered satellite radio frequency model includes frequency filtering, phase noise and amplifier non-linearity impairments. Applied phase noise and amplifier non linearity models are compliant to the definition provided in [9].

We would like to recall that in satellite microcosm technical conventions, Input Back Off (IBO) is defined as the power difference between input power giving maximum output power and current input power. Output Back Off (OBO) equals to the difference between maximum deliverable power in continuous wave signal and current output power of modulated signal.

TABLE II. MAIN PARAMETERS FOR PERFORMED SIMULATIONS

Parameter name	**Value**
Center frequency	2,2 GHz
Bandwidth	5 MHz
Constellation and rotation	QPSK, with Φ= 29.0 degrees rotation or without rotation
Coder	LDPC 16200 bits length. No BCH encoder
Propagation channel	LMS-ITS 3 states model, 50 km/h, 40° satellite elevation
OFDM	2K FFT size, 1/4 guard interval
Satellite RF model	with and without, variable Input Back Off for the amplifier (IBO)

TABLE III. summarizes considered IBO values and their respective OBO. These values condition obtained results and are of great importance when dimensioning a satellite link.

TABLE III. IBO VS OBO FOR CONSIDRED S-BAND AMPLIFIER

IBO (dB)	**OBO (dB)**
0	1.37
2	1.6
4	2.2

B. *System dimensioning and optimization*

From a satellite system optimization/dimensioning point of view, two metrics have usually to be considered : power loss, corresponding to OBO value, and signal quality loss, equal to the performance gap considering RF sub-block or not. Total loss metric is then defined as:

$$Total\ loss = signal\ quality\ loss + power\ loss \qquad (10)$$

A good choice would minimize Total loss value. But for different combinations giving nearly the same *Total loss* amount, preferred solution would be the minimization of signal quality loss, corresponding to a better carrier over intermodulation power ratio. In our case, total loss is around 1.85 dB obtained by the combination of 0.25 dB of signal quality loss and 1.6 dB of power loss.

C. *Performance Results*

Figure 7. and Figure 8. show FER simulation results at the output of the LDPC decoder for the considered DVB-T2-like chain with a code rate R = 11/15 and R = 2/3 respectively. Corresponding results confirm channel capacity predictions. Indeed at 10^{-4} of FER, rotated constellation brings a 1.3 dB gain over a LMS-ITS channel with R = 11/15 and 1.0 dB with R = 2/3 when no impairments are considered. Additional gains ranging from 0.15 to 0.5 dB are achieved when filtering, phase noise and amplifier non-linearity with 4.0 and 0.0 dB IBO are introduced. Therefore, rotated constellation seems to improve robustness to different types of impairments.

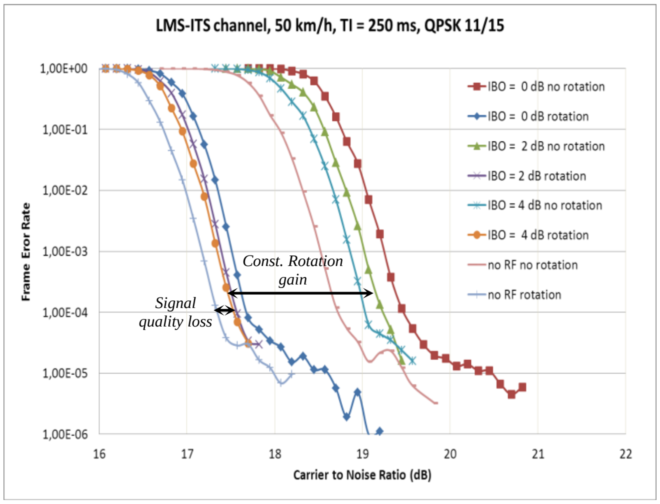


Figure 7. FER comparison results between a rotated and a classical QPSK for R=11/15 with and without impairments.

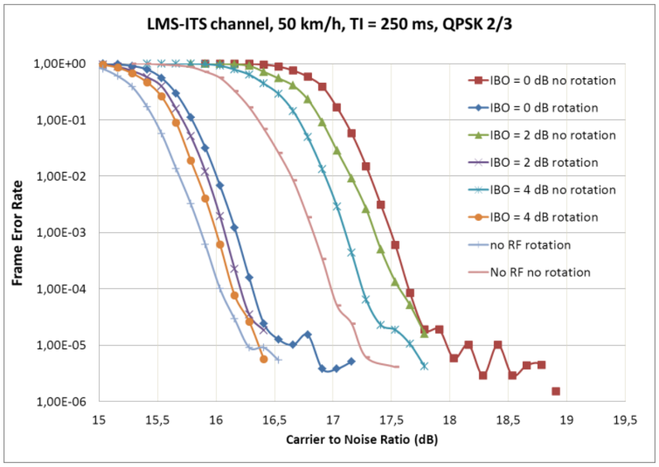


Figure 8. FER comparison results between a rotated and a classical QPSK for R=2/3 with and without impairments.

VI. CONCLUSION

In this paper, the study of rotated constellation was performed over a satellite-to-vehicle link in S-band. In addition, impairments of a satellite payload have been considered. Backed by channel capacity computations and coded simulation results, a substantial gain greater than 1.5 dB was pointed out for medium to high spectral efficiencies/coding rates. Moreover, larger gains were obtained when channel impairments were considered. Complexity overhead is negligible provided that interleaving blocks process I/Q cells as in the DVB-T2 standard. The use of this standard in a satellite context may appear in a satellite-terrestrial hybrid system, where for market considerations, there is a single chipset to receive terrestrial and satellite link. Through this paper, a way to make DVB-T2 waveform better optimized for satellite link has been shown. Lastly, whereas this paper was limited to QPSK modulation and DVB-T2 adopted rotation angle, other studies are ongoing considering interleaver design, rotation angle choice, and other QAM modulations over such satellite channels.

ACKNOWLEDGMENT

The first author thanks the Research and Technology decision board of French Space Agency allowing this kind of applicative research.

REFERENCES

[1] ETSI (European Telecommunications Standards Institute), “Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)”, ETSI, EN 302 755 V1.2.1, Sophia Antipolis, France, October 2010.

[2] ETSI, “Digital Video Broadcasting (DVB); Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)”, ETSI, TS 102 831 V1.1.1 , Sophia Antipolis, France, October 2010.

[3] C. Abdel Nour and C. Douillard, “Rotated QAM Constellations to Improve BICM Performance for DVB-T2”, 5th Symp. on Turbo Codes and Related Topics, Lausanne, Switzerland, Sept. 2008, pp. 55-60.

[4] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," IEEE Trans. on Infor. Theory, vol.44, no.3, May 1998, pp.927-946.

[5] J. Boutros and E. Viterbo, "Signal space diversity: a power- and bandwidth-efficient diversity technique for the Rayleigh fading channel," IEEE Trans. on Infor. Theory, vol.44, no.4, Jul 1998 , pp.1453-1467.

[6] F. Pérez-Fontán, M. Vázquez-Castro and C. Enjamio Cabado, “Statistical Modeling of the LMS Channel”, IEEE Trans. on veh. technol., vol.50, No. 6, Nov. 2001, pp. 1549-1567.

[7] J.Olmos, S. Ruiz, M. García-Lozano and D. Martin Sacristan “Link Abstraction Models Based on Mutual Information for LTE Downlink” European cooperation in the field of scientific and technical research, 2010.

[8] W. Chauvet, C. Amiot-Bazile and J. Lacan, “Prediction of performance of the DVB-SH system relying on Mutual Information”, in 11th signal processing for space communications workshop, Cagliari, Italy, Sept 2010, , pp. 413-420

[9] ETSI, “DVB-SH implementations Guidelines”, ETSI TS 102 584 V1.3.1, Nov 2011.

[10] IMT-advanced “Detailed specifications of the satellite radio interfaces of International Mobile Telecommunications-Advanced (IMT-advanced)”, document 4/40-E, revision 1, oct 2013.

Vehicular Ad Hoc Network Security and Privacy: A Second Look

Jesse Lacroix, Khalil El-Khatib
Faculty of Business and Information technology
University of Ontario Institute of Technology
Oshawa, Canada
Emails: {Jesse.Lacroix, Khalil.El-Khatib}@uoit.ca

***Abstract*—VANETs are an emerging infrastructure that makes use of vehicles as the main objects within a network. These networks use either peer-to-peer communications to communicate with other vehicle objects directly or a more centralized client/server approach to communicate with its road side infrastructures to either authenticate, send or receive information. With this added ability implemented into modern and upcoming vehicles, the transportation infrastructure would greatly improve in terms of efficiency, safety and user-friendliness. Although communication introduces better ways of traveling, adding a network infrastructure to vehicles and their environments also introduces the possibility of security breaches inside the vehicles and respective surroundings through internal and external components embedded in Vehicular Ad Hoc Networks. It has been shown that multiple attack surfaces exist and proper defence mechanism must be implemented to properly secure and deploy this type of network. This survey will present an overview of VANETs and synthesize related works to demonstrate new security mechanisms and how much this type of network and in-house components of vehicles are exposed.**

Keywords—Vehicular Ad Hoc Networks; Security; Vulnerabilities.

I. INTRODUCTION

Modern vehicles are now embedded with Electronic Control Units (ECUs) and On-Board Units (OBUs) to send and receive information to other vehicles or Road Side Units (RSUs). RSUs and vehicles are used to send critical information to other peers and to communicate to other parts and types of network infrastructures such as the Internet. RSUs are important in the operation of VANETs because they are used as relays to send information to all vehicles (for e.g., safety-related messages such as an accident occurring within a specific region and authentication messages for system validation). This type of communication is called Vehicle-to-Infrastructure (V2I) communication. Since these are not mobile, it is much easier to have them deliver the messages to affected cars because RSUs are deployed in such a way that vehicle objects can maintain a constant connection or have an indirect way of communicating with them. RSUs are not the only way vehicles can communicate inside the VANET; Vehicle-to-Vehicle (V2V) communications will also allow vehicle objects to communicate together and exchange information. Vehicle tracking, vehicle speed, Basic Safety Messages (BSMs) and other related information can all be exchanged between the vehicles themselves directly to ensure efficient and safe operation of the vehicles in their respective environments. What is important about VANETs is that they incorporate other means of communications to facilitate their operation. Examples of these as shown by Checkoway et al. [4] are: Bluetooth; broadcast channels, such as radio and GPS channels); addressable channel, such as OnStar [4]; and cellular channels, including 3G/4G LTE and basic voice channels for cellular communications. Combining all of these technologies together offers much more robustness to VANETs; however, on a security aspect, it does compromise security standards since more attack surfaces are introduced in the formula.

Each vehicle in VANET has a number of components that are used by vehicles for internal operations and data flow presented by Everett and McCoy [7]. The internal components work in conjunction with the OBUs so that proper information is transmitted from one vehicle to another. The components are:

- CANs (Control Area Networks) – used as backbone channels
- LINs (Local Interconnect Networks) – used for low speed and low bandwidth applications
- FlexRay – used for high speed and high bandwidth safety critical applications
- MOST (Media Oriented System Transport) – used for high speed and high bandwidth media applications
- TPMS (Tire Pressure Monitoring System) – used to monitor tire condition, precise pressure, etc.
- HSM (Hardware Security Module) – Stores and secures sensitive data (for e.g., private keys)

These components produce the overall infrastructure implemented inside vehicles to properly function and work directly with ECUs to perform proper operations. Compromising one of these components potentially leads to the full compromise of the vehicle; proper mechanisms must therefore be implemented for safeguarding.

The IEEE 1609 standard, shown by the IEEE Standards Association [10], known as the Wireless Access in Vehicular Environments (WAVE) is a service recognized by the Intelligent Transportation System (ITS). It is employed in the United States and similar infrastructures employed around the world for VANETs so that vehicles and respective infrastructure can communicate. This standard can also be associated to the Dedicated Short Range Communications (DSRC) protocol for radio spectrum allocation used by WAVE technologies. WAVE embodies many standards for its secure and efficient communications. They are as followed (this survey relates to the 1609.2 standard):

- IEEE Std 802.11 (2012)—Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for metropolitan and local networks as well as data exchange between systems
- IEEE Std 1609.2 (2013)—WAVE Security Services for applications and Management Messages; makes use of Elliptic Curve Cryptographic (ECC) as an encryption standard
- IEEE Std 1609.3 (2010)—WAVE Networking Services
- IEEE Std 1609.4 (2010)—WAVE Multi-Channel Operations
- IEEE Std 1609.11 (2010)—WAVE ITS over-the-air payment data exchange protocol
- IEEE Std 1609.12—WAVE Identifier Allocations

The European Telecommunications Standards Institute (ETSI) has developed its set of standards for VANET communication and information exchange for the ITS based off IEEE 802.11 technologies shown by Rizzo and Brookson [21]. ETSI ITS standards will take in consideration the IEEE 1609.2 data sets, but it will adopt them to fit explicit protocols developed for ETSI standards and will collaborate closely with the IEEE community. Here are some of the current ETSI ITS security standards:

- ETSI TS 102 867—ITS Security Service IEEE 1609.2 stage 3 mapping
- ETSI TS 102 940—ITS Security Service for communications security architecture and management
- ETSI TS 102 941—ITS Security Service for Trust and Privacy Management
- ETSI TS 102 942—ITS Security Service for Access Control
- ETSI TS 102 943—ITS Security Service for Confidentiality Services
- ETSI TS 103 097—ITS Security Service for headers and certificate formats

RSUs are also responsible for authenticating vehicles when one connects to the VANET network. At this moment, a centralize authentication scheme in combination with a Public Key Infrastructure (PKI) is the approach used to authenticate vehicles in the network with the use of trusted third parties (TPPs) as well as a central Certificate Authority (CA). This system would provide every vehicle with a valid certificate as long as they are part of the legitimate list of users located in the CA. This infrastructure makes use of what is known as a “legal authority” that binds certificates to the actual identities of drivers but is only accessible by the Central Authority. If a certificate is recovered, the driver’s personal details are not revealed and only linked to a given pseudonym. This authentication scheme is robust but faces computational issues when it comes to handling pseudonyms in an efficient and timely manner.

The contribution of this paper is to look at the current state of VANET security and privacy mechanisms and issues regarding all of its internal and external components so that a proper overview is provided to its audience. This paper can be used as reference to the challenges presented for future research when tackling presented problems and developing future platforms and systems. This survey will present and discuss the following topics to give a proper overview of Vehicular Ad Hoc Network security-related concepts and vulnerabilities at their current states of research. First, related works will be presented and split in subsections covering the main security concepts and new potential security mechanisms in each respective fields, including authentication and confidentiality, availability, non-repudiation, data trust and privacy. Proven vulnerabilities inside vehicles and on the network will then be discussed followed by the conclusion and future work.

II. SECURITY AND PRIVACY CHALLENGES IN VANET

The following section will present works and their research states with new security mechanisms in core concepts, such as authentication and confidentially, availability, non-repudiation, data trust and privacy.

A. *Authentication and Confidentiality*

Authentication plays a huge role in network security regardless of the infrastructure it is implemented in. Many methods and schemes exist to authenticate legitimate users to services, but a popular scheme employs the use of the public key infrastructure, as discussed by Fuentes, González-Tablas and Ribagorda [9]. This scheme employs a CA, TPPs and pseudonyms that form the Vehicular Public Key Infrastructure (VPKI). There are two proposals for the actual authentication process in the VPKI. The first method suggests that vehicle creates the pseudonyms and public keys themselves and send the information to RSUs so that they can be authenticated. The other method suggests that the HSM inside the vehicles takes care of authentication, since all stored information is already secured, and private keys are always contained inside the HSM; signed pseudonyms sent to the system are therefore much more secure. The main goal is to ensure that pseudonyms are changed at a regular interval so that enough confusion is caused within a network if someone is attempting to link acquired information to a specific car. Of course, other levels of addresses must be changed to limit how an attacker can link stolen information to cars such as the vehicle's IP (Internet Protocol) and MAC (Media Access Control) addresses. An added proposal was suggested by Adigun et al. [2] for helping the VPKI re-allocate and change certificates to ensure that captured information does not link electronic license plate information to private information about drivers. The first method suggested is pretty straightforward and requires a regional CA to issue a new pseudonym to vehicle objects the instant a time "*t*" threshold is reached. The second approach presented is basically the same as above except that the vehicle itself generates a new pseudonym after the threshold "*t*" is reached. The difference in both approaches in this scheme is that the use of speed and distance between a vehicle object and RSU is the defining factor in determining the threshold. The bandwidth of the environment is also a factor that is considered to calculate the time. It was assumed that RSUs were equally distributed in the environment (for this example and presented scheme). The reasoning of this scheme suggests that if speed, distance and bandwidth are sufficient with respective categories, changing pseudonyms could be done more often and faster without impacting the overall authentication process of the vehicle object inside the VANET. This would make the current authentication scheme more robust.

As much as authentication is essential, certificate revocation methods must be properly implemented to remove illegitimate users off VANETs. Legitimate infrastructure and vehicle objects can malfunction so false information must be flagged and certificates revoked. The issue with this scheme is the maintenance poses a challenge because revocation lists, also known as CRLs, can be huge and hard to update on traveling vehicles. They might also not have access to RSUs depending on the area (for e.g., rural areas might have unequally distributed road side infrastructure in contrast to urban areas that have RSUs well distributed for constant communication). Compressed Certificate Revocation Lists (RC2RLs) presented by Fuentes, González-Tablas and Ribagorda [9] and Salem, Abdel-Hamid and El-Nasr [23] are the proposed mean for fixing this issue through the use of filters and subsets to speed up revocation lookup and updating revocation lists. This compressed method allows broadcast channels such as Radio Data Systems (RDS) to transmit the information through FM waves. In any case, most vehicles, regardless of their location, have access to radio waves so they can have their certificate revoked if malicious/accidental malfunctions occur. An extended method has also been presented by Zhang et al. [29] in hopes of making certificate revocation more efficient. It introduces a new algorithm that makes uses of a concept called "k-Means" clustering that basically uses nodes and centric points to create groups of vehicles to spread CRLs so that all vehicles are checked against these revocation lists. This scheme also adds two new fields to CRLs, and these are composed of an "Issued Data" and "Credibility" field. The credibility field receives an assigned value of 0 to 100 (0 means non-trustworthy and 100 means credible source) that will be based off how the source is perceived by the vehicular network. It uses correct and historical behaviour as factors in determining if the source is faulty, malicious or credible. The issued date field helps determine how long a certificate has been issued to determine if it should still be valid or not in regards to credibility.

Work shown by Whyte et al. [25] demonstrate the current leading authentication design for V2V communications. This scheme uses a PKI architecture for bootstrapping, provisioning of pseudonym certificates, reporting misbehaviour and certificate revocation. The designed Security Credential Management System (SCMS) used by this design would also allow for safe, reliable and private means of communicating and exchanging BSMs; it is similar in design to its European counter-part V2X PKI infrastructure. The SCMS mechanism would prevent privacy exposures from SCMS insiders and outsiders as well as mitigating false warning. These attacks are prevented with CRLs, constant changing of certificates and dividing operations with organizational separation employed by the SCMS. These work in conjunction with multiple authority figures to ensure efficient operation of the suggested scheme.

Xiong et al. [26] present the use of group signatures to achieve confidentiality and authenticity. In this work, it is shown that vehicles sending messages anonymize themselves for authentication among their respective groups and can only be identified by a trusted authority while doing this efficiently and secretly. To achieve this type of authentication, all vehicles' OBUs within a group load public parameters from another group entity called the Member Manager (MM) to generate a private key. All private keys are assumed to be stored into their respective vehicle's safe tamper-proof devices. This allowed the concept of "signcryption". This concept makes it possible for a group to receive a message from the original sending one and have the sending one's MM verify the true identity of the sender if dispute arises from the receiving group since it can only identify which group a message originates from. This allows confidentiality requirements to be achieved as well as conditional privacy/anonymity while maintaining efficiency for proper operation. The MM, by then, has all OBUs within the group registered and reveals identities of vehicles when required.

There can also be a dynamic approach to the PKI architecture elaborated by Salem, Abdel-Hamid and El-Nasr [23], which works with vehicles requesting dynamically for keys as they pass RSUs that are retrieved from CAs. The scheme shown in this work helps mitigate non-repudiation attacks, masquerade, man-in-the-middle attacks, Sybil attacks and replay attacks through the use of unique identifiers, nonces and information known between the source vehicle and certificate authority.

Public/private key infrastructures have shown to be effective in traditional networks that do not have much mobility. VPKI schemes show promise for authenticating vehicle objects efficiently with different options, but the amount of calculations and computation required to do this can be time consuming and compromise performance depending on network load and specific scheme used. Sulaiman et al. [24] present a different approach in authentication schemes to validate users on VANETs. They introduce methods that use one-way hashing chain methods or also known as Hashed-Chain based Authentication Protocol (HAP). The method employed by this type of authentication works in the following way:

- RSUs use HAP to generate public/private keys
- These keys are then distributed to newly introduced vehicle objects on the network
- These keys are also paired with a variable sized hash value and proof cipher
- Synchronized clocks employed by this scheme allows vehicles to verify each other using a combination of their respective public with the variable sized hash code

With this mentioned technique, computations are reduced, and authentication has less overhead, which are desirable characteristics in a mobile network. This protocol shows efficiency in regards to computation and processing when vehicle objects would authenticate. Although the PKI method shows more security, HAP does employ constant key changes to ensure randomization so that an attacker would have difficulty compromising key sets for attacks on the VANET. Since the traditional public key infrastructure offers more efficient security standards, HAP could potentially be deployed side-by-side to ECDSA (Elliptic Cryptography Digital Signature Algorithms) methods when delays in the network are inevitable and faster processing is required by the system, similar to a backup to the VPKI. The results shown by Sulaiman et al. [24] are promising and demonstrate that HAP has the potential to become a new method of authentication in VANETs; however, it will unlikely fully replace the already robust PKI architecture.

B. *Availability*

VANETs are mobile networks that require some data (for e.g., BSMs) to be sent and delivered to them in real time since some crucial decisions need to be done by end users. VANETs must be fully operational with barely any downtime if drivers are to become more dependent on them in the case of having the roadside infrastructure evolve and become more efficient. If any network entity was taken offline due to malfunction or lack of processing power, the vehicle objects would be directly affected since RSUs provide important information to vehicles. In a nutshell, availability is firstly about designing a network that is capable of handling the intended and predicted network load, properly processing it and remaining scalable while migrating any type of interference and malicious Denial-of-Service (DoS) attacks, distributed or not. The issue with VANETs is that they deploy prevalent wireless technologies and are therefore more susceptible to DoS attacks since wireless technologies are easier to access and exploit.

The issue with availability is that methods to mitigate these types of attack are harder to implement than it is to find new ways of attacking a network and attempting to block legitimate services, if not all. Kang, Lee and Gligor [11] present a new and ground-breaking type of Denial-of-Service attack that does not take in consideration the physical server it wants to bring down (or service in any case) but indirectly attacks it by forming a target area around it. This attack is called “The Crossfire Attack”. It mainly disconnects services and servers by attacking key links in the infrastructure with the use of layer 3 mapping. This work was presented in a non-VANET environment, but these concepts can be applied the same way since Road Side Units and its infrastructure are not mobile like vehicles. Here is a summary how such an attack works:

1. Select the target area where the desired servers/services are located
2. Select the links to attack (after doing a layer 3 mapping of the target area)
3. Select and attack decoy servers so traffic is directed to the target area and redirect flows from decoy server to the targeted links
4. Attacker then has the rest of the botnet target disjoint target links so that the targeted services/servers lose Internet connection

The employed technique clearly demonstrates how effective it can be in bringing down networks. The most efficient part of this attack is that only decoy servers and services are targeted, which makes it more difficult to pinpoint the source of origin of the attack. These decoy servers could so well spread out that the traffic to them might seem legitimate, but would instead be a target of this indirect Distributed Denial-of-Service attack. The evolution of attacks demonstrate that proper counter-measures must be taken if they are all to be mitigated, especially if attacks can be carried over to different types of infrastructures such as VANETs.

As mentioned above, Denial-of-Service attacks are hard to mitigate but not impossible. Methods do exist in attempts of stopping or rendering them less efficient so that network operations maintain tolerable flow and functionality with minimum requirements. Although the method presented by Abumansoor and Boukerche [1] is for unintended DoS attacks caused by high level congestion of vehicles, the presented concepts can be used to mitigate intended attacks. One way an attacker can deny service to an area or specific victim would be through the use of sending excessive amount of information and probes to the victim(s) and/or surroundings. The amount of cars in the area that send probes for useful information such as localization data and reply to other probes would obviously slow down the network infrastructure if there was an over-saturated amount of information being sent and received from any source, including malicious ones. The method used to counter such means would make use of vehicles using their sensors and acquired neighbour information to determine area congestion. Probing rates would then be adjusted so that network load heavily reduces. This method is entitled Adaptive Group Beaconing shown by Abumansoor and Boukerche [1]. Authority management nodes in charge of monitoring network activity and sending periodic messages can monitor the amount of probing and adjust its periodic message notifications and control Quality of Service (QoS) so that certain applications inside vehicles continue to operate properly regardless of the probing rate in the network. This type of adaptive probing behaviour can positively affect Denial-of-Service mitigation even if it is based off unintentional DoS attacks caused by a congested network.

Since general Denial-of-Service attacks tend to leave heavy traces of network traffic to successfully disable services, existing mechanism can be implemented with Vehicular Ad Hoc Network infrastructures to successfully detect abnormalities inside it. Intrusion Detection Systems (IDSs) make the use of signatures to detect and report attacks and malicious intent to system administrators or automated security service so that action can be taken. Even better would be Intrusion Prevention Systems (IPSs) since they can not only detect malicious attacks but stop them at the same time. The work discussed by Coussement, Bensaber and Biskri [5] demonstrates that these systems can be implemented into the vehicles themselves or into RSUs to ensure safety. There is only one issue with this approach and that is that these systems do not have a control mechanism with VANETs; this would need to be implemented. The employed mechanism presented by this paper would use a probabilistic scheme to determine incoming attacks. Normal behaviour of vehicles and known responses to vulnerabilities would be traced and recorded (in our case, high traffic load and congestion can be added in this probabilistic model to determine whether a DoS attack is occurring). Vehicles, when analyzed, would also be grouped into clusters to get more generalized predictions and behaviour in helping to determine if a vulnerability is exposed or services are being disrupted to targets in the same vicinity. With the use of this protocol/mechanism, IDSs/IPSs could potentially be implemented within VANETs to add more layers of security and rendering it safer. The true way to mitigate Denial-of-Service attacks is to ensure that all traffic is authenticated and that only legitimate users can send information. The use of message capacity mechanisms associated to each vehicle and network object can also be used with previous methods in attempts of mitigating the damages caused by DoS attacks, if not eliminating the threat altogether. The implementation of existing DoS mitigation methods must be considered to help eliminate them in VANETs since availability is crucial.

C. *Non-Repudiation*

Non-repudiation of origin consists of having vehicles acknowledge that they have sent messages to wherever it is destined. The use of digital signatures is employed to sign all messages that are being sent. The main encryption method employed by VANETs, which is mainly used by the IEEE 1609.2 standard because of high performance and complex cryptographic scheme, is ECC as shown by Fuentes, González-Tablas and Ribagorda [9] and the IEEE Standards Association [10]. This encryption type combined with non-repudiation of origin make it a strong method for sending and verifying messages' signatures. Signature checking isn't the only required step in signature verification: the certificate of the sender must also be checked to ensure that it is valid and isn't part of CRLs. A group signature method has also been suggested to add privacy to the non-repudiation of origin process. Clusters would only send one digital signature to destinations that represents the group that sent it. TPPs are the only entities that would have access in determining individual objects within the source cluster. Non-repudiation of origin allows every vehicle object to be held accountable for all action it performs on the network so this helps identify attackers when attempting to send bogus and/or harmful information, which is flagged by vehicle objects and road side infrastructure. Attackers would need to find a way to retrieve digital signatures or to duplicate another legitimate user's signature so that he/she can impersonate an unsuspecting victim and get away with the attack. Work developed in this area working in conjunction with security standards in all other security related concepts will help identify the sources of attacks so that mitigation is done much more easily and have the hackers held accountable. Strong authentication message as well as credential managements methods must be implemented in the authentication schemes to circumvent this type of vulnerability.

D. *Data-Trust*

Elliptic Curves Cryptography is the main encryption scheme used by the DSRC protocol for encryption and has proven to be efficient in terms of overall security and computation. Hash-based authentication presented by Sulaiman, Raja and Park [24] shows promise for speed and processing but lacks security compared to proven ECDSAs and other methodologies, such as those used by the 1609.2 standard, for example. ECC, being standardized, makes it hard for attackers to exploit, so data will remain unaltered and trustworthy. Methods shown by Fuentes, González-Tablas, and Ribagorda [9], such as two-direction reporting, threshold-based trust and the use of group signatures, which all incorporate static and dynamic factors, are some methods that can be used to ensure this aspect of security in VANETs.

To ensure that data trust is properly implemented as mentioned above, the correct approach would be to have a framework of trust implemented. Such a framework would greatly improve the performance of trust determination and credibility checks of data being received by vehicles. The work presented by Rostamzadeh et al. [22] proposes the implementation of such a framework called FACT. Successfully implementing this would greatly reduce delays in network communications and maximize performance since trust verification would greatly be minimized. This paper proposes that network segmentation should be implemented based on individual roads, neighbourhoods and road segments, to name a few. Then, depending on known reputation of the area and risk factors, these segmentations would be assigned a trust factor that reflects upon messages a vehicle sends when in a specific area. FACT would then classify all traffic into three categories ranking their overall importance. They are as follows:

- Category A – Holds all the critical infrastructure messages
- Category B – Holds all road side service information
- Category C – Holds all third party service messages

All of these categories employ QoS priorities to ensure that the respective messages are sent accordingly to their respective destinations in regards to their level of trust assigned. For category "A" messages, delay, data integrity and reliability are the key security concepts that must be considered when delivering these messages since the delay with critical information can directly affect a driver's ultimate reaction and decision in the given framework. Category "B" messages are more concerned with reliability, access control, source anonymity and authentication to ensure the source is legitimate as well as access to the information is available and accurate. Source authentication and reliability are the main security concepts applied to category "C" messages since the third parties must be legitimate and allow message to be delivered efficiently in large numbers. The mechanisms being developed in this area of VANET will directly impact a user's reaction and decision making process so data trust must be kept under constant check. This field shows that is on the right direction. This framework will further strengthen authentication schemes and underlying data trust protocols so that data being sent and received by legitimate users is trustworthy, authentic and authorized, especially if data trust protocols are supported by robust standardized protocols that employ ECC or other ECDSA methodologies.

E. *Privacy*

Users are legally entitled to know how their provided information will be used, stored and secured when agreeing to use a provided Internet service. Law and regulations have been put in place by some national/regional governments, but these laws vary and are applied in different ways based off the user's location. Some organizations are enforced to communicate how they will protect the sensitive data and how they are implementing these procedures. The work elaborated by Kosa, Marsh and El-Khatib [12] proposes a framework that will be used to calculate the privacy states that would be automated and used for representing privacy concerns and states in VANETs, which are shown in the figure below. This would then lead to determine how data is collected and handled by its respective regulators across the system. The framework developed by Kosa, Marsh and El-Khatib [12] uses Canadian standards and laws as an example but can be extended to fit other country laws for adaptation into their transportation system's VANET.

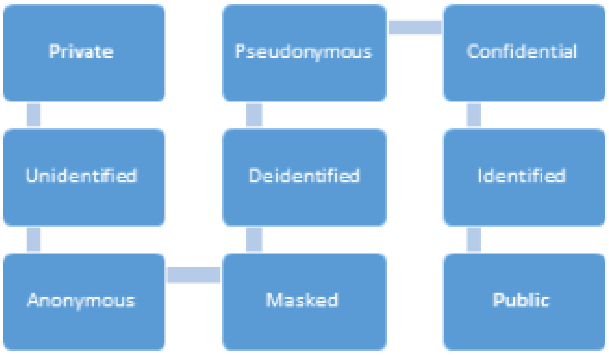


Figure 1. Different privacy states [12]

In Canada, there are multiple legal documents that regulate privacy concerns, what organizations need to do to protect this data, and how to communicate this to the users so they are made aware. The documents are as follows:

- PIPEDA (Personal Information Protection and Electronic Document Act) – covers non-profit organization and the private sector [17]
- Privacy Act – Information collected by the federal Government [19]
- MFIPPA (Municipal Freedom of Information and Protecting Privacy Act) – covers municipal organizations [15]
- FIPPA (Freedom of Information and Protection of Privacy Act) – covers provincial organizations [8]
- PHIPA (Personal Health Information and Protection Act) – covers all health organizations [18]

These acts are used to protect information collected by each respective organization according to their specific criteria that must be followed and, for this specific case, be used in Canada for privacy representation in VANETs. Any other country would adopt their laws and regulation documents with this framework to determine its privacy representation. Five privacy requirements must be respected when organizations collect information in this framework, shown by Kosa, Marsh and El-Khatib [12]. They are as follows:

- Privacy Regulation
- Inter-Jurisdiction

- Model Complexity
- Interoperability
- Access Regulation

The framework would employ a Finite-State-Machine (FSM) to determine what the result of a given communication between two entities would be in when an information transaction occurs. The given framework would help identify what to expect in terms of what type of communications are occurring so that users are made aware of how information is being handled. It basically gives consent on what users are expected to disclose and what users are expecting in regards to what sensitive information is being protected when collected. Privacy in VANETs must have the correct guidelines applied to them and must rely on current technologies and mechanism to ensure that private data is kept safe in accordance to their respective state and ruling government. Any country could adopt this framework to its respective laws and regulations for determining privacy representation for VANETs. Overall, such a framework helps determine privacy evaluation and decision making for end users regarding when and how data is collected/stored by respective governments, and how it would be handled.

III. PROVEN VULNERABILITIES

VANET security shows that there are many mechanisms being developed to ensure that all security concepts are enforced and maintain a standard of efficiency for the operation of vehicular networks. The mechanisms being developed do have specific reasons and are made to fend off many types of attacks that are present and could potentially target a VANET. Work shown by Rawat, Sharma and Sushil [20] demonstrates these types of attacks. Here is a list of network attacks that affect network communications when it comes to V2V and V2I communications:

- DoS attacks – as mentioned above, these attacks can target any specific object within the network in hopes of disrupting network service and functionality so that all operations are delayed or rendered useless with the use of excessive traffic and/or over-utilizing key resources in the infrastructure
- Sybil attacks – also elaborated by Yu, Xu and Xiao [28], Sybil attacks are done when a malicious users impersonates multiple identities hoping to fool legitimate users in taking different routes due to traffic congestion protocols
- Message suppression/alteration/falsification – a malicious user manages to drop legitimate traffic in the network in attempts of falsifying road conditions. Alteration is when legitimate messages are altered to fool legitimate users with incorrect data. Falsification is when an attacker broadcasts false information to influence the traffic to his liking or cause havoc
- Replay attack – legitimate messages are captures and used later in legitimate circumstances for illegitimate means
- GPS spoofing – an attacker falsifies GPS information to fool other vehicles into thinking they are at a different location with his/her own GPS simulator
- Tunneling attack – two physically separated parts of a VANET are connected through a tunnel thinking they are neighbours so an attacker could analyze the traffic of perform selective forwarding attacks
- Timing attack – time slots are altered so that safety critical message are delayed and received after their useful lifetime is outlived
- Man-in-the-middle attack – the attacker is between a legitimate communication session and intercepts traffic to see the content but forwards it to the right end destinations to remain invisible
- Home attacks – malicious user attempts to take control of the vehicle’s internal components with the use of the Internet
- ID disclosure – a target’s location is disclosed and made publicly available so that anyone can view the location of the vehicle
- Brute-force attack – an illegitimate user attempts to break cryptographic keys used in secure communication sessions

All these types of attacks have the potential of affecting VANETs and end users. That is why security standards are being developed so that all fronts are reinforced and that these attacks greatly reduced, if not rendered completely ineffective. These attacks, if well-coordinated, could also lead to the compromise of internal components if vehicles are lured to specific locations which allow an organized attacker to perform more sophisticated types of attacks.

A different attack has been introduced that basically fully compromises a node in the Vehicular Ad Hoc Network. The work presented by Lin et al. [13] demonstrates that a malicious user attempts to physically capture nodes inside the network. Once physical access is acquired, the adversary implements malware as well as attempts to reveal secret keys so that all communications with compromised nodes are known and traffic is exposed. Privacy concerns also arise as location could then be disclosed, not to mention other attacks could be launched including Denial-of-Service, spamming, Sybil attacks, etc. A compromised node could then affect further nodes attached to it so it can spread into the network and increase its potential regional reach, if not global up until the entire network is compromised. The only downfall to this attack is that physical access is required so some parts of the infrastructure are not reachable (e.g., highway RSUs) and/or publicly exposed; however, if managed correctly and not caught in the act, RSUs that are not easily physically accessible could fall prey to one that is and compromised. This type of attack is dangerous as it enables a platform to launch all mentioned above attack through a seemingly legitimate node of the infrastructure. Overall, many methods are available for attacking a network. Many methods and mechanisms must therefore be deployed and further researched to ensure end user security.

Vehicular Ad Hoc Networks show definite promise in the functionality it is intended to provide. The ability to send information about road conditions, accidents, congestion warnings from indirect neighbours, to name a few, is useful as discussed by Younes and Boukerche [27]. The wireless technology employed to do this is quite efficient in enabling the operations of VANETs, but like any other infrastructure, specifically wireless oriented ones, vulnerabilities to attack the network arise. Rawat, Sharma and Sushil [20] present home attacks that are directed towards taking control of vehicles using, and not limited to, the Internet, so that internal vehicle components are exploited and taken over. Works shown in [4][6][11][14][16] present multiple attack surfaces that are exposed through external components and allow compromise of the internal network components of the vehicle objects. There are many attack vectors that are of the following:

- OBD II port – direct physical access to internal components of the vehicle
- “PassThru” device – Device that connects to OBD II port for analysis of system buses and firmware updates
- Media devices (e.g., MP3s, USBs, CDs, etc.) – direct physical access to internal components of the vehicle
- Bluetooth – short range communications access to internal components of the vehicle
- Cellular – long range communications access to internal components of the vehicle
- Broadcast Channels – long range communications access to internal components of the vehicle

The work presented by Checkoway et al. [4] explains how full vehicle compromise (for e.g., vehicle acceleration and braking, to name a few) was attained and all possible ways they have managed to successfully do it. Figure 2 shows all the multiple attack surfaces. Vehicle objects have shown vulnerability from direct physical access to the vehicle’s OBD II port. If an attacker manages to get access to this port when the driver is not present, he can listen in on the internal network components and debug the communication in attempt of reverse-engineering the internal protocols. The user can use packets that he/she crafts, based off the debug output, to make the vehicle do as he/she pleases. This is the most efficient way of compromising a vehicle, but physical access to the port is hard and is noticeable by users since the car has to be broken into. “PassThru” devices, which are used by vehicle manufacturers, authorized dealerships and mechanic shops, are used to update and gain access to a vehicle’s internal network components (CANs, LINs, FlexRay, etc.). Once this device is connected, they can update and maintain the firmware, which would be periodically done when a vehicle comes in for maintenance schedules and safety checks. These devices can also use wireless communications and allow an untrusted third party to connect to it. When the device connects to the OBD II port, the attacker could gain access to the internal components of the vehicle shown by Checkoway et al. [4]. No authentication checks are done by the internal components when a PassThru device connects to it, meaning anyone connected to the PassThru gains automatic access to the OBD II port. Authentication means would need to be implemented to stop this from happening. An attacker could also upload malicious packages to the PassThru device so that whenever it connects to a vehicle, the files are uploaded to the vehicle to compromise the internal network. This method would allow multiple unsuspecting vehicle objects to be infiltrated. Media devices such as CDs and USB devices can also be used to upload malicious information to vehicles if inserted in the proper access channels. It has been show by Checkoway et al. [4] that if CDs contain malformed audio files, they can update the firmware inside the vehicle through a buffer overflow attack.

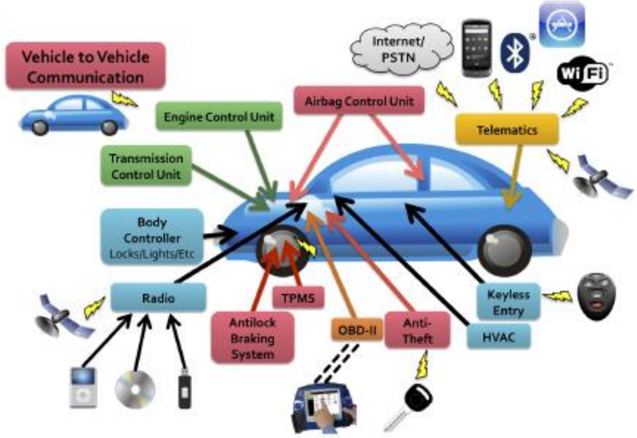


Figure 2. Attacks surfaces in VANETs [4]

Bluetooth has also shown to be a vulnerability inside vehicles. This work demonstrates that through device and car pairing, the vehicle can be compromised. With the use of “Bluesniff”, which is used to sniff and capture Bluetooth MAC addresses, brute-forcing methods can be done to pair to the vehicle. Approximately 9 PINs per minute can be brute forced to pair to a vehicle according to trials successfully made by Checkoway et al. [4]. Although this does not sound like a lot of attempts in the given time frame, this technique could be done in a public garage where thousands of vehicles may be present, brute-forcing one within seconds. This is plausible as tests presented by Checkoway et al. [4], demonstrate that a single vehicle was compromised within 15 minutes. This time significantly reduces when there's more than one vehicle, and, once the device of the attacker is paired to the vehicle, custom applications can be used to gain access to the vehicle's network.

Just like Bluetooth, long range communication means can also be used to exploit buffer overflow vulnerabilities inside the vehicle to fully compromise it. Cellular communication can be used to breach a vehicle's security, which demonstrate how volatile and dangerous wireless communications can be when exploited. AqLink, a protocol used to send and receive voice communication on cellular channels, has been reverse-engineered by Checkoway et al. [4]. This protocol changes analog bits to digital ones so that they can be interpreted by the internal systems. This opens up the possibility of using audio playback to trigger an exploit that was successfully done in the presented research. They were able to phone a remote vehicle that is within cellular range and play an audio file through an audio device, and it compromised the vehicle through a buffer overflow exploit. The issue with this type of attack is that the transmission speed is limited and can only send data at a certain limited rate; a certain amount of data must therefore be delivered before a timeout occurs to trigger the attack. Well-crafted and short code must be done to successfully exploit the vehicle object. Other mediums such as 3G or addressable channels, such as OnStar, allow for faster delivery mechanism with a much bigger payload, but the vehicle must be within range of 3G transmitters to be contacted. Work discussed by Cai et al. [3] shows that Bluetooth can further be exploited with the use of antennas to boost signals and coupling with devices that have more than one antenna (in this case vehicles). The vehicles do not need to be in line of sight, and with the use of multiple antennas, the vehicle object and Bluetooth device can be paired, making it much harder for an attacker of being detected since visual cues are not available. Many attack surfaces exist in vehicles for targeting internal components of vehicles that interact with the vehicular network of this infrastructure, if not targeting the external components of VANETs to launch attacks through them, and many different types of attacks exist. Security standards must be kept under constant revision to ensure that all components are secured and cannot be easily exploited, if not impossible, since security in VANETs is extremely important to ensure end user safety and well-being.

IV. Conclusion AND future work

A. Conclusion

There is a little doubt that VANETs offer great potential in the advancement of vehicles and the development of the ITS. The functionality of this network architecture does come at a price since it requires high efficiency with no room for security flaws. Current technologies in authentication, localization, information access and so forth show promise, but better mechanisms must be implemented to ensure that all standards are met. Current research works have shown present alternate solutions with promising potential that will possibly be implemented in future instances of VANETs as its development cycle extends and nears completion. Plenty of vulnerabilities have also been discovered and reported to ensure that none of them are present when VANETs are publicly available to the masses. These vulnerabilities demonstrate to what extent a VANET can be exploited, even to the point of full car control that is unacceptable considering the damage it could cause to the end users. As much as the efficiency of the transportation system would increase if the deployment of VANET was sooner than later, extended research in its security related aspects must be done before being fully implemented.

B. Future Work

Future work from the VANET research community must put emphasis not only on developing security mechanisms to counter all potential vulnerabilities, but also on platforms that could be used by researchers to perform further testing on the internal and external components of VANETs. These types of networks are not as readily available as the more standard Internet architecture platforms so it is important that research goes into developing ways for researchers to be able to directly test potential solutions to VANET issues, security related or not. Work must also be put into testing all these proposed solutions unto larger and scalable models to ensure that the mechanisms work as predicted. It also goes to show that future research must be done to test out all the possible security angles of VANETs since room for vulnerabilities cannot be tolerated. The extent of such work would help ensure the protection of VANET users, which is paramount in an architecture that is directly tied to the transportation system.

Acknowledgment

This work is partially sponsored by NSERC DIVA Research Network grant.

References

[1] O. Abumansoor and A. Boukerche, "Preventing a DoS threat in Vehicular Ad Hoc Networks using Adaptive Group Beaconing." Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 233-240.

[2] A. Adigun B. A. Bensaber, and I. Biskri, "Protocol of change pseudonyms for VANETs." Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 150-155.

[3] L. Cai, K. Zeng, H. Chen, and P. Mohapatra, "Good neighbor: ad hoc pairing of nearby wireless devices by multiple antennas." in NDSS, 2011.

[4] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces." in USENIX Security Symposium, 2011.

[5] R. Coussement, B. A. Bensaber, and I. Biskri, "Decision support protocol for intrusion detection in VANETs." Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 41-47.

[6] L. Ertaul and S. Mullapudi, “The security problems of Vehicular Ad Hoc Networks (VANETs) and proposed solutions in securing their operations.” in ICWN, 2009, pp. 3–9.

[7] C. E. Everett and D. McCoy, “OCTANE: Open Car Testbed And Network Experiments bringing cyber-physical security research to researchers and students.” in Cyber Security Exper. and Test, 2013.

[8] Freedom of Information and Protection of Privacy Act of Canada (2011). [Online]. Available: http://www.e-laws.gov.on.ca/html/statutes/english/elaws_statutes_90f31_e.htm [accessed: May, 2014].

[9] J. M. de Fuentes, A. I. González-Tablas, and A. Ribagorda, “Overview of security issues in Vehicular Ad hoc Networks.” *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*, 2010.

[10] IEEE Standards Association, “IEEE guide for wireless Access in vehicular environments (WAVE) architecture,” 2013.

[11] M. S. Kang, S. B Lee, and V. D. Gligor, “The crossfire attack.,” in *IEEE Symposium on Security and Privacy*, 2013.

[12] T. A. Kosa, S. Marsh, and K. El-Khatib, “Privacy representation in VANET.” Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 48-52.

[13] C. Lin, G. Wu, F. Xia, and L. Yao, “Enhancing efficiency of node compromise attacks in Vehicular Ad hoc Networks using connected dominating set,” *Mobile Networks and Applications*, vol. 18, no. 6, pp. 908–922, Dec. 2013.

[14] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, “Security in Vehicular Ad Hoc Networks.” Communications Magazine, IEEE, vol. 46, no. 4, pp. 88–95, 2008.

[15] Municipilaty Freedom of Information and Protection of Privacy Act of Canada (2007). [Online]. Available: http://www.e-laws.gov.on.ca/html/statutes/english/elaws_statutes_90m56_e.htm [accessed: May, 2014].

[16] B. Parno and A. Perrig, “Challenges in securing vehicular networks.” in Workshop on hot topics in networks (HotNets-IV), 2005, pp. 1–6.

[17] Personal Information Protection and Electronic Document Act of Canada, (2011). [Online]. Available: http://laws-lois.justice.gc.ca/eng/acts/P-8.6/page-1.html [accessed: May, 2014].

[18] Personal Health Information Protection Act of Canada (2010). [Online]. Available: http://www.e-laws.gov.on.ca/html/statutes/english/elaws_statutes_04p03_e.htm [accessed: May, 2014].

[19] Privacy Act of Canada (2014). [Online]. Available: http://laws-lois.justice.gc.ca/eng/acts/P-21/ [accessed: May, 2014].

[20] A. Rawat, S. Sharma, and R. Sushil, “VANET: Security attacks and its possible solutions.,” *Journal of Information & Operations Management*, vol. 3, no. 1, 2012.

[21] C. Rizzo and C. Brookson, “ETSI white paper No. 1 security for ICT – the Work of ETSI.,” ETSI, 2014.

[22] K. Rostamzadeh, H. Nicanfar, S. Gopalakrishnan, and V. C. M. Leung, “A context-aware trust-based communication framework for VNets.” Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 156-161.

[23] V A. H. Salem, A. Abdel-Hamid, and M. A. El-Nasr, “The case for dynamic key distribution for PKI-based VANETS,” *International journal of Computer Networks & Communications*, vol. 6, no. 1, pp. 61–78, Jan. 2014.

[24] A. Sulaiman, S. V. Kasmir Raja, and S. H. Park, “Improving scalability in vehicular communication using one-way hash chain method,” Ad Hoc Networks, vol. 11, no. 8, pp. 2526–2540, Nov. 2013.

[25] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, “A security credential management system for V2V communications,” in *Vehicular Networking Conference (VNC), 2013 IEEE*, 2013, pp. 1–8.

[26] H. Xiong, G. Zhu, Z. Chen, and F. Li, “Efficient communication scheme with confidentiality and privacy for vehicular networks,” *Computers & Electrical Engineering*, vol. 39, no. 6, pp. 1717–1725, Aug. 2013.

[27] M. B. Younes and A. Boukerche, “Efficient traffic congestion detection protocol for next generation VANETs.” Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 208-212.

[28] B. Yu, C.-Z. Xu, and B. Xiao, “Detecting Sybil attacks in VANETs,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 746–756, Jun. 2013.

[29] Q. Zhang, M. Almulla, Y. Ren, and A. Boukerche, “An efficient certificate revocation validation scheme with k-means clustering for Vehicular Ad Hoc Networks.” Proc. 3rd Annual DIVA Workshop, NSERC DIVA workshop, 2013, pp. 249-254.

Nonlinear Error Modeling of Reduced GPS/INS Vehicular Tracking Systems Using Fast Orthogonal Search

Mohamed Maher Atia, Aboelmagd Noureldin
Department of Electrical & Computer Engineering
Royal Military College of Canada
Kingston, ON, Canada
Mohamed.maher.atia@gmail.com;
Aboelmagd.Noureldin@rmc.ca

Michael Korenberg
Department of Electrical & Computer Engineering
Queen's University
Kingston, ON, Canada
Korenber@queensu.ca

***Abstract*— Land Vehicle Tracking systems depend mainly on Global Navigation Satellite Systems (GNSS), such as Global Positioning System (GPS). However, GNSS suffer from signal blockage and degradation in urban areas. At the same time, most land vehicles, nowadays, come with low-cost low-power Inertial Measurement Units (IMU). Although these IMU can be used an accurate short-term tracking system using Inertial Navigation Systems (INS) technology, they are currently mostly used only for safety applications. This paper proposes an enhanced land-vehicles tracking system by integrating a reduced IMU system with GPS to enhance the tracking accuracy of land vehicles in downtown and urban areas. Commonly, GPS/INS integration is based on Kalman Filter (KF), where a linearized dynamic models for INS errors is utilized. If Low-Cost MEMS-based inertial sensors with complex stochastic error nonlinearity are used, performance degrades significantly during short periods of GPS-outages. This paper presents a nonlinear INS-errors modelling using a fast nonlinear identification technique called fast orthogonal search (FOS). During reliable GPS coverage, the corrected vehicle state and sensors measurements are input to FOS and the FOS models outputs are trained to predict the INS deviations from GPS. During GPS-outages in urban areas, the trained FOS models along with the most recent vehicle state are used to predict INS deviations from GPS. The predicted INS deviations are then feedback to the system Kalman Filter, as updates to estimate all INS errors. The experimental setup of this work used a very low-cost IMU from Crossbow Inc. (USA based), the vehicle odometer measurements along with a GPS receiver from Novatel, Inc. (Canada based). Experiments were performed in Kingston, Ontario, Canada. Initial results show promising improvement of tracking accuracy in challenging GNSS-denied areas.**

Keywords-Land Vehicles Tracking; Reduced IMU; GPS; INS/GPS integration.

I. INTRODUCTION

Inertial Navigation Systems (INS) utilize inertial sensors to provide navigation information continuously with time [1][12][24]. In a Strapdown 3D INS with full Inertial Measurements Unit (IMU) [24][25], three acceleration sensors (Accelerometers) and three angular rate sensors (Gyroscopes) are utilized. The accelerometers measure the acceleration of the moving body in three orthogonal directions. Gyroscope measures the rotation rate around these three basic orthogonal axes. The essential functions in INS are defined as follows: 1) Determination of the angular motion of a vehicle using gyroscopic sensors, from which its attitude relative to a reference frame may be derived. 2) Measure the acceleration using accelerometers. 3) Resolve the acceleration measurements into the reference frame using the knowledge of attitude. 4) Account for the gravity component. 5) Integrate the resolved accelerations to estimate the velocity and position of the vehicle. Although INS systems have good short term accuracy, there are two main problems in using such a scheme. The first problem is the sensor imperfections and drifts [2][8]. The second problem is that the measurements of such sensors must be mathematically integrated to provide velocity, position, and attitude information. Integration causes errors to accumulate [2][8] resulting in huge drifts over time that growth without bounds.

On the other side, GPS systems provide consistent long term accuracy giving position and velocity updates using GPS satellites signals processing [1][12]. A major problem of GPS is signal blockage and multi-path in urban canyons, under buildings, and tunnels. In these environments, signal may be difficult to acquire or number of satellites available may be not sufficient to provide position information [25].

Based on the complementary error characteristics of INS and GPS, an integrated solution using both systems is often used. Although there are many approaches to fuse data from both systems, KF is most widely used [1][12][19]. KF utilizes an error dynamic model of the INS system errors to implement two main steps: Prediction step and Update step. Prediction step is done as long as no GPS update is available. In this step, the system uses the error dynamic model to estimate the INS errors. In the update step, GPS velocity and position measurements are used to get optimal estimate of INS errors. Thus, by subtracting INS errors from the INS output, accurate navigation information is obtained. This integration scheme is called loosely coupled which is utilized here in this work. This scheme is shown in Fig. 1. The challenge with INS/GPS systems is that during GPS

ISBN: 978-1-61208-348-3

outages, the system depends only on the INS error dynamic model which is, in most of the cases, an approximate linearized model. This leads to poor errors estimates during GPS outages. Thus, the performance degrades significantly during GPS outages. This paper presents an enhanced GPS/Reduced INS integrated navigation system that is based on nonlinear systems identification technique called Fast Orthogonal Search (FOS). The novelty aspects of this work lies in the utilization of fast nonlinear modeling of INS errors using FOS. Compared to existing linear estimation, such as KF [1], and existing nonlinear filtering techniques such as Particle Filter [28], the utilization of FOS is significantly faster and more reliable.

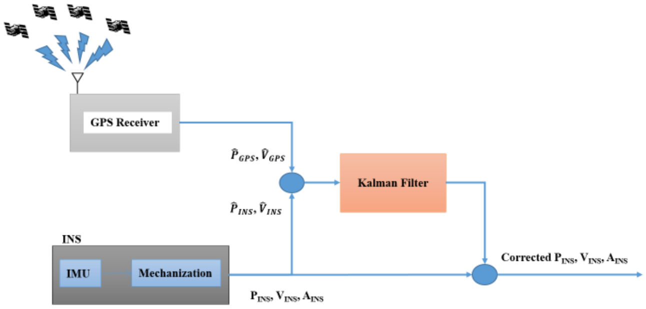

GPS Receiver
$\hat{P}_{GPS}, \hat{V}_{GPS}$
Kalman Filter
$\hat{P}_{INS}, \hat{V}_{INS}$
INS
IMU
Mechanization
Corrected P_{INS}, V_{INS}, A_{INS}
P_{INS}, V_{INS}, A_{INS}

Figure 1. INS/GPS Integration in Loosely Coupled scheme

The remaining of the paper is organized as follows: Section II describes the problem. Section III describes the methodology including the reduced IMU/GPS navigation system, the FOS algorithm, and the proposed bridging technique. Section IV describes experimental work and results.

II. PROBLEM DEFINITION AND RESEARCH OBJECTIVES

GPS/INS integration is based on KF, where a linearized dynamic models for INS errors is utilized. If Low-Cost MEMS-based inertial sensors with complex stochastic error nonlinearity are used, performance degrades significantly during GPS-outages. Although several solutions to bridge GPS outages were introduced [4][5][6][8][9]. Majority of these solutions are based on utilizing Artificial Intelligence (AI) techniques to train INS errors estimation model that can be used during GPS outages instead of KF update step. One problem of these bridging schemes is that the resulting models may be over-learned the data records that they were trained on. This leads to another problem which is the short availability period of the models. Thus, these models may be useful in short GPS outages, but degrade significantly if outages periods are several minutes [3-6][9]. In addition, the scheme in which these bridging techniques is used is to totally depend on such AI-trained model separately, without interaction with KF. This scheme prevents such bridging techniques from the optimal estimation that KF provides. Moreover, these methods use models with sophisticated parameters that need to be estimated during good GPS availability, which add complexity and computational load to the navigation system.

The primary objectives of this research is as follows:

- To propose a KF algorithm with new GPS outage bridging scheme to mitigate large drifts during GPS outages.
- The bridging technique should not add much complexity to the integrated INS/GPS solution to be suitable for real-time realization.
- To provide this INS/GPS vehicular navigation system at low cost using a reduced IMU consists of single vertical gyro and two level accelerometer aided by vehicle speed measurements.

III. METHODOLOGY

A. GPS/Reduced IMU System

The proposed GPS outages bridging technique is realized on low-cost 3D land-vehicles tracking system using Reduced IMU integrated with GPS, based on Kalman Filtering. A low-cost 3D Reduced IMU platform consists of one MEMS grade vertically aligned gyroscope, two horizontal accelerometers, and vehicle odometer. This platform is shown in Fig. 2.

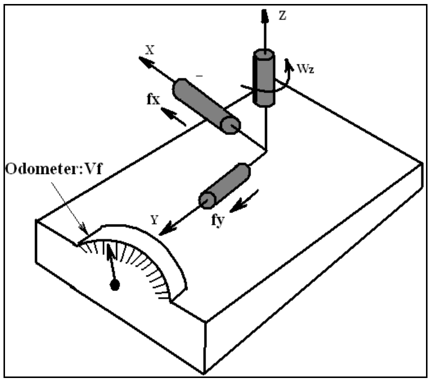


Figure 2. 3D Reduced IMU platform

The state of the vehicle is determined by the vector: $\{A_k, r_k, p_k, \phi_k, \lambda_k, h_k, Ve_k, Vn_k, Vu_k\}$, where φ_k is the latitude of the vehicle, λ_k is its longitude, and h_k is its altitude, Ve_k, Vn_k, Vu_k are the East, North, and up velocity, respectively, p_k is the pitch angle (inclination), r_k is the roll angle, and A_k is the azimuth angle (heading from north). The INS error state vector $x_k = [\ \delta A_k\ , \delta r_k\ ,$ $\delta p_k\ ,\ \delta\phi_k\ , \delta\lambda_k\ , \delta h_k\ , \delta Ve_k\ , \delta Vn_k\ , \delta Vu_k\ , \delta a_{od}$ $, \delta f_x\ , \delta f_y\ , \delta w_z\ ]$ where $\delta a_{od}, \delta f_x, \delta f_y, \delta w_z$ are errors of the odometer-derived acceleration, transversal accelerometer, forward accelerometer, and the gyroscope, respectively.

The nonlinear vehicle state dynamic model is generally given by

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (1)$$

where u_k are the sensors reading, and w_k is the noise contaminating sensors measurements (process noise). The detailed mathematical equations of this dynamic model can be found in [28]. The measurement model involves GPS velocity, position, and azimuth updates and it is given generally as

$$z_k = h(x_k, v_k) \quad (2)$$

where z_k are sub-set of Reduced INS system error state vector whose elements can be directly observed from the difference between Reduced INS output and GPS measurements which are velocity, position, and azimuth. The v_k is the GPS measurement noise.

In KF reduced INS/GPS integrated system, both Reduced INS errors and GPS measurement dynamic models in (1) (2) are linearized using Taylor series expansion [1] to apply Kalman Filtering. After linearization, systems models are given by

$$x_k = \Phi x_{k-1} + G u_{k-1} + w_{k-1} \quad 3)$$

$$z_k = H x_k + v_k \quad (4)$$

When GPS observations are available, deviations of Reduced INS output (position, velocity, and azimuth) from GPS measurements (z_k^{GPS}) is used as observations to KF which use the difference between the actual system output (z_k) and the observation (z_k^{GPS}) to derive the system to correct its state which is Reduced INS error state vector. Having the observations ($\delta z_k = z_k - z_k^{GPS}$), Reduced INS error state x_k is now partially known from δz_k. Hence, KF performs the update step to estimate the complete Reduced INS error state vector x_k as follows:

$$x_k = x_k + K\delta y_k \quad (5)$$

where K is the Kalman gain. Hence, Reduced INS navigation output is corrected by subtracting errors state from it. For more details about KF equations and Kalman gain derivations, we refer the reader to [1][12][19].

B. *Fast Orthogonal Search (FOS)*

Orthogonal Search [26][27] is a general purpose nonlinear systems identification tool that can model any general system as seen in Fig. 3, and as explained in the following figure, using the following general model:

$$Y_j[n] = \sum_{m=0}^{C-1} a_{jm} P_m[n] + e_j[n] \quad (6)$$

where $P_m[n]$ is a set of arbitrary candidates , a_{jm} are coefficients and $e_j[n]$ is the residual errors. The purpose of FOS is to select the best set of candidates $P_m[n]$ and the coefficients a_{jm} that minimizes $e_j[n]$. The candidates $P_m[n]$ can be any arbitrary function of system inputs and outputs. For example, in an autoregressive model, the candidates $P_m[n]$ would be the system input delayed with specific number of samples ($x[n-l], l = 1, 2, ..., \mathrm{L}$). In Orthogonal Search techniques, a Gram–Schmidt procedure [26][27] is used to replace the functions $P_m[n]$ by a set of orthogonal basis functions $W_m[n]$ where the model for a specific j is represented by the following corresponding model:

$$Y[n] = \sum_{m=0}^{C-1} g_m W_m[n] + e[n] \quad (7)$$

In orthogonal basis function space, the coefficients g_m that minimize the mean square error over the observations is given by

$$g_m = \frac{\overline{Y[n]W_m[n]}}{\overline{W_m^2[n]}} \quad (8)$$

where the over-bar in denotes the time average. The mean square error is given by:

$$\overline{e^2} = \overline{\left[Y[n] - \sum_{m=0}^{C-1} g_m W_m[n]\right]^2} = \overline{Y^2[n]} - \sum_{m=0}^{C-1} Q_m \quad (9)$$

Where

(10)

$$Q_m = \frac{\left[\overline{Y[n]W_m[n]}\right]^2}{\overline{W_m^2[n]}} \quad (10)$$

The reduction in mean square error resulting from adding a term $a_m P_m[n]$ is Q_m . The fast orthogonal search procedure makes use of the fact that it is not necessary to create the orthogonal functions $W_m[n]$ explicitly. Only their correlations with $P_m[n]$, the data $Y[n]$, and with themselves are required. By eliminating the generation of the orthogonal functions $W_m[n]$ explicitly, the FOS performance is much faster than existing traditional modeling techniques. This enables the FOS to work well in real-time applications that require superior performance, such as video streaming, for high speed networks, such as ATM and Internet.

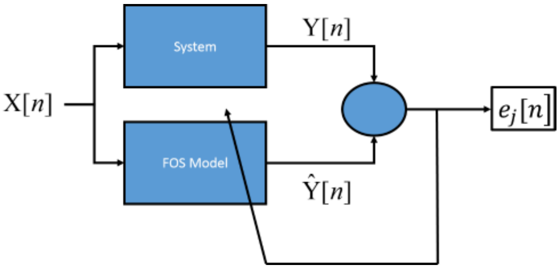


Figure 3. FOS Nonlinear Systems Identification Technique

C. *Bridging GPS outages using FOS*

During GPS reliable availability, we have valuable information that should not be ignored. We have the corrected vehicle state which is the most accurate state estimation according to all available knowledge till the moment. In addition, we have the current sensors readings and the Reduced INS output deviations from GPS measurements. This valuable information represents the error characteristics of INS or Reduced INS solution by giving us data points that map the current vehicle state with current sensors readings (as input) and the INS or Reduced INS errors in position, velocity, and attitude (as output). If enough number of data points is collected in a data set in the format shown in Table 1, a FOS model can learn this data set mapping [18] and provide predictions of data points that are not seen before in the data set we already collected.

TABLE 1. INPUT/OUTPUT DATA FOR FOS MODELING

INPUT		OUTPUT
Corrected Vehicle State (velocity and Attitude)	**Sensors Readings**	**Reduced INS output deviations from GPS velocity and attitude**
…….	…….	…….

During GPS outages, FOS equations (3) and (4) are used to predict the Reduced INS output deviations (part of error state vector *x*), which then are fed to KF as a *virtual GPS updates* to estimate all Reduced INS output errors. The mechanism is shown in Fig. 4 and Fig. 5.

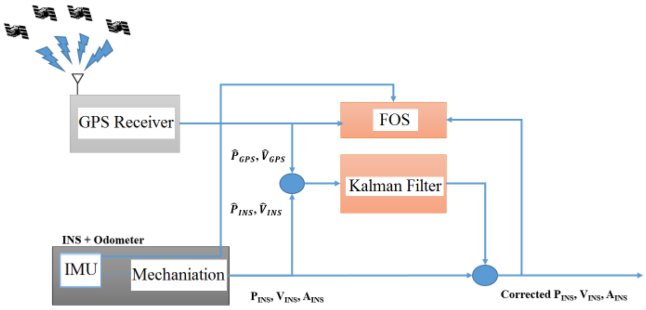

GPS Receiver
FOS
Kalman Filter
$\hat{P}_{GPS}, \hat{V}_{GPS}$
$\hat{P}_{INS}, \hat{V}_{INS}$
INS + Odometer
IMU
Mechaniation
$P_{INS}, V_{INS}, A_{INS}$
Corrected $P_{INS}, V_{INS}, A_{INS}$

Figure 4. FOS-Aided Reduced INS/GPS Mechanism in training

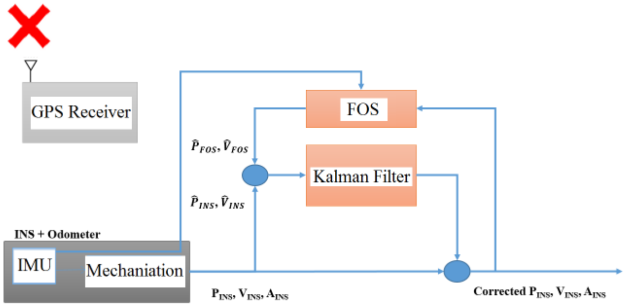

GPS Receiver
FOS
Kalman Filter
$\hat{P}_{FOS}, \hat{V}_{FOS}$
$\hat{P}_{INS}, \hat{V}_{INS}$
INS + Odometer
IMU
Mechaniation
$P_{INS}, V_{INS}, A_{INS}$
Corrected $P_{INS}, V_{INS}, A_{INS}$

Figure 5. FOS-Aided Reduced INS/GPS Mechanism in service

IV. EXPERIMENTAL WORK AND RESULTS

The developed INS/GPS loosely coupled KF algorithm was tested on physical road data records collected over two different trajectories. The set of equipment used in experiments are as follows: Honeywell HG1700 AG11 tactical grade Inertial Measurement Unit (IMU) , Novatel GPS receiver, CarChip E/X (8225) data logger [17] of a General Motors Passenger Van, and Laptop computer to control the equipments and log recorded data. Novatel CDU interface software was used to record GPS and IMU data which provide USB ports interface. G2 Pro-Pack Span unit developed by Novatel provides a tightly coupled INS/GPS navigation solution, which was used as a reference to evaluate proposed technique. Fig. 6 shows the testing trajectory as it appears in GPS Visualizer tool.

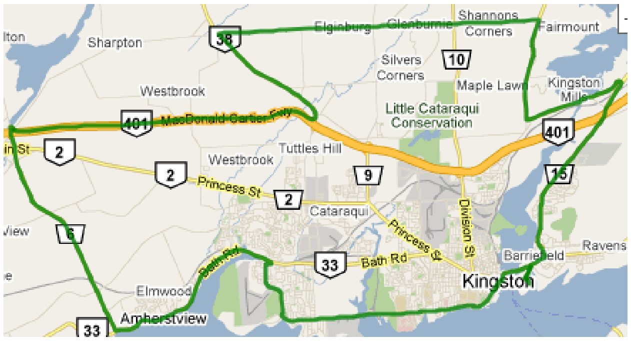


Figure 6. Testing Trajectory

The Root Mean Square error (RMSE) [8] of the horizontal position of the vehicle during GPS outage was used as a performance measure. RMSE during 20 minutes of GPS-outage is shown in Fig. 7, which is compared with and without the FOS-bridging technique. The FOS was trained for only 6 minutes of good GPS availability period before the GPS-outage starts.

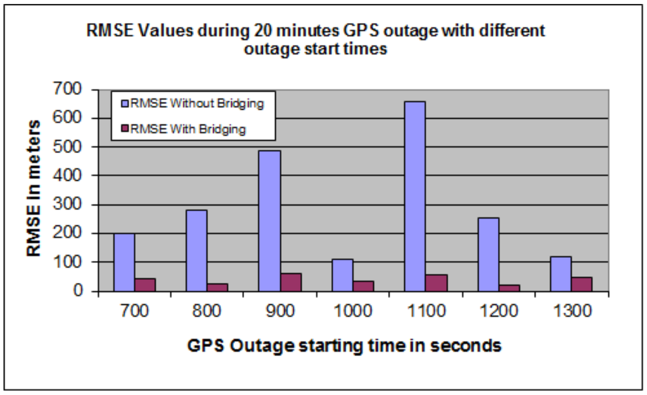


Figure 7. RMSE during 20 minutes of GPS-outage

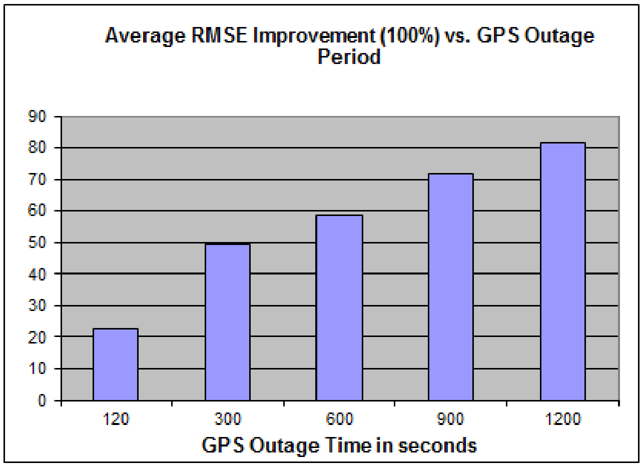


Figure 8. Relationship between RMSE percentage improvements vs GPS-outage period

Fig. 8 shows the relationship between the GPS-outage period and the improvements in RMSE obtained by applying the proposed FOS-based bridging technique. Obviously the FOS performs better with longer GPS-outages.

V. CONCLUSION AND FUTURE WORK

This work presented an enhanced multi-sensors INS/GPS tracking systems for land vehicles using Fast Orthogonal Search (FOS) as a nonlinear identification technique. the proposed bridging scheme of Kalman Filter INS/GPS tracking systems successfully prevents the large drifts that occur during long GPS outages periods. The bridging scheme utilized FOS-based measurements prediction to enable Kalman Filter to perform update step on virtual aiding measurements. Experimental results show great RMSE improvement in longer GPS-outages. The proposed bridging scheme can be used with any AI-based modeling method or non-linear systems identification technique. Future work includes applying the same mechanism on full 3D IMU/GPS configuration [2] instead of the reduced IMU configuration. In this case, it is expected that more FOS candidates may be required [26].

REFERENCES

[1] A. Farrell, Aided Navigation – GPS with High Rate Sensors, McGraw Hill, 2008.

[2] Y. X. Niu and N. El-Sheimy, “Real-time MEMS based INS/GPS integrated navigation system for land vehicle navigation application,” Proc. Navigation, National Technical Meeting (ITM), Monterey, CA, United States, January 2006, pp. 501-507.

[3] E. D. Kaplan, Understanding GPS Principles and Applications, Artech House, Boston, 1996.

[4] L. Semeniuk and A. Noureldin, “Bridging GPS outages using neural network estimates of INS position and velocity errors,” Measurement Science and Technology, vol. 17, no. 9, September 2006, pp. 2782–2798.

[5] A. Cole, J. Wang, C. Rizos, and A.G. Dempster, “Bridging GPS outages in the agricultural environment using virtualite measurements,” Symp. Position, Location and Navigation (PLANS), IEEE/ION, May 2008, pp. 497–504.

[6] P. Chen-peng and L.Zao-zhen, “Bridging GPS outages of tightly coupled GPS/SINS based on the Adaptive Track Fusion using RBF neural network,” IEEE International Symp. Industrial Electronics (ISIE), July 2009, pp. 971-976.

[7] M. Shaik, O. Das, L. Zhao, and Z. Liao, ”Inter-vehicle range smoothing for NLOS condition in the persistence of GPS outages,” Proc. 4th IEEE Conference on Industrial Electronics and Applications (ICIEA), May 2009, pp. 3904–3909.

[8] W. Abd-Elhamid, A. Noureldin, and N. El-Sheimy, “Adaptive Fuzzy Modeling of Low Cost Inertial Based Positioning Errors,” IEEE Transactions on Fuzzy Systems, vol. 15, no. 3, June 2007, pp. 519–529.

[9] K. Kim and C. G. Park, “INS/GPS tightly coupled approach using an INS error predictor,” Proc. 18th International Technical Meeting of the Satellite Division of The Institute of Navigation, Long Beach, CA, United States, September 2005, pp. pp. 488-493.

[10] U. Iqbal, T. Karamat, A. Okou, and A. Noureldin, “Experimental Results on an Integrated GPS and Multi Sensor System for Land Vehicle Positioning,” Hindawi Publishing Corporation, International Journal of Navigation and Observation, vol. 2009, Article ID 765010.

[11] B. M. Scherzinger and S. Woolven, “POS/MV-handling GPS outages with tightly coupled inertial/GPS integration,” Proc. OCEANS, MTS/IEEE Prospects for the 21st Century, vol. 1, September 1996, pp. 422–428

[12] S. Mohinder, S. Grewal, R. Lawrence, and A. P.Andrews, “Global Positioning Systems, Inertial Navigation Systems, and Integration,” Wiley & Sons Inc. 2nd ed., 2007.

[13] L. Yong, P. Mumford , and C. Rizos, ”Performance of a low-cost field re-configurable real-time GPS/INS integrated system in urban navigation,” Symp. Position, Location and Navigation (PLANS), IEEE/ION, May 2008, pp. 878–885.

[14] Z. Berman and J.D. Powell, ”The role of dead reckoning and inertial sensors in future general aviation navigation,” Symp. Position Location and Navigation (PLANS), April 1998, pp. 510–517.

[15] V. Malyavej, P. Torteeka, S. Wongkharn, and T. Wiangtong, “Pose estimation of unmanned ground vehicle based on dead-reckoning/GPS sensor fusion by unscented Kalman

filter," Proc. International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), vol. 1, May 2009, pp. 6-9.

[16] L. M. Ha, "MEMS INS/GPS integration using Kalman Filters," Proc. International Conference on Advanced Technologies for Communications, October 2008, pp. 449-449.

[17] CarChip E/X OBDII-Based Vehicle Data Logger Software, http://www.davisnet.com/product_documents/drive/spec_sheets/8211-21-25_carchip_specsB.pdf, [retrieved: May, 2014].

[18] A. Noureldin, T. B. Karamat, M. D. Elberts, and E. Shafie, "Performance Enhancement of MEMS-Based INS/GPS: Integration for Low Cost Navigation Applications," IEEE Transactions on Vehicular Technology, vol. 58, no. 3, May 2008, pp. 1077-1096.

[19] R. V. C. Wong, K. P. Schwarz, and M. E. Cannon, "High-accuracy kinematic positioning by GPS–INS," J. Inst. Navigat., vol. 35, no. 2, Summer 1988, pp. 275–287.

[20] K. W. Chiang, "INS/GPS integration using neural networks for land vehicular navigation applications," PhD dissertation, Dept. Geomat., Univ. Calgary, Calgary, AB, Canada, 2004.

[21] M. D. Eberts "Performance Enhancement of MEMS based INS/GPS integration for low cost navigation applications," MSc. thesis, Dept. Elect. Computer. Eng., Roy. Mil. College, Kingston, ON, Canada, 2007.

[22] P. Zarchan, "Fundamentals of Kalman Filtering: A Practical Approach," 2nd ed., Reston, VA: AIAA, 2005.

[23] A. Hiliuta, R. Landry, and F. Gagnon, "Fuzzy corrections in a GPS/INS hybrid navigation system," IEEE Transactions on Aerospace and Electronic Systems, vol. 40, no. 2, April 2004, pp. 591–600.

[24] D. H. Titterton, and J.L. Weston, "Strapdown inertial navigation technology," 2nd ed., The Institution of Electrical Engineers, 2004.

[25] A. Noureldin, "Mobile Multi-Sensor System Integration," Course Notes, Royal Military College, EE 513, fall 2009.

[26] M. J. Korenberg and L. D. Paarmann, "Applications of fast orthogonal search: Time-series analysis and resolution of signals in noise," Analysis of Biomedical Engineering, vol. 17, no. 3, 1989, pp. 219–231.

[27] M. J. Korenberg and L. D. Paarmann, "Orthogonal approaches to time-series analysis and system identification," IEEE Signal Processing Magazine, vol. 8, no. 3, 1991, pp. 29–43.

[28] J. Georgy, A. Noureldin, M. Korenberg, and M. Bayoumi, "Low Cost 3D Navigation Solution for Reduced INS/GPS Integration Using Mixture Particle Filter," IEEE Transactions on Vehicular Technology, vol. 59, no. 2, February 2010, pp. 599-615.

Application of Rate Adaptation Algorithm on Road Safety in Vehicular Networks

Kenneth S. Nwizege
College of Engineering
Swansea University, UK
kennethsorle@yahoo.com

Mauro Bottero
Independent Consultant,
Mulan
mauro_bottero@yahoo.it

Abstract **- Vehicular communications occur when two or more vehicles come into range with one another, to share data over wireless media. The applications of vehicular communication are far-reaching, from toll collection to collision avoidance. One of the goals of rate adaptation is to maximize throughput by exploiting the multiple transmission rates available for 802.11 devices by adjusting their transmission rates dynamically, based on to the time-varying and location dependent wireless channel conditions.**

In this paper, we present and study in detail Adaptive Context-Aware Rate Selection (ACARS) algorithm that is efficient in data transfer, energy utilization and road safety applications. The goal of ACARS is to select the rate that will yield a good throughput performance, with transmit power control and access point (AP) coordination to improve data transfer performance and safety application in Dedicated Short Range Communication (DSRC). From results obtained, ACARS is able to minimize the total transmit power in the presence of propagation processes and mobility of vehicles, by adapting to the rapidly varying channels conditions compared to other rate adaptation algorithms.

Keywords- DSRC; Vehicular Communication; IEEE802.11p; Rate Adaptation; Road Safety.

I. INTRODUCTION

Safety of lives is one of the primary concerns for the evolution of Dedicated Short Range Communication (DSRC) technology in vehicular networks. The rate at which accidents occur on our roads increase proportionately, especially as the number of vehicles on the road is on the increase. According to recent news, the number of connected cars is expected to grow from 45 million at the end of 2011 to 210 million by 2016 [17]. Carelessness from drivers, inexperience, mal-function of vehicles and the 'act of god' are some popular reasons for cause of road accidents, which leads to death due to its regular occurrences. For this reason, DSRC has proposed a technology to be built into every vehicle [1] since safety in vehicular communication is inevitable. DSRC is designed to offer complete solution for mobile data broadcast, and also to active Wireless Access to Vehicular Environment (WAVE) protocol. It is a short to medium range communication service that supports applications like: electronic toll collection, Collision Avoidance (CA), Quality of Service (QoS) and public safety etc.

The focus of DSRC for any Rate Adaptation Algorithm (RAA) is low latency and high data rate, experienced due to the fast varying condition of vehicles as they change speed and environments. DSRC supports both safety (control channel) and non-safety (service channel) to enable its effectiveness. Vehicles must be able to switch between Control Channel (CCH) and Service Channels (SCHs) several times a second. Both of these share the limited resources of DSRC. Control Channel Interval (CCHI) has direct impact on reliability; the larger CCHI, the lower collision probability.

The rest of the paper is organized as follows: Section 2 is review of literature, while Section 3 deals with vehicular communication, Section 4 is an overview of ACARS, Section 5 presents the network concept and the results and discussions are presented in Section 6. Finally, Section 7 concludes the paper.

II. REVIEW OF LITERATURE

Auto Rate Fallback (ARF) [18] was first proposed in 1996, as the simplest and first rate adaptation algorithm. Other popular existing RAAs in wireless networks are Adaptive Auto Rate Fallback (AARF) [16] (ACARS) [11], Channel-Aware Rate Adaptation (CARA) [13], Context-Aware Rate Adaptation Algorithm (CARA) [15], Robust Rate Adaptation Algorithm (RRAA) [16] etc.

In ARF the decision whether to increase or decrease the transmission rate is based on the number of consecutive successfully or unsuccessfully transmission attempts. This algorithm is widely adopted because it is simple. In this algorithm, the sender tries to send a packet at a higher rate after a fixed number of continuously successful transmissions at a given rate. The sender decreases the rate after one or two consecutive failures. If the probe packet is successful, the next packet will be sent at higher rate and if not, the sender will immediately lower the rate. The sender also lowers the rate after two consecutive failures.

In [2]11]15][16], AARF was implemented with mobility concept. From these references, some of the implementations were with context- information [2] [11], while others implemented with power control in analysing the performance of rate adaptation algorithm.

ONOE [16] is a very slowly adapting algorithm whose implementation is available in the MADWifi driver code [5]. It tries to change the rate after one second interval. ONOE is a credit-based algorithm that maintains the credit score of the current rate for every destination and after the end of a second; it calculates the credit and makes the rate change decision. This rate selection scheme has also been implemented with mobility and power control in [2][11] [15] where performances of various rate schemes were analysed.

In [4], SampleRate algorithm was proposed in order to maximise throughput in wireless networks. It is based on transmission statistics over cycles. In every tenth packet data, it picks a random rate that may do better than the current one to send the data packet. If it occurs that the selected rate provides smaller transmission time, it will switch to this rate. The performance of this rate selection scheme was evaluated in [2], with different performance metrics for various RAAs.

Furthermore, in [2] another RAA was proposed known as Context-Aware Rate Selection (CARS). This algorithm performs better in stressful scenarios, since it adapts its bit-rate faster in varying conditions. From [2], results show that CARS performs better than ONOE and SampleRate by using optimum higher data rates which allows CARS scheme to reduce network load. The limitation in this scheme is that, it lacks the ability to dynamically tune the estimation window size using the context-information, and it is not robust to adapt to shadowing effect available in wireless and mobile environments and detecting channel errors.

In [12] [13] [14] , another rate selection scheme known as Context-Aware Rate Algorithm (CARA), also considered other RAAs by implemented mobility in existing RAAs and CARA itself., but there was no implementation of power control scheme to enable existing RAAs estimate Signal-to-Noise Ratio (SNR) from the Physical (PHY) layer.

Adaptive Context-Aware Rate Selection (ACARS) algorithm is a SNR-based rate selection scheme that relies on the Request-To-Send/Clear-To-Send (RTS/CTS) mechanism to provide instantaneous receiver-side Signal-to-Interference Noise Ratio (SINR) information to the transmitter. With the knowledge of the SINR at the receiver, the transmitter directly sets the transmission rate without wasting time to probe. But the trade-off in a SNR-based rate selection scheme is that in trying to solve the hidden node problem using RTS/CTS mechanisms, introduces significant overhead because of the time it takes in communicating with the receiver to estimate SNR from the PHY layer. Although ACARS performs well in the presence of fading processes, it has slow response to path loss exponents and hence does not perform very well [11] [15].

In this paper, we will evaluate the performances of various rate selection schemes and discuss results obtained as it applies to road safety.

III. VEHICULAR COMMUNICATION

Vehicular communication can either be an Ad-Hoc network where all vehicles communicate with each other directly or infrastructure network where vehicles communicate via an Access Point (AP). We have only shown a Vehicle-to-Infrastructure (V21) in Figure 2, because that is the only network configuration we have used in this paper.

The growth of connected cars associated with Mobile Information and Communication Technologies (MICT) will change the way vehicular environments will be planed and maintained. In this scope, context-awareness rises as an important technology so as to achieve optimal vehicular-centric information , such as assisting vehicular applications with meaningful information.

A. IEEE 802.11p Multichannel Operation

The PHY layer is responsible for transmitting raw bits in wireless channels; this is achieved via channel assignment. IEEE 802.11p is an extension of 802.11 wireless LAN Medium Access (MAC) and PHY. Three different PHY Layer modes have been defined by 802.11-2007 standards. They are the 20 MHz, 10 MHz and 5 MHz These modes can be achieved by using a reduced clock/sampling rate [3] [4] [8] [9].

B. Medium-Access Control (MAC) Layer

The function of the MAC layer is to coordinate the use of the communication medium. MAC layer protocol decides which node will access the shared medium at any given time. The MAC layer uses the Collision Avoidance (CSMA/CA) mechanism to regular access to the channel. The physical CSMA/CA does not rely on the ability of stations to detect a collision by hearing their own transmission; an Acknowledgement (ACK) is transmitted by the destination station to signal the successful reception of the transmitted packets and then transmission of ACK is immediately done following the packet reception after a Short Interframe Space (SIFS).

IV. OVERVIEW OF ACARS

In vehicular communication, context-information include speed, acceleration of the vehicle, position, distance from the neighbouring vehicle, environmental factors such as location, time of day, weather, type of road traffic density. In ACARS, we only used two significant parameters: speed of vehicles and the distance of the vehicles from the AP. This algorithm is based on CARS scheme with some assumptions, and modifications to the original CARS algorithm. The full implementation of CARS algorithm is not known from [2] because, information such as context-information, Packet Error

Rate (PER) were not discussed, making it difficult to implement line 4 of CARS algorithm [2].

In this design, we used some mathematical illustrations to derive parameters for context-information and use them in implementing the CARS algorithm. For this reason, we re-named the original CARS algorithm as seen in [2] as modified CARS, because it is not identical to the original CARS. The function $\mathbf{E_C}$ uses context-information, transmission rate and packet length as input parameters and estimated packet error rate as output. $\mathbf{E_H}$ uses Exponentially Weighted Moving Average (EWMA) of past transmission statistics for each bit rate which has same working principle as SampleRate [7]. To predict the link quality of the channel, we used **cars.α** which is based on speed of the vehicles. When speed is zero, there is no prediction of link quality using context-information; hence EWMA will be given preference at that condition. But when vehicles are moving with high speed, **cars.α** is given preference, and this relationship **α =max(0,min(speed/S))** helps to determine the values of **cars.α** with different speed normalizers. The values of **S=30m/s** is chosen based on research in [2].

The three basic layers among the seven layers of the Open System Interconnection (OSI) reference model that implement ACARS are the application, MAC and PHY layers. The vehicles also known as Mobile Nodes (MN) use information from the application layer available in each MN, while the MAC layer handles the rate selection algorithm, and the PHY layer handles RTS/CTS frame exchange, SNR estimation and power control.

Algorithm 1. The Adaptive Context-Aware Rate Selection Algorithm

```
Function: ACARS_GetRate
 Output:  Rate
Input : ctx, α, len

1:  Update   counter of   packet transmissions
2:  Update average RSSs of  recent  ACKS(RSS)
3:  Best$_{Rate}$ = Find$_{Best}$_Rate try[ ]
4 : Determine α  by  using α= max(0,min(1, speed/S))
5:  Compute   backoff  using
6:  backoff = CW$_{size}$ x slot time
7:  Decrement all  backoff  counters
8:  Update  the  simulation   time  accordingly
9:  Requires: Mob$_{Model}$
10:  (t,v,old$_{pos}$ ,ap CommRange, n, x$_{max}$   for   Context-information)
11. Compute  Bper  from  the  PER  table
  p(n,:) = polyfit (Snr $_{(Ber)Mode1}$:,1), log(n($_{Ber)Mode1}$(:,2)),exp$_n$)
12 : Compute α  using
 cars. α = 1 - E$_H$>0) x max(0,min(1, v(iTx)/cars.S))
13:  E$_c$= cars.Ec (iTx,:)
14: Determine E$_H$ using
15:  E$_H$= cars. E$_H$ (iTx,:)
16: Compute E_C
17:  cars.E$_c$ (iTx,jj) = min(1, exp (polyval(Phy.p(jj,:), snr{$_{temp}$}))}
 18: Compute  E$_H$ using
 19: cars.EH (TxVehic,RateLevel (TxVehic))= cars. E$_H$ (TxVehic, RateLevel(TxVehic)) x (1-a)+ Bper x a
 20: Compute  PER  using
  PER=E$_c$ cars. α +(1-cars. α) x E$_H$
  Avgr$_{etries}$   = (N.PER^{(N+1)}  -(N+1).PER^N+1)/(1-PER)+N.PER^N)
  Thr =Rate/avg$_{retries}$.(1-PER^N)^/ρ
 21: Select  rate
 22:  IF Thr  > Max_Thr
 23:  Update link condition
 24: Best$_{Rate}$ ← bit-rate
 25: Max_Thr ← Thr
 26: ENDIF.
```

V. NETWORK CONCEPT

In this section, we will describe the network configuration used in implementing of our algorithm. We will also highlight on the parameters used in our simulation in MATLAB.

A. *Simulation Scenario*

In this network configuration, each time a vehicle enters into the communication range; it communicates with the Road Side Unit (RSU). It adds new vehicle information such as vehicle speed, position, distance, etc. This information helps the RSU to broadcast the emergency information to the vehicles. Every minute, vehicles leave and enter the communication range with high speed. RSU will communicate to vehicles as soon as they enter the communication range. For example, if there are no vehicles within range, and there is an accident or emergency message at that time , as soon as any vehicle enters the range, message will propagate through the first entered vehicle in that communication range. This RSU communication helps in communication when there is no vehicle in cluster range.

In this scenario, all vehicles act as clients. We use a fixed base station as server, which is similar to what is obtained in cities and highways having RSUs (e.g., kiosks and cafes) with wireless services. Our scenario consists of a road of length 1000 m with multiple lanes. The base station is located at the middle of the road. Vehicles select their speeds uniformly over the range [Speed$_{avg}$ *0.75, Speed$_{avg}$ *1.25] km/h. Tables I shows the simulation parameters used., while Figure 1 shows the V2I configuration used in our simulation.

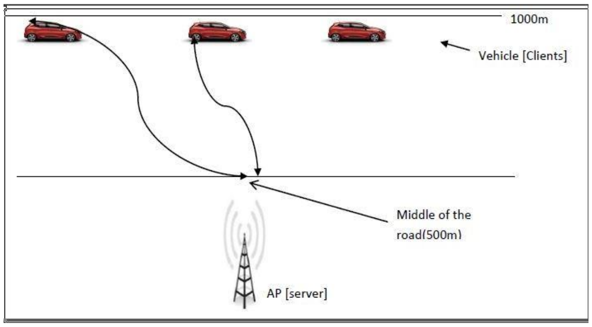


Figure 1. Network Setup.

Speed$_{avg}$ x 0.75, Speed$_{avg}$ x1.25km/h (1)

d= $\sqrt{(x_2 - x_1\ )^2 + (y_2 - y_1)^2 + \cdots + (x_n - y_n)^2}$ (2)

where **d** is distance in meters, **x** and **y** are vehicle positions in the **x** and **y** coordinates and **Speed$_{avg}$** is average speed of vehicles. **Speed$_{avg}$** of 55 km/h was used for the different number of vehicles. The distance **d** between vehicles and Access Point (AP) or RSU with vehicle positions **x** and **y** were determined from equation (2). Parameters used in this simulation are listed in Table I.

TABLE 1. CONFIGURATION PARAMETERS

Parameters (Units)	**Values**
Length of Road (m)	1000
Number of Vehicles	150
Position of AP (m)	500
PHY and MAC Protocol	802.11p
Frequency (GHz)	5.89
Normalized Transmit Power (mW)	40
Noise Power (dBm)	-90
ϒ (Path Loss exponent, Urban area cellular radio)	2.0, 2.7-3.5
Sigma (dB)	6- 8
Communication Range (m)	300
DIFS (µs)	50
SIFS ((µs)	30
HPHY (bits)	192
HMAC (bits)	200
Data rate (Mbps)	3, 4.5, 6,9,12,24,27
Maximum Retransmission	3

B. Propagation Environment

Simulation was carried out by implementing a V2I network so that analysis of AP coordination for data transfer protocol and context-information can be evaluated. Mathematical calculations were integrated into our MATLAB code and were used in this implementation. To ensure accuracy of results, simulation was done to **4** iterations and for a highly dense network of **150** vehicles.

C. Free Space Path Loss

Free space path loss model is a power off that relates to distance. Due to high mobility of vehicles as speed changes, the distance between the transmitter and receiver changes. This makes empirical free space path loss necessary in order to model the effect of distance on packet delivery probability. This space loss accounts for the loss due to spreading of Radio Frequency (RF) energy as transmission of signals propagates through free space. From the equation (3) of path loss, it is seen that the power density is reduced by $\frac{1}{R^2}$ as distance is increased.

P$_{rx}$= $p_{tx}\left(\frac{\lambda_o}{4\pi R}\right)^2$ (3)

where $\frac{P_{tx}}{4\pi R^2}$ is the power density, λ_o is wavelength in meters, P$_{rx}$, p_{tx} are received and transmit power respectively. In free space, the power of electromagnetic radiation varies inversely with the square of distance, making distance an ideal indicator of signal level as well as loss rate. Due to imperfect propagation environment, in practice, it is not exactly the inverse square. Distance between sender and receiver gives a high correlation between signal level and error rate as this affects the number of transmitted packets that will be received [19].

$g_{(t)}$ = $g_{p(t)} + g_{s(t)} + g_{m(t)}$ (4)

P_{rx} = $P_{tx} - g_t$ (5)

RSS $= P_{rx} - P_{noise}$ (6)

where g_t is power gain, $g_{p(t)}$is path loss, $g_{s(t)}$ is shadowing and $g_{m(t)}$ is multipath fading P_{noise} is noise power while RSS is the Received Signal Strength.

D. Log-Normal Shadowing

Communication channel is a time varying power gain which consists of path loss, log-normal shadowing and multipath fading. The receivers experience a desired signal gain with respect to the transmit power $\mathbf{P_t}$ used by the transmitter. We used shadowing deviation and path loss exponent in our simulation as shown in Table I to evaluate the impact of environmental factor on RAAs.

ISBN: 978-1-61208-348-3

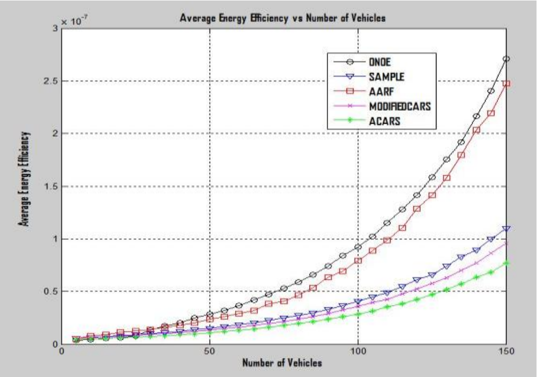


Figure 2. Average Energy Efficiency vs Number Of Vehicles.

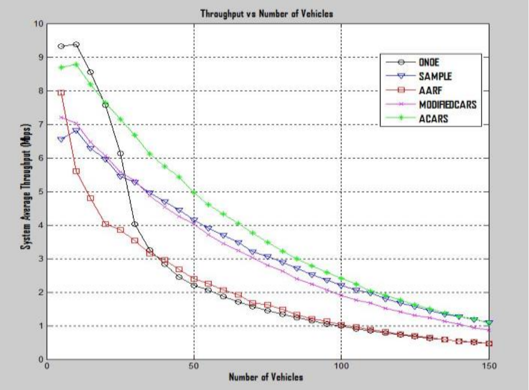


Figure 3. Average System Throughput vs Number of Vehicles.

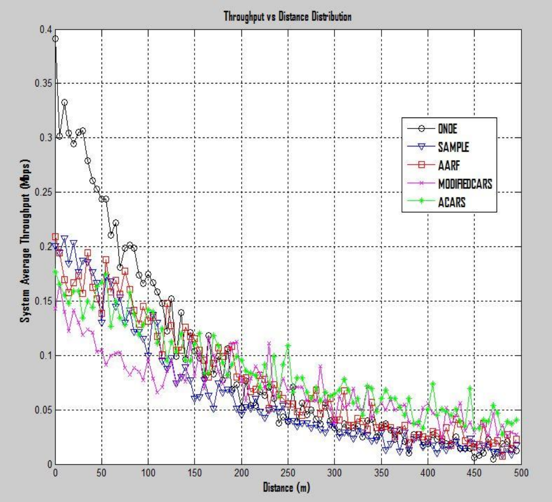


Figure 4. Average System throughput vs Distance.

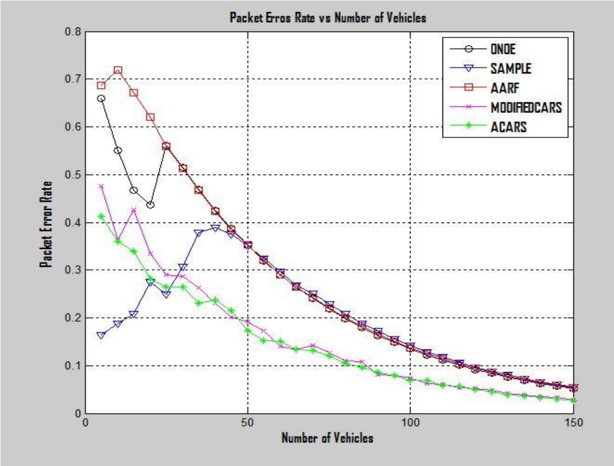


Figure 5. Packet Error Rate vs Number of Vehicles.

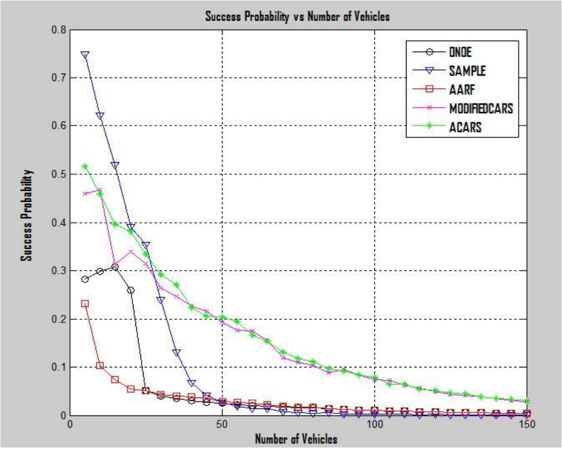


Figure 6. Success Probability vs Number of Vehicles.

VI. RESSULTS AND DISCUSSIONS

AARF has poor energy efficiency than all other rate selection schemes from Figure 2. From this figure, SampleRate performs better than the other rate schemes followed by ACARS. ONOE struggled to increase its efficiency at higher vehicle density. This trend is different from the behaviour of AARF that degrades its efficiency as the density of the vehicle increases.

ONOE is a credit-based rate adaptation algorithm; it spends about 10 seconds on each bit-rate before it increases rate and then scaling up to the highest bit-rate of 27 Mbps, therefore does not perform so well in this scenario. AARF and ONOE have same poor performances compared to the others, as seen in Figure 3. AARF waits for 10 consecutive successful transmission attempts before increasing rate. Since it is a transmitter-based rate adaptation algorithm, it cannot adapt fast in selecting the proper transmission rate that will match the channel condition, this may be one of the reasons for its low performance. ACARS performs better than all other rate adaptation schemes. Observation from this figure shows that SampleRate performs better than AARF, ONOE and MODIFIEDCARS.

As observed in Figure 4, mobility results in more rapid channel variations that are very challenging for rate selection algorithms. In such dynamic environments, we expect that it is critical to gain accurate channel information quickly in order to effectively utilize the channel. From this figure, ACARS and MODIFIEDCARS show some fluctuations in performance as the distance increases and perform better than the other rate selection schemes. ONOE performs worse than the others rate selection schemes. From this figure, it is observed that ONOE is affected greatly by vehicle mobility, because up to about **120** meters, it performs better than all others, but dropped greatly from **150** meters. This shows that ONOE cannot perform well in high mobility scenario especially as the inter-node distance increases

From Figure 4, ACARS and MODIFIEDCARS compete with each other and almost overlap as the network congestion increases due to increase of vehicles. Both of them did not perform badly as can be observed from this figure. SampleRate performs better than all other rate schemes, while AARF and ONOE did not perform well compared to the others. The reason for this may be because, both of them spend much time before changing rate in this circumstance thereby increasing the waiting time.

The performance measure between SampleRate and ACARS/MODIFIEDCARS is large. AARF also degrades fast as the number of vehicles increases. We can observe how the network congestion greatly affect AARF and ONOE.

The throughput performance of a network is affected by the rate at which packets are in error. From Figure 5, ACARS has a low PER rate, which helps in the overall throughput performance of this algorithm. On the other hand, AARF and ONOE perform very poorly compared to all other rate selection schemes. SampleRate struggles to compete with ACARS as can be seen from this figure. It also performs better that AARF, ONOE and MODIFIEDCARS in this scenario.

The success probability for both ACARS and MODIFIEDCARS are better than the others Figure 6. Success rate is proportional to network throughput. If the success rate is high then the final throughput of the system will be better. SampleRate performs poorly compared to the other rate selection schemes. The channel condition in this scenario may have greatly affected its ability to choose an appropriate transmission time in order to change rate.

From these results, we have seen that ACARS can be a good RAA, to be used for vehicle safety and data transfer in DSRC, because of its good throughout performance and high success rate.

VII. CONCLUSION AND FUTURE WORK

One of the key contributions in this paper is the implementation of a SNR-based rate adaptation algorithm that estimates SNR from the PHY layer, so as to be effective in packet delivery probability. With this feature, ACARS performs better than existing rate adaptation algorithms (RAAs) and this can be seen from most of our simulation results.

Another key contribution in this paper is the integration of power control into the design of ACARS algorithm and other existing rate adaptation algorithms. From literatures, it has either been rate adaptation analysis, or power control analysis respectively, without a combination of these two. We have combined these two in the design and implementation of ACARS. Results obtained show that ACARS can minimize energy consumption which is one of the major challenges of wireless mobile nodes. It can also reduce network congestion and enhance QoS with help of the power control scheme.

In the future, we will evaluate the performance of ACARS on other context- information and also consider a Vehicle-to-Vehicle (V2V) scenario.

.

ACKNOWLEDGEMENT

This research was supported by Education Trust Fund (ETF) through the Rivers State Polytechnic, Nigeria.

REFERENCES

[1] J. He, Z. Tang, T. O'Farrell, and T. Chen. Performance Analysis of DSRC Priority Mechanism for Road Safety Applications in Vehicular Network. Aston University, United Kingdom. IEEE Wireless Communication and Mobile Computing, 2009, pp 980-990.

[2] P. Shankar, T. Nadeem. J. Rosca , and I. Iftode. Context Aware Rate Selection for Vehicular Networks. Department of Computer Science, Rutgers University, IEEE, 2008, pp 1-10.

[3] L. Armstrong. Dedicated Short Range Communications(DSRC). http://www.leearmstrong.com/dsrc/dsrchomeset.htm.

[4] S. Wu. High Performance Rate Adaptation on IEEE 802.11 Networks. PhD Thesis, Auburn University, 2008, pp 34-60

[5] A. Kamerman , and L. Monteban. A High-Performance Wireless LAN for the Unlicensed Band. AT&T Bell Laboratories Technical Journal, 1997, pp.118-133.

[6] C. Sommer, S. Joerer, and F. Dressler. On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation. IEEE Vehicular Networking Conference 2012, pp 65-68.

[7] L. Stibor, Y. Zang, and H. Reumerman. Evaluation of communication distance of broadcast messages in a vehicular ad-hoc network using IEEE 802.11p. IEEE communication society, 2007, pp 254-256.

[8] K. S. Nwizege, F. M. Good, and S. Neenwni. Performance Analysis of Adaptive Rate Mechanism for IEEE 802.11p in DSRC for Road Safety Application in Vehicular Networks. Computational Intelligence, Modelling and Simulation (CIMSIM), 2011 Third International Conference, pp 362-366.

[9] K. S. Nwizege, J. He, and M. Shedrack,. Optimizing Rate Algorithms in Wireless Networks. European Modelling and Simulation (EMS) , Madrid, 16-18 November, 2011, pp 2-6.

[10] S. R . Saunders, and A . A. Zavala. Antennas and Propagation for Wireless Communication systems. John Wiley and Sons , Ltd, second edition, 2007, pp 172-178.

[11] K. S. Nwizege, and J. He. ACARS. Adaptive Context Aware Rate Selection Algorithms in Vehicular Networks. International

Journal of Convergence Information Technology , April 2013, pp 1.5.

[12] X. W. G. Judd, and P. Steenkiste. Efficient Channel-Aware Rate Adaptation in Dynamic Environments. (Carnegie Mellon University, Pittsburgh,PA, USA.), June 2008, pp. 118-131,

[13] S. C. J. Kim, S. Kim, and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 801.11 WLANs. IEEE Communication Society, 2006, pp 1-3.

[14] S. H. Wong, H. Yang, S. Lu, and V. Hargavan. Robust Rate Adaptation for 802.11 Wireless Networks. Proceedings of Mobicom, 2006, pp 146-152.

[15] K. S. Nwizege , and M. MacMammah,. Performance Evaluation of Path Loss Exponents on Rate Algorithms in Vehicular Networks International Journal of Emerging Science and Engineering (IJESE), August 2013, vol.1, pp 103-108.

[16] F. Karnadi, Z. H. Mo, and K. C. Lan. Rapid Generation of Realistic Mobility Models for VANET. March 2007, pp 2509-2513.

[17] http://umtri.umich.edu/cvnews/?p=54.0, Connected Vehicle News (2011). Intel Invests in Connected Cars: Connected Vehicle Market Expected to Reach 210 Million Units in 2016, Accessed January, 2014.

[18] K. Zhang, A. Lim, S. Wu, and Q. Yang. A High TCP Performance Rate Adaptation Algorithm for IEEE 802.11 Networks. International Journal of Computer Networks & Communications (IJCNC). Department of Computer Science and Software Engineering, Auburn University, Vol.2, No.6, 2010, pp 32-34.

[19] A. Sheth, and R. Han. An Implementation of Transmit Power Control in 802.11b Wireless Networks. Department of Computer Science University of Colorado, CU-CS-934-02 August 2002, pp 3-6.

Finding in Multimodal Networks without Timetables

Pedro Oscar Pérez Murueta
TECNOLÓGICO DE MONTERREY
Querétaro, México
pperezm@itesm.mx

Edgar García
IBM
México, México
garciate@mx1.ibm.com

Ma. de los Angeles Junco Rey
TECNOLÓGICO DE MONTERREY
Estado de México, México
ajunto@itesm.mx

***Abstract*— In this paper, we present an algorithm that finds a set of possible routes in a public transport network with no fixed timetables, and ranked according individual preferences and restrictions. A case study based on Mexico City transit network is presented.**

Keywords-Shortest path algorithm; public transport network; intermodal trip planning

I. INTRODUCTION

Moving in a city using public transportation requires for a commuter to plan ahead on the path to follow and the modes to use, as well as the start time of the trip if there is a schedule restriction for arriving at the destination. In this planning process, the commuter uses available network data personal preferences and past experiences.

By representing the network as a graph and using path finding algorithms, tools can be created to help users find the best path according to individual preferences and restrictions [3][6]. These tools can be used as well by transportation agencies to study the transit network in order to improve it.

Usually, these models rely on a deterministic representation of the network where timetables can be defined for the transit routes; however, another level of complexity is added when we deal with cities in developing countries, where a usual problem is the low reliability of public transportation, especially regarding timetables, caused not only by serious urban traffic congestion, but also by the fragmented nature of the industry [1]. Thus, a different approach is needed to design path finding models suited for this environment.

In this paper, an algorithm is proposed to find a set of alternative paths in a multimodal network with no fixed timetables, ranked according to individual preferences and restrictions.

The organization of this paper is as follows. Section II provides the background of our work. Section III describes the algorithm, its implementation and a case study used to test it. And, finally Section IV presents the conclusions and future works.

II. BACKGROUND

Finding the path to get from one point to another in a city using a multimodal network involves more than finding the shortest path between those points through the network; you have to take in account the possibility (and limitations) to use different ways with different results according to user preferences and restrictions. For a user the best route may be the fastest one, while another one would prefer the more cost effective or the less crowded.

This problem has been usually handled using multi-criteria path finding algorithms on graphs representing the structure of the multimodal network [2][3], and often adding time-wise weights to deal with differences in routes over time [4][5].

The super-network approach [6], where different modal networks are linked using transfer nodes, is useful to handle the multimodal network as a single graph in order to apply known path finding algorithms.

In order to calculate the departure time, it’s needed to add the travel time for each segment and the transition between modes, including waiting times. Usually these models require defining timetables for public transportation, so an expected travel time can be computed for the full path. But, what happens when a timetable is not available? This is usually the case in developing countries’ cities due to overall road characteristics, network design and public transportation ownership structures [1].

When no timetables are available, users need to rely on time expectation to compute transition time. This expectation can be the product of individual experience or crow-sourced information, but in any case it won’t be a deterministic value, but a probabilistic distribution of the time the user has to wait for the transport in a given stop/station, or the time it takes for a vehicle to go from one station to another at a given time of the day.

III. ALGORITHM

Let $S = (N,L)$ be a directed graph composed of a set N of nodes and a set of links $L \subset N \times N$, represent a super-network where each node is an entry point to the network.

A preprocessing phase is needed in order to prepare some required information. As the network is expected to remain static and there is a finite number of origins and destination points, it’s feasible to compute and store the K shortest paths for any origin-destination node pair to produce a set of paths $P=\{(o,d,i,l,n,C) \mid o,d \in N, i = 1...K, l \subset L, n \subset N\}$, where l is the set of links making the path, n includes the nodes where

ISBN: 978-1-61208-348-3

the |user will have to wait for a vehicle, and *C* is a vector of quantitative characteristics of the path (as total distance, cost or mode switch count). Even if this process is time consuming, it will only be done once.

The proposed path finding algorithm takes as input a pair of geographical points to be used as origin and destination, and a set of restrictions and preferences. Restrictions define threshold values to be contrasted with path characteristics in order to discriminate them. Preferences express the weight of each path characteristic used to decide between alternative paths.

In order to find a set of feasible alternative paths the algorithm follows the next steps:

- Origin-Destination selection – For every geographical point to be used as origin or destination, there is a variety of entry points to the multimodal network at walking distance. An entry point $e \subset N$ is considered reachable by walking if the distance from geographical reference is lower than a specified parameter *w*. So, for an origin-destination pair (*o*,*d*), let $eO = \{e \mid e \in N, \text{dist}(o,e) < w\}$and $eD = \{e \mid e \in N, \text{dist}(d,e) < w\}$ be the list of all reachable entry points from the origin and destination respectively. $Pr = \{p \mid p \subset P, p.o \in eO, p.d \in eD\}$ contains all the possible paths to get from o to *d*.
- Feasible routes – For all the paths in *Pr*, the user restrictions are evaluated, so only the paths meeting all the specified criteria are kept.
- Travel time – for each path in *Pr* a travel time value is computed as a random variable following a gamma probability distribution. Let $w_{it} = \Gamma(k_{it}, \theta_{it})$, a random variable that is gamma-distributed with shape k_{it} and scale θ_{it}, represent the expected waiting time in spot *i* at time *t*, and likewise $m_{jt} = \Gamma(k_{jt}, \theta_{jt})$ the expected travel time for link *j* at time *t*. The travel length value for a path *p* in a given timeframe *t* corresponds to the sum of the expected travel time for all the segments and waiting times included in path: $tl(p, t)=\sum_{i\in p.l} w_{it} + \sum_{j\in p.n} m_{jt}$. This value is added to vector *C* of respective path to be taken into account for path ranking.
- Based on travel time probability distribution, departure time can be calculated specifying a required arrival time and risk propensity. In the same way, an expected arrival time distribution can be computed if departure time is specified.
- Finally the individual preferences are used to weight characteristics in V for all the paths included in Pr, so a multivariate ranking algorithm [9] can be applied to decide which path is better for this user to get from origin to destination.

A. *Implementation*

The preprocessing phase requires computing the K shortest paths for any origin-destination node pair. This problem has been extensively studied since the early fifties and several implementations have been proposed, including works by Yen [8], Lawler [10], Katoh [11], Hoffman [12], Ahuja [13], Eppstein [7] and Hershberger [14] among others. All of them extend the shortest path algorithm defined by Dijkstra [15] and their efficiency is based on how they are constructed each of the K paths. For this implementation Yen's algorithm was preferred because of the ease with which it could develop a multithreaded implementation. Application was developed in the Java language.

For each pair of nodes in the network, the K shortest paths are found and stored in a database; along with estimated travel time computed using a gamma probability distribution, monetary cost of trip, number of mode shifts and line changes. All of these values are based on structural information contained in network definition.

When a user specifies a pair of origin-destination points, the path finding module uses a set of network entry and exit points to form a list of possible routes to be filtered by user constraints and then sorted by user's preferences. The following considerations are considered to choose these points: i) A person never walks more than an hour to gain access to any stop of any transport mode of the multimodal network; ii) The system will always look up to two entry and exit points to the transport system.

ALGORITHM

INPUT: An origin vertex *V*, a set *S* with all possible destinations, a set *R* of restrictions and number *K* of path to obtain for each origin-destination pair.

OUTPUT: A set *W* with de *K* best evaluated paths for each origin-destination pair.

METHOD:

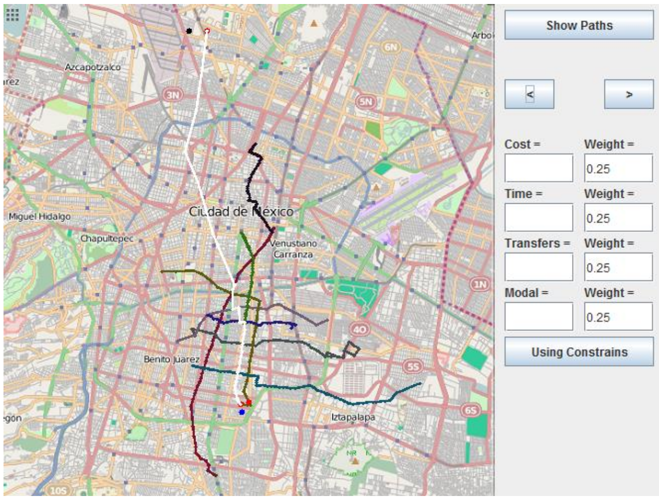
```
    for (each vertex $I$ in $S$) {
        create_thread($J[I]$ := Yen's Algorithm($V, I, K, R$));
    }
    while (!finish_all_threads) {}
    $W$ := join_all_sets($J$);
    return $W$;
```

B. *Case Study*

In order to test the implementation, a multimodal network model was built using structural data from Mexico City's public transport network publicly available from Datos Abiertos [16]. The test network includes five different modes: subway, BRT, bus, trolley and suburban train, for a total of 433 nodes. None of these modes offer timetable information or real time data to their users.

In the preprocessing step, the 10 shortest paths were computed for each pair of nodes in the network. Using an Intel Xeon server with 8 GB of RAM, OpenSUSE 12.1 operating system, the full set was generated in 703 seconds.

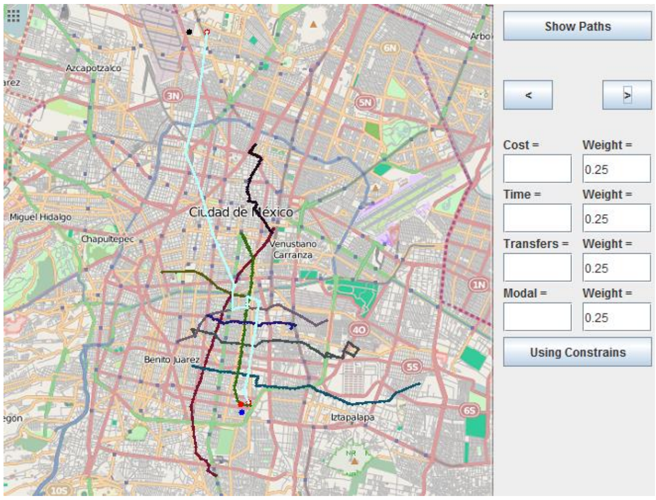
A GUI was developed to offer a visual, interactive, interface to confirm the correct operation of the path finding algorithm. This application displays the public transportation network and, given a pair of origin-destination points in the city, uses the presented algorithm to compute the list of paths and highlights the best path to get from one point to the other, allowing viewing alternative paths with higher costs. Fig. 1 shows the best path for a certain set of points.



Cost = | Weight = 0.25
Time = | Weight = 0.25
Transfers = | Weight = 0.25
Modal = | Weight = 0.25

Figure 1. -The best multimodal path between origin-destination pair of points (without restrictions).

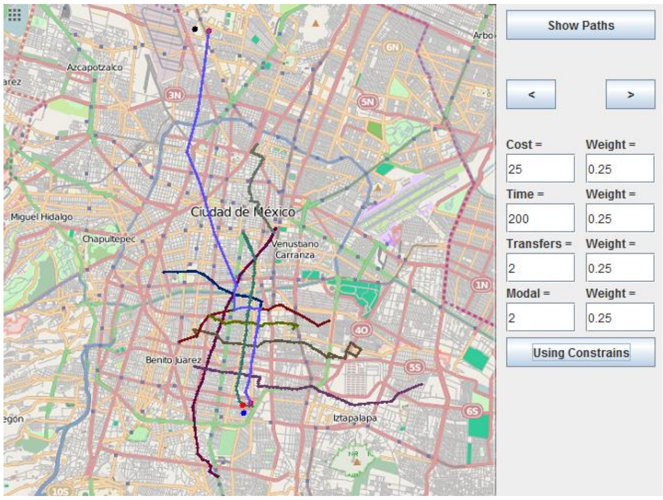
The application also shows us the next best routes. For example, Fig. 2 shows the second best path for the same defined set of points.



Cost = | Weight = 0.25
Time = | Weight = 0.25
Transfers = | Weight = 0.25
Modal = | Weight = 0.25

Figure 2. The second best multimodal path between origin-destination pair of points (without restrictions).

It is also possible to set restrictions (cost, time, number of transfers, number of mode changes) in the searches. Fig. 3 shows one path for a defined set of points with certain restrictions.



Cost = 25 | Weight = 0.25
Time = 200 | Weight = 0.25
Transfers = 2 | Weight = 0.25
Modal = 2 | Weight = 0.25

Figure 3. Computed path using restrictions.

IV. Conclusions and Future Work

In countries where public transport systems have no defined time tables is very difficult to determine the travel time between some source-destination pair. We present an algorithm that is able to estimate this time based on probability distributions. At the same time, it generates K best routes taking into account a set of specific restrictions. In a future work, this development will be used as a planning tool for multimodal travel systems and multi-agent traffic simulation.

References

[1] R. Iles, Public Transport in Devoloping Countries, Emerald Group Publishing Limited, 2005.

[2] L. Antsfeld and T. Walsh, "Finding Multi-criteria Optimal Paths in Multi-modal Public Transportation Networks using the Transit Algorithm," in Proceedings of ITS World Congress 2012, Octuber 2012, pp. 25.

[3] J. Jariyasunant, E. Mai, and R. Sengupta, "Algorithm for finding optimal paths in a public transit network," Transportation Research Record: Journal of the Transportation Research Board, no. 2256, 2010, pp. 34-42.

[4] A. Orda and R. Rom., "Shortest-path and minimumdelay algorithms in networks with time-dependent edge-length," Journal of the ACM, vol. 37, no. 3, July 1990, pp- 607-625.

[5] E. Pyrgay, F. Schulzz, and D. Wagnerz and C. Zaroliagisy, "Experimental Comparison of Shortest Path Approaches for Timetable Information," ALENEX/ANALC, January 2004, pp. 88-99.

[6] K. Carlier, S. Fiorenzo-Catalano, C. Lindveld, and P. Bovy, "A supernetwork approach towards multimodal travel modeling," in TRB 2003 Annual Meeting, January 2003, pp. 10-2460.

[7] M. Köppen and R. Vicente-Garcia, "A Fuzzy Scheme for the Ranking of Multivariate Data and its Application," in Fuzzy Information, 2004. Processing NAFIPS '04. IEEE Annual

Meeting of the, Fraunhofer IPK, Berlin, Germany, June 2004, pp. 140-145.

[8] J. Y. Yen, “Finding the K Shortest Loopless Paths in a Network,” Manage. Sci., vol. 17, no. 11, 1971, pp. 712–716.

[9] E. L. Lawler, “A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem,” Manage. Sci., vol. 18, no. 7, 1972, pp. 401–405.

[10] N. Katoh, T. Ibaraki, and H. Mine, “An efficient algorithm for K shortest simple paths,” Networks, vol. 12, no. 4, 1982, pp. 411–427.

[11] W. Hoffman and R. Pavley, “A Method for the Solution of the Nth Best Path Problem,” J. ACM, vol. 6, no. 4, 1959, pp. 506–514.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Networks flows: Theory, Algorithms and Applications, New Jersey: Prentice Hall, 1993.

[13] D. Eppstein, “Finding the K Shortest Paths,” SIAM J. Comput., vol. 28, no. 2, 1999, pp. 652–673.

[14] J. Hershberger, M. Maxel, and S. Suri, “Finding the K Shortest Simple Paths: A New Algorithm and Its Implementation,” ACM Trans. Algorithms, vol. 3, no. 4, 2007, p. 45.

[15] E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numer. Math., vol. 1, no. 1, 1959, pp. 269–271.

[16] Dirección de Sistemas de Comunicación para la Atención Ciudadana de la Coordinación General de Modernización Administrativa, "Datos Abiertos," 11 Junio 2014. [Online]. Available: http://datosabiertos.df.gob.mx. [Accessed 11 Mayo 2014].

Vehicle Online Monitoring System Based on Fuzzy Classifier

Diana María Gómez Jaramillo and Claudia Victoria Isaza Narváez
Grupo SISTEMIC, Facultad de Ingeniería
Universidad de Antioquia
Medellín, Colombia
e-mails: {dmaria.gomez, cisaza}@udea.edu.co

***Abstract*— In the automotive sector, electronic, mechanical, and software components have evolved significantly, resulting in increased complexity in vehicle fault diagnosis. The use of fuzzy classification techniques has been adapted for the online diagnosis of complex systems. In particular, Learning Algorithm for Multivariate Data Analysis (LAMDA) fuzzy classifier provides additional information through the Global Adequacy Degree (GAD) allowing to perform early preventive actions and supporting the operator in the decision-making process. This paper presents a car fault diagnosis system based on the LAMDA fuzzy classifier. The algorithm identifies, while the vehicle is in motion (online monitoring), the state of the vehicle, i.e., normal driving behavior, aggressive driving (driving behavior reflecting an impatient or angry driver) or mechanical failure. The implementation of the monitoring system implementation is performed in a midrange Renault vehicle. The algorithm achieves a 92.52% correct functional state identification with a low computational cost.**

Keywords—Fuzzy classifier; on-board diagnostics; online monitoring.

I. INTRODUCTION

The monitoring process provides information about the system functional state (situation); this information is then used as a tool to perform troubleshooting tasks and scheduling, among others [1]. In the automotive sector, electronic automotive control has led to significant changes in technology, requiring costly scanning systems for fault diagnosis and detection in vehicles [2]. For pollutant emission detection in vehicles, On Board Diagnostics (OBD) systems were introduced in 1988 [3]. In 1996, OBD II was created in order to further restrict emissions [4]. The basic operation of these systems is to activate a malfunction warning light, Malfunction Indication Light (MIL) upon detecting a fault. Recently, third generation on board diagnostics (OBD III) identifies failures by satellite, in order to report emission problems to the regulator and to identify the position of the vehicle and the fault code in order to proceed to repair it [5]. The disadvantage of OBD systems is that critical faults are not detected early (i.e., they are only detected when they have already occurred).

In 2011, Hasan et al. [6] developed a system based on OBD system operation principle using a microcontroller which integrates the scanner to the vehicle, providing the driver a graphical interface for monitoring interesting signals in real time. The systems do not generate alarms to warn about the presence of failure.

In 2004, He and Feng [7] proposed a method based on fuzzy pattern recognition and the use of similarity measurements, for diagnosis and fault detection in combustion engines. The online diagnosis test to detect abnormal operation of fuel injection showed 80% correct fault detection.

In 2008, Schilling [8][9] implemented an insulation system and engine fault detection using Kalman filters [10]. Thus, when the filter residue exceeds a threshold, the presence of engine failure is detected. In this case, it is only possible identify two states, namely, normal or fault.

The study of the angular velocity signal has also been used for detecting engine faults. The algorithm proposed by Gani and Manzie [11] verifies certain thresholds to warn about the presence of failure; the disadvantage is that this algorithm has good performance at low speeds, but it is difficult to correct the influence of the engine torque inertia at high speeds.

On the other hand, fault diagnosis has been conducted in internal combustion engines valves based on vibrations, aiming to distinguish between normal or failure states through digital image processing. However, this method is not useful for differentiating between failure classes [12]. Fault detection from vibration allows detecting incipient faults in rotating mechanical systems using Probabilistic Neural Networks (PNN). Slowness in classifying new data is one of the disadvantages of the PNN [13].

In 2011, Wenqiang et al. [14] used Bayesian networks [15] and machine learning techniques [16] for detecting fault diagnosis in vehicles. They compared the diagnosis based on time-varying Bayesian networks with the traditional static method. With a percentage of 85.7% classification accuracy, the time-varying Bayesian network presents better performance than with the static method. The test was conducted under stationary conditions at a speed of 2,000 rpm, but the vehicle was not in motion.

Recently, the use of the Hilbert-Huang transform (HHT) and Support Vector Machines (SVM) have led to engine fault diagnosis using the engine’s sound. It attains a percentage of 91.43% correct classification [17]. In this case, only one signal (microphone signal) was used; moreover, SVMs did not allow multiple classes, and the required calculation involved a high computational cost.

The above proposals are limited to identify engine failure and others extend to other parts of the vehicle, but only performing troubleshooting in a static vehicle. Moreover, proposals do not provide additional information about whether the system is in a normal state or have progression toward a fault condition.

To monitoring complex systems, fuzzy clustering techniques have been used, which have demonstrated good performance in industrial settings [18]. Fuzzy clustering algorithms allow, from a historical data, grouping similar data in the same class or functional state (e.g., normal, alarm, fault, etc.) and determining the degree of membership of a new data to all classes. Each class is associated to a functional state of the system. The LAMDA fuzzy clustering technique [19] has been widely used for process monitoring; especially due to its low computational cost [20] and because it allows the identification of new states which were not in the historical data, through the Non Informative Class (NIC) class.

In this work, a methodology for monitoring a vehicle online is proposed. The objective is to recognize the functional states (faults or not) online. The monitoring is based on a fuzzy classifier to estimate the GAD (Global Adequacy Degree) of a data vector (instantaneous values of the measured variables) to each class or functional state. Then, the data vector is associated with the class with the maximum GAD. The GAD may provide information that a system is in a normal state, but moving away from this class indicates the start of a fault, allowing early action to be taken.

The rest of this paper is organized as follows: Section II describes the monitoring systems and the method for acquiring and pre-treating data is explained. In the same section, the fuzzy clustering algorithm Learning Algorithm for Multivariate Data Analysis (LAMDA) is discussed and used to identify the functional states online in a vehicle. Then, the experimental setup is explained; and finally the results and discussions are analyzed and conclusions are described.

II. MONITORING SYSTEM

A monitoring system provided information in real time about the status of the process variables and location of faults [21]. Fuzzy clustering algorithms enabled monitoring, diagnosis and fault detection from *n-dimensional* analysis, independent of time [22]. Using LAMDA fuzzy clustering algorithm, the degrees of membership of a data vector to its classes is defined, providing important information for decision making in any system.

With the offline analysis of data and using the fuzzy clustering algorithm, a classifier was obtained with which it was possible to monitor vehicle operating status online. The diagram in Figure 1 corresponds to the proposed methodology used for the vehicle monitoring online.

The monitoring systems included a data acquisition phase where the critical variables of the vehicle (see Table 1) were analized. A pre-treatment of each signal was performed, and this made for each data vector in each sampling time. In an offline phase, the features of each class were identified with historical data of the vehicle and then, at an online phase, online recognition of the states of the vehicle was identified. This way was possible to early identify, the functional state in which the vehicle was located, before an incipient failure could generate a more serious fault. The following subsections explain each one of the phases.

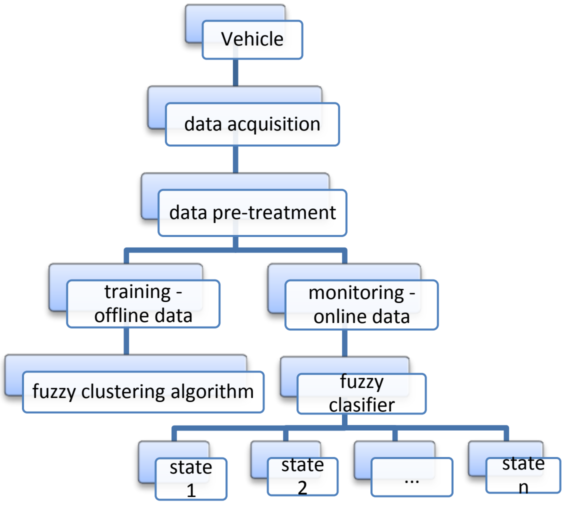


Figure 1. Methodology for the classifier.

A. *Data acquisition and pre-treatment*

Sensors required to measure the signals were installed on the vehicle. At each sampling time (one sample every 250ms), recorded values of the variables in the data vector ***x*** were analyzed.

Signals measured (shown in Table 1) were carried by the sensors to an onboard computer through the data acquisition card NI USB-6218. The monitoring and online classification of signals were performed by a Supervisory Control and Data Acquisition (SCADA), developed in Labview®, the interface of which is shown in Figure 2.

TABLE 1. SENSORS AND VARIABLES MEASURED

Measured variables	Sensor
Air flow	Mass air flow (Toyota - Denso Air flow meter 22250-45040)
Engine speed, measured in RPM (revolutions per minute)	Hall effect sensor
Butterfly valve position, indicates the percentage of accelerator opening	Throttle potenciometer
Voltage	Lambda or oxygen sensor
Vibrations from mechanical deformations	Piezoelectric accelerometer (AC150-2C Accelerometer)

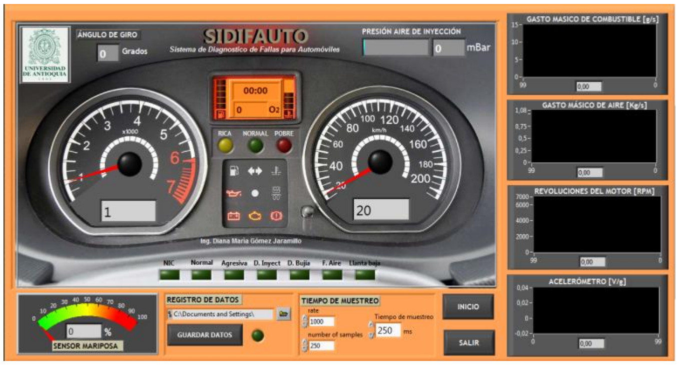


Figure 2. User interface designed in Labview®.

Data acquisition card is configured to acquire sensor data every 250ms. The pre-treatment of data was performed using a low pass Chebyshev filter of order 6 [23], which eliminates the high frequency components. Then the average of each signal is calculated by means of a sliding window taking 300 samples of the same class for each average. Each sample (individual) corresponds to a data vector $\boldsymbol{x} = [x_1, x_2, \dots, x_d]$, with (*d*=5), with the information from the five variables (descriptors) systems.

B. *Fuzzy clustering algorithm*

In order to find classes or functional states in the training stage, it is possible to use fuzzy clustering algorithms such as Fuzzy C-means (FCM) [24], GK-means (GKM) [25], or Learning Algorithm for Multivariate Data Analysis (LAMDA) [26], among others.

The LAMDA fuzzy clustering technique has been widely used for process monitoring especially for its low computational cost [27][28][29] and because it allows identifying new states which were not in the historical data through the NIC.

LAMDA fuzzy clustering algorithm was employed with the aim of finding the degrees of membership (GAD) of a data vector to a class or functional state at each instant of time. This algorithm also takes into account the contribution of each descriptor (a variable measured in the vehicle) to other classes. For the contribution of a determined sample in the time *n*, the value of Marginal Adequacy Degree (MAD) is estimated in each variable for each class [30][31].

1) Training Phase (Offline)

From historical data, at each sample time, a data vector $\boldsymbol{x} = [x_1, x_2, \dots, x_d]$ (with *d*=total number of measured variables) is obtained. These vectors are used for training.

For each sample (data vector *x*) the Marginal Adequacy Degree is calculated. To calculate the MAD, possibility functions are used; in this case, the following probabilistic function was used (1) [32].

$$MAD_{j,l} = \rho_{j,l}^{\bar{x}_{i,l}} (1 - \rho_{j,l})^{(1-\bar{x}_{i,l})} \tag{1}$$

where $\bar{\boldsymbol{x}}_{i,l}$ is the normalized value of the descriptor *l* of a particular data vector *i*, with $l = 1, \dots, d$ and $i = 1, \dots, n$. $\rho_{j,l}$ is the mean value for the *j* class and the descriptor *l*, with $j = 1, \dots, k$; this parameter was calculated using the historical samples belonging to each class (see (2)) [33].

$$\rho_{j,l} = \frac{1}{T_j} \sum_{i=1}^{i=T_j} \bar{x}_{i,l} \tag{2}$$

where T_j is the total number of historical individuals belonging to class *j*. As each of the historical data is analyzed, the value of $\rho_{j,l}$ is updated using an estimate of the moving average of the data for each descriptor in each class.

The membership from a data vector to a *j* class is estimated with the Global Adequacy Degree. The GAD_j for a *j* class is computed from the $MAD_{j,l}$ (see (3)). This interpolation is performed between the fuzzy operators *T-norm* (in this case *MIN*), which corresponds to a logical intersection operation, and *S-norm* (in this case *MAX*), which corresponds to a logical union operation. The Exigence Index α is a value between 0 and 1 that indicates the exigence with which an individual is attributed to a class; the closer this parameter to 1, the more demanding the classification.

$$GAD_j = \left(MAD_{j,1}, \dots, MAD_{j,d}\right) =$$
$$\alpha T\left(MAD_{j,1}, \dots, MAD_{j,d}\right) + (1-\alpha) S\left(MAD_{j,1}, \dots, MAD_{j,d}\right) \tag{3}$$

In the trained classifier, a data vector is associated to a class if the maximum GAD calculated corresponds to that class.

Each class represents a functional state (i.e., normal driving behaviour, aggressive driving or a type of mechanical or electrical failure).

2) Online monitoring

While the vehicle was in motion, current status (failure or not) could be identified using the trained classifier.

The online monitoring consisted of the *GAD*$_j$ with $j = 1, \dots, k$ calculation at each sample time.

This way, at each sample time, the monitoring system estimates the membership to each class. The vehicle behavior online was classified in the class (functional state) with the maximum GAD value (for example, normal state) and the other GADs were useful to identify if there was a movement away from this class which indicated the start of a fault, allowing early action to be taken. If the vehicle had a failure, the data vector was classified into the class associated with this failure.

To prevent a misclassification, when a new state (not included in the training phase) is present, a Non Identification Class (NIC class) is included. For this class, the average value for all descriptors ($l = 1, \dots, d$) is $\rho_{NIC,l} = 0.5$, and the MADs and GAD$_{\mathrm{NIC}}$ values are estimated with the equations 1 and 3 respectively. The NIC automatically defines a threshold for classifying a data vector into the defined classes. Then, the behaviors that are not associated with any of the defined classes are classified into the NIC class.

III. EXPERIMENTAL SETUP

The failures to be identified in the proposed monitoring system were chosen as reported in Section I. The vehicle condition under normal and aggressive driving conditions was also taken into account, in such a way that aggressive driving was not confused with a failed state.

To build the database, the test protocol consisting of a distance of 1,700m was established. The time of a round trip was about 5min. For each state, 2 to 3 replicates were made. Each repetition consisted of a complete tour of the 1,700m. As a conditions of the terrain, the route was a paved runway with ridgesand slopes in some areas. Each failure was caused before starting the tour.

From historical data, 7,666 samples were obtained, where each functional state has approximately 1,100-1,300 data. By applying the pre-treatment, explained in Section II, samples were reduced to 5,866, of wich 70% (4,016 samples) were used for the training phase, each sample corresponded to a vector with the 5 variables described in Section II (see Table 1). The 4,016 data vectors $\boldsymbol{x} = [x_1, x_2, \dots, x_d]$ with d=5 were classified with a fuzzy clustering algorithm.

The classifier was obtained using the LAMDA fuzzy clustering algorithm with an exigency index α =0.5. Each class had an associated state. The classes considered in the case study are described in Table 2.

TABLE 2. DESCRIPTION OF CLASSES

Abbreviation of Classes	Class description
C1	Normal state vehicle – normal driving
C2	Normal state vehicle – aggressive driving
C3	Disconnect injector cylinder 1
C4	Disconnect spark plug cylinder 1
C5	Clogging of the air filter
C6	Lower rim
C7	NIC

Each class differs from the others according to a profile that characterizes it (see Figure 3), where each bar corresponds to the value of ρ (mean value of the descriptor in each class).

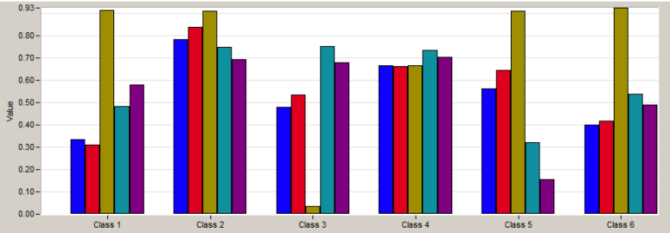


Figure 3. Profile classes

After the training phase, the vehicle was analyzed online using the same testing protocol implemented for the training phase, but only one full tour for each functional state was carried out. 1,760 samples were analyzed in the on-line phase. At each sample time (each 250ms), the data vector was analyzed by the monitoring system and the different functional states were induced and identified.

IV. RESULTS AND DISCUSSIONS

Figure 4 shows the classification obtained with the training data and the verification of the 6 classes corresponding to each functional state of the vehicle and the NIC class, as established in Table 2.

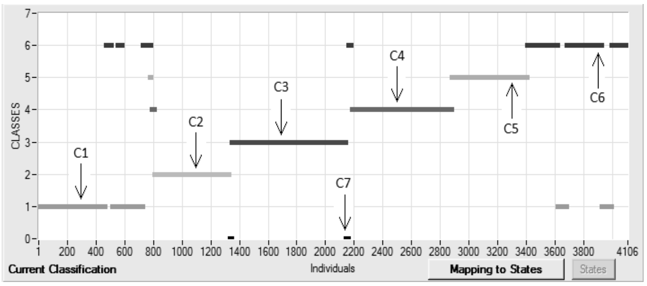


Figure 4. Classifier training data.

The X axis of Figure 4 indicates the number of individuals or samples used in classification and the Y axis shows the 7 classes identified. The graph shows some small groups of samples, different from those grouped in the NIC, which do not correspond to the class in which they were classified.

This occurs because data start to be acquired when the vehicle is idling, i.e., the engine is running but the vehicle is not moving; therefore, the first samples of classes 4, 5 and 6 are confused with other states. On the other hand, class 1 (Normal state vehicle-normal driving) and class 6 (lower rim) tend to merge because of the pressure the tyre loses when extracting the air to simulate the failure; it was not enough to ensure that system was fully differentiated in these two states of the vehicle.

The percentage of correctly classified individuals was 92.45% for the 4,016 data used in the training stage (see Figure 3).

Once the classifier was trained, the next step was monitoring performed when the vehicle was in motion to observe if the class that registers the SCADA system matches the functional state in which the vehicle was operating. At each sampling time, the recorded data vector was analyzed and the functional status that occurred in the vehicle was calculated. The graphical interface indicates the user, online, the current functional state of the system via a flashing light (see Figure 5), since this testing was performed under standard conditions and different faults were generated.

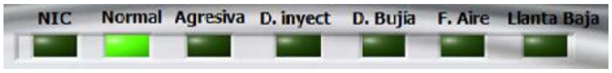


Figure 5. Functional state in which the vehicle is operating.

For the online phase, the percentage of correctly classified individuals was 92.52% for the 1,760 data (see Figure 6).

When comparing Figures 3 and 4, it can be seen that small test groups were not grouped in the class they really belonged to; this corresponds to situations similar to those already dealt with by the training data.

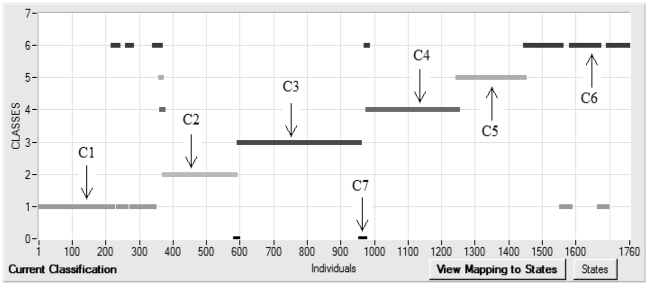


Figure 6. Online data classification.

In the online classification, the functional state of the system was correctly identified. Also, when entering new data, different from those recorded in the historical classification, the fuzzy classification algorithm groups them into the NIC class, generating in this way, a new class that had not been considered during training. This case identified a fault which occurred that was not included in the historical classification, causing the interface to indicate that the system was in the NIC class. The vehicle was then inspected at an Automotive Diagnostic Center and an electrical failure was diagnosed.

The LAMDA fuzzy clustering algorithm allows online identification of different states or classes through the GAD, which provides information about the possible change from a normal state to a failure state, allowing early action to be taken. This is possible by identifying the class to which an individual belongs (defined by the highest degree of membership) and the class that this individual could migrate to, by knowing the next lowest GAD and its associated class.

Table 4 shows the degree of membership associated with each of the two sample types, and it can be observed that while in a normal driving state, the sample has a degree of membership to an aggressive driving or a failed class.

TABLE 4. MEMBERSHIP DEGREES ASSOCIATED TO EACH CLASS

Sample	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
1070	0.5013	**0.5719**	0.4846	**0.5414**	0.5003	0.5089
1778	0.4702	0.4954	**0.5549**	0.5169	**0.4400**	0.4724

Sample 1,070, for example, has a degree of membership to class 2 (Aggressive driving) of 0.5719, while for class 4 (disconnect spark plug) the degree of membership is 0.5414 and in other classes the degree of membership is lower compared to the two previous classes. This way, it is established that the 1,070 samples belong to class 2 because their degree of membership to this class is higher than to the others. Therefore, if the system is in class 2, is more likely to go to class 4 than any of the other classes.

In the 1,778 sample, the highest degree of membership is to class 3 (Disconnect injector cylinder 1), while the degree of membership immediately below corresponds to class 4 (Disconnect spark plug) and the lowest of all the membership degrees corresponds to class 5 (Clogging of the air filter). This indicates that if the system is in class 3, it is more likely to change to class 4, and is less likely to change to class 5.

Moreover, this proposal has the advantage that the variables analyzed are easily accessible, since it is not necessary to open the ECU (Electronic Control Unit); this allows analysis of more system components apart from the engine. Additionally, the results were obtained with a low computational cost (to identify the situation in a sample, instant calculation requires no more than a few milliseconds). The data processing is performed on a laptop with Intel Core 2 Duo of 2 GHz and 4 GB of RAM, located at the front of the car. The analysis and classification of a data vector is executed in a much shorter time compared with the sampling interval (250ms).

The system correctly identifies new data that enters the algorithm and classifies correctly. Through the graphical interface shown in Figure 2, a flashing light warns the driver of the vehicle about the type of fault that the system has, so that a driver can check the type of fault identified and contact an Automotive Diagnostic Center.

The developed system is useful for Renault vehicle, if you want to replicate the experiment in a different system, the classifier must be trained again.

V. Conclusion and future work

A useful system is proposed for online monitoring of a vehicle, using a LAMDA fuzzy clustering algorithm. A warning light advises the driver by a graphical interface about of the functional state of the vehicle, thanks to the online monitoring of the variables.

LAMDA fuzzy classifier provides information about how the system evolves, enabling identification of the current status of the vehicle and the possibility of migration to another state, fault or not, based on the degree of membership associated with each class.

LAMDA fuzzy clustering has a low computational cost and allows the identification of new classes that were not in the historical data, through the NIC class. Additionally, the algorithm achieves a correct functional state identification, in front of other techniques.

In the future, this algorithm will allow the inclusion of unforeseen situations, as it defines all kind of degrees of membership, including the NIC, and from this, it is possible to identify that there is not a high degree of membership to situations registered in the historical data. Possible future work would be to predict these states using prior knowledge of the degrees of membership obtained with the LAMDA algorithm.

Acknowledgment

We thank the group of Efficient Management of Energy-GIMEL of Universidad de Antioquia, for allowing us to perform the test in the Renault LOGAN vehicle intended for academic projects.

This work was supported by CODI at Universidad de Antioquia.

References

[1] R. Sarrate, J. Aguilar-Martin, and F. Nejjari, “Event-based process monitoring,” Engineering Applications of Artificial Intelligence, 20, 2007, pp. 1152-1162.

[2] S. M. Namburu, S. Chigusa, D. Prokhorov, L. Quiao, K. Choi, and K. Pattipati, “Application of an effective data-driven approach to real-time fault diagnosis in automotive engines,” Aerospace Conference IEEE. Big Sky, MT, 2007, pp. 1-9.

[3] Y. Hyun-Jeong, L. Shin-Kyung, and K. Oh-Cheon, “Vehicle-generated data exchange protocol for remote OBD inspection and maintenance,” 6th International Computer Sciences and Convergence Information Technology (ICCIT 2011). Seogwipo, South Korea, 2011, pp. 81-84.

[4] M. Nyberg and T. Stutte, “Model based diagnosis of the air path of an automotive diesel engine,” Control Engineering Practice, 12, 2004, pp. 513-525.

[5] J. Mohammadpour, M. Franchek, and K. Grigoriadis, “A survey on diagnostics methods for automotive engines,” International Journal of Engine Research, vol. 13, 2011, pp. 41-64.

[6] N. Hasan, A. Arif, U. Pervez, M. Hassam, and S. Shabeeh, “Microcontroller based on board diagnostic (OBD) system for non-OBD vehicles,” 2011 UKSim 13th International Conference on Modelling and Simulation. Cambridge, 2011, pp. 540-544.

[7] Y. He and L. Feng, “Diesel fuel injection system faults diagnosis based on fuzzy injection pressure pattern recognition,”

5th World Congress on Intelligent Control and Automation, vol. 2, 2004, pp. 1654-1657.

[8] A. Schilling, “Model-based detection and isolation of faults in the air and fuel paths of common-rail DI diesel engines equiped with a lambda and a nitrogen oxides sensor,” ETH Zurich: Zurich, Switzerland, 2008.

[9] A. Schilling, A. Amstutz, and L. Guzzella, “Model based detection and isolation of faults due to ageing in the air and fuel paths of common rail direct injection diesel engines equipped with a lambda and a nitrogen oxides sensor,” Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2008, pp. 101-117.

[10] T. Lefebvre, H. Bruyninckx and J. De Shutter, “Kalman filters for non-linear systems: a comparison of performance,” International journal of control, vol. 77, 2004.

[11] E. Gani and C. Manzie, “Misfire-misfuel classification using support vector machines,” Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2007, pp. 1183-1195.

[12] X. Yong, G. Feng, H. Guiyou, N. Zhibing, and H. Xinhan, “Application the wavelet packet and image processing to fault diagnosis for I.C. engines,” Chinese control and decision conference. Xuzhou, 2010, pp. 1223-1228.

[13] R. Moreno, “Detección y clasificación de fallos incipientes en engranajes y rodamientos”. Universidad Castilla, La Mancha, Spain, 2010.

[14] G. Wenqiang, Z. Zoe, and H. Yongyan, “A novel fault diagnosis for vehicles based on time-varied bayesian network modeling,” Chinese control and decision conference (CCDC 2011), Mianyang, 2011, pp. 1504-1508.

[15] N. Friedman and M. Goldszmidt, “Building classifiers using Bayesian networks,” AAAI-96 Proceedings, 1996, pp. 1247-1284.

[16] D. Goldgerg and J. Holland, “Genetic Algorithms and Machine Learning,” Machine learning, vol. 3, 1988, pp. 95-99.

[17] Y. Wang, Q. Ma, Q. Zhu, X. Liu, and L. Zhao, “An intelligent approach for engine fault diagnosis based on Hilbert-Huang transform and support vector machine,” Applied Acoustics, vol. 75, 2014, pp. 1-9.

[18] J. Waissman, B. Cherif, and G. Vázquez, “Fuzzy automata identification based on knowledge discovery in datasets for supervision of a WWT process,” 3rd International Conference: Sciences of electronic, technologies of information and telecommunications (SETIT 2005), Tunisia, 2005.

[19] J. Aguilar-Martín and R. Lopez De Mantaras, “The process of classification and learning the meaning of linguistic descriptors of concepts,” Approximate reasoning in decision analysis. North Holland, 1982, pp. 165-175.

[20] K. Jyoti and S. Singh, “Data clustering approach to industrial process monitoring, fault detection and isolation,” International Journal of Computer Applications, vol. 17, No 2, 2011, pp. 41-45.

[21] F. Ly, A. Toguyeni, and E. Craye, “Indirect predictive monitoring in flexible manufacturing systems,” Robotics and Computer-integrated Manufacturing, vol. 16, No5, 2000, pp. 321-338.

[22] L. Hedjazi, T. Kempowsky-Hammon, L. Despènes, M-V Le, S. Elgue, and J. Aguila-Martin, “Sensor placement and fault detection using an efficient fuzzy feature selection approach,” 49th IEEE Conference on Decision and Control, Atlanta, USA, 2010.

[23] A. Katiyar and M. Katiyar, “Design of Butterworth and Chebyshev lowpass filter for equalized group delay,” International journal of advanced research in computer science and software engineering, vol. 2, 2012, pp. 524-528.

[24] J. C. Bezdek, “Pattern recognition with fuzzy objective function algorithms,” Plenum Publishing Corporation, New York, USA, 1981.

[25] D. E. Gustafson and W.C. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” Proceedings of IEEE Conference on Decision and Control, 17, 1978, pp. 761 -766.

[26] J. Aguilar-Martín, “Knowledge-based supervision and diagnosis of complex process,” IEEE International Symposium of Intelligent Control, Intelligent Systems and Semiotics, Cambridge, USA, 1999, pp. 225-230.

[27] C. Uribe and C. Isaza, “Unsupervised feature selection based on fuzzy partition optimization for industrial processes monitoring,” IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA), Ottawa, 2011, pp. 1-5.

[28] J. Botia, C. Isaza, T. Kempowsky-Hammon, M-V. Le, and J. Aguilar-Martin, “Automaton based on fuzzy clustering methods for monitoring industrial processes,” Engineering Applications of Artificial Intelligence, 26, 2013, pp. 1211-1220.

[29] H. O. Sarmiento, “Metodología para el establecimiento y ponderación automática de conexiones entre estados funcionales de un proceso como herramienta para el diagnóstico y predicción de fallos”, Universidad de Antioquia, Colombia, 2013.

[30] T. Kempowsky-Hammon, “Surveillance de procédés à base de méthodes de classification: conception d'un outil d'aide pour la détection et le diagnostic des défaillances, ” Toulouse, France, 2004.

[31] C. V. Isaza, “Diagnostic par techniques d´apprentissage floues: conception d´une methode de validation et d´pptimissation des partitions, ” Toulouse, France, 2007.

[32] T. Kempowsky-Hammon, A. Subias and J. Aguilar-Martin, “Process situation assessment: from a fuzzy partition to a finite state machine,” Engineering Applications of Artificial Intelligence, 19, 2006, pp. 461–477.

[33] T. Kempowsky-Hammon, J. Aguilar-Martin, A. Subias, and M-V Le, “Learning methodology for a supervision system using LAMDA classification method,” IBERAMIA´02 – VII Iberoamerican Conference on Artificial Intelligence, Sevilla, Spain, 2002.

Verification of Transport Protocol's Parallel Routing of a Vehicle Gateway System

Hassan Mohammad and Piyush Patil
Body & EE
MBtech Group GmbH & Co. KGaA
Sindelfingen, Germany
Hassan.mohammad@mbtech-group.com
Piyush.patil@mbtech-group.com

Abstract—Transport protocol (TP) routing is a routing functionality of a vehicle gateway system enabling TP data to be transferred to different types of connected networks. This routing functionality is required for intra- as well as inter-vehicular communications such as flashing new software onto Electric Control Units (ECUs) and collecting status information (large data packets) and routing them for diagnostic purposes. Transport protocol's parallel routing is the scenario of establishing multiple TP routing instances in parallel in order to save time and resources. This article addresses the issue of verifying parallel routing of TP for a gateway system. To achieve this goal, a new recursive test case selection and generation strategy along with a suitable input parameter model is introduced. The new strategy is a combination test strategy guided by the category partition method to overcome the combinatorial explosion problem raised by testing of TP parallel routing functionality. Additionally, the new strategy enables user to analyze the performance of the gateway related to TP parallel routing. This is achieved by providing statistical information in diverse scenarios and determining the maximum number of guaranteed TP parallel routing instances. The statistical information can help in optimizing the configuration or the implementation of the transport protocol by providing hints about error causes.

Keywords- Parallel Routing of Transport Protocols, Combination Test Strategies, Input Parameter Models (IPM).

I. INTRODUCTION.

Today's vehicles Electric/Electronic (E/E) system is designed as a distributed system in order to overcome the increasing complexity and meet the diversity of requirements such as performance, comfort, safety and costs. In a vehicle distributed system, gateways are indispensable. They enable Electric Control Units (ECUs) within connected networks to interchange information necessary for accomplishing specified functionalities. A modern E/E system has multiple gateways, e.g., central gateway, telematic gateway, etc. During information interchange, the gateway routes data between its connected networks although they work on different communication protocols.

Mainly, two types of data routing can be established over the gateway. The first type is frame routing and concerns with routing of data that fit into one frame. This kind of routing is simple and out the focus of the article. The second type is TP routing and concerns with routing of data packets which do not fit into one frame. This routing functionality is required for intra-vehicular communications such as flashing new software onto Electric Control Units (ECUs), variant coding, and software update or even reading vehicle status. For such use cases, an external device "Tester" is connected via an external interface "OBD-connector" to the central gateway in order to access and communicate with the ECUs in the network. In inter-vehicular communication, TP routing is also required when large data packets need to be exchanged between the ECUs. For this type of routing, the gateway utilizes transport protocols, such as CAN transport protocol [1] and FlexRay transport protocol [2], which provide features for segmentation, reassembling, flow control and error detection. TP parallel routing is the scenario of routing TP data between multiple communicating ECUs located on different networks in parallel in order to save time and resources, as for example the case of flashing multiple ECUs in parallel.

Verifying TP parallel routing of a gateway system is not a trivial problem, since a huge number of possible combinations of communicating ECUs can be built when test cases have to be selected, especially if the combination of communicating ECUs demands transport protocol change, i.e., converting one transport protocol to another, when different types of protocols are involved during the routing process.

This article presents a new recursive test case selection and generation strategy along with a suitable input parameter model (IPM) to overcome the combinatorial explosion problem raised by testing TP parallel routing on a vehicle gateway system while still having a very good thoroughness of the test.

The remainder of this article is organized as follows. Section II describes the combinatorial explosion problem of TP parallel routing test, its differentiation from described problem in literature and techniques existed to fight it. The proposed approach is presented in details in Section III. Section IV discusses the approach and Section V concludes the article.

II. Background and Related Work

A. The Combinatorial Explosion Problem of TP Parallel Routing Test

The gateway is part of a distributed network system. It communicates with its environment over communication channels via buses. Communication channels of a gateway system, e.g., CAN or FlexRay, are mostly heterogeneous, i.e., each has its own characteristics and behavior. A group of up to *c* channels shall be defined as the Communication Channels of a gateway system (1).

$$Communication\ Channels = (Ch_1, Ch_2, ..., Ch_c) \qquad (1)$$

The environment consists of multiple ECUs, e.g., engine Control Module, interacting with the gateway and among each other. Each ECU is located on a specific channel and utilizes it to establish the required communication. If an ECU connected on a channel of the gateway needs to communicate with another ECU located on another channel, then the gateway establishes a routing process between the two ECUs. It receives the data from sending ECU on the source channel and routes it to the receiving ECU on the destination channel. The environment of the gateway shall be described as a group of *e* ECUs connected to Communication Channels (2).

$$Environment = (ECU_1, ECU_2, ..., ECU_e) \qquad (2)$$

$$e >= c \qquad (3)$$

Generally, ECUs in the environment exchange data over the gateway in predefined Fashions. Each Fashion is characterized through a set of configuration parameters which are required to establish TP routing relationships between communicating ECUs. A Fashion *F* shall be described as a set of configuration parameters *P* (4). (see [1] for configuration parameter of CAN TP).

$$Fashion_F = (P_{F_1}, P_{F_2}, ..., P_{F_p}) \qquad (4)$$

As described before, ECUs in the environment communicate over the gateway in diverse Fashions. Fashions are a superset of Scenarios, where one Scenario is constructed with the set of parameters for a Fashion. Scenarios are related to ECUs, i.e., some ECUs could communicate only in some Scenarios. All possible Scenarios shall be described as a group of *s* Scenarios (5).

$$Scenario = (Scenario_1, Scenario_2, ..., Scenario_s) \qquad (5)$$

A Connection Channel is an instance of a Fashion and it has the same set of parameters defined for that Fashion. A Connection Channel in the Fashion *F* shall be described (6).

$$Connection\_Channel_x = (P_{F_{1x}}, P_{F_{2x}}, ..., P_{F_{px}}) \qquad (6)$$

A Routing Instance is a relationship between a specific Connection Channel and a possible Scenario and shall be described (7).

$$Routing\_Instance_x = (P_{F_{1x}}, P_{F_{2x}}, ..., P_{F_{px}}, Scenario_x) \qquad (7)$$

The gateway can be configured to serve *y* Routing Instances in parallel. The number *y* of parallel Routing Instances is a configuration parameter which needs to be verified. In the case of errors, the next verified *y* should be determined. To verify parallel routing of *y* instances, the combinatorial explosion problem arise. By considering all variables described before and assuming that all ECUs communicate with the same number of Scenarios; the theoretical number of combinations for routing instances can be calculated (8).

$$X = \left(\frac{e!}{y!(e-y)!}\right).y^s + \left(\frac{e!}{(y-1)!(e-(y-1))!}\right).(y-1)^s + ... + \left(\frac{e!}{1!(e-1)!}\right).1 \qquad (8)$$

Let us take a simple example and assume that the E/E system has 50 ECUs, the gateway is configured to support only 3 Routing Instances in parallel and each Connection Channel can be mapped to only 3 Scenarios. The number of theoretical combinations in this example is 5390050. Assume that to test each possible combination 10 seconds are needed, testing the theoretical number of combinations will require 62.39 days. This duration is not acceptable.

B. Combination Testing Strategies

To overcome the combinatorial explosion problem of testing distributed systems that consist of a number of interacting elements, combination testing strategies have been devised in literature [3]-[11]. A chronological overview with a comparison of diverse strategies can be found in [12]. Combination testing strategies are category partition [13] based methods that supports the finding of a trade off between test coverage and available resources by providing techniques for selecting combinations of parameter.

The combinatorial explosion problem mentioned in literature shall be explained as in the following example:

Assume a distributed system consisting of a central unit interacting over communication channels with *n* units of the environment u_1, u_2,..., u_n. Each unit u_i in the environment uses a defined parameter p_i for communication. The parameter p_i shall have v_i possible configuration values. By assuming that configuration values of parameters are independent from each other, the number of configuration possibilities of the system would be $v_1 \times v_2 \times ... \times v_n$. If each possible configuration requires *c* test cases to verify it, the number of test cases for exhaustive test would be $c \times v_1 \times v_2 \times ... \times v_n$. In a nontrivial software system, the values of n and v_i are large which leads to a huge number of possible combinations of parameter values.

In order to find a trade off between test thoroughness and test resources, combination test strategies define coverage criteria needed to be satisfied. Coverage criteria can be varied between *1-wise* to *N-wise*.

1-wise coverage criterion requires that, each interesting value of each parameter must be included at least in a test case to reach 100% coverage. Whereas, *N-wise* coverage or exhaustive test requires that all possible combinations of interesting values must be included in generated test cases. Decision on which coverage criterion to be used depends on several factors, such as the effort required to construct each test case, time, resources and budget. Studies on reported bugs and failures [14][15] have shown that *2-wise* or *pair-wise* combinations are very effective in finding failures of parameter interaction. However, as shown in [16], the quality of *2-wise* combination testing is affected strongly by diverse factors which can be only partially influenced by the technique.

Related to testing TP parallel routing, the goal of test is to measure the performance of the gateway to handle multiple parallel Routing Instances. The problem is more complex because:

- n is not a constant. It is a configuration parameter which can be different for every new release of the system.
- Each unit of the environment can have multiple sets of configuration parameters that can be used in defined scenarios to establish a communication.
- Sets of configuration parameters also include timing parameters and the interactions between different values of timing parameters are difficult to resolve.
- The number y of parallel instances, which is also a configuration parameter, can be any subset of n. In case of errors, one of the test goals is to find the next verified y.
- In TP parallel routing, each additional instance will consume resources of the system and may lead to errors. Hence, it is not only a specific combination of input parameter values which can affect the behavior and may reveal errors, but also the number of included input parameter sets and their values.

Based on these factors and other uncontrolled factors mentioned in [16], the proposed combination testing strategies revealed in literature are not suitable for fighting the combinatorial explosion problem in our case.

C. Related Work

During the literature research, no combination testing technique was found that is designed to support testing parallelism of applications. Only systems accepting a fixed number of input parameters and techniques to solve the problem of handling combinations of interesting values for those parameters have been discussed in literature.

To solve the combinatorial explosion problem raised by testing TP parallel routing, a new recursive test case selection and generation approach is proposed. The approach is based on the category partition method and utilizes *N-wise* coverage criterion. A suitable IPM [17]-[20] is also defined and serves as an input for the test case generation strategy.

Recursively generation and running of combinatorial test cases gives the ability to analyze the results from executed test cases and collect symptoms. Information gained can help deciding the next parameter sets to combine.

Although the proposed approach is guided by the category partition method, it differs from it in diverse aspects. One aspect is that the proposed approach deals with testing of parallelism, which is described through combinations of instances. Another aspect is the definition of interesting parameter values in category partition, which is different in the proposed approach.

An important difference to existing combination test strategies is the usage of semantic information in a recursive approach. Semantic information is not part of existing combination test strategies in their basic form. It is utilized here to build an IPM and to guide the selection of combinations which can reduce the test suite size.

III. Proposed Strategy for Test Case Selection and Generation

In this section, steps for the test case selection and generation methodology are discussed. The methodology shall select and generate test cases to test the gateway system for its user test requirements confined to TP parallel routing with an efficient number of test cases. The methodology is categorized in 5 steps as depicted in Fig. 1.

1. Determining existing scenarios and defining constraints.

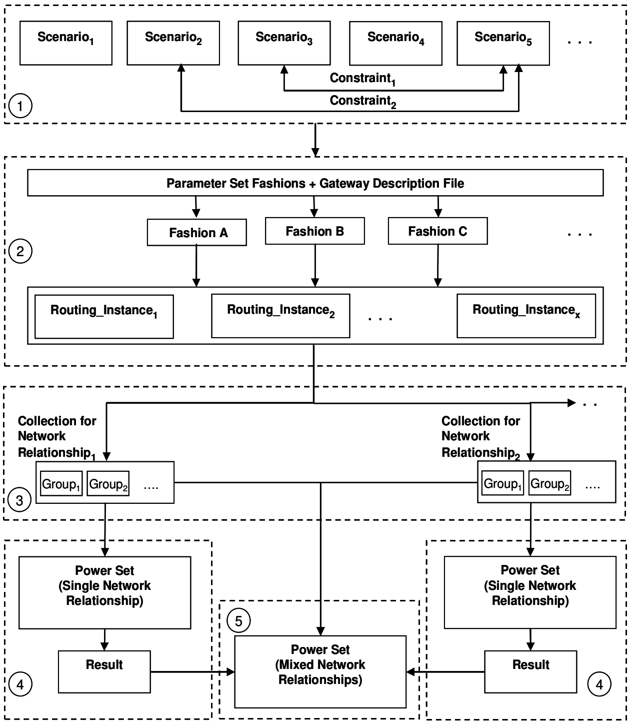


Figure 1. Test Case Selection and Generation Steps.

2. Mapping TP Scenarios onto Connection Channels in order to construct Routing Instances.
3. Collection and completion based on similarity criteria.
4. Testing TP parallel routing for Single Network Relationships (SNRs).
5. Testing TP parallel routing for Mixed Network Relationships (MNRs).

The first 3 steps are concerned with constructing an IPM and the following steps with the new combination testing strategy.

A SNR mentioned in step 3 is an abstract term, which describes all Routing Instances between two specific networks of the gateway.

A MNR in step 5 comprises Routing Instances from different SNRs.

A. Determining Existing Scenarios and Defining Constraints

In this step, TP routing scenarios are discussed and analyzed with persons practicing TP functionalities. At the end of this step, real use case scenarios are defined and described in a selectable format. Several advantages are gained from this step:

- Real use case scenarios are mostly not described and can not be recognized or built automatically from gateway description file.
- Analyzing can help in avoiding scenarios which are not practiced but theoretically conceivable.
- Future extensions for scenarios can be discussed and defined.

Along with determining existing scenarios, constraints can also be defined.

- Constraints for combinable or non-combinable Scenarios.
- Constraints for combinable or non-combinable Connection Channels.
- Configuration Constraints, e.g., maximum configured parallel routing instances.

The user has the ability to construct *preventing* or *allowing* Constraints. *Preventing* Constraints are responsible for preventing specific combinations to be constructed and included in a test case. Whereas, *allowing* constraints define conditions used to build certain combinations.

The decision on using *preventing* or *allowing* constraints depends on the case study. If the number of *allowing* constraints is bigger than the number of *preventing* constraints, then it is better to use *preventing* constraints for selection in order to reduce manual effort. Constraints are crucial for combination selection. They can reduce the number of combinations to a large extend. Defined constraints shall be described in a suitable format.

Examples for *preventing* constraints between scenarios:

$$Scenario_2 + Scenario_5 = NC \tag{9}$$

$$Scenario_3 + Scenario_5 = NC \tag{10}$$

NC: Not Combinable.

Examples for *preventing* constraints between Connection Channels:

$$\begin{aligned} &Connection\_Channel_x + \\ &Connection\_Channel_z = NC \end{aligned} \tag{11}$$

Example for configuration constraints:

$$Maximum\_Parallel\_Instances\_CAN\_TP = y \tag{12}$$

B. Mapping TP Scenarios onto Connection Channels in Order to Construct Routing Instances

The gateway is described on a certain abstraction level by means of a description file. ECUs communicating with the gateway have parameters defined in the description file. These parameters define the behavior of ECUs from the gateway point of view. If an ECU communicates using the transport protocols in a specific scenario, a related set of configuration parameters called *Connection_Channel* are utilized.

Mapping TP Scenarios onto Connection Channels is a step in which TP parameters of defined Connection Channels are extracted and then mapped to TP Scenarios. As a result of this step, each Connection Channel included in the gateway description file must be related to minimum one specific TP Scenario. Resulted relationships are called Routing Instances. Examples of mapping can be formulated as in the following (13) (14).

$$Routing\_Instance_A = (P_{F_{11}}, P_{F_{21}}, \ldots, P_{F_{p1}}, Scenario_x) \tag{13}$$

$$Routing\_Instance_B = (P_{F_{12}}, P_{F_{22}}, \ldots, P_{F_{p2}}, Scenario_x) \tag{14}$$

C. Collection and Completion Based on Similarity Criteria

The goal of collection is to cluster similar Routing Instances which stimulate the same or similar behavior in the gateway when TP routing is established. In the proposed approach, collection is performed in two steps as depicted in Fig. 2.

- Creating groups out of Routing Instances. Routing Instances of each created group must have the same values for all related parameters such as routing parameters, network relationships and mapped scenarios.
- Creating SNR collections out of constructed groups. Groups of a SNR collection must have the same network relationships, i.e., the same source and destination networks for all of their Routing Instances. SNR collections are the base for TP parallel routing test of single network relationships.

Collection process is part of designing the IPM and helps in reducing the number of combinations required for test.

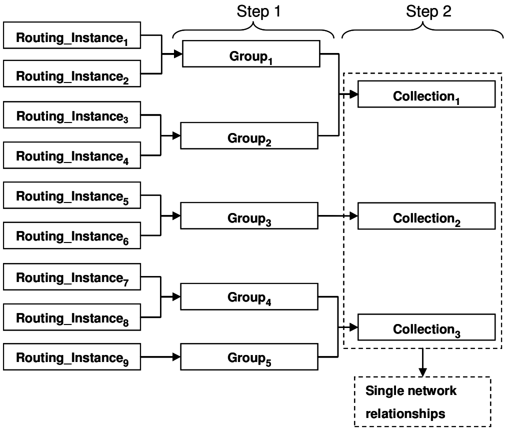


Figure 2. Steps of Collection

The following example explains reduction achieved when groups are constructed:

Assume that 4 Routing Instances A, B, C and D are constructed and the gateway is configured to support 2 Routing Instances in parallel. The number of possible combinations of 2 Routing Instances out of 4 would be 6 (the order has no effect). Groups can be constructed based on similarity criteria such that *Group$_1$* consists of instance A and instance B, *Group$_2$* consists of instances C and D. After grouping, the number of combinations could be rather reduced to 3, because all other possible combinations would resemble the same effects on the gateway, i.e., combinations of instances (A, C), (A, D), (B, C) and (B, D) are all similar and can be replaced by only one representative as in example (A, C). (see Fig. 3).

In completion, Similarity Numbers and Stress Factors are assigned to constructed groups. The same Similarity Number will be assigned to groups if their Routing Instances have the same routing parameters, the same scenarios and the same characteristics for network relationships. Concerned characteristics are the protocol type and channel bandwidth.

The Stress Factor is calculated initially based on aspects such as the expected processing time, memory usage, network bandwidth and other network specific aspects. Stress Factor shall be also adjusted during the test run based on variance. Table 1 represents an output example of the grouping and completion process.

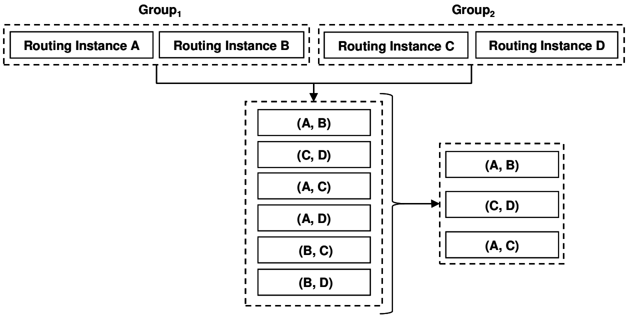


Figure 3. Advantages of Building Groups

Similarity Numbers and Stress Factors are required for the combination selection during TP parallel routing test for MNRs.

The number of formed groups in step 1 of collection depends on the complexity of the gateway under test with respect to connected networks, their heterogeneous level, the number of configured Connection Channels and the heterogeneous level of their parameters.

Grouping process has several benefits in addition to reducing the number of combinations required for testing:

- It assists in analyzing the parallel routing behavior of the gateway under diverse combinations of configuration parameters or combinations of network relationships.
- It can facilitate the search for error causes by enabling the user to compare routing results under particular scenarios, and gain information about the relationship between specific attributes of the included Routing Instances and raised errors.

D. Testing TP Parallel Routing for Single Network Relationships (SNRs)

The proposed test case selection and generation approach is a recursive technique consisting of two main test phases. The first phase deals with TP parallel routing test for SNRs by means of constructed SNR Collections. The second phase deals with TP parallel routing test for MNRs based on Similarity Numbers and Stress Factors. This separation is very practical for analyzing the behavior of the gateway and can provide information about possible reasons for errors if they can be revealed.

In the first phase, formed SNR Collections are picked up successively. For each SNR Collection, a power set of its groups shall be constructed. Power set is the set of all subsets of input elements without the empty set, and serves as a medium to check if the coverage criteria can be completely achieved. Subsets are then categorized into levels, where each level consists of all subsets with the same number of elements (see Fig. 4 as an example of a SNR Collection with 3 groups). Subsets on a specific level substitute implicitly all subsets on the successive levels. This feature shall be used to

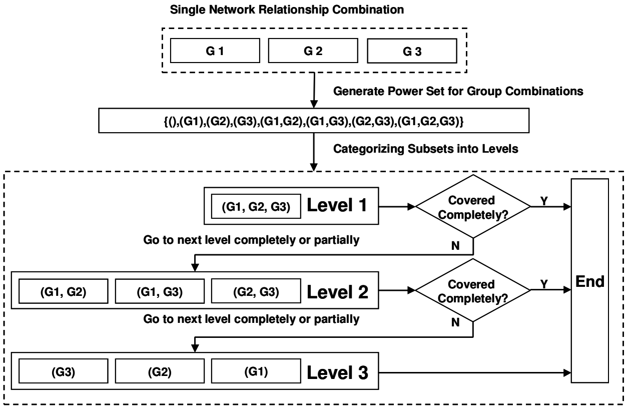

Single Network Relationship Combination
G 1
G 2
G 3
Generate Power Set for Group Combinations
{(),(G1),(G2),(G3),(G1,G2),(G1,G3),(G2,G3),(G1,G2,G3)}
Categorizing Subsets into Levels
(G1, G2, G3)
Level 1
Covered Completely?
Y
N
Go to next level completely or partially
(G1, G2)
(G1, G3)
(G2, G3)
Level 2
Covered Completely?
Y
N
Go to next level completely or partially
(G3)
(G2)
(G1)
Level 3
End

Figure 4. Checking the Combination Coverage with the Help of Power Set

TABLE I. OUTPUT EXAMPLE OF FIRST GROUPING STEP

	Parameter$_1$	Parameter$_2$	Parameter$_3$	Parameter$_4$	Scenario	Similarity number	Stress Factor
Group$_1$ (n$_1$ Routing Instances)	1,2	Any	Any	-	Scenario 1	1	3
Group$_2$ (n$_2$ Routing Instances)	1,2	Any	Any	-	Scenario 1	2	1
Group$_3$ (n$_3$ Routing Instances)	1,2	Any	Any	-	Scenario 1	3	2
Group$_4$ (n$_4$ Routing Instances)	1,2	Any	Any	-	Scenario 1	4	4
Group$_5$ (n$_5$ Routing Instances)	1,3	Any	Any	Any	Scenario 2	1	3

reduce the combinations in the successive levels when test cases for all combinations on a specific level can be built.

Levels shall be handled using a top to bottom processing strategy. Since combinations on one level comprise implicitly all combinations on successive levels, the capability of generating test cases for combinations on a specific level would be sufficient to finish the parallel routing test for the related SNR. Considering the example in Fig. 4, if the combination (G1, G2, G3) on level 1 can be built in a test case, the processed SNR test can be completed because all other combinations on levels 2 and 3 are included implicitly in the combination on level 1.

Generally, when test cases are generated, two possibilities can be distinguished for each combination of groups on a processed level. Either a test case can be generated for that combination completely or partially.

A completely generated test case describes the situation when all Routing Instances of all groups for a selected combination is included in one test case (constraints and gained information are criterion for the construction of test cases). A partially generated test case describes the situation when Routing Instances of groups for a selected combination can only be partially included. For such situations, the algorithm shall proceed to the next lower level to cover missing combinations.

Reasons for utilizing power set are:

1. In the formulation of subsets, the order of elements has no effect.
2. The count of elements in subsets varies between one to a maximum number.

These two features are correlated to parallel routing, because the order of Routing Instances in a test case has no effect on the test and the count of Routing Instances can be varied from one to a maximum configured number.

Introducing a recursive testing technique for parallel routing can help in analyzing the results from executed test cases. By analyzing the results, Stress Factors shall be corrected if a variance is observed. Additionally, groups that stress the gateway more than others shall be isolated for testing TP parallel routing for MNRs.

E. Testing TP Parallel Routing for Mixed Network Relationships (MNRs)

Parallel routing test for SNRs helps in correcting group's Stress Factors. From groups of each SNR Collection, a representative with the best Stress Factor shall be marked when combinations have to be selected for testing TP parallel routing for MNRs. Since networks can also have similarities among each other, the number of combinations can be further reduced by omitting similar combinations for MNRs. This can be achieved based on the Similarity Number of SNR collection's groups. In the selection of combinations for MNRs, the power set of available networks shall be constructed and similar subsets shall be deleted. Resulted combinations for MNRs shall be the base to check if the coverage criteria can be completely achieved.

The described concept for testing TP parallel routing for MNRs shall be explained in the example in Fig. 5:

Network 1 (N1) consists of formulated groups G1, G2 and G3 along with their respective Similarity Numbers (Si. N.) and Stress Factors (S.F.). Network 2 (N2) and Network 3 (N3) also contains similar information. Representative groups shall be selected based on the groups having best Stress Factors from N1, N2 and N3. Thereby, N1 (G1), N2 (G1) and N3 (G2) can be selected for optimizing the number of combinations. Other groups shall be omitted because of following reasons:

- Groups excluding representatives have higher Stress Factor and shall affect the behavior of gateway and hence they should be used for defining worst case scenarios.
- Representative groups implicitly resemble the excluded groups from each of the networks and hence can reduce the duplication of the process.

Power set shall be formulated from the representatives. Based on the Similarity Numbers, subsets from the power set shall be omitted, thereby resulting into an optimized formulated power set. This optimized power set shall be further used for categorizing into levels as explained in the previous section.

Testing TP parallel routing for MNRs then follows the same concept as of testing TP parallel routing for SNRs.

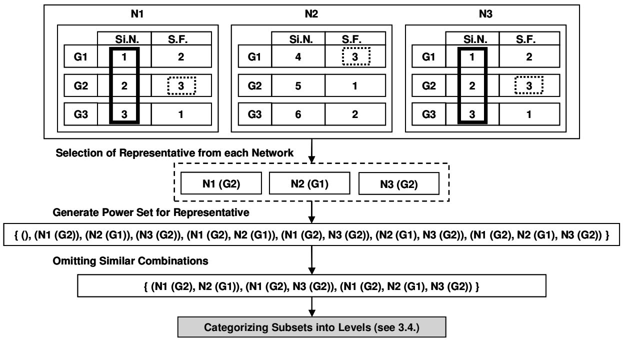

N1
N2
N3
Si.N.
S.F.
G1
G2
G3
1
2
3
2
3
1
4
5
6
3
1
2
1
2
3
2
3
1
Selection of Representative from each Network
N1 (G2)
N2 (G1)
N3 (G2)
Generate Power Set for Representative
{ (), (N1 (G2)), (N2 (G1)), (N3 (G2)), (N1 (G2), N2 (G1)), (N1 (G2), N3 (G2)), (N2 (G1), N3 (G2)), (N1 (G2), N2 (G1), N3 (G2)) }
Omitting Similar Combinations
{ (N1 (G2), N2 (G1)), (N1 (G2), N3 (G2)), (N1 (G2), N2 (G1), N3 (G2)) }
Categorizing Subsets into Levels (see 3.4.)

Figure 5. Optimizing Power Set for Mixed Network Relationships

Finally, the number of combinations required to test TP parallel routing for a given system shall be calculated from the number of combinations for testing SNRs along with the number of combinations for testing MNRs.

IV. Discussion

Depending on the grade of diversity in parameters for Connection Channels and for the gateway connected networks, the number of resulted groups can increase. The idea is to use combinations of groups instead of combining Routing Instances to reduce the number of generated combinations. If the number of groups still higher, then Stress Factors shall be required within groups for testing SNRs. Another drawback of this approach is the need of system functionality expertise to define similarity criterion and calculate the Stress Factors of the groups. However, this needs to be performed only once. Later on, combinations to be tested and test cases can be generated automatically for each new release of the system.

V. Conclusion and Future Work

In this paper, a recursive test case selection and generation methodology has been proposed to overcome the combinatorial explosion problem in testing TP parallel routing on a gateway. Based on similarities between parameter values of Connection Channels, the methodology collects Connection Channels into groups which serve as input for building combinations to verify TP parallel routing for SNRs. Similarities between networks together with Stress Factors gained from verifying SNRs provides the base for building combinations to testing TP parallel routing for MNRs. The two phases for testing TP parallel routing are very practical and can provide information for optimizing the TP configurations and error analysis. After group selection, power set is used to construct combinations of groups which is required to completely achieve the *N*-wise coverage criteria. Subsets of power set are divided into levels to give orientation for constructing combinations and contribute in reducing combinations for testing. An Implementation of the Methodology is currently under development to test TP parallel routing on a central gateway with five networks (3 CAN networks with 500 kilo baud, 1 CAN network with 250 kilo baud and 1 FlexRay network with 10 Mbps).

References

[1] “Road vehicles-Diagnostics on Controller Area Network (CAN)-,” ISO 15765:2004(E), Switzerland, 2004.

[2] “Road vehicles-Communication on FlexRay-,” ISO 10681:2010(E), Switzerland, 2010.

[3] P. E. Ammann and A. J. Offutt, “Using Formal Methods to Derive Test Frames in Category Partition Testing,” In Ninth Annual Conference on Computer Assurance (COMPASS’94), Gaithersburg, MD, Jun. 1994, pp. 69–80.

[4] D. M. Cohen, S. R. Dalal, A. Kajla, and G. C. Patton, “The Automatic Efficient Test Generator (AETG) System,” Proc. of the 5th International Symposium on Software Reliability Engineering, Monterey, CA, Nov. 1994, pp. 303–309.

[5] M. B. Cohen, J. Snyder, and G. Rothermel, “Testing Across Configurations: Implications for Combinatorial Testing,” Proc. of the 2nd Workshop on Advances in Model Based Software Testing, Raleigh, North Carolina, USA, Nov. 2006, pp. 1–9.

[6] C. J. Colbourn, M. B. Cohen, and R. C. Turban, “A Deterministic Density Algorithm for Pairwise Interaction Coverage,” Proc. of the IASTED Intl. Conference on Software Engineering, Innsbruck, Austria, 2004, pp. 345–352.

[7] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, “IPOG: A General Strategy for T-Way Software Testing,” Proc. of the 14th Annual IEEE Intl. Conf. and Workshops on the Engineering of Computer-Based Systems, Tucson, AZ, Mar. 2007, pp. 549–556.

[8] Y. Lei and K. C. Tai, “In-Parameter-Order: A Test Generation Strategy for Pairwise Testing,” Proc. of the 3rd IEEE Intl. High-Assurance Systems Engineering Symposium, Washington, DC, Nov. 1998, pp. 254–261.

[9] Y. K. Malaiya, “Antirandom Testing: Getting the Most Out of Black-Box Testing,” Proc. Of the 6th International Symposium on Software Reliability Engineering, Toulouse, Oct. 1995, pp. 86–95.

[10] T. Shiba, T. Tsuchiya, and T. Kikuno, “Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing,” Proc. of the 28th Annual Intl. Computer Software and Applications Conference (COMPSAC 2004), Hong Kong, Sept. 2004, pp. 72–77.

[11] K. C. Tai and Y. Lei, “A Test Generation Strategy for Pairwise Testing,” IEEE Transactions on Software Engineering 28, Jan. 2002, pp. 109–111.

[12] M. Grindal, J. Offutt, and S. F. Andler, “Combination Testing Strategies: A survey,” Software Testing, Verification, and Reliability, 2005, pp. 167–199.

[13] T. J. Ostrand and M. J. Balcer, “The Category-Partition Method for Specifying and Generating Functional Tests,” Communications of the ACM, 31(6), New Yourk, Jun. 1988, pp. 676–686.

[14] D. R. Kuhn and M. J. Reilly, “An Investigation of the Applicability of Design of Experiments to Software Testing,” Proc. of 27th NASA Goddard/IEEE Software Eng. Workshop, Dec. 2002, pp. 91–95.

[15] D. R. Wallace and D. R. Kuhn, “Failure Modes in Medical Device Software: An Analysis of 15 Years of Recall Data,” Int. J. of Reliability, Quality and Safety Eng., vol. 8, no. 4, 2001, pp. 351–371.

[16] J. Bach and P. J. Shroeder, “Pairwise Testing: A Best Practice that Isn’t,” Proc. of the 22nd Pacific Northwest Software Quality Conference, 2004, pp. 180–196.

[17] M. N. Borazjany, L. S. G. Ghandehari, Y. Lei, R. N. Kacker, and D. R. Kuhn, “An Input Space Modeling Methodology for Combinatorial Testing,” Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference, Luxembourg, Mar. 2013, pp. 372–381.

[18] M. Grindal and J. Offutt, “Input Parameter Modeling for Combination Strategies,” Proc. SE'07 Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering, Anaheim, CA, 2007, pp. 255-260.

[19] M. Grindal, J. Offutt, and J. Mellin, “Handling Constraints in the Input Space when Using Combination Strategies for Software Testing,” Technical Report HS-IKI-TR-06-001, School of Humanities and Informatics, University of Skövde. 2006.

[20] S. A. Vilkomir, W. T. Swain, and J. H. Poore, “Software Input Space Modeling with Constraints among Parameters,” Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International, Seattle, WA, Jul. 2009, pp. 136–141.

A Functional Requirement Traceability Management Methodology for Model-based Testing Framework of Automotive Embedded System

Kabsu Han, Jiae Youn, Jeonghun Cho
School of Electronics
Kyungpook National University
Daegu, Republic of Korea
{kabus, jiae0620}@knu.ac.kr, jcho@ee.knu.ac.kr

***Abstract*— We present an automated functional requirement traceability generation and management methodology for model-based testing framework. Traceability of software was recognized in 1960s and international standard was established in 1980s. In automotive industry, lots of researches for the requirement traceability are performed but not practical for testing. This paper presents traceability fundamental and practical case study for model based testing of automotive embedded system that includes generation of the functional requirement traceability.**

Keywords - Model-based testing; Rrequirement management; Test automation; Traceability; Functional requirement ;

I. INTRODUCTION

The traceability was pointed as an issue of interest in software engineering and recognized to discuss the problem of software engineering in 1968 [1]. In 1980s, traceability was founded as a requirement in lots of national and international standards for software and system development. In automotive industry, automotive embedded systems increase steadily as the requirements and functionalities increase. Furthermore lots of companies, such as OEM, suppliers, are involved in developing the automotive embedded system. Although model-based development and testing are widely used [5][6], the requirements and traceability of automotive embedded system cannot be managed easily. This paper introduces the concept of required traceability for model-based testing and proposes practical framework that include bidirectional traceability among requirements, models and test cases. Also, practical requirements tracing with commercial tools are described.

Section 2 describes entire model-based testing process. Section 3 and Section 4 describe background knowledge about requirement engineering and traceability with standard and COTS tools. Section 5 shows case study for model-based testing of automotive embedded system. Finally, Section 6 describes conclusion.

II. MODEL-BASED TESTING

In model-based testing (MBT), the test developer simply describes a functional model of the system under test (SUT). A test sequence generation algorithm that can be selected by hand in the test case generator creates test cases to verify and validate the functional model of the SUT. A test case generator creates test cases that can run on the SUT from the functional test cases. After that, a test automation tool executes the test cases on the SUT automatically. Reports that compare each output from the SUT and the expected results are generated automatically. Test coverage and reliability of the test depend on the model of the SUT and the test sequence generation algorithm; even test cases can be generated manually and automatically. Figure 1 shows the entire process of MBT [2].

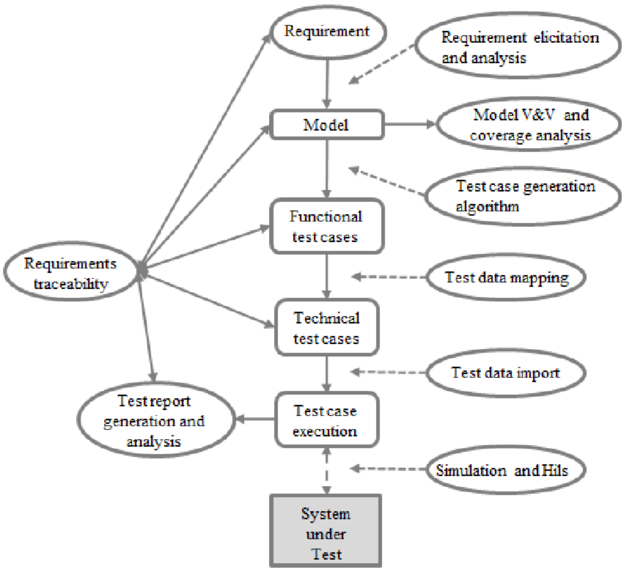

Requirement
Requirement elicitation and analysis
Model
Model V&V and coverage analysis
Test case generation algorithm
Functional test cases
Requirements traceability
Test data mapping
Technical test cases
Test data import
Test report generation and analysis
Test case execution
Simulation and HiIs
System under Test

Figure 1 Model-based testing process

The MBT method requires more steps and tools than the manual testing method, such as modeling, test case generation, and test case execution. Making a model of the SUT is describing a functional model of the system that needs to be tested. The modeling has to focus on the functional requirements of the system that the test developer wants to test. The model of the SUT may omit a lot of the details of the SUT that are not related to the testing. After describing the model, it has to be verified and validated for MBT. Most modeling tools provide automated verification

ISBN: 978-1-61208-348-3

and validation tools. Also, a graphical verifier is very useful to easily check the model.

The next step is generation of functional test cases from the model. The test developer has to decide the test selection criteria in order to generate efficient test cases. Because infinite numbers of test case are available, a plan to test all cases is impractical. Through selection criteria, coverage of the test cases is decided, and functional test cases that are test sequences of the model are generated. Figure 2 shows a transition based test coverage of black-box testing. The functional test cases are a kind of simple view of the SUT, so they do not contain detailed information to execute test cases directly on the SUT.

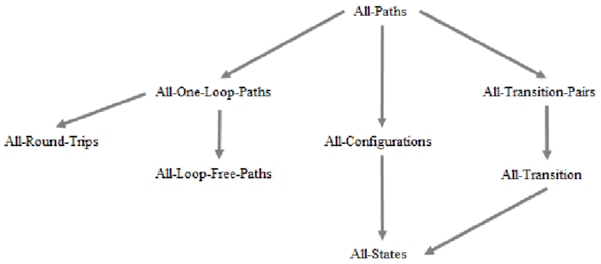


Figure 2 Transition based test coverage

The generation of an executable test case, called a test script, is required to execute the generated test cases on the SUT. The adaptation and transformation approach can execute test cases on the SUT. The test case generation tools have to fill in detailed information of a low-level SUT that are not described in the functional model.

One of the benefits of MBT is independence between test cases and test environment. By regeneration of executable test cases, the same set of test cases that includes the models can be reused in different test environments.

III. Requirement Engineering

The requirement engineering phase is the first step of model-based testing. The requirement engineering process can be divided into 6 processes like below [1][3][5].

- Requirement elicitation
- Requirement analysis
- Requirement specification
- System modeling
- Requirement validation
- Requirement management

During early phase of the requirement engineering, user requirements are elicited and analyzed. The requirement elicitation is about the understanding the problems to solve. Because user requirements can be conflicting among them, requirment engineer have to make decisions to elicit and analyze the requirements that have to be specified. After the problems to solve are understood and analyzed, they have to be described for the requirement specification. The requirement specification has to describe the product to be developed not the process. In automotive industry, some certification standards, such as IEC 61508 and ISO26262 for the product, are proposed. To specify requirements, lots of techniques can be used, such as informal and formal description. In model-based testing, system modeling will be described with appropriate modeling language, such as FSM, MSC and UML, according to the requirement specification. After that, the requirement specification can be verified and validated through the system modeling. Depending on the modeling language, lots of verfication and validation method can be used, such as simulation and formal verification. Also, the requirement specification has to be managed during the entire project. These requirements consist of functional things that have to be provided and non-functional thing such as performance, reliability, cost. Throughout in this paper, the functional requirements are considered and the requirement management tool is used to manage the requirements.

In many cases, requirements are elicited as documents format, such as MS word and excel. But these cannot be used for requirement specification and requirement management tool directly. Also, the requirement specification in requirement management tools cannot be exchanged easily. To solve this problem, automotive industry proposed requirement exchange format, called Rule Interchange Format (RIF) [7]. The new name Requirement Interchange Format (ReqIF) was introduced by OMG in 2011 [8]. RIF/ReqIF is an XML file format that can exchange the requirements between requirement management tools from different vendors. Also, the requirement exchange format defines a process to transform the requirements between partners. EAST-ADL2, a kind of European architecture description language, proposed a RIF importer/exporter extension already. IBM DOORS, the requirement management tool, supports RIF/ReqIF importer and exporter and MS documents importer/exporter. Also, the Requirement Modeling Framework (RMF), open-source-framework with requirements, supports ReqIF standard [9].

IV. Traceability

In a software and system engineering area, the trace can be defined like below [1].

- *A specified triplet of element comprising: a source, a target and a trace link which connecting a source and a target. When more than a source and a target are associated by a trace link, such as a sub-pair of a source and a target, the sub-pair are treated as a single source or a target.*
- *The action of following a trace link from a source to target.*

The trace can either be atomic or chained. The traceability is the potential ability for traces. To assure the traceability, each of the sources, targets and trace links have to be acquired and stored. After that, software and system engineering activities and task can be traced as shown in Figure 3. The traces exist within specific development and maintenance life cycles. Also, the trace can be reused in different life cycles. The requirement traceability is the ability to describe and follow the requirement lifecycle in forwards and backwards direction. The tracing is the activity of either establishing or using traces. The tracing can be divided into 3 types, manual, automated and semi-automated.

- *Manual tracing – traceability is established by human tracer. Traceability creation and maintenance with drag and drop user interfaces are used in requiremnt management tools commonly.*
- *Automated tracing – traceability is established via automated tools and methods. Typically, traceability creation and trace link maintenance are automated.*
- *Semi-automated tracing – traceability is established via combination of automated tools and human activities. For example, automated tools suggust candidate trace links and human tracer verify them.*

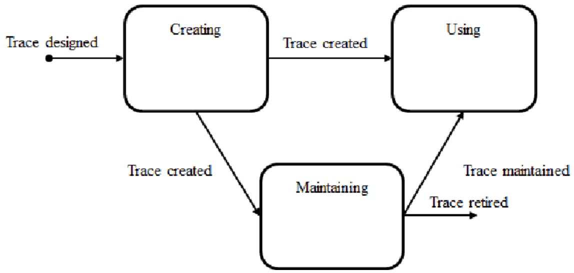


Figure 3 Traceability Model

In model-based testing, lots of traceability links are required like below [3].

- *traceability between requirements*
- *traceability between requirements and system model*
- *traceability between requirements and test cases*
- *traceability between requirements and test reports*

The traceability between requirements can be supported by the requirement document tools and the requirement management tools. In case of MS documents, MS office XML format are XML-based document formats and XML schema introduced in Office 2007. MS word and MS excel documents can import from and export to XML format. IBM DOORS can import from MS document and export to MS document. If importer and exporter between tools are not supported directly, RIF/ReqIF can be used to exchange the requirements, such as Papyrus MDT and plug-in. Figure 4 shows exporter of MS word and IBM DOORS.

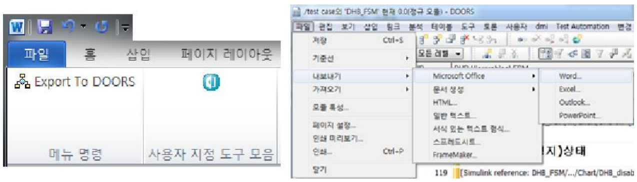


Figure 4 Requirement importer and exporter between MS word and IBM DOORS

The traceability between requirements and system model also can be supported the requirement management tools and modeling tools. Mathwork MATLAB/SIMULINK with verification and validation toolbox supports traceability link to MS word, MS excel and IBM DOORS. This toolbox can generate multiple traceability links with MS word bookmark, MS excel cell and DOORS object semi-automatically. When traceability links are generated, MS documents and DOORS objects are indicated with MATLAB/SIMULINK icon. Figure 5 shows traceability links on Stateflow model and Figure 6 shows Traceability links on MS documents and DOORS objects.

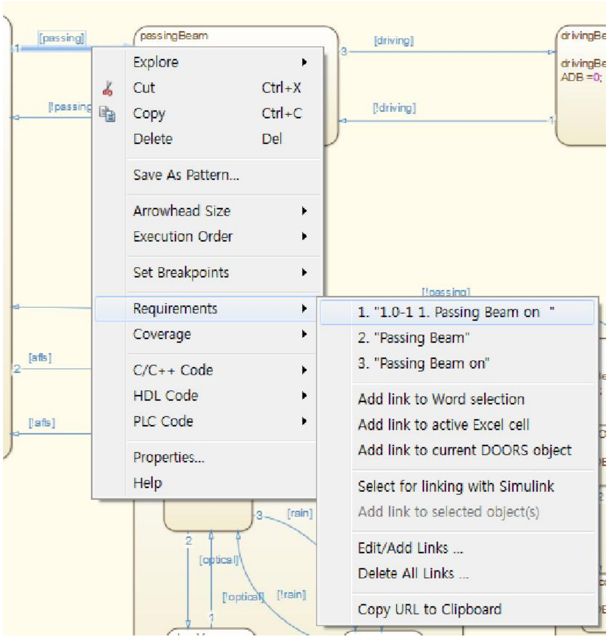
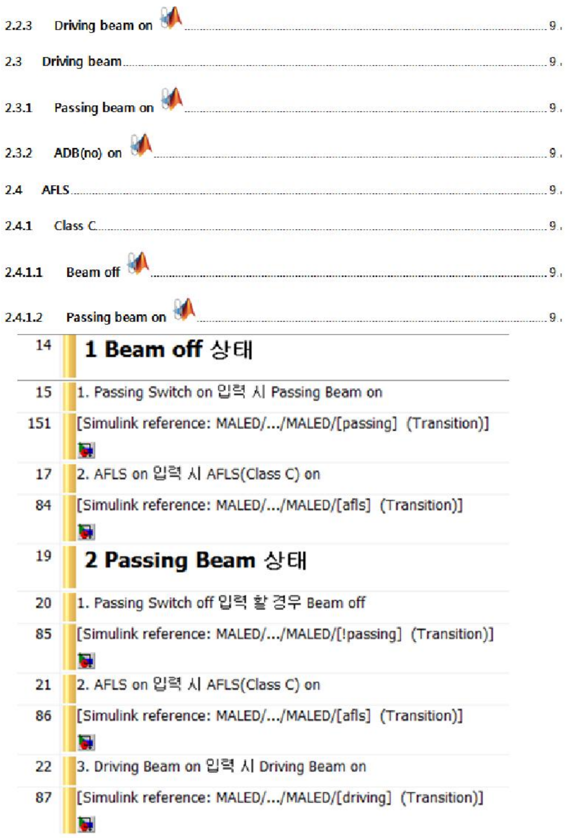


Figure 5 Traceability links on Stateflow model

In case of automotive embedded system, to execute the test cases that are generated from the functional requirements with System under test (SUT), I/O ports and in-vehicle network (IVN) interfaces are required [4]. Appropriate commercial-off-the-shell (COTS) tools as a de-facto in the automotive industry, such as Vector CANoe and dSPACE microautoboxII, can be used to execute test cases. Vector Test Automation Editor can generate executable test cases with XML format and supports traceability between requirements and test cases/test reports. The generated test cases can be executed on Vector CANoe through IVN. Depending on the DOORS objects, test groups and test cases are generated automatically in XML test module. The title of test groups and test cases are object id of DOORS. The test descriptions are imported from DOORS and external reference to DOORS are generated automatically.

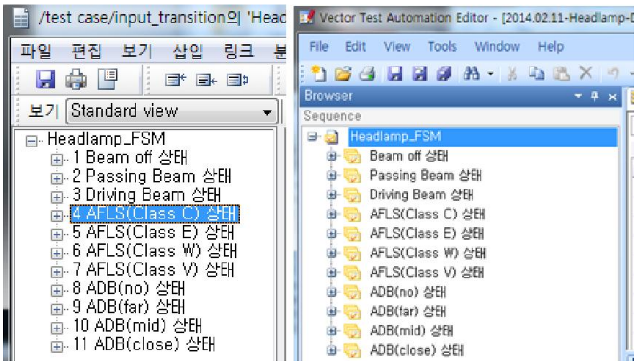


2.2.3 Driving beam on 9
2.3 Driving beam 9
2.3.1 Passing beam on 9
2.3.2 ADB(no) on 9
2.4 AFLS 9
2.4.1 Class C 9
2.4.1.1 Beam off 9
2.4.1.2 Passing beam on 9

14	**1 Beam off 상태**
15	1. Passing Switch on 입력 시 Passing Beam on
151	[Simulink reference: MALED/.../MALED/[passing] (Transition)]
17	2. AFLS on 입력 시 AFLS(Class C) on
84	[Simulink reference: MALED/.../MALED/[afls] (Transition)]
19	**2 Passing Beam 상태**
20	1. Passing Switch off 입력 할 경우 Beam off
85	[Simulink reference: MALED/.../MALED/[!passing] (Transition)]
21	2. AFLS on 입력 시 AFLS(Class C) on
86	[Simulink reference: MALED/.../MALED/[afls] (Transition)]
22	3. Driving Beam on 입력 시 Driving Beam on
87	[Simulink reference: MALED/.../MALED/[driving] (Transition)]

Figure 6 Traceability links on MS documents and DOORS objects

Test reports can be exported to DOORS through test report data mapping and import test report data. Also, test reports contain external reference to DOORS for traceability between requirements and test reports. Figure 7 shows traceability between DOORS and TAE as traceability between requirements and test cases. Figure 8 shows generated test groups and test cases include description and external references.



/test case/input_transition의 'Head

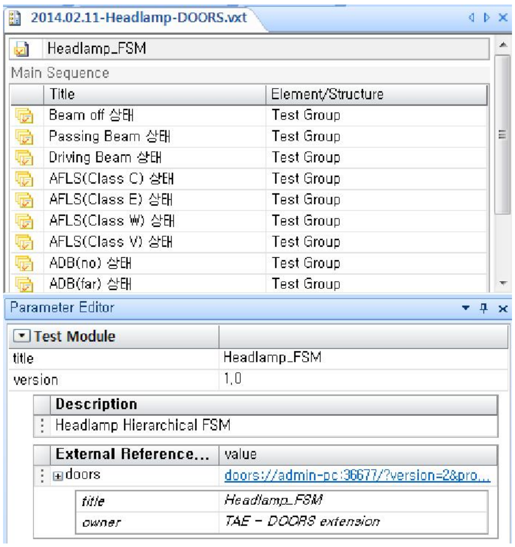
Standard view

Headlamp_FSM
- 1 Beam off 상태
- 2 Passing Beam 상태
- 3 Driving Beam 상태
- 4 AFLS(Class C) 상태
- 5 AFLS(Class E) 상태
- 6 AFLS(Class W) 상태
- 7 AFLS(Class V) 상태
- 8 ADB(no) 상태
- 9 ADB(far) 상태
- 10 ADB(mid) 상태
- 11 ADB(close) 상태

Vector Test Automation Editor - [2014.02.11-Headlamp-D

Headlamp_FSM
- Beam off 상태
- Passing Beam 상태
- Driving Beam 상태
- AFLS(Class C) 상태
- AFLS(Class E) 상태
- AFLS(Class W) 상태
- AFLS(Class V) 상태
- ADB(no) 상태
- ADB(far) 상태
- ADB(mid) 상태
- ADB(close) 상태

Figure 7 Traceability between requirements and test cases



2014.02.11-Headlamp-DOORS.vxt

Headlamp_FSM

Main Sequence

Title	Element/Structure
Beam off 상태	Test Group
Passing Beam 상태	Test Group
Driving Beam 상태	Test Group
AFLS(Class C) 상태	Test Group
AFLS(Class E) 상태	Test Group
AFLS(Class W) 상태	Test Group
AFLS(Class V) 상태	Test Group
ADB(no) 상태	Test Group
ADB(far) 상태	Test Group

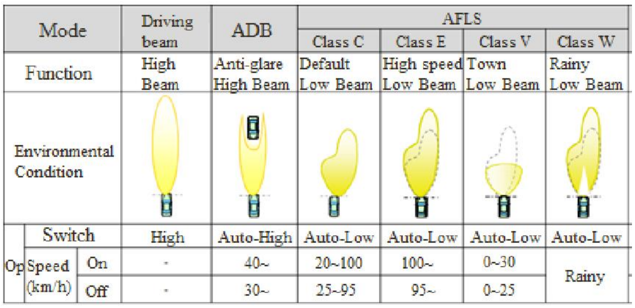
Parameter Editor

Test Module	
title	Headlamp_FSM
version	1.0
Description	
Headlamp Hierarchical FSM	
External Reference...	value
doors	doors://admin-pc:36677/?version=2&pro...
title	Headlamp_FSM
owner	TAE - DOORS extension

Figure 8 Generated test groups and test cases

V. CASE STUDY

To create and manage functional requirement traceability of model-based testing framework, intelligent headlamp system that includes adaptive front lighting system (AFLS) and adaptive driving beam (ADB) is adopted. The functional requirements elicited from a part of vehicle regulation of UNECE, such as R.48 and R.123, and requirements of OEM. The target system is an ECU of intelligent headlamp system. The main functional requirements of intelligent headlamp system consist of passing beam, AFLS, driving beam and ADB. The functional requirements of AFLS consist of class C, class E, class V and class W that elicited from the regulation of UNECE. The functional requirements of ADB are elicited from OEM. Figure 9 shows the functional requirements of AFLS and Figure 10 shows the functional requirements of ADB.



Mode			Driving beam	ADB	AFLS			
					Class C	Class E	Class V	Class W
Function			High Beam	Anti-glare High Beam	Default Low Beam	High speed Low Beam	Town Low Beam	Rainy Low Beam
Environmental Condition								
Op	Switch		High	Auto-High	Auto-Low	Auto-Low	Auto-Low	Auto-Low
	Speed (km/h)	On	-	40~	20~100	100~	0~30	Rainy
		Off	-	30~	25~95	95~	0~25	

Figure 9 Functional requirements of AFLS

The ECU of intelligent headlamp system receives environmental information, such as vehicle speed, illumination and other vehicle, from other ECUs and controls the headlamps of vehicle. At the first phase, the functional requirements are elicited from informal documents that contain functional and non-functional requirements for ECU, R.48 and R.123 of UNECE, and described in MS word. The functional requirements in MS word are exported to DOORS for requirement management.

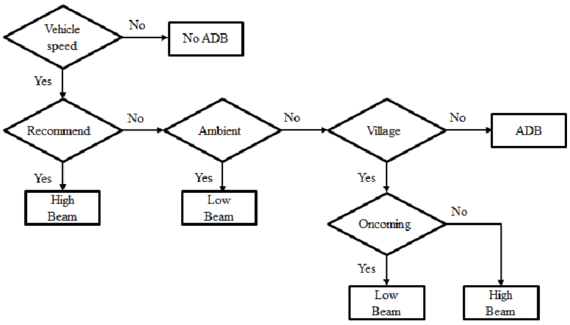
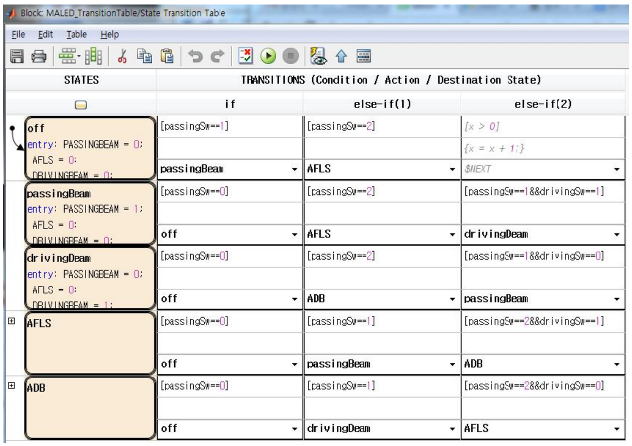


Figure 10 Functional requirements of ADB

During this phase, 117 functional requirements for headlamp system and 60 functional requirements for ADB are generated as DOORS objects. After that, the functional requirements in MS word are described in MS excel to generate a functional model. Because the functional requirements of ECU can be modeled as a discrete system, Stateflow are used to generate the functional model. Transition table function in Stateflow can generate the functional model with tabular description automatically, shown as Figure 11.

When the traceability links between the functional model and requirement are generated, the functional requirements verification and validation can be done through the functional model. If any inconsistency and corruption exist in the functional model, model analyzer will find it. SIMULINK design verifier analyzes the function model and generates test cases for structural coverage, such as condition, decision and MC/DC. During this phase, 167 test cases are founded and 48 test cases are generated. Figure 12 shows the report of SIMULINK design verifier.



STATES	TRANSITIONS (Condition / Action / Destination State)		
	if	else-if(1)	else-if(2)

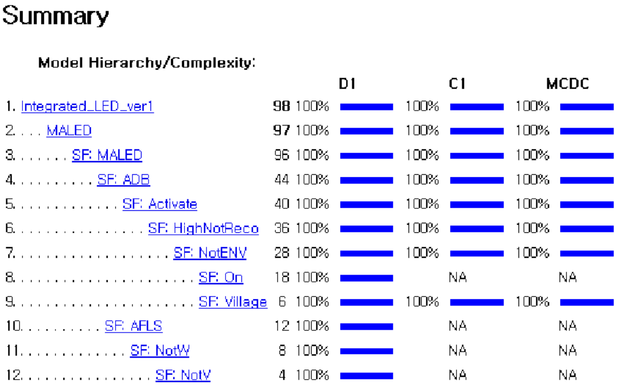
off
entry: PASSINGBEAM = 0;
AFLS = 0;
DRIVINGBEAM = 0;
	[passingSw==1]
	passingBeam
		[passingSw==2]
		AFLS
			[x > 0]
			{x = x + 1;}
			$NEXT

passingBeam
entry: PASSINGBEAM = 1;
AFLS = 0;
DRIVINGBEAM = 0;
	[passingSw==0]
	off
		[passingSw==2]
		AFLS
			[passingSw==1&&drivingSw==1]
			drivingDeam

drivingDeam
entry: PASSINGBEAM = 0;
AFLS - 0;
DRIVINGBEAM = 1;
	[passingSw==0]
	off
		[passingSw==2]
		ADB
			[passingSw==1&&drivingSw==0]
			passingBeam

AFLS	[passingSw==0] off	[passingSw==1] passingBeam	[passingSw==2&&drivingSw==1] ADB
ADB	[passingSw==0] off	[passingSw==1] drivingDeam	[passingSw==2&&drivingSw==0] AFLS

Figure 11 Functional modeling with transition table



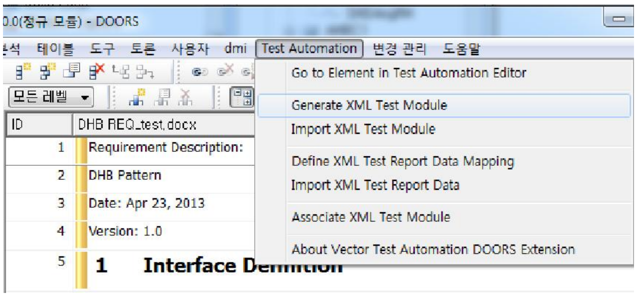
Summary

Model Hierarchy/Complexity:

		D I		C I		MCDC	
1. Integrated_LED_ver1	98	100%	▬	100%	▬	100%	▬
2. . . . MALED	97	100%	▬	100%	▬	100%	▬
3. . . . . . . SF: MALED	96	100%	▬	100%	▬	100%	▬
4. . . . . SF: ADB	44	100%	▬	100%	▬	100%	▬
5. . . . . SF: Activate	40	100%	▬	100%	▬	100%	▬
6. . . . . SF: HighNotReco	36	100%	▬	100%	▬	100%	▬
7. . . . . SF: NotENV	28	100%	▬	100%	▬	100%	▬
8. . . . . SF: On	18	100%	▬	NA		NA	
9. . . . . SF: Village	6	100%	▬	100%	▬	100%	▬
10. . . . . SF: AFLS	12	100%	▬	NA		NA	
11. . . . . SF: NotW	8	100%	▬	NA		NA	
12. . . . . SF: NotV	4	100%	▬	NA		NA	

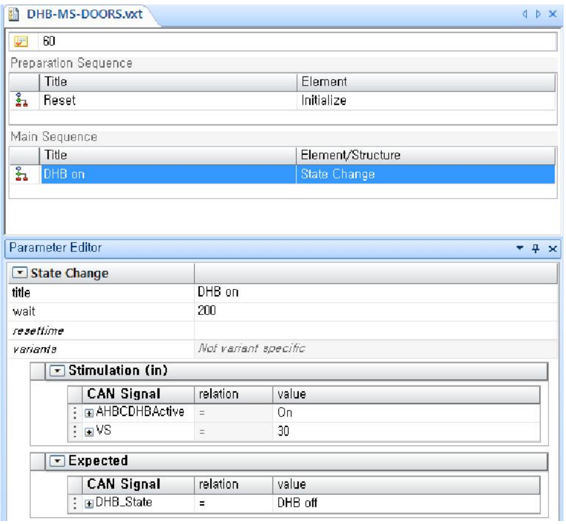
Figure 12 Validation result of the functional model

When the validation of the functional requirements through the functional model is finished, test cases can be generated from the functional requirements. Through the DOORS interface, XML test modules can be generated and associated automatically. Since the title of each test case is an object ID of DOORS module, traceability between requirements and test cases can be managed easily. Vector TAE is used to edit the XML test modules and Vector CANoe is used to execute the XML test modules. Because generated XML test modules contain test sequence, description and external reference to DOORS only, test engineer have to develop each test case according to functional requirements. During this phase, states and transitions in the functional model are mapped to technical signals in technical model. Depending on the technical model, various signal format, such as CAN, LIN and FlexRay, can be used. In this case study, headlamp ECU is connected with other ECU through CAN network. 12 messages with 49 signals are in CAN database file and 30 environment variables are developed to controls the CAN message and test environment. Figure 13 shows test case generation with DOORS-TAE interface and Figure 14 shows developed test cases with technical signals.


0.0(정규 모듈) - DOORS
분석
테이블
도구
토론
사용자
dmi
Test Automation
변경 관리
도움말
모든 레벨
Go to Element in Test Automation Editor
Generate XML Test Module
Import XML Test Module
Define XML Test Report Data Mapping
Import XML Test Report Data
Associate XML Test Module
About Vector Test Automation DOORS Extension

ID	DHB REQ_test.docx
1	Requirement Description:
2	DHB Pattern
3	Date: Apr 23, 2013
4	Version: 1.0
5	1 Interface Definition

Figure 13 Test case generation with DOORS interface



DHB-MS-DOORS.vxt

60

Preparation Sequence

	Title	Element
	Reset	Initialize

Main Sequence

	Title	Element/Structure
	DHB on	State Change

Parameter Editor

State Change	
title	DHB on
wait	200
resettime	
variants	*Not variant specific*

Stimulation (in)

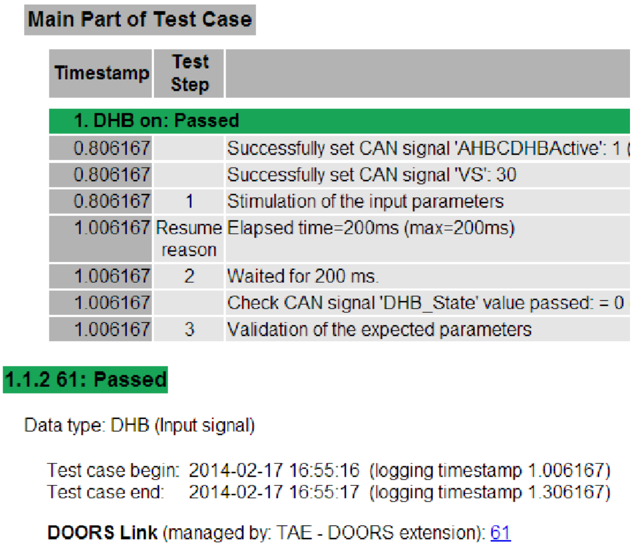
CAN Signal	relation	value
AHBCDHBActive	=	On
VS	=	30

Expected

CAN Signal	relation	value
DHB_State	=	DHB off

Figure 14 Developed test cases

When test cases are developed, each test case can be executed on the Vector CANoe with SUT. If real SUT is not available yet, simulation model can replace the real SUT as well as other ECUs. In the test environment, 9 ECUs are simulated that are not available in the LAB., such as Transmission control Unit (TCU), Engine Management System (EMS) and camera module, and a prototype and simulation model of ECU of intelligent headlamp are used to test. After execution of test cases, test report of the test cases that includes test verdict and traceability links are generated automatically. Also, the test report contains detail test step with time stamp and statistics. Figure 15 shows a part of test report that includes timestamp, test step, verdict and traceability link. Also, DOORS can import XML test report data through DOORS interface. With the test report, traceability between functional requirements and test reports can be established and managed.



Main Part of Test Case

Timestamp	Test Step	
1. DHB on: Passed		
0.806167		Successfully set CAN signal 'AHBCDHBActive': 1 (
0.806167		Successfully set CAN signal 'VS': 30
0.806167	1	Stimulation of the input parameters
1.006167	Resume reason	Elapsed time=200ms (max=200ms)
1.006167	2	Waited for 200 ms.
1.006167		Check CAN signal 'DHB_State' value passed: = 0
1.006167	3	Validation of the expected parameters

1.1.2 61: Passed

Data type: DHB (Input signal)

Test case begin: 2014-02-17 16:55:16 (logging timestamp 1.006167)
Test case end: 2014-02-17 16:55:17 (logging timestamp 1.306167)

DOORS Link (managed by: TAE - DOORS extension): 61

Figure 15 Test report

VI. CONCLUSION

To create and manage functional requirement traceability for model-based testing framework of automotive embedded system, automated and semi-automated tracing is considered. Bidirectional traceability between functional requirements, MS documents and IBM DOORS, are created through IBM DOORS interface. Also, traceability between functional requirements and functional model and traceability between requirements and test cases are created through COTS tools, such as MATLAB SIMULINK and Vector CANoe, for practical requirement tracing. The case study shows discrete system only but applicable to continuous system. Automated tracing for model-based testing framework is very helpful to verify and validate automotive embedded system.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC(Convergence Information Technology Research Center) support program (NIPA-2013-H0401-13-1005) supervised by the NIPA(National IT Industry Promotion Agency.)

REFERENCES

[1] J. Huang, O. Gotel, and A. Zisman, Software and Systems traceability. Springer-Verlag, London, 2012.

[2] M. Utting and B. Legeard, Practical model-based testing, 1st ed., vol. 1. Elsevier: San Francisco, pp.19–35, 2007,

[3] M. Adedjouma, H. Dubois, and F. Terrier, “Requirements exchange:from specification documents to models” The 16th International Conference on Engineering of Complex Computer System (ICECCS 2011) IEEE, April, 2011, 27-29, pp. 350-354, ISBN:978-1-61284-853-2.

[4] K. Han, I. Son, and J. Cho, “A study on test automation of IVN of intelligent vehicle using model-based testing” The Fifth International Conference on Ubiquitous and Future Networks (ICUFN 2013) IEEE, July, 2013, 2-5, pp. 123–128, ISSN:2165-8528, doi:10.1109/ICUFN.2013.6614794.

[5] R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. Akbar Raja, and K. Kamran, “Requirement traceability: A systematic review and industry case study” Int. J. Soft. Eng. Knowl. Eng., May, 2012, vol. 22, pp. 1-49, ISSN:0218-1940, doi: 10.1142/S02181940120 05846.

[6] M. Weber and J. Weisbrod, “Requirement engineering in automotive development-experiences and challenges” IEEE Joint International Conference on Requirement engineering, 2002, Sep, 9-13, pp. 331-340, ISSN:1090-705X, doi: 10.1109/ICRE.2002.1048546.

[7] World Wide Web Consortium. *RIF Overview*. [Online]. Available from: http://www.w3.org/TR/2010/NOTE-rif-overview-20100 622/ 2014. 05. 07

[8] Object Management Group. *Requirement Interchange Format*.[Online]. Available from: http://www.omg.org/spec/ReqIF/ 2014.05.07

[9] Eclipse Incubation. *Requirement Modeling Framework*. [Online] Available from: http://www.eclipse.org/rmf/ 2013.05.07

Establishing Personalized IVI Features on Distributed Open Source Webinos Middleware Using Low-cost Devices

Krishna Bangalore, Daniel Krefft, Uwe Baumgarten
Informatik
Technische Universität München
München, Germany
Email:{krishna.bangalore, krefft, baumgaru}@in.tum.de

***Abstract*— An observable trend in the automotive area leads to a growing demand for personalized in-vehicle infotainment (IVI) systems, but mostly they are closed, therefore integrating custom made apps were not possible with existing IVI-systems. So we need to extend the existing closed IVI-system with an Open source based system supporting custom made apps e.g., with diagnostic and location features, independent of the car manufacturers and the used bus-system. This paper presents a solution of a web app using low-cost devices with an Open source web-based webinos middleware. To evaluate the functionality and feasibility of the system with respect to in-car diagnostics data and location features, we present a prototype webinos *Vehicle Hub* application that runs on a HTML5 web browser which showcases the implementation of diagnostics data in dashboard view on top of the webinos middleware. Additionally for the developers to enhance the given open system we provide a *Vehicle Testbed* to test the APIs and necessary drivers. The application runs on three different device types – IVI-system, PC and smartphone/tablet. Users control the view of the dashboard and the data that they want to view with drag and drop on their PCs and dynamic streaming data with gauges on the IVI-system and smartphone/tablet.**

Keywords – *Middleware; In-Vehicle Infotainment System; Browser; HTML5; OBD-II; Raspberry Pi; Distributed Systems; Distributed applications.*

I. INTRODUCTION

IVI-systems are following a trend in enabling in-car browser-based runtime environments [2]. Supporting IVI-systems with an embedded web browser that appears as a combined interface for cloud services and personalized features could be a starting point. Therefore, it would be important to see the capability for executing web applications or JavaScript respectively in a sufficient manner [4]. Accessing vehicle data from the Controller Area Network (CAN) bus is propriety based and is not easily available [1]. It would be important to build an open and cost effective system to access required engine data from the vehicle, following the safety and security regulations that can be used with any car. The webinos middleware provides a solution for open and web-based communication for heterogeneous devices in a distributed manner [16]. The Internet of Things Application Programming Interface (IOT API) [14] provided by webinos allows us to build drivers to connect devices like On board Diagnostics (OBD-II) [21] to be used as sensors. Based on webinos technology, the webinos *Vehicle Hub* app provides a solution for showing the OBD-II vehicle parameter values that can be viewed on the graphs and gauges, which are integrated with the webinos dashboard as an interface for managing devices and to register services in the user's personal zone or services that the user's friends provide [3]. The webinos *Vehicle Hub* app provides 18 OBD-II parameters where we can choose the parameters and set the intervals to see the values accordingly.

In summary, this paper makes the following contributions:

- We present webinos as a middleware for permitting vehicle data from the car to be viewed on a browser locally or remotely.
- We evaluate the functionality and feasibility of such a middleware approach by webinos *Vehicle Hub* application as an automotive use case.
- We additionally provide webinos *Vehicle Testbed* page for testing the APIs and necessary drivers.

The following Section 2 presents webinos personal zone concept. Section 3 shows related work about different ways of working with cars. Section 4 shows the webinos approach for connecting the in-car head unit with other devices required to build a personalized IVI-system. Sections 5 and 6 outlines the implementation and evaluation of webinos approach with a common use case. Section 7 concludes with an outlook on future work.

II. *WEBINOS* PERSONAL ZONE CONCEPT

The core webinos architecture is based on state of the art widget and HTML5 web runtime environment. The critical innovation of webinos is to place an embedded server on the devices, and place all extended APIs, policies and packaging logic behind the server [15]. By tying the functionality to a server, rather than binding it to the traditional runtime, these services become addressable by other devices, not just the device the browser is running on. As shown in the Figure 1, the personal zone concepts within webinos are built up from internet agents Personal Zone Hub (PZH) and device agents Personal Zone Proxy (PZP). The way these agents communicate and identify each other, is at the heart of webinos mechanics.

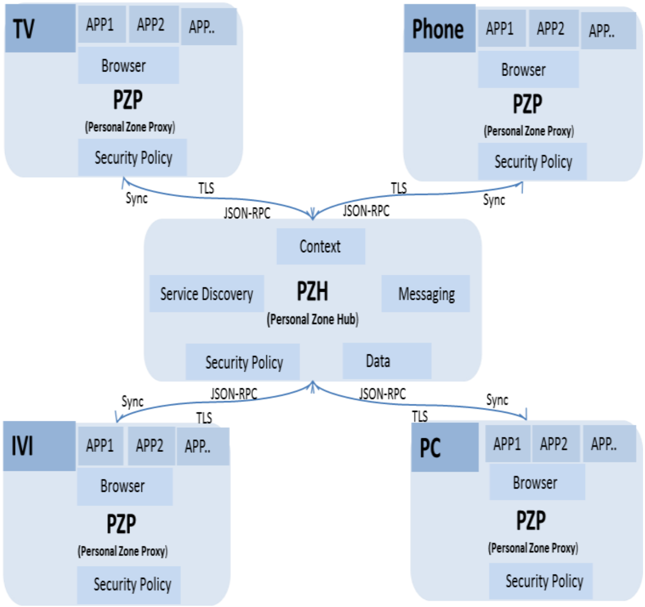


Figure 1: Personal Zone concept introduced by the *webinos* middleware

The aim of webinos is to provide a secure platform to connect heterogeneous devices like TV, IVI-system, PC and Smartphone for a multi-user and for any Operating system that supports web browsers in a web-enabled federated framework (means domains can exchange messages). The JavaScript Object Notation-Remote Procedure Call (JSON-RPC) [26] is used to get remote JavaScript and Transport Layer Security (TLS) [27] mutually authenticates the connection and gives the overlay network security and attestation. The secure session allows for transport of messages and synchronization. Apps run on top of webinos, when a webinos API is invoked, the invoked methods and its parameters are sent over a web socket, where the listening PZP checks them and returns a callback with a result locally. When it comes to remote communication the PZP forwards the request to the PZH, which finds the correct PZP that is connected to it and forwards the request. If PZP belongs to another PZH then it is forwarded under the security and policy control checks.

III. DIFFERENT WAYS OF WORKING WITH CARS

The market for personalized solutions, especially in the automotive area, is set to explode over the next few years with AutolinQ opening doors for future of mobility inside the car [17].

The following opportunities are offered by the upcoming trends:

- **Vehicle IVI-System:** A complete web based technology stack is provided to implement in-vehicle entertainment, navigation and real time driving consoles. It also provides an environment for third party application development.
- **In car communications:** A secure local communication stack allows phones, tablets, satellite navigation systems and other devices to interact seamlessly with the vehicle’s entertainment and telematics systems.
- **Remote sensing:** Secure interoperable remoting protocols allow these same capabilities to be accessed by trusted remote third parties, thereby enabling remote vehicle and driver diagnostic scenarios.

A. *What is so special about webinos vehicle?*

Since there has been a huge trend for HTML5-based applications [6] Sonnenberg, presents an approach for embedding a web application server into a native application, running on a portable device [9]. Similar to that, Open source webinos middleware provides a practical solution for in-car application development and porting on to new devices. Like Chrome OS [11], Firefox OS [18] and Tizen [19] it is HTML5 based, and indeed largely compatible with these technologies. It differs in that, it is not tied to a specific application ecosystem, and comes with a suite of vehicle specific additions, which speeds up automotive application development.

A strong emphasis has been placed on the webinos security model. This is important because vehicle informatics subsystems are extremely sensitive, and grant access to highly sensitive data. Security plus interoperability is the key here. The webinos protocols are unusual in that the same mechanism that allows device interaction over the cloud can be reused for local, in-vehicle networks. For real world deployment where internet in-car is unreliable this is essential. In practice, this means that “permissions permitting” any phone or tablet in car can securely and interoperably interact with the core infotainment systems of the car. Imagine pushing the location of the destination directly from a tablet to the in-vehicle navigation or even using this exact same technology to push locations from a remote desktop [28].

Out of all the use cases that webinos supports, remote analytics and sensing are the most interesting and disruptive aspects. Fleet management, real-time logistics, remote vehicle diagnostics (automatically alerts issues similar to mbrace2) [7] and more recently, behavioral driver monitoring are all existing and in some cases quite mature technologies. The webinos technology stack is interesting in this context because it can support all of these use cases, by using entirely commoditized and Open source stacks, which not only break apart existing locked in systems, but do so in a way that grants explicit control of sensitive data to the end user.

IV. THE *WEBINOS* DOMAIN FOR VEHICLE

The Personal zone concept that webinos provides, the PZP and PZH are built on top of node.js [5]. Node.js is based on Google's V8 JavaScript engine, all the devices (Personal Zone Proxies) belonging to the same zone support and expose a set of standard APIs for accessing services such as device features IOT [14], Geolocation [13], Device Orientation [12], networking with other devices and cloud services.

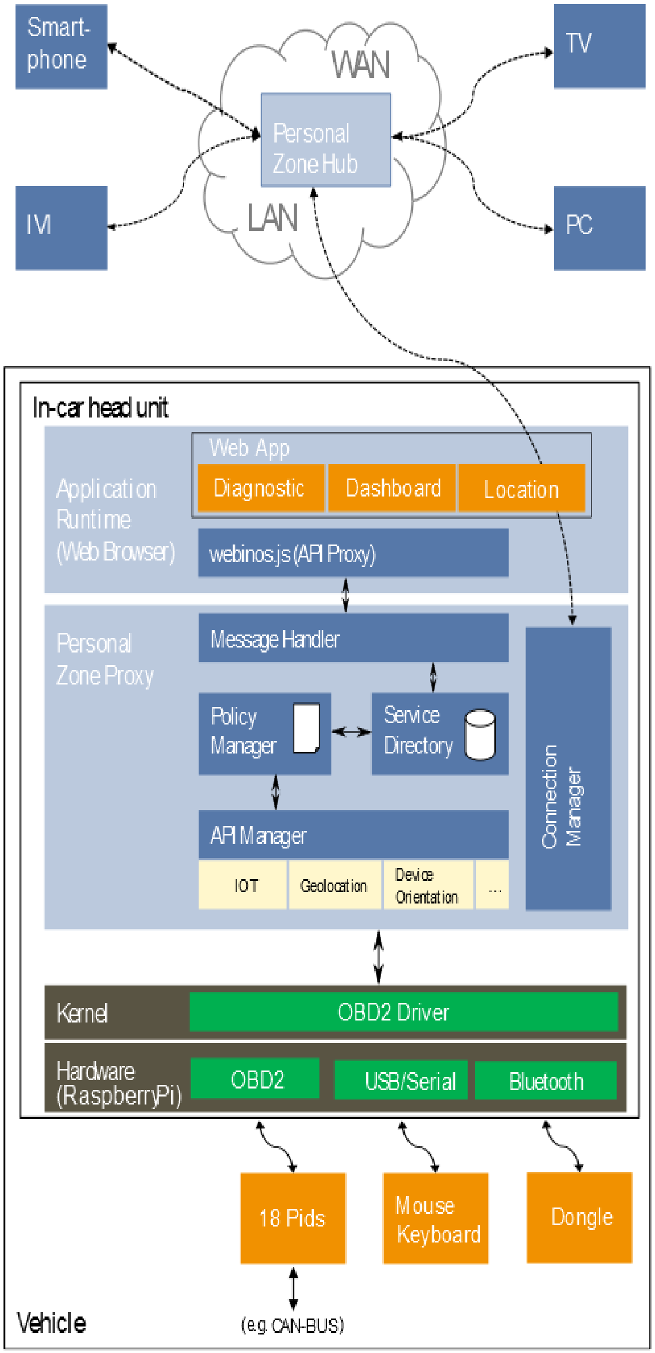


Figure 2: Architecture view how the *webinos* Middleware connects with OBD-II in a distributed automotive environment, similar to architecture [16]

As shown in Figure 2, the architecture describes the local interaction. The PZH is the center point of the personal zone between PZH and PZPs and the PZPs in the same zone if they try to share the resources. The PZPs can be more than one type of device, for example in-car devices, PCs, smartphones and so on. The interactions can take place between PZHs if the personal zones try to communicate with each other PZH.

Applications run on top of the webinos stack. PZH runs the server resident webinos applications using node.js, unpacking and performing security checks on packed widgets, authenticates users to set up trusted sessions, stores the policy files, routes messages, processes synchronization protocol messages so that the PZPs can synchronize the data from the devices [15]. The architecture describes the working of webinos middleware for the vehicle, before an API can be used by an application it has to query the Service Discovery for the available APIs for that particular PZP. Once the API is discovered it has to bind to the discovered API. In case of remotely available features the message is passed through Connection Manager, the requested feature is routed through PZH. The Policy manager checks for requests from the local or remote devices and grants access based on the policy settings set by the user.

The vehicle prototype uses Raspberry Pi [20] as hardware and the webinos IOT API that supports OBD-II driver to retrieve data from the streaming OBD-II sensor. Look for Section 5 for implementation notes.

V. IMPLEMENTATION NOTES

The interoperable specifications for the device APIs, the security model and the remoting interoperability layer have all been made available under royalty free terms. The software is highly flexible and can be deployed on several operating systems and in several configurations including Pandaboard [8], Android and Raspberry Pi.

Our current prototype deployment scenario for vehicle environment includes the following listed components attached to it (see Figure 3):

- Raspberry Pi with 5V Power-supply and a SD-Card. The Raspberry Pi does not have any internal storage, the SD-card is used to store the image of Linux version Raspbian wheezy [20]. It uses SD-card for booting and for storage. We recommend 16 GB.
- Bluetooth Dongle.
- OBD-II (Bluetooth) [21].
- Surf stick or Wi-Fi (mobile hotspot).
- Compact/mini PC display or TFT-Screen (for a closer automotive touch, we recommend to use

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-348-3

suitable car TFT with 8" to 10" display size and 16:9 ratio) [22].

- DVI-to-HDMI cable required for attaching the screen to the Raspberry Pi.
- Compact/mini PC keyboard and mouse or a combination of keyboard and mouse (pad) [23].

See our website [28] for available documentation. The source code is available under Apache 2.0 license terms.

The Bluetooth OBD-II connector is connected to the Raspberry Pi using the Bluetooth dongle. Since Raspberry Pi has two USB ports in our implementation we have connected the Bluetooth dongle and Wi-Fi stick to the keyboard which has USB hub. For the Vehicle environment we would propose to use a mini PC keyboard and mini PC mouse or a combination of keyboard and mouse (pad) [23].

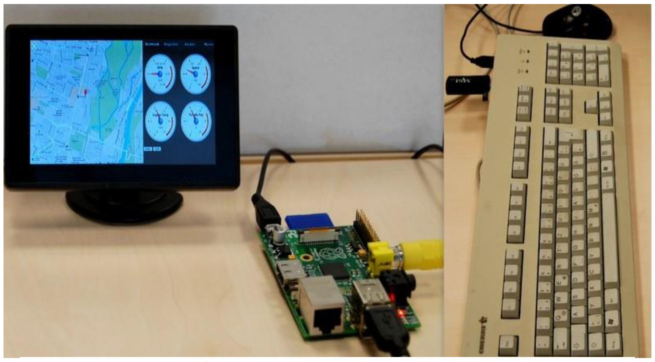


Figure 3: Prototype of Modularized webinos, running on a Raspberry Pi Hardware and showcasing IVI-system view connected to an OBD simulator

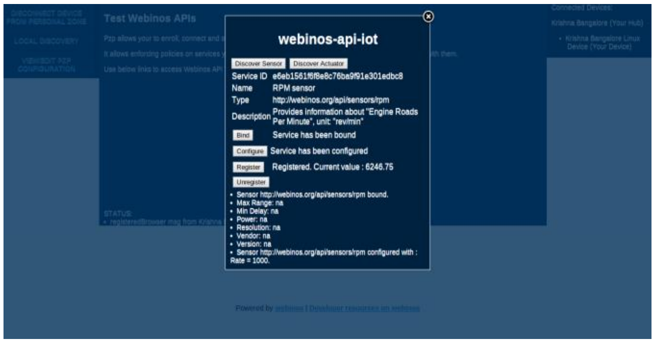
Figure 4: ELM 327 OBDII device

The ELM 327 OBD-II device (shown in Figure 4) standard specifies the type of diagnostic connector and its pinout, the electrical signaling protocols available, and the messaging format [21]. It also provides a list of vehicle parameters to monitor along with how to encode the data for each. There is a pin in the connector that provides power for the scan tool from the vehicle battery. The parameters defined in the OBD-II driver, that the IOT API uses looks for the streaming parameter messages from the OBD-II. The Geolocation and Device Orientation API offer the relevant data.

VI. EXAMPLE SCENARIO

A. *webinos Vehicle Testbed*

The webinos platform provides a *Vehicle Testbed* for testing its APIs that runs on a HTML5 browser to display test results from the OBD-II. In this particular example we connected the OBD-II to the webinos *Vehicle Testbed*, we could either test in a virgin PZP mode (not enrolled to the PZH) or enroll it to the PZH. We chose the latter and by doing this we register the browser to a particular OpenID [10] that the user wants.



Service ID e6eb1561f6f8e8c76ba9f91e301edbc8
Name RPM sensor
Type http://webinos.org/api/sensors/rpm
Description Provides information about "Engine Roads Per Minute", unit: "rev/min"
Service has been bound
Service has been configured
Registered. Current value : 6246.75

- Sensor http://webinos.org/api/sensors/rpm bound.
- Max Range: na
- Min Delay: na
- Power: na
- Resolution: na
- Vendor: na
- Version: na
- Sensor http://webinos.org/api/sensors/rpm configured with : Rate = 1000.

Figure 5: webinos Testbed shows RPM value

Figure 5 shows the working of webinos-api-iot, we show an example scenario of Revolutions per minute (RPM) value being streamed. We have to follow certain steps to retrieve the OBD-II value. First, find IOT API service through the service discovery. Second, bind to the found and selected service. Third, call the IOT API method to display the result. We can choose the register button as shown in the Figure 5 to connect to a listener to retrieve the RPM sensor streaming every few seconds.

B. *webinos Vehicle Hub app*

The webinos *Vehicle Hub* app shows OBD-II vehicle parameter values that can be viewed on the graphs and gauges as shown in the Figure 6 that depicts the IVI-system view and Figure 7 depicts the PC view, both use RGraph [25], which is integrated with the webinos dashboard as an interface for managing devices and to register services in the user's personal zone or services that the user's friends provided. The dashboard provides all the OBD-II parameters where we can choose the parameters and set the intervals to see the values accordingly. For demonstrating the webinos *Vehicle Hub* app we chose 5 useful parameters e.g., Engine RPM, Vehicle Speed, Throttle position, Engine Temperature and Fuel Rail Pressure that matches the webinos specifications and are useful diagnostics data.

The OBD-II drivers that are written for the webinos IOT API are registered within user's personal zone and allow the application to listen for values from OBD-II as sensors data. The application uses web technologies such as HTML5, JavaScript and CSS. For e.g., JSPlumb library [24] is used for the drag & drop feature.

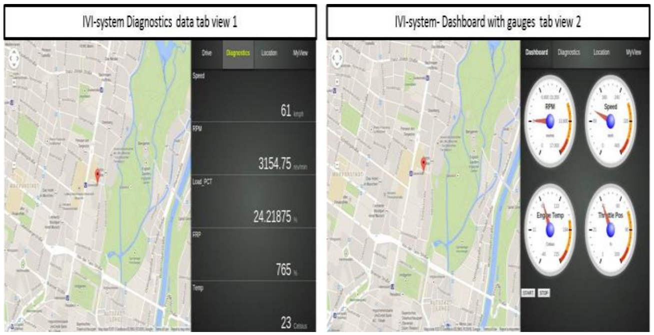


Figure 6: IVI-system Diagnostics data tab view 1 and IVI-system Dashboard with gauges tab view 2 [28]

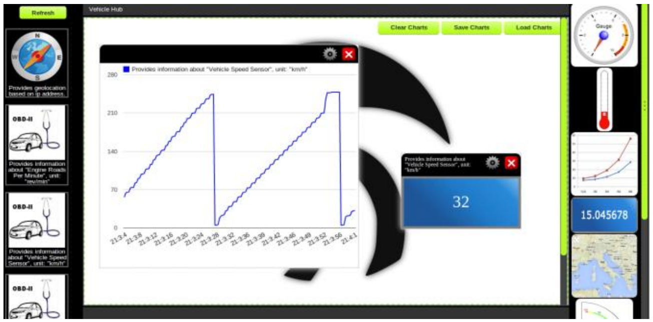

Refresh
Vehicle Hub
Clear Charts
Save Charts
Load Charts
Provides geolocation based on ip address.
OBD-II
Provides information about "Engine Roads Per Minute", unit: "rev/min"
Provides information about "Vehicle Speed Sensor", unit: "km/h"
Provides information about "Vehicle Speed Sensor", unit: "km/h"
280
210
140
70
0
21:3:4
21:3:8
21:3:12
21:3:16
21:3:20
21:3:24
21:3:28
21:3:32
21:3:36
21:3:39
21:3:42
21:3:46
21:3:49
21:3:52
21:3:56
21:4:1
32
Gauge
15.045678

Figure 7: PC view - drag and drop OBD parameters as sensors [28]

Vehicle Hub Features:

- In the PC view we can drag and drop the OBD-II parameters on the graphs and gauges, on the right side as shown in the Figure 7 graphs and gauges are present that are to be dragged and dropped to the middle of the window screen. Each OBD-II parameter act as sensors as seen on the left side of the Figure 7 and needs to be dragged and dropped on the graph window pane to view the results.
- Shows Live streaming data both on the PC view as well on the IVI-system.
- Historical view can record and play the historical data on IVI-system, collected logs of the data from the OBD-II that could be used to show as reports, we can show them on a graph as trip data and some vehicle specific data.
- Diagnostics data shows the 18 OBD-II supported parameter values that are useful for the mechanics to diagnose the vehicle. By connecting to webinos, mechanics can view the vehicle data from elsewhere and diagnose the vehicle before it’s brought to the garage.
- Insurance data collection – It was learnt from one of our workshops that the insurance companies install and maintain devices to get the vehicle reports. By using the current app and by further enhancing the webinos Vehicle Hub app we could create reports that will be useful for the insurance companies. This proposal from our workshops is now an ongoing work and is in its requirements phase.

C. Running webinos Vehicle Hub

Steps to run webinos *Vehicle Hub* app using OBD device with modularized webinos codebase on Linux distributions:

1. The demo works on the latest webinos version. It can be executed using firefox/Chrome browser. To make it run we need to clone the hub-webinosVehicle repository from the github [34] inside the *web_root* folder of the *webinos-pzp* or copy the content of the *hub-webinosVehicle* folder into the *web_root* folder in the *webinos-pzp*.
2. Place the *webinos-api-iot* API [30], *webinos-api-deviceOrientation* API [31] and *webinos-api-geolocation* API [32] in the *webinos-pzp* [29] folder. After that, in the *webinos-api-iot/node modules* path, clone the OBD-II drivers to get it run.

After, installing the required APIs and drivers do an *npm install* and change settings in *config.json* in *webinos-iot-driver-obd2* [33] to set the connector parameter for the Vehicle to *OBD* or if using the simulator then to *obdsim*.

Starting OBD-II simulator - The demo Figure 3 runs with an OBD-II simulator. To install the OBD-II simulator on to the Linux/Raspbian machine follow the instructions as described below.

Installing on the terminal

- *apt-cache search obd*
- *sudo apt-get install obdgpslogger*

To run with simulator

- *obdsim or obdsim –o*

Note: Change the */dev/pts/(Port Number)* in *config.json* in *webinos-iot-driver-obd2*

When connecting with a car to retrieve real time car values - Connect OBD-II (Bluetooth/Serial) into the OBD slot and rest similar to Figure 3, it runs on /dev/USB0 for serial and for Bluetooth, change the parameters of the OBD *params* settings present in the *config.json* in the *webinos-iot-driver-obd2* folder to retrieve the data.

VII. CONCLUSIONS AND FUTURE WORK

The presented work strongly focuses on providing custom apps e.g., diagnostic and location features with low-cost devices based on distributed Open source *webinos* middleware, connecting to the vehicle environment across heterogeneous devices.

This approach allows various in-car infotainment concepts. Firstly, it allows the execution of web applications that have access to the vehicle data via the introduced IOT API and OBD-II driver to support it and this implementation, can be tested on a *webinos* testbed. Secondly, the *webinos* platform provides the components to build and communicate with the vehicle system. The implemented webinos *Vehicle Hub* highlights that the *webinos* middleware is capable and applicable to aggregate data seamlessly across heterogeneous devices, with the help of webinos dashboard. The user can control the OBD parameters that the user wishes to see.

However, the application outlines items for future work. The devices mentioned in the paper were used as a prototype to showcase the usage of *webinos* middleware. We could use a smartphone with Android operating system and OBD-II to stream the data by connecting via Bluetooth interface (which is in development) to show the application running with an automatic user interface adaption without distracting the car driver. Instead of OBD-II it would be interesting to use different devices like vehicle Black box that match the webinos specification with similar parameters. We can build innovative applications like *Insurance apps* and *Traffic apps* to support the open and web world of communication that the webinos middleware presents.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Program (FP7-ICT-2009-5, Objective 1.2) under grant agreement number 257103 (webinos project). We thank all our project partners within the webinos consortium towards the contribution and realization of this work.

REFERENCES

[1] http://www.dgtech.com/images/primer.pdf [retrieved: May, 2014].

[2] http://www.we-conect.com/cms/media/uploads/events/31/ dokumente/QNX__Why_Automakers_(Should)_Care_ about_HTML5.pdf [retrieved: May, 2014].

[3] http://dev.webinos.org/deliverables/wp3/Deliverable35/ wiki$t3-5$Deliverable_Specifications_Personal_Zone _Security.html [retrieved: May, 2014].

[4] S. Isenberg, M. Goebl, and U. Baumgarten. Is theWeb Ready for In-Car Infotainment? A Framework for Browser Performance Tests Suited for Embedded Vehicle Hardware. In 2012 14th IEEE International Symposium on Web Systems Evolution (WSE). IEEE, 2012.

[5] http://nodejs.org/ [retrieved: May, 2014].

[6] G. Lawton. Moving the OS to the Web. Computer, 41(3):16{19, Mar. 2008.

[7] Mercedes-Benz, mbrace. http://www.mbusa.com/mercedes/mbrace#!layout=/mbrace/remote_access&waypoint=mbrace-remote_access [retrieved: May, 2014].

[8] Pandaboard, Pandaboard Reference. http://pandaboard.org/content/resources/references [retrieved: May, 2014].

[9] J. Sonnenberg. A distributed in-vehicle service architecture using dynamically created web Services. In IEEE International Symposium on Consumer Electronics (ISCE 2010), pages 1-5. IEEE, June 2010.

[10] OpenID, http://openid.net/ [retrieved: May, 2014].

[11] Chrome OS,https://www.google.com/intl/en/chrome/browser/ [retrieved: May, 2014].

[12] W3C, DeviceOrientation Event Specification. http://dev.w3.org/geo/api/spec-source-orientation.html [retrieved: May, 2014].

[13] W3C, Geolocation API Specification. http://www.w3.org/TR/geolocation-API/ [retrieved: May, 2014].

[14] webinos, Specifications, 2014. http://dev.webinos.org/specifications/api/sensors.html [retrieved: May, 2014].

[15] webinos, Architecture, 2014. http://dev.webinos.org/deliverables/wp3/Deliverable31/wiki$ wp3-1$Webinos_key_architectural_components.html [retrieved: May, 2014].

[16] Isenberg, Simon and Bangalore, Krishna and Goebl, Matthias and Haberl, Wolfgang and Baumgarten, Uwe. Towards a Personalized and Distributed In-car Infotainment Experience Using the Open and Web-based Webinos Middleware, Multi-Device '2012 [retrieved: May, 2014].

[17] Continental Corporation. AutolinQ, http://www.continental-corporation.com/www/pressportal_us_en/themes/press_releases/3_automotive_group/pr_2009_06_02_en.html [retrieved: May, 2014].

[18] http://www.mozilla.org/en-US/firefox/os/ [retrieved: May, 2014].

[19] Tizen IVI Architecture, http://events.linuxfoundation.org/images/stories/pdf/lceu2012 _haitzler.pdf [retrieved: May, 2014].

[20] Raspberry Pi, www.raspberrypi.org [retrieved: May, 2014].

[21] OBD-II, http://www.obdii.com/, http://en.wikipedia.org/wiki/On-board_diagnostics [retrieved: May, 2014].

[22] TFT-Screen, http://www.cartft.com/catalog/il/1213 [retrieved: May, 2014].

[23] http://www.logitech.com/de-de/product/wireless-touch-keyboard-k400r [retrieved: May, 2014].

[24] JSplumb, http://jsplumbtoolkit.com/demo/home/jquery.html [retrieved: May, 2014].

[25] RGraph, http://www.rgraph.net/ [retrieved: May, 2014].

[26] JSON-RPC, http://www.jsonrpc.org/specification [retrieved: May, 2014].

[27] TLS, http://www.techsoup.org/support/articles-and-how-tos/introduction-to-transport-layer-security [retrieved: May, 2014].

[28] https://developer.webinos.org/vehicle-hub [retrieved: May, 2014].

[29] https://github.com/webinos/webinos-pzp [retrieved: May, 2014].

[30] https://github.com/webinos/webinos-api-iot [retrieved: May, 2014].

[31] https://github.com/webinos/webinos-api-deviceOrientation [retrieved: May, 2014].

[32] https://github.com/webinos/webinos-api-geolocation [retrieved: May, 2014].

[33] https://github.com/webinos/webinos-iot-driver-obd2 [retrieved: May, 2014].

[34] https://github.com/webinos/hub-webinosVehicle [retrieved: May, 2014].

Performance Comparision of Encoding Schemes for ETSI ITS C2X Communication Systems

Sebastian Bittl and Arturo A. Gonzalez and Wolf A. Heidrich
Fraunhofer ESK
Munich, Germany
Email: {sebastian.bittl, arturo.gonzalez, wolf.heidrich}@esk.fraunhofer.de

Abstract—**Wireless Car-to-X communication is about to enter the mass market in upcoming years. Thereby, available bandwidth is small with only a low number of usable channels and many communicating entities. Therefore, efficient data encoding schemes are required to allow bandwidth saving use of the wireless channel by embedded devices. No detailed analysis of different encoding schemes for Car-to-X communication regarding important properties like runtime, memory consumption and encoded output length has been published so far. We provide such analysis for standardized ASN.1 and binary representations as well as Google Protocol Buffers as an alternative approach to the data encoding problem. Standardised data content for CAM, DENM and the security envelope are used in the conducted performance study. We show that ASN.1 encoding outperforms usage of Google Protocol Buffers, but is outperformed by a binary encoding scheme in most cases. This implies that standardization efforts for the security envelope should reconsider the recent shift from binary encoding towards usage of ASN.1.**

Keywords-ETSI ITS, data encoding, performance metrics, ASN.1, Google Protocol Buffers.

I. Introduction

Car-to-X (C2X) communication systems are gaining attention in the awake of their upcoming deployment as ETSI Intelligent Transport Systems (ITS) in Europe and Wireless Access in Vehicular Environments (WAVE) in the United States [1].

C2X communication happens digitally, meaning that messages between the involved nodes are represented as a series of bits, i.e., as bit streams. As in any software implementation of a communication system, the format of the messages exchanged between two communication end points must be well known by them. That means that nodes should be able to represent messages as bit streams and to interpret them as the original messages as well. The generation of a bit stream from a message is defined as encoding. Hence, we refer to an encoded message as the bit stream representation of such message. Following the same logic, decoding is defined as the generation of the original message out of its bit stream representation.

Several encoding schemes exist nowadays, and some of them are used extensively in everyday data communications. Depending on the application requirements, one scheme may be suited better than another. The requirements for these encoding schemes range from human readability (e.g., XML[2], JSON[3]), through the space the bit stream takes up in memory (e.g., ASN.1 PER encoding, binary encoding), up to system performance, i.e., encoding/decoding processing delay (e.g., binary encoding, ASN.1 OER encoding).

In the C2X realm, it is significantly relevant to use a bandwidth efficient encoding scheme since C2X communications operate under quite strict bandwidth constraints. As an example, in Europe only one 10 MHz channel is available for safety critical applications [4]. Therefore, an encoding that generates short bit streams out of messages is favoured. Moreover, safety C2X applications have strict end-to-end delay requirements. Therefore, encoding/decoding delays should be minimal such that their contribution to the end-to-end delay can be considered neglible.

In this work, we focus on the comparison of the performance metrics of two coding schemes, namely Abstract Syntax Notation 1 (ASN.1) encoding rules and Google Protocol Buffers applied to the two most common C2X message types in C2X communications: Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Basic Service (DENM). Since both encoding schemes support the Time Optimized and Space Optimized variants, performance metrics are obtained for both cases on both schemes. Furthermore, three different encoding schemes for the ETSI ITS Security Envelope, which are binary encoding, ASN.1 encoding rules and Google Protocol Buffers performance metrics, are also compared.

The remaining part of this work is organized as follows; an overview of related work is given in section II, the performance requirements and measurements are described in detail in section III and the target platforms description is summarized in section IV. The obtained results are described in section V. Finally, section VI provides a conclusion about the achieved results.

II. Background

The background of this work regarding platform independent data encoding, especially in the area of ETSI ITS, is provided in this section. Additionally, a comparision to the limited number of other published performance studies is given.

A. Data Encoding Rules

CAM and DENM are two standardized ITS messages defined in [5] and [6] respectively. According to these standards, the encoding of CAM and DENM is done using ASN.1 UPER encoding rules. There are several encoding rules which ASN.1 specifies: Basic Encoding Rules (BER), Packed Encoding Rules (PER), Canonical Encoding Rules (CER), Distinguished Encoding Rules (DER), Octet Encoding Rules (OER) among many other flavours, each providing advantages and disavantages from the point of view of a specific application.

Since PER provides a more compact encoded message than the older BER and its subsets DER and CER, it is often used in systems where bandwidth conservation is important [7]. This might be the reason why the CAM and DENM standards specify that the encoding rules to be used should be Unaligned PER (UPER). In aligned PER fields are aligned to 8-bit octet boundaries by inserting padding bits whereas in UPER padding bits are never inserted between fields, hence allowing a higher bit stream size reduction.

A widely used alternative to ASN.1 encoding rules are the so called Google Protocol Buffers [8], [9], which are used by Google extensively in their production environment. Therefore, they can be regarded as a stable and reliable library. Google Protocol Buffers offer a more simplistic approach to the platform independent data encoding, making them easier to manipulate and implement [8]. Additionally, they can be configured to do encoding optimized for either fast processing or small memory footprint. The latter is also a common feature provided by standard ASN.1 implementations. For example, the software provided by OSS Nokalva [7] provides this feature. Therefore, Google Protocol Buffers can be seen as a comparable alternative for message implementation. Hence, the performance study provided in section V makes use of these two technologies.

Furthermore, for the ETSI ITS security envelope two different sets of encoding rules have been proposed so far. At first, binary encoding with explicit definition of all data fields was proposed in [10]. Additionally, encoding using ASN.1 rules was proposed recently in [11]. As a further reference, Google Protocol Buffers will also be used in the following for the security envelope.

Further publicly available data serialization tools for converting arbitrary data into a platform independent binary representation include systems like Apache Avro, Apache Thrift or Message Pack [12][13][14]. These systems are either less mature or deployed to a much smaller extent in professional environments compared to ASN.1 and Google Protocol Buffers (see e.g., [15] for protobuf vs. Thrift). Therefore, they are not studied in detail. Additionally, serialization technologies like XML or JSON which aim to achieve a human readable and easy to parse data representation at the price of increased encoding length are out of the scope of this work. They are simply not appropriate to be used in bandwidth constrained communication systems.

B. State of the Art and Contribution of this work

There are several publications comparing other encoding schemes, such as XML with ASN.1. For example, the authors in [16] compare the performance between binary encoded XML and ASN.1 by running the tests on PC machines. In [17] the authors compare the performance of XML against ASN.1 BER on digitally signed data. They conclude that for applications where high performance is required, ASN.1 BER may be a better choice.

In [18] authors compare the performance of XML, JSON and Google Protocol Buffers in terms of data size and coding speed. The authors conclude that Google Protocol Buffers requires less bytes for the message representation in comparison with XML or JSON. The authors also explore the possibility of compressing XML and JSON messages using gzip [19]. In the latter case, both compressed text formats perform better than Google Protocol Buffers in terms of data size. In terms of speed, the authors show that Google Protocol Buffers perform better than both text schemes. In [20], authors perform a similar study showed in [18] and expand it for performance in energy consumption, relevant for the smartphones case. They also show that gzip-compressed Google Protocol Buffers, variant not explored in [18], performs better in terms of encoded data size in comparison with compressed XML, but worst that compressed JSON. When the authors measure performance in respect to encoding time, they concluded that for the data set they used Google Protocol Buffers performed better. On the parsing process on the receiver side, i.e., decoding, JSON perform slightly better than the other two.

To the understanding of the authors at the time of writing this work, there are no previous studies focusing on a quantitatively comparison of performance measurements between ASN.1 and Google Protocol Buffers, in specific on the field of C2X commmunications. Although the ETSI standard defines the encoding mechanisms as ASN.1, this work should provide some insight for the viability of an alternative based on an open source development as well as to provide some information on the performance comparison of these encoding schemes on different computer platforms such as embedded systems.

III. Performance Requirements and Measurements

In this work, we consider three main aspects in the performance evaluation of the different encoding schemes. These are:

1) computation time,
2) memory footprint on computation and
3) encoded data length

Aspects 1 and 2 clearly focus on the required computing power for the encoding and decoding process. As ETSI ITS technology shall be implemented in embedded systems e.g., in vehicles, these criteria are quite important due to the limited resources typically available in such systems.

The length of the encoded data is a criteria which mostly influences the required communication bandwidth on the wireless channel. It directly determines how long it takes to communicate a data packet over the air. Given that a communication channel has a limited capacity, the lenght of the encoded messages directly influences the number of possible transmissions over the air in a specific time span. Additionally, ETSI ITS uses only a single control channel to distribute important CAMs and DENMs. Therefore, an increased size of the encoded data packet directly leads to a decrease in system performance and scalability.

IV. Target Platforms

A. Hardware

To execute our performance measurements of the encoding schemes in question we have used three different platforms. The reason is to show the influence of different used hardware technologies as well as to exclude effects on the overall performance study caused by a single processor technology. Table I summarizes the main characteristics of the three platforms used in our experiments.

TABLE I. Used CPU hardware and achievable measurement accuracy via Linux clock counters.

type	AMD Geode LX	Intel Atom Z520PT	Intel Core i7-2640M
clock speed	500 MHz	1.33 GHz	2.8 GHz
clock res.	2 ns	1 ns	1 ns

More details about the used processor technologies can be found in references [21], [22], [23].

The clock resolution given in table I was obtained by using the clock_getres() [24] function on the individual platforms in the software environment described in the next section.

B. Software

On all platforms, a standard Debian Linux [25] system with kernel version 3.2.23 was used as the underlying operating system during the performance study. Furthermore, ASN.1 related functionality was provided by the OSS Nokalva library [7]. Google protocol buffers were used in version 2.4.1 as provided by the Debian distribution. For binary encoding of the security envelope the implementation from the ezCar2X framework [26] was used. All used software was compiled on the target with the GCC compiler version 4.7.2 [27]. Thereby, strong optimization was enabled with the *-O3* compiler flag.

For timing measurements the Linux kernel high performance counters have been used, which can be used from userspace by calling the clock_gettime() function [24]. Thereby, CLOCK_PROCESS_CPUTIME_ID was used as the clock ID in order to determine only the time spent in the process which contains the algorithm to be measured. An accuracy of up to 1 ns can be achieved, if the underlying hardware permits such accurate measurements [28]. In order to make the measurements more accurate, the suggestions from [29] for avoiding effects of out-of-order execution have been applied. Therefore, the CPUID instruction was executed before and after calling the clock_gettime() function.

The described methodology for time measurements is preferred over directly reading the CPUs time stamp counter (TSC), which is e.g., used in [29]. The reason is that while [29] uses operations only available inside the Linux kernel, the measurements in our performance study are done in the user space. Therefore, certain prerequisites of the approach from [29] like disabling of interrupts or scheduling cannot be fullfilled. Hence, we rely on the implementation of the clock counter in the Linux kernel.

An algorithm's main memory footprint (heap as well as stack usage) was measured by the help of the so called *malloc_count* framework [30]. This framework allows arbitary parts of a program to be traced by inserting dedicated function calls into it. These calls where only used during memory measurements and were removed during timing measurements as they would introduce overhead. Other memory tracing tools like *massiv* from the valgrind framework [31] do not allow adjustment of the measurement procedure with such fine granularity. Therefore, malloc_count was used to obtain the results presented in section V-C.

V. Performance Study

A. Content for Encoding and Decoding

We have used CAM [5] and DENM [6] messages containing only values in the mandatory fields. For this messages, we have used real data within the message content as far as possible e.g., the included time stamps.

The studied security envelopes consist of the message fields as specified in [10] and [11]. Thereby, all three defined security profiles are taken into regard. Additionally, for security profile number 1 two cases have to be distinguished. The corresponding envelope can hold a signed certificate or just an eight byte hash value of the certificate. Both cases have been included in the performance study.

In order to separate the security component tests from others, no real payload was used on these tests. For the case of binary encoding, the envelope only includes the mandatory one byte dummy payload as specified in the standard [10].

B. Encoding Rules for Google Protocol Buffers

The definition files for the Google Protocol Buffers (Protobuf) were derived from the ASN.1 definitions given in standards [5], [6], [11]. Thereby, transformation is straight forward due to the low number of available data types in protobuf. During the transformation process always the smallest Protobuf data type which is able to hold the corresponding ASN.1 data type was selected to avoid unneccessary overhead.

C. Results of Performance Study

The results of the conducted performance study regarding memory consumption and encoded output length are summarized below in tables II (CAM), III (DENM) and IV (security envelope). In the following, individual results for these message contents are studied in detail.

Memory requirements, as well as encoding length, are independent of the used CPU architecture. Therefore, just a single result is given for these criterias in the following. Runtime performance, which is clearly processor specific is looked at afterwards.

In the following, we use TOED as a short for time optimized encoder and SOED for space optimized encoder. All encoding length and memory consumption measurement results are given in bytes.

At first, encoding performance for CAMs is studied in detail. The achived results are summarized in table II. From table II it is clear that Protobuf generates almost four times more output bytes than ASN.1 for an encoded CAM. The space optimized code is roughly on par with the ASN.1 code, as Protobuf uses less heap but more stack space and

TABLE II. PERFORMANCE RESULTS FOR CAMS.

enc. type	heap / stack TOED	heap / stack SOED	encoded length
protobuf	469 / 1564	2450 / 6580	165
ASN.1	1066 / 1300	4120 / 4600	42

ASN.1 uses roughly the same amount of heap and stack space. Additionally, the time optimized Protobuf code uses less space as the space optimized code and even less than the ASN.1 time optimized code.

In the following, we study the encoding performance of DENMs. The corresponding results are given in table III. As one can clearly see, the memory consumption is similar to the encoding of CAMs but somewhat lower. This is in line with the smaller size of encoded data. As less data has to be encoded, a lower memory consumption can be expected. Additionally, the time optimized Protobuf encoding shows again the smallest memory footprint of all of the shown four encoding schemes. Furthermore, Protobuf performs worst in encoded length, however it only needs roughly three times as much space as ASN.1 compared to almost four times for CAMs.

TABLE III. PERFORMANCE RESULTS FOR DENMS.

enc. type	heap / stack TOED	heap / stack SOED	encoded length
protobuf	306 / 1532	2181 / 6580	114
ASN.1	1067 / 1252	4163 / 4184	43

Table IV gives the performance results for main memory consumption as well as encoding length for the ETSI ITS security envelope. In table IV the profile column gives the number of the applied security profile as defined in [10]. As described above in section V-A, the two cases of an envelope with and without certificate have to be distinguished for security profile number 1.

TABLE IV. PERFORMANCE RESULTS FOR THE SECURITY ENVELOPE.

enc. type	profile	heap/stack TOED	heap/stack SOED	enc. length
binary	1 no cert.	220 / 5348	same as TOED	96
	1 cert.	412 / 6420	same as TOED	178
	2	412 / 6420	same as TOED	189
	3	412 / 6420	same as TOED	186
protobuf	1 no cert.	1282 / 6452	1286 / 6452	127
	1 cert.	2065 / 9892	2065 / 9892	231
	2	2244 / 9892	2244 / 9892	243
	3	2094 / 9892	2094 / 9892	237
ASN.1	1 no cert.	1927 / 6500	5123 / 6504	87
	1 cert.	2309 / 6676	5505 / 8296	197
	2	2309 / 6676	5515 / 8296	207
	3	2309 / 6676	5515 / 8296	207

The encoding lengths for security profiles 2 and 3 are only different for the case of binary encoding and not for ASN.1 encoding, as the data field called *message type* is optional according to [10] but required according to the ASN.1 definition given in [11]. As the only difference between these two security profiles is the presence of the message type data field, this difference vanishes in the case of ASN.1 encoding. Therefore, no separate data for computation time and memory consumption is given for security profile number 3 and ASN.1 encoding, as it would be identical to the case of security profile number two. In order for a difference between the two security profiles to exist, our Protobuf definition declares the message type field as being optional.

One can see from the most right column that in all cases binary encoding clearly outperforms Protobuf in respect to achieved encoding length. Additionally, it outperforms ASN.1 encoding in three out of four cases, the only exception being the case of security profile number 1 without certificate. In this case ASN.1 encoding is only 9 bytes less than binary encoding. However, for the case with certificate and security profile one, ASN.1 requires 19 more bytes than binary encoding. Furthermore, binary encoding requires 18 bytes less for security profile number two against ASN.1 and 21 bytes less for security profile number 3, respectively.

To obtain results for the computation time we ran the measurement procedure described in section IV-B 10,000 times and computed the average of the measured outcome. Corresponding results for all processor types from table I are shown in Figures 1, 2 and 3. Please note that the vertical axis of the graph is on a logarithmic scale. Additionally, for binary encoding only four runtime measurement results are provided per processor as this scheme is not defined for encoding of CAMs and DENMs. Therefore, only the four different kinds of security envelope encoding have been measured.

An overview about the achieved runtime performance measurements on a Intel Core i7 processor is provided in Figure 1 (see also Table I). The obtained results clearly show that, for the security envelope, binary encoding is significantly faster than the two other encoding schemes. Additionally, ASN.1 encoding outperforms Protobuf for both cases of TOED w.r.t. SOED optimization.

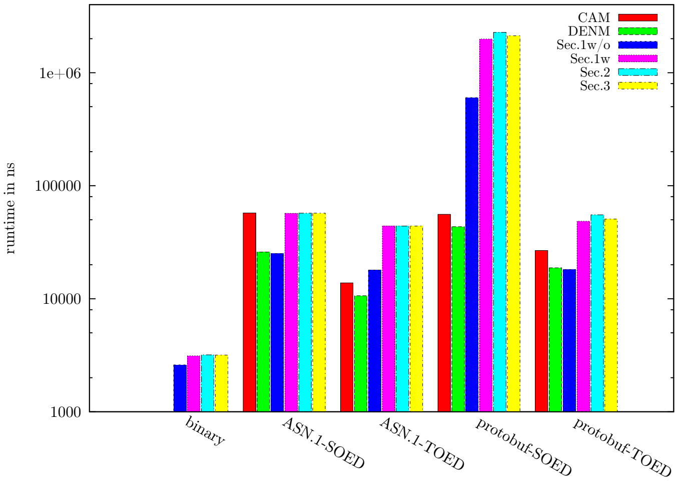


Figure 1. Runtime performance of ETSI ITS CAM, DENM and security envelope encoding on an Intel i7 processor.

An interesting result is that the difference regarding the runtime of TOED and SOED versions is significantly different for ASN.1 and Protobuf. Thereby, the results for ASN.1 encoding differ far less than the corresponding results for Protobuf do. Furthermore, the difference in runtime between TOED and SOED versions for Protobuf is much bigger for encoding of the security envelope then it is for CAM and DENM encoding. The documentation of Protobuf mentions that space optimized encoding relys on reflection instead of using dedicated data access methods [8]. From the achieved results one can expect that the optimization strategy of the used

ASN.1 library works differently, unfortunately, no in detail description regarding this point is available (see also [32]).

A significant difference between the definitions of CAMs (or DENMs) and the security envelope is the higher number of small and deeply nested data fields used for defining the security envelope ([10][11][5][6]). The achieved results depicted in Figure 1 indicate that binary as well as ASN.1 SOED encoding can handle this kind of structure better than Protobuf SOED can do. Thereby, the reflection based access scheme is likely the source of excess in runtime increase when comparing Protobuf SOED with the TOED variant.

To avoid overloading the figures, the computed standard deviation of the measured runtimes are not shown. In general the standard deviation was quite low, e.g., a value of 152 ns was found for binary encoding of the security envelope with security profile one and no included certificate. The differences between the obtained results of different encoding schemes for same encoded data content are always bigger than three times the standard deviation of the corresponding runtimes. Therefore, the achieved measurement results can be regarded as reliable.

The results obtained from the runtime measurements on a Intel Atom processor are depicted in Figure 2 (see also Table I). Comparing Figure 2 to preceeding Figure 1 one can see that except of a general increase in runtime (note the different scaling of the vertical axis of both figures), the overall results are the same for the Atom and the i7 processor technology. Due to the lower processor speed (see also Table I) such an increase in runtime can be expected. However, the increase is somewhat bigger than what can be calculated by just determining the factor one obtains from dividing the respective processor clock speeds. It is reasonable to observe an advantage in the runtime performance of the i7, which is due to the improved processor technology such as precaching algorithms, as it was introduced to the market significantly later than the Atom processor.

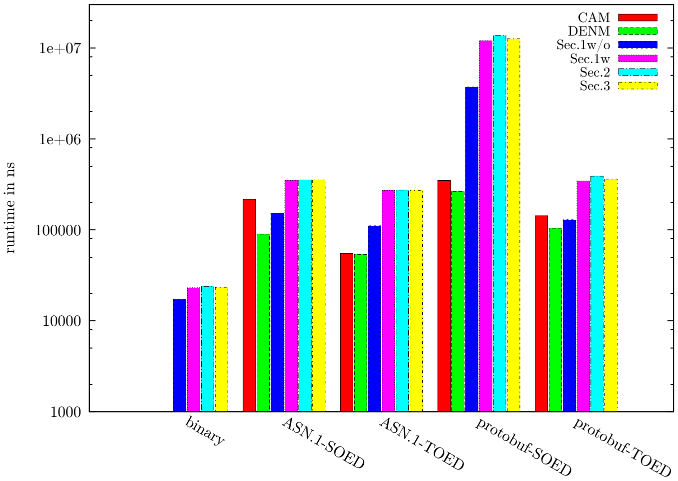


Figure 2. Runtime performance of ETSI ITS CAM, DENM and security envelope encoding on an Intel Atom processor.

Finally, Figure 3 provides the results of runtime measurements conducted using an AMD Geode processor (see also Table I). From the comparison of results shown in Figure 3 to the results given in Figures 1 and 2, one can see that the overall outcome of the performance study does not change by switching from a modern high speed processor (like the i7) to a quite old and low speed processor, like the AMD Geode.

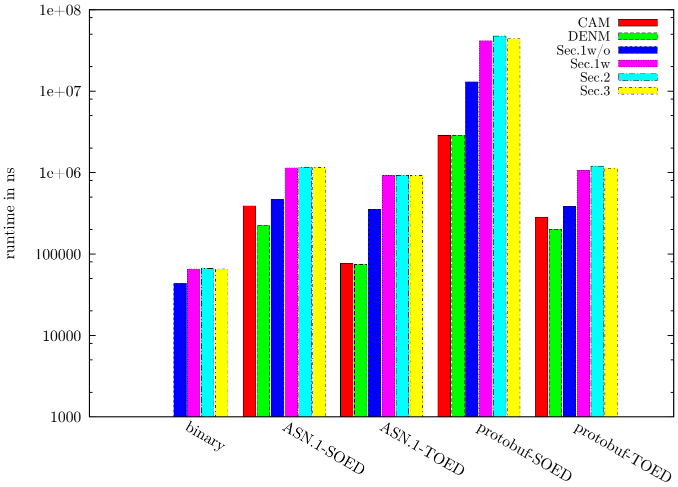


Figure 3. Runtime performance of ETSI ITS CAM, DENM and security envelope encoding on an AMD Geode processor.

Given the latter statement, we conclude that the achieved results can also be used to interpret the behaviour of the studied encoding algorithms within embedded systems using medium speed processors, nevertheless low end, low power processors may possibly behave differently.

In summary, it has been shown that regarding runtime and memory consumption, binary encoding outperforms all other studied encoding schemes running on all platforms. Only in the case of security profile number 1 without certificate, ASN.1 achieves a shorter encoding length than binary encoding. Is worth to note that, the timing interval for including a certificate in the security envelope of a CAM is equal to the default sending interval of CAMs (see [10][5]). The latter means that normally CAMs are sent with a certificate included in the envelope. Therefore, the results show that the newer standard [11] defining the security envelope using ASN.1 significantly deteriorates the performance of its encoding compared to the preceeding standard [10] using a binary encoding scheme. Furthermore, as ASN.1 does not provide a forward compatability functionality, like e.g., Protobuf would do, there is almost no reason why one should prefer ASN.1 over binary encoding. The conducted performance study also shows that Protobuf cannot be seen as a real alternative to ASN.1 for ETSI ITS data encoding. Protobuf is outperformed by ASN.1 on almost all of the selected important performance criteria on any of the platforms used and for all kinds of data types considered. Protobuf was found to be somewhat smaller compared to the respective ASN.1 counterpart only on the memory footprint parameter for some kinds of data types. Nevertheless, also in those particular cases, Protobuf is not able to outperform the binary encoding scheme.

VI. Conclusion and Future Work

Efficient data encoding schemes are required for future bandwidth-limited C2X communication. In this work, we have addressed three main performance metrics in C2X communications: encoded data length, runtime and memory footprint.

A study on these metrics for the ASN.1 and Google Protocol Buffers encoding schemes, for the time and space optimized variants, has been performed on the ETSI CAM and DENM messages as well as their Security Envelope. On the latter, we have further evaluated these metrics also for the case of binary encoding. To make the study as independent on the hardware as possible, the evaluation was done using three different processor technologies. Our work also presents the followed methodology for obtaining the mentioned performance metrics.

The results presented here show that the outlined measurement methodology is able to provide the required performance characteristics in a reliable way. Additionally, it was found that the performance of the different encoding technologies is independent of the used processor technology. From the presented results, it is clear that the performance of Google Protocol Buffers (Protobuf) is always outperformed by ASN.1 encoding w.r.t. the required encoding delay or runtime. Only in a minor amount of the studied cases, Protobuf outperformed ASN.1 encoding with regard to its memory footprint.

An important result of the conducted performance study is that binary encoding greatly outperforms ASN.1 encoding in the clear majority of cases for the security envelope. ASN.1 actually outperformed its binary counterpart with respect to encoded data length only in one of the studied cases. Regarding runtime and memory footprint: binary encoding performs significantly better in all studied cases. The latter implies that the recent shift from binary towards ASN.1 encoding (from [10] to [11]) is not justified at least by the mentioned performance metrics. Therefore, the authors propose to conduct either extensive simulations or field tests using both technologies before finalizing the corresponding standard in order to determine which encoding scheme should be used for mass rollout of the future ETSI ITS system.

Directions on future work may include an extension of the provided performance study regarding new upcoming platform independent encoding schemes like Apache Avro [12]. Such systems may provide more flexiblity regarding how to organize the encoded data. However, future research has to show whether these improvements have to be paid for by a performance degradation limiting practical usablity.

REFERENCES

[1] "Memorandum of Understanding for OEMs within the CAR 2 CAR Communication Consortium on Deployment Strategy for cooperative ITS in Europe," June 2011, v 4.0102.

[2] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Std., Rev. 5th, Nov. 2008.

[3] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," Network Working Group, IETF, RFC 4627, July 2006.

[4] Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band, ETSI European Standard 202 663, Rev. V1.1.0.

[5] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, ETSI European Standard 302 637-2, Rev. V1.3.0, Aug. 2013.

[6] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service, ETSI European Standard 302 637-3, Rev. V1.2.0, Aug. 2013.

[7] OSS Nokalva, Inc, "ASN.1 Tools for C Overview," online: http://www.oss.com/asn1/products/asn1-c/asn1-c.html, Jan. 2014, retrieved: 05.2014.

[8] Google, "Protocol Buffers - Google Developers," online https://developers.google.com/protocol-buffers/, Apr. 2012, retrieved: 05.2014.

[9] ——, "Protocol Buffers. Googles Data Interchange Format." online http://code.google.com/p/protobuf/, Jan. 2014, retrieved: 05.2014.

[10] Intelligent Transport Systems (ITS); Security; Security header and certificate formats, ETSI Technical Specification 103 097, Rev. V1.1.1.

[11] Intelligent Transport Systems (ITS); Security; Security header and certificate formats, ETSI Technical Specification 103 097, Rev. V2.1.1.

[12] J. Russell and R. Cohn, Apache Avro. Book on Demand, 2012.

[13] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow, Apache Thrift. VDM Publishing, 2010.

[14] S. Furuhashi, "MessagePack: It's like JSON, but fast and small," online: http://msgpack.org/, Jan. 2014, retrieved: 05.2014.

[15] D. Gupta, "Thrift vs. Protocol Buffers," online: http://old.floatingsun.net/articles/thrift-vs-protocol-buffers/, May 2011, retrieved: 05.2014.

[16] OSS Nokalva, Inc, "Alternative Binary Representations of the XML Information Set based on ASN.1," online: www.w3.org/2003/08/binary-interchange-workshop/32-OSS-Nokalva-Position-Paper-updated.pdf, Aug. 2013, retrieved: 05.2014.

[17] M. C. Smith, "Comparing the Performance of Abstract Syntax Notation One (ASN.1) vs eXtensible Markup Language (XML)," in In Proceedings of the Terena Networking Conference, 2003.

[18] "Using Internet data in Android Applications," online: http://www.ibm.com/developerworks/xml/library/x-dataAndroid/x-dataAndroid-pdf.pdf, June 2010, accessed: February 27th, 2014.

[19] "The gzip homepage," online: http://www.gzip.org, July 2003, accessed: February 27th, 2014.

[20] B. Gil and P. Trezentos, "Impacts of data interchange formats on energy consumption and performance in smartphones," in Proceedings of the 2011 Workshop on Open Source and Design of Communication, 2011, pp. 1–6.

[21] 2nd Generation Intel Core Processor Family, Datasheet, Vol.1, 8th ed., Intel, June 2013, doc. No. 324641-008.

[22] Intel Atom Processor Z5XX Series, Datasheet, 3rd ed., Intel, June 2010, doc. No. 319535-003US.

[23] AMD Geode LX Processor Family, AMD, Feb. 2014, doc. No. 33358E.

[24] ISO, "ISO/IEC 9945:2008 Information technology – Portable Operating System Interface (POSIX®)," May 2009, international Organization for Standardization, Geneva, Switzerland.

[25] "Debian – The Universal Operating System," online: http://www.debian.org/, Dez. 2013, retrieved: 05.2014.

[26] Fraunhofer ESK, "ezCar2X: Streamlining application development for networked vehicles," online: http://www.esk.fraunhofer.de/en/projects/ezCar2X.html, Feb. 2014, retrieved: 05.2014.

[27] R. M. Stallman and the GCC Developer Community, Using the GNU Compiler Collection, For GCC version 4.7.2, Free Software Foundation, Sept. 2012.

[28] M. T. Jones, "Kernel APIs, Part 3: Timers and lists in the 2.6 kernel," online: http://www.ibm.com/developerworks/library/l-timers-list/, Mar. 2010.

[29] G. Paoloni, "How to Benchmark Code Execution Times on Intel IA-32 and IA-64 Instruction Set Architectures," Intel, White Paper 324264-001, Sept. 2010.

[30] T. Bingmann, "malloc_count - Tools for Runtime Memory Usage Analysis and Profiling," online: http://panthema.net/2013/malloc_count/, Mar. 2013, retrieved: 05.2014.

[31] J. Seward, N. Nethercote, J. Weidendorfer, and V. D. Team, Valgrind 3.3, 1st ed. Network Theory Ltd., May 2008.

[32] OSS Nokalva, Inc, "What do I gain by using the time-optimized encoder/decoder (TOED)? What do I lose?" online: http://www.oss.com/asn1/knowledge-center/asn1-c/91.html, Feb. 2014.

ISBN: 978-1-61208-348-3