UBICOMM 2021

The Fifteenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies

ISBN: 978-1-61208-886-0

October 3 - 7, 2021

Barcelona, Spain

UBICOMM 2021 Editors

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Dmitry Korzun, Saint Petersburg State University (PetrSU), Russia

UBICOMM 2021

Forward

The Fifteenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2021) continued a series of events on the fundamentals of ubiquitous systems and the new applications related to them. The conference provided a forum where researchers were able to present recent research results and new research problems and directions in all aspects of ubiquitous techniques.

The rapid advances in ubiquitous technologies are based on more than 35 years of research in distributed computing systems, and more than two decades of mobile computing. The ubiquity vision is becoming a reality. Hardware and software components evolved to deliver functionality under failure-prone environments with limited resources. The advent of web services and the progress on wearable devices, ambient components, user-generated content, mobile communications, and new business models generated new applications and services. The conference makes a bridge between issues with software and hardware challenges through mobile communications.

Advances in web services technologies along with their integration into mobility, online and new business models provide a technical infrastructure that enables the progress of mobile services and applications. These include dynamic and on-demand service, context-aware services, and mobile web services. While driving new business models and new online services, particular techniques must be developed for web service composition, web service-driven system design methodology, creation of web services, and on-demand web services.

As mobile and ubiquitous computing becomes a reality, more formal and informal learning will take pace out of the confines of the traditional classroom. Two trends converge to make this possible: increasingly powerful cell phones and PDAs, and improved access to wireless broadband. At the same time, due to the increasing complexity, modern learners will need tools that operate in an intuitive manner and are flexibly integrated in the surrounding learning environment.

Educational services are becoming more customized and personalized, and more frequently subjected to changes. Learning and teaching are now becoming less tied to physical locations, co-located members of a group, and co-presence in time. Learning and teaching increasingly take place in fluid combinations of virtual and "real" contexts, and fluid combinations of presence in time, space and participation in community. To the learner full access and abundance in communicative opportunities and information retrieval represents new challenges and affordances. Consequently, the educational challenges are numerous in the intersection of technology development, curriculum development, content development and educational infrastructure.

We take here the opportunity to warmly thank all the members of the UBICOMM 2021 technical program committee, as well as all the 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 effort to contribute to UBICOMM 2021. We truly believe that, thanks

to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the UBICOMM 2021 organizing committee for their help in handling the logistics of this event.

UBICOMM 2021 Chairs

UBICOMM 2021 General Chair
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

UBICOMM 2021 Steering Committee
Vitaly Klyuev, University of Aizu, Japan
Wladyslaw Homenda, Warsaw University of Technology, Poland
Stéphane Galland, Belfort-Montbéliard University of Technology, France
Dmitry Korzun, Petrozavodsk State University, Russia

UBICOMM 2021 Publicity Chairs
Lorena Parra, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

UBICOMM 2021
Committee

UBICOMM 2021 General Chair

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

UBICOMM 2021 Steering Committee

Vitaly Klyuev, University of Aizu, Japan
Wladyslaw Homenda, Warsaw University of Technology, Poland
Stéphane Galland, Belfort-Montbéliard University of Technology, France
Dmitry Korzun, Petrozavodsk State University, Russia

UBICOMM 2021 Publicity Chairs

Lorena Parra, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

UBICOMM 2021 Technical Program Committee

Afrand Agah, West Chester University of Pennsylvania, USA
Wafaa Ait-Cheik-Bihi, Itris Automation by Schneider Electric, France
Mehmet Akşit, TOBB ET University, Ankara, Turkey / University of Twente, The Netherlands
A. B. M. Alim Al Islam, Bangladesh University of Engineering and Technology, Bangladesh
Sadam Al-Azani, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia
Mrim Alnfiai, Dalhousie University, Canada
Tahssin Altabbaa, Istanbul Gelisim University / Huawei Istanbul, Turkey
Nafisa Anzum, University of Waterloo, Canada
Paramasiven Appavoo, University of Mauritius, Mauritius
Mehran Asadi, Lincoln University, USA
F. Mzee Awuor, Kisii University, Kenya
Muhammed Ali Aydin, Istanbul University - Cerrahpasa, Turkey
Chiara Bachechi, University of Modena and Reggio Emilia, Italy
Matthias Baldauf, FHS St.Gallen, Switzerland
Anoud I Bani-hani, Zayed University, Dubai, UAE
Luca Bedogni, University of Modena and Reggio Emilia, Italy
Oladayo Bello, New Mexico State University, Las Cruces, USA
Imed Ben Dhaou, University of Turku, Finland
Djamal Benslimane, Université Claude Bernard Lyon 1, France
Aurelio Bermúdez, Universidad de Castilla-La Mancha, Spain
Javier Berrocal, University of Extremadura, Spain
Nik Bessis, Edge Hill University, UK
Robert Bestak, Czech Technical University in Prague, Czech Republic
Sourav Kumar Bhoi, Parala Maharaja Engineering College, India
Nadia Bouassida, Higher Institute of computer science and Multimedia, Sfax, Tunisia
Azedine Boulmakoul, Université Hassan II de Casablanca, Morocco
Lars Braubach, Complex Software Systems | Bremen City University, Germany
Erik Buchmann, Hochschule für Telekommunikation Leipzig, Germany
Joseph Bugeja, Malmö University, Sweden
Christian Cabrera, Trinity College Dublin, Ireland

Juan Vicente Capella Hernández, Universitat Politècnica de València, Spain
João Carreira, Instituto de Telecomunicações, Portugal
Debatri Chatterjee, TCS Research - Tata Consultancy Services Ltd., India
Chao Chen, Purdue University Fort Wayne, USA
Michael Collins, Technological University Dublin, Ireland
André Constantino da Silva, IFSP & NIED/UNICAMP, Brazil
Giuseppe D'Aniello, University of Salerno, Italy
Roland Dodd, Central Queensland University, Australia
Ivanna Dronyuk, Lviv Polytechnic National University, Ukraine
Wael M. El-Medany, University of Bahrain, Bahrain
Francisco Falcone, ISC-UPNA, Spain
Andras Farago, University of Texas at Dallas, USA
Olga Fedevych, Lviv Polytechnic National University, Ukraine
Niroshinie Fernando, Deakin University, Australia
Renato Ferrero, Politecnico di Torino, Italy
Olivier Flauzac, University of Reims, France
Franco Frattolillo, University of Sannio, Benevento, Italy
Stéphane Galland, Belfort-Montbéliard University of Technology, France
Crescenzio Gallo, University of Foggia, Italy
Jose Garcia-Alonso, University of Extremadura, Spain
Vassilis C. Gerogiannis, University of Thessaly, Greece
Chris Gniady, University of Arizona, USA
Mikhail Gofman, California State University, Fullerton, USA
Ann Gordon-Ross, University of Florida, USA
Clementine Gritti, University of Canterbury, New Zealand
Weixi Gu, UCBerkeley, USA
Qiang (Nathan) He, Swinburne University of Technology, Australia
Songlin He, New Jersey Institute of Technology (NJIT), USA
Wladyslaw Homenda, Warsaw University of Technology, Poland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Sergio Ilarri, University of Zaragoza, Spain
Yasser Ismail, Southern University and A&M College, USA
Jacek Izydorczyk, Instytut Elektroniki Politechniki Śląskiej, Gliwice, Poland
Mehmet Karakoç, Alanya Hamdullah Emin Paşa University - Antalya, Turkey
Reinhard Klemm, Avaya Labs, USA
Vitaly Klyuev, University of Aizu, Japan
Dmitry Korzun, Petrozavodsk State University, Russia
Konstantinos Kotis, University of the Aegean, Greece
Abhishek Kumar, University of Helsinki, Finland
Gyu Myoung Lee, Liverpool John Moores University, UK
Pierre Leone, University of Geneva, Switzerland
Wenjuan Li, The Hong Kong Polytechnic University, China
Mauro Henrique Lima de Boni, Federal Institute of Education, Science and Technology of Tocantins - IFTO, Brazil
Jianqing Liu, University of Alabama in Huntsville, USA
Xiaodong Liu, Edinburgh Napier University, UK
Xiaoli Liu, University of Helsinki, Finland
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Giuseppe Loseto, Polytechnic University of Bari, Italy
Aliane Loureiro Krassmann, Federal Institute Farroupilha, Brazil
Derdour Makhlouf, University of Tebessa, Algeria
Meriem Mandar, National School of Applied Sciences Bd Béni Amir, Khouribga, Morocco
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Stella Markantonatou, Institute for Language and Speech Processing, Greece
Francesca Martelli, Istituto di Informatica e Telematica (CNR), Italy
Weizhi Meng, Technical University of Denmark, Denmark
Philippe Merle, Inria Lille - Nord Europe, France
Daniela Micucci, University of Milano - Bicocca, Italy
Mona Minakshi, Intel Corporation, USA
Habib Mostafaei, Universität Berlin, Germany
Mjumo Mzyece, University of the Witwatersrand, Johannesburg, South Africa
Tamer Nadeem, Virginia Commonwealth University, USA
Ryo Nishide, Shiga University, Japan
Kouzou Ohara, Aoyama Gakuin University, Japan
Satoru Ohta, Toyama Prefectural University, Japan
Hamza Ouarnoughi, INSA Hauts-de-France, Valenciennes, France
K. K. Pattanaik, ABV-Indian Institute of Information Technology and Management, Gwalior, India
Giovanni Pau, Kore University of Enna, Italy
Isidoros Perikos, University of Patras, Greece
Laura Po, University of Modena and Reggio Emilia, Italy
Rashed Rahman, Georgia State University, USA
Luca Reggiani, Politecnico di Milano, Italy
Elena Renda, IIT - CNR, Italy
André Restivo, University of Porto, Portugal
Amine Rghioui, EMI - Mohamed V University, Morocco
Vanlin Robinson, University of Chicago, USA
Federica Rollo, University of Modena and Reggio Emilia, Italy
Michele Ruta, Politecnico di Bari, Italy
Khair Eddin Sabri, The University of Jordan, Jordan
Prasan Kumar Sahoo, Chang Gung University, Taiwan
Josep Maria Salanova Grau, Center for Research and Technology Hellas, Greece
Mohsen Amini Salehi, University of Louisiana at Lafayette, USA
Moid Sandhu, University of Queensland | Data61 - Commonweath Scientific and Research Organization (CSIRO), Australia
José Santa, Technical University of Cartagena, Spain
Sarwar Sayeed**,** The University of the West of Scotland, UK
Peter Schneider-Kamp, University of Southern Denmark, Denmark
Floriano Scioscia, Polytechnic University of Bari, Italy
Luca Sciullo, University of Bologna, Italy
Hugo Sereno Ferreira, Universityof Porto, Portugal
Alireza Shahrabi, Glasgow Caledonian University, Scotland, UK
Jianchen Shan, Hofstra University, USA
Ahmed S. Shatnawi, Jordan University of Science and Technology, Jordan
Shih-Lung Shaw, University of Tennessee, Knoxville, USA
Haichen Shen, Amazon Web Services, USA
Michael Sheng, Macquarie University, Australia

Copyright Information

Table of Contents

Testing Deterministic Avionics Networks Using Orthogonal Arrays

Muhammed Efe Altuntaş[1,2], Muhammed Şeker[1,2]
Pınar Kışla[1,2], Eyüp Can Akpolat[2]

[1]*Yildiz Technical University*
Dept. of Avionics Engineering
Istanbul, Turkey
[2]*Turkish Aerospace Industries, Inc.*
Ankara, Turkey
email: muhammedefe.altuntas@tai.com.tr

İbrahim Hökelek
Muhammet Selim Demir

TÜBİTAK, BİLGEM
Kocaeli, Turkey
email: ibrahim.hokelek@tubitak.gov.tr

Erhan Akdoğan

[1]*Yildiz Technical University*
Dept. of Mechatronics Engineering
[2]*Health Institutes of Turkey*
İstanbul, Turkey
email: eakdogan@yildiz.edu.tr

***Abstract*—ARINC 664 is a widely used Ethernet-based Deterministic Network (DTN) standard providing the bounded latency and jitter for safety critical applications in the avionics systems. It is not possible to generate a test suite with a feasible size without employing a formal method such as Orthogonal Arrays (OAs) if we consider all possible combinations of DTN switch configuration parameters such as the number of Virtual Links (VLs), the frame size, and bandwidth allocation gap for each connection. However, using OAs is not sufficient to generate test cases representing real-life usage of DTN switches since many combinations of VL parameters may be inadvertently eliminated by the OA generation process. In this paper, we present an approach, called *chained* OAs, where multiple applications of OAs are applied to the test factors to generate test cases that are representative of real-life usage of DTN switches. The test cases are converted into executable test scripts to be run by the AIM commercial test system. This study shows that chained OAs method is capable of reducing prohibitively large number of tests required for complex systems, while covering pair-wise combinations of all test parameters.**

***Index Terms*—orthogonal array; test generation; deterministic network; ARINC 664; avionics**

I. INTRODUCTION

Ethernet-based Deterministic Network (DTN) technologies have been developed to cope with the high speed data communication requirements of the modern avionics applications. ARINC 664, which is a commonly used DTN technology, provides the bounded latency and jitter when safety critical applications communicate with each other over the avionics networks within an aircraft. Because of their safety-critical nature, the availability of large number of configuration options and high speed operation using thousands of virtual links, testing ARINC 664 switches is a challenging task.

Prohibitively great numbers of combinations of DTN configurations necessitates that formal test generation techniques be employed to effectively reduce the number of tests. Orthogonal Array (OA) is a popular technique to optimize test suite size for systems with many features while maintaining their coverage [1]–[7]. However, using a single level OAs is not sufficient to generate test cases representing realistic operating conditions of DTN switches. Many combinations of switch configuration parameters may be inadvertently eliminated by the OA optimization process. In this paper, we present an approach, called *chained* OA*s*, where multiple and consecutive applications of OAs are employed to generate tests that are representative of real-life usage of DTN switches. Chained OAs method allows for preserving the number of combinations for a set of selected features, while eliminating redundant combinations for others as decided by test engineers. We define two stages for the chained OA method in test generation of DTN switches. First, combinations of different number of VLs that can be assigned to each switch port are generated by OAs. Then, the OA output elements are used as inputs to a set of second stage OAs to determine the remaining switch parameters such as transmission time gaps between consecutive frames, minimum and maximum frame sizes, and VL priorities. The OA outputs are generated using an open source Python library, called *oapackage* [8]. The tests generated by the chained OAs method are converted into executable scripts, which are applied to a DTN switch under test by a commercial test system, called AIM [9].

This paper is organized as follows. ARINC 664 avionics deterministic networks are outlined in Section II. The fundamental features of OAs are introduced in Section III. Section IV presents the complexity of test space for avionics deterministic networks and impact of the chained OA method to reduce the test suite size while preserving important test features. The architecture of our DTN switch test system is summarized in Section V.

II. ARINC 664 - AVIONICS DETERMINISTIC NETWORK

The most important elements of an ARINC 664 DTN include Switch (SW), End Systems (ESs), and Virtual Links (VLs) [10]. Applications running on different ESs communicate with each other using VLs that are routed by SW. The bounded latency and jitter requirements for VLs are met by offline planning of ARINC 664: the number of VLs, the source and destination ESs, the minimum and maximum frame sizes, and Bandwidth

 ISBN: 978-1-61208-886-0

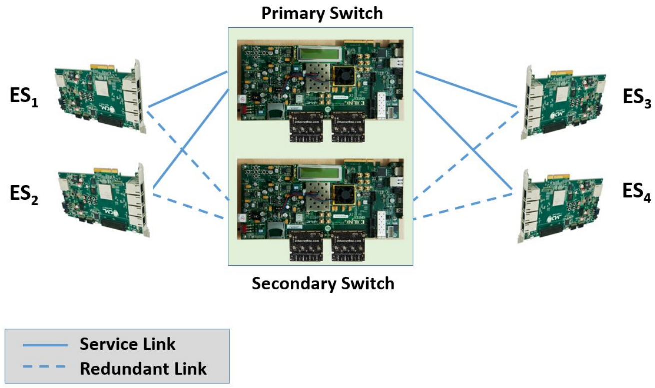

Fig. 1. An example ARINC 664 DTN with 2 SWs and 4 ESs

Allocation Gap (BAG) for all VLs [11]–[13]. The main idea behind providing deterministic communication is to limit the amount of traffic injected by each VL to the network by setting allowed frame size and BAG values, to achieve the desired delay and jitter objectives. BAG specifies the minimum transmission time between the first bits of two consecutive frames. A traffic shaping mechanism running over each VL will ensure that the amount of traffic injected by each node will be controlled and the traffic exceeding the average and peak rates will be shaped. The switch will enforce filtering and policing requirements, limiting fault propagation, and ensuring determinism.

In the ARINC 664 redundancy management architecture, the same frame is transmitted over two independent SWs and one of the received copies is sent to the application layer at the receiver. ARINC 664 does not provide a reliable transmission guarantee that all frames will be delivered to their destinations. However, in a properly planned network, the amount of congestion can be predicted and limited during the planning phase and the priority concept along with the redundancy management provide a reliable data transmission service. For the safety certification of ARINC 664, it is a common practice that the Network Calculus (NC) framework can be applied to calculate the worst-case delay and jitter bounds for a given network configuration [14] [15]. Figure 1 depicts an ARINC 664 DTN with two SWs, one primary and the other secondary, and four ESs, each of which is connected to the SWs via service and redundant VLs.

III. Orthogonal Arrays

Orthogonal Arrays (OAs) [7] are among mathematical tools to design an optimum mix of multiple variables in a set of experiments. They are effective in reducing the number of experiments, where, otherwise, a prohibitively large number of combinations would be needed [1]–[3] [7]. They are classified as the same and mixed level OAs as described in Sections III-A and III-B, respectively.

A. *Same level* OAs

In general, an OA is defined as a 4-tuple:

$$OA(N, F, L, S), \tag{1}$$

where

- N is the number of experiments (also referred to as *runs*) which represents the number of rows in an OA.
- F, called a *factor*, is the number of columns in an OA, where each column corresponds to a variable of an experiment.
- L is the number of *levels*, representing the number of distinct values that a factor can take. It is possible that, in a given experiment, different factors may possess different levels.
- S is called the *strength* of an OA ($0 < S \leq F$). An OA is designed in such a way that the number of rows needed to cover S-wise combinations of variables is given as L^S [1]. When $S = F$, all possible combinations of variables will be included in the experiment. A user can select the degree of combinations of variables to be considered by setting smaller values for S.

When all F factors have the same number of L levels, the OAs are referred to as the *same level* OA*s*. For a given experiment having F factors with L levels, there are L^F possible runs to cover all combinations of variables (assuming that all factors have the same number of levels L). OAs allow for reducing the number of runs to L^S by selecting an S value to cover S-wise combinations of variables. For example, all pairwise combinations of variables can be covered when S is set to 2. The number of runs can be further reduced as $N = \lambda \cdot L^S$ due to time limitations, where λ is a real number between 0 and 1 [5] [16]. Selection of λ is determined by testers depending on the complexity and the number of parameters of a given system under test such that smaller values are preferred for complex systems.

Let us consider an experiment with $F = 3$ factors, each with $L = 2$ levels. Without using an OA, the number of runs required to cover all possible combinations of variables is 2^3 as shown in Table I (a). We can observe that all possible situations between any two factors appear more than once in Table I (a). For example, $(f_1 = 0, f_2 = 1)$ is listed twice for factors 1 and 2 in runs 3 and 4, respectively. Orthogonal arrays eliminate such repetitions and reduce the total number of runs for experiments with large number of factors and levels. If we design an OA for this example with all factors having 2 levels and selecting the strength as $S = 2$, we will have a total of $N = 2^2$ runs. In this case, the OA will be represented as $OA(4, 3, 2, 2)$ as shown in Table I (b).

B. *Mixed level* OAs

In the OAs described in Section III-A, all factors have the same number of levels. However, it is possible that, in real-life situations, some factors may have different number of levels to be tested. For example, in an ARINC 664 switch, mapping incoming traffic flows to outgoing ports may have a large number of possible assignments to be tested (e.g., up to 24 ports), whereas the number of different priorities for a flow

TABLE I. EXAMPLE FOR SAME LEVEL OA: (a) AN EXPERIMENT WITH 3 FACTORS EACH WITH 2 LEVELS AND (b) RUNS FOR $OA(4, 3, 2, 2)$

Runs	f_1	f_2	f_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

(a)

Runs	f_1	f_2	f_3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

(b)

TABLE II. EXAMPLE RUNS FOR A MIXED LEVEL $OA(12, 2^4 3^1, 2)$

Runs	f_1	f_2	f_3	f_4	f_5
1	0	0	0	0	0
2	0	1	0	1	0
3	1	0	1	0	0
4	1	1	1	1	0
5	0	0	1	1	1
6	0	1	1	0	1
7	1	0	0	0	1
8	1	1	0	1	1
9	0	0	1	1	2
10	0	1	0	0	2
11	1	0	0	1	2
12	1	1	1	0	2

may require much fewer options (e.g., two priorities of *High* and *Low*). OAs whose factors have different levels are called the *mixed level* OA*s*. The tuple defining a mixed level OA is given as [1]:

$$OA(N, L_1^{|F_1|} L_2^{|F_2|} \dots L_v^{|F_v|}, S) \tag{2}$$

where $|F_i|$ is the number of different factors with the same number of levels L_i for $(i = 1, 2, \dots, v)$ and $\sum_{i=1}^{v} |F_i| = F$. N and S are defined in Eq. (1). For a factor F_i with the number of levels L_i, the number of possible runs will be $L_i^{|F_i|}$, which may be restricted by selecting an appropriate value of strength S as L_i^S as discussed above for the same level OAs. N, the number of runs, is selected in such a way that it is the least common multiplier of the number of $S-$wise factor level combinations. For a pairwise coverage, the combinations between any two factors can be generated as $L_i \cdot L_j$ for $i \neq j$ and $i, j = 1, \dots, F$. The generalization to $S-$wise combinations is studied in [1].

As an example, consider an experiment with $F = 5$ factors, where $F_1 = 4$ of them are with $L_1 = 2$ levels and one factor $F_2 = 1$ with $L_2 = 3$ levels. For the factors having 2 levels, the number of possible runs is $L_1^{|F_1|} = 2^4$, whereas, for the factor having 3 levels, there are $L_2^{|F_2|} = 3^1$ possibilities. Without using OAs, there are a total of $L_1^{|F_1|} \cdot L_2^{|F_2|} = 48$ runs for this example experiment. Suppose that we are interested in pair-wise combinations of these factors (i.e., $S = 2$). A mixed level OA can be defined as $(12, 2^4 3^1, 2)$, where $N = 12$ is calculated as the least common multiplier between the pair-wise combinations of any two factors. Since there are two levels (i.e., 2 and 3), the number of different combinations between any two factors can only be $2 \cdot 2 = 4$ or $2 \cdot 3 = 6$ with the least common multiplier is 12, which is the number of runs in Table II.

IV. Testing Avionics Networks Using Orthogonal Arrays

A. *Test Space for* DTN *Switch*

While traditional Ethernet switches perform switching function by routing frames from input ports to output ports by means of MAC tables, DTN switches utilize statically configured VL tables to route frames from input to output ports, where each frame carries VL identifier within its frame header to be used for switching purpose. An example DTN switch configuration, which is selected as the System Under Test (SUT), uses 4 ports and operates at the speed of 1 Gbps. The DTN switch includes filtering, policing, and switching features. A selected subset of parameters defining these features such as minimum and maximum frame lengths, BAG, and priority are shown in Table III.

For this example switch configuration, there are eight different BAG values (in milliseconds) each indicating the time slot in which a VL needs to be transmitted. For example, VLs with BAG = 2ms are guaranteed to be transmitted within an interval of 2ms. L_{min} and L_{max} values define the minimum and maximum frame sizes, respectively ($L_{min} < L_{max}$). Although the ARINC 664 standard allows for frames between 64 and 1518 bytes [11], different minimum and maximum frame sizes within these limits are permitted. For example, if a VL selects $L_{min} = 100$ bytes, the switch will not permit a 99-byte packet to be transmitted in this VL. A small sample of L_{min} and L_{max} values are shown in Table III. Determining the BAG values for all VLs, together with L_{min} and L_{max} values, will specify the maximum speed of operation for a switch. In deterministic networks, it is important to classify packets by their importance so that they are prioritized in switch queues. For this example configuration, priority values are defined as *Low* and *High*. The number of incoming and outgoing VLs into a given port are selected to be eight distinct values between 192 and 3840.

B. *Complexity of the Test Space*

The number of VLs is an important parameter possibly indicating the traffic utilization rate of the ports and eventually the determinism features of a switch. Based on Table III, there are $4 \cdot 8 \cdot 8 \cdot 8 \cdot 2 \cdot 8 \cdot 8 = 262{,}144$ different configurations for one VL. Since the maximum number of VLs in our example SUT implementation is 4096, the number of possible combinations of different VL configurations is prohibitively large. In an example configuration driven from Table III, port 2 sends 1152 VLs to output ports 3 and 4 with BAG = 64, $L_{min} = 128$, $L_{max} = 1518$ and with high priority, whereas port 3 receives 384 VLs and port 4 the remaining 768 VLs (the total number of VLs sent and received will be equal among all ports). For the parameters shown in Table III, the total number of combinations C for a given VL becomes

$$C = B^V \cdot L_s^V \cdot L_u^V \cdot p_i^V \cdot P \tag{3}$$

TABLE III. SAMPLE CONFIGURATION PARAMETERS FOR DTN SWITCH

Number of Ports	BAG (ms)	L_{min} (Bytes)	L_{max} (Bytes)	Priority	Incoming VLs to a Port	Outgoing VLs from a Port
1	1	64	1518	High	192	192
2	2	128	1400	Low	384	384
3	4	150	1300		576	576
4	8	200	1200		960	960
	16	300	1100		1152	1152
	32	400	1000		1920	1920
	64	512	900		2880	2880
	128	750	800		3840	3840

where B, V, L_s, L_u, p_i and P are the number of different values for BAG, VLs, L_{min}, L_{max}, priorities, and the number of ports, respectively. As an example with 1152 VLs for each port, the resulting complexity of the test space can be expressed using Eq. (3) as:

$$C = 8^{1152} \cdot 8^{1152} \cdot 8^{1152} \cdot 2^{1152} \cdot 4 \tag{4}$$

It is clear from Eq. (4) that it is not possible to generate a test suite with a feasible size without employing formal methods such as OAs. However, using OAs given in Table III may not generate test cases representing real-life usage of DTN switches. For example, a run generated from Table III with 2 ports, BAG = 4ms, $L_{min} = 300$, $L_{max} = 1000$, and high priority implies that all VLs will have the same number of BAG, L_{min} L_{max} and priority values. But, in a realistic scenario, each VL can independently have any of the parameter values given in Table III. In Section IV-C, we present an approach, where a sequence of OAs are employed to remedy this limitation.

C. Using Chained OAs for DTN Test System

As described in Section IV-B, the DTN network features define a prohibitively large test space. Although OAs can be used to reduce the test space significantly by limiting the coverage to a subspace with pairwise combinations of factors, directly using the parameters in Table III may inadvertently eliminate important test cases representing real-life usage of DTN switches. In this paper, we use *chained* OA*s*, where the test factors are divided into groups and OAs are generated sequentially such that the outputs of the first OA is used as the inputs for the next OA. Without loss of generality, let us first present the chained OA for two groups of input parameters, namely F_{g1} and F_{g2}, where $F_{g1} \cup F_{g2} = F$. With chained OAs, first, the parameters for F_{g1} that are considered to be more influential in system performance are selected and OA runs are generated with possibly higher values of strength S_{g1}. This way, suppression of different combinations is kept at minimum for this subset. After F_{g1} is handled, subsequent OAs are generated using the remaining parameters.

Let us consider an OA, called OA_{g1}, with N_{g1} runs, F_{g1} factors, each with level L_{g1}, and strength of S_{g1}:

$$OA_{g1}(N_{g1}, L_{g1}^{F_{g1}}, S_{g1})$$

OA_{g1} generates an orthogonal array with N_{g1} rows and F_{g1} columns as follows:

TABLE IV. DTN SWITCH VARIABLES FOR OAs: (a) THE NUMBER OF INCOMING VLs TO PORTS FOR OA_{g1}, AND (b) TEST PARAMETERS FOR VLs FOR $OA_{i,j}$

Number of VLs			
Port 1	Port 2	Port 3	Port 4
192	192	192	192
384	384	384	384
576	576	576	576
960	960	960	960
1152	1152	1152	1152
1920	1920	1920	1920
2880	2880	2880	2880
3840	3840	3840	3840

(a)

Number of Ports	BAG (ms)	L_{min} (Bytes)	L_{max} (Bytes)	Priority
1	1	64	1518	High
2	2	128	1400	Low
3	4	150	1300	
4	8	200	1200	
	16	300	1100	
	32	400	1000	
	64	512	900	
	128	750	800	

(b)

$$\begin{matrix} n_{1,1} & n_{1,2} & \dots & n_{1,F_{g1}} \\ n_{2,1} & n_{2,2} & \dots & n_{2,F_{g1}} \\ \vdots & \vdots & \vdots & \vdots \\ n_{N_{g1},1} & n_{N_{g1},2} & \dots & n_{N_{g1},F_{g1}} \end{matrix}$$

where $n_{i,j}$ is the element of run i for factor j. Using the chained OAs approach, for each element $n_{i,j}$ of OA_{g1}, we generate a secondary OA, called $OA_{i,j}$:

$$OA_{i,j}(n_{i,j}, L_{g2}^{F_{g2}}, S_{g2})$$

where F_{g2}, L_{g2} and S_{g2} are the number of factors, levels and the strength selected for the secondary level orthogonal arrays, respectively ($S_{g2} \leq S_{g1}$). After applying the chained OAs, the total number of runs will be:

$$\sum_{i=1}^{N} \sum_{j=1}^{F_{g1}} (OA_{i,j}) \tag{5}$$

where $OA_{i,j}$ generates a matrix with dimensions of $(n_{i,j} \times F_{g2})$. For the general case with R subsets of F factors, where $F_{g1} \cup \cdots \cup F_{gR} = F$, subsequent OAs can be generated in a similar manner.

In testing a DTN switch, the number of VLs is the most critical parameter since it directly impacts the determinism of traffic flows through port utilization. Therefore, the first OA, called OA_{g1}, is defined to generate runs with different number of VLs for each of the four incoming and outgoing ports. Once these runs are generated by OA_{g1}, another OA, called OA_{g2}, is needed to cover the individual parameters of each of these VLs.

In defining OA_{g1}, the range of 1 to 4096 possible VLs for each of the four ports of a switch is arranged for this

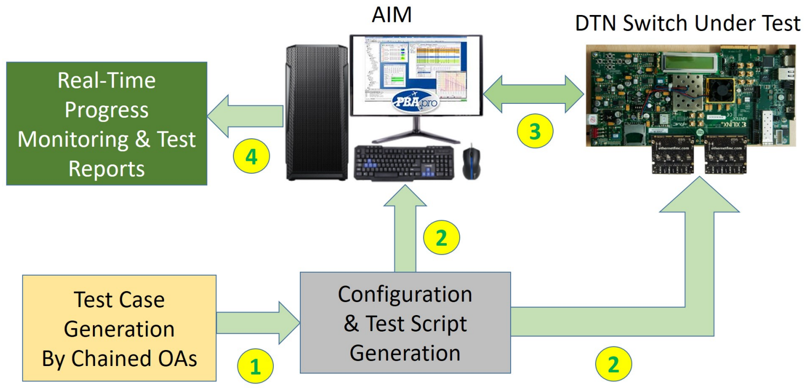

Fig. 2. Architecture and information flow of DTN test system

study by eight different values from 192 to 3840 as shown in Table IV(a): $OA_{g1}(256, 8^4, 2)$, where 256 is selected as the feasible number of runs and the strength is chosen as $S = 2$. Note that since the switch supports a maximum of 4096 VLs, any cases that exceed this upper limit in OA_{g1} are eliminated, resulting in 256 rows being reduced to 143.

VL configurations assigned by OA_{g1} to each switch port are determined by $OA_{i,j}$ using the five (5) parameters shown in Table IV(b). These parameters are selected as the most influential ones on testing switch performance: $OA_{i,j}(n_{i,j}, 4^1 8^3 2^1, 2)$, where $n_{i,j}$ is the number of runs for element at i^{th} row and j^{th} column generated by OA_{g1}.

For the sake of simplicity, this study assumes that there are 4 ports in this implementation of a DTN switch and, hence, there are a total of $143 \cdot 4 = 572$ orthogonal arrays generated based on Table IV(a):

$$\sum_{i=1}^{143} \sum_{j=1}^{4} OA_{i,j} \tag{6}$$

where $OA_{i,j}$ is a matrix with the dimensions of $(n_{i,j} \times F_{g2})$, $n_{i,j}$ ranging from 192 to 3840 from Table IV(a) and $F_{g2} = 5$ is the number of columns in Table IV(b). Therefore, 572 different matrices with the dimensions of $n_{i,j} \times 5$ are generated for this simplified version of DTN switch, where each matrix corresponds to a different switch configuration. The total number of VL configurations ranges approximately from $143 \cdot 4 \cdot 192 =$ 109,824 to $143 \cdot 4 \cdot 960 =$ 549,120. Note that there are multiple test cases in each configuration as described in Section V.

V. Implementation of DTN Test System

A. System Architecture

High level architecture of a DTN switch test system and its information flow are presented in Figure 2. First, OA model parameters representing realistic operation of the switch are determined as presented in Tables III, IV(a) and IV(b). These parameters include the number of switch ports, possible number of VL combinations to be used for each port and the set of VL parameters to be included in OA array runs. Selection of switch parameters is an important step in test generation process since there are typically multiple trade-offs between a feasible number of test cases and a desired coverage of switch features that need to be decided together by the systems engineers, developers and test engineers. The current implementation of the DTN test system generates executable test scripts using the chained OAs method described in Section IV-C, which are applied to an SUT using a commercial tool called AIM [9].

Once the OA parameters are determined (e.g., the number of runs and the factors with their levels), the next step is to construct the OAs with the desired level of strength. In step 1 of Figure 2, the OA tables were generated using an open source Python library, called *oapackage* [8], whose capabilities were suitable for the study presented in this paper. The chained OA method uses two different stages of OAs for this case study. In the first OA output, combinations of different number of VLs that can be assigned to each of the four switch ports are generated. The inputs used in this paper for generating the first OA are given in Table IV(a). Then, each element of first OA output is used as the number of runs for the second stage OA to determine the remaining parameters such as BAG, L_{min}, L_{max}, and priority as shown in Table IV(b).

The outputs of the chained OA method are now ready to be converted to SUT configurations, which define the operational characteristics of a DTN switch. Since AIM is used as the tool to apply the tests to the SUT, similar conversions are performed to prepare the corresponding AIM configurations, as depicted in step 2 of Figure 2. With these configurations, test cases are executed by generating data frames, sending and receiving them to and from the SUT ports and checking their validity with respect to the expected outputs (step 3 in Figure 2).

Because there are large number of tests to be executed on SUT, logging the test results and generating appropriate reports are important for the test system (step 4 in Figure 2). In addition, since the tests are expected to run for many hours in a continuous manner, observing the status of the test progress in real-time is an important requirement for the test system. The GUI shown in Figure 3 is an example for test monitoring capabilities of the test system, which depicts switch configurations, the number of tests that are already run,

 ISBN: 978-1-61208-886-0

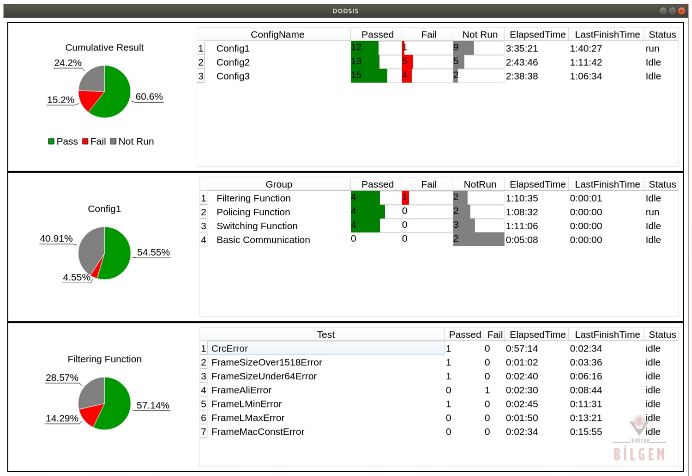

Fig. 3. GUI sample for real-time status monitoring of DTN test system

pass/failure ratios, and the expected time of the completion. By clicking on any of the tabs on the GUI, test engineer will have access to more detailed outcomes of a given test.

B. Current Status of Test System

We estimate that, using chained OAs, the test suite will deploy hundreds of thousands of VLs to verify real-life scenarios that the SUT is expected to be exposed. Currently, the test suite includes 143 switch configurations, each of which represents a different setting for distinct usage of the DTN switch in the field. Each switch configuration will be run continuously for 3-4 hours, representing a realistic load for typical avionics applications.

VI. Conclusions and Future Work

Test generation for network switches used in safety critical avionics systems is a significant challenge since these systems possess a large number of configuration options and high speed operation using thousands of virtual connections. There are prohibitively large number of test cases if one wishes to include all possible combinations of switch configuration parameters. One popular tool to reduce number of tests is OAs, especially suitable for complex industrial applications. In this paper, we presented chained OAs method, where test parameters are divided into multiple groups and the OA generated for a group is used as an input for the next group. Independent of implementation or employed test architecture, chained OAs method can be applied to any SUT with large number of test parameters. For test generation of DTN switches, we defined two groups, where the first group deals with different number of virtual connections to be assigned to each switch port and the second one determines the remaining switch parameters such as transmission time gaps, minimum and maximum frame sizes and priorities. The test cases generated by the chained OAs method were converted into executable test scripts, which were applied to an SUT by a commercial test system.

Future extension of this work includes development of OAs that incorporate parameters defined for single and multiple end systems and reliability test suites for interoperability of switch and end systems for avionics applications. We expect that these extensions will be capable of representing real-life usage of deterministic avionics network systems and, hence, provide more effective test cases. We also plan to present a formal study to explicitly quantify the complexity and efficiency trade-off achieved by the chained OAs approach.

References

[1] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. Springer, ISBN 978-1-4612-1478-6, 1999.

[2] M. S. Phadke, *Quality Engineering Using Robust Design*. Prentice-Hall, ISBN 0-13-745167-9, 1989.

[3] K. Gopalakrishnan, D. Stinson, and D. Cheriton, "Applications of Orthogonal Arrays to Computer Science," in *Proc. of ICDM.*, 2006, pp. 149 – 164.

[4] G. Taguchi and S. Konishi, *Taguchi Methods: Orthogonal Arrays and Linear Graphs - Tools for Quality Engineering*. American Supplier Institute, Center for Taguchi Methods, ISBN 978-0-94-1243-01-8, 1987.

[5] R. Kacker, E. Lagergren, and J. Filliben, "Taguchi's Orthogonal Arrays are Classical Designs of Experiments," *Journal of Research of the National Institute of Standards and Technology*, vol. 96, no. 5, p. 577, 1991.

[6] E. D. Schoen, P. T. Eendebak, and M. V. M. Nguyen, "Complete Enumeration of Pure-Level and Mixed-Level Orthogonal Arrays," *Journal of Combinatorial Designs*, vol. 18, no. 2, pp. 123–140, 2010, retrieved 15.05.2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcd.20236

[7] S. Yazar, "Reconstruction of the Taguchi Orthogonal Arrays with the Support Vector Machines Method," *Balkan Journal of Electrical and Computer Engineering*, vol. 9, pp. 129 – 137, 2021.

[8] P. Eendebak, "The Orthogonal Array Package," retrieved 04.04.2021. [Online]. Available: http://www.pietereendebak.nl/oapackage/index.html

[9] "AFDX / ARINC 664P7 Modules with a Difference," retrieved 15.05.2021. [Online]. Available: https://www.aim-online.com/products-overview/modules/afdx-arinc664p7/

[10] J.-P. Moreaux, "Data transmission system for aircraft," U.S. Patent US6925088B1, Nov. 1999.

[11] *ARINC 664, P7: Avionics Full Duplex Switched Ethernet (AFDX) Network*. INC Aronautical Radio, 2005.

[12] E. C. Akpolat et al., "Genetic Algorithm Based ARINC 664 Mixed Criticality Optimization Using Network Calculus," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[13] M. N. Durceylan, O. F. Gemici, I. Hokelek, and H. Asmer, "Reconfigurable arinc 664 end system implementation on fpga," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020, pp. 1–4.

[14] M. Yeniaydin et al., "Priority Re-assignment for Improving Schedulability and Mixed-Criticality of ARINC 664," in *2021 IFIP Networking Conference (IFIP Networking)*, 2021, pp. 1–6.

[15] A. Bouillard, M. Boyer, and E. Le Corronc, *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons, ISBN 978-1-119-56341-9, 2018.

[16] H. Bayrak and A. Alhan, "On the Construction of 2-Symbol Orthogonal Arrays," *Hacettepe Journal of Mathematics and Statistics Volume*, vol. 31, pp. 57–62, 01 2002.

A Review of Application Protocol Enhancements for Internet of Things

Fatma Hmissi
Cristal Laboratory, ENSI
Manouba University Campus
Manouba 2010, Tunisia
e-mail: fatma.hmissi@ensi-uma.tn

Sofiane Ouni
Cristal Laboratory, ENSI
Manouba University Campus
Manouba 2010, Tunisia
e-mail: soufiane.ouni@insat.rnu.tn

***Abstract*—Nowadays, all sectors utilize devices that are part of the Internet of Things (IoT) for the purpose of connecting and exchanging information with other devices and systems over the Internet. This increases the diversity of devices and their working environments, which, in turn, creates new challenges, such as real-time interaction, security, interoperability, performance, and robustness of IoT systems. To address these, many applications protocols were adopted and developed for devices with constrained resources. This paper surveys communication protocols divided according to their goals along with their merits, demerits, and suitability towards IoT applications. We summarize the challenges of communication protocols as well as some relevant solutions.**

***Index Terms*—Internet of Things (IoT), Messaging Protocol, Device Management Protocol, Service Discovery Protocol, Constrained devices, Interoperability, Security, Quality of Service (QoS).**

I. INTRODUCTION

The Internet of Things (IoT) refers to the physical objects that are connected to the Internet. The IoT is used to make people's lives and businesses easier in many areas. Generally speaking, there is no standard architecture for the Internet of Things systems, but what is certain is that all architectures are composed of several parts which interact and communicate with each other without human intervention. The Internet of Things connects the real world of things to the virtual world of networks and the cloud.

Figure 1 introduces a typical scenario of the IoT system where the interaction between the different parts is clearly presented. The Devices are the most important elements in the system. They are equipped with embedded sensors, actuators, and controllers to interact with the physical environments to collect information or to change the actual status. A device can exchange data either with other devices or with datacenter, the Cloud, or other servers. The Gateway represents a physical entity that is composed of several electronic devices. The main purpose of the Gateway is to connect to a different networks having different typologies. It contains software that translates the protocols to establish communication between the things and the network. The number of connected devices is expected to grow rapidly, with a predicted 75 billion devices worldwide expected to be connected to the Internet by 2025 [1]. This great number of connected devices is expected to generate unlimited data. As a result, an enormous amount of data to be stored, processed, and made available in a continuous, efficient, and easily interpretable manner is growing rapidly, which puts a lot of pressure on the Internet infrastructure. To solve this problem, companies combined the capabilities of IoT and cloud computing. The technology of cloud computing assists in alleviating the pressure on the Internet infrastructure by storing, processing, and transferring data to the Cloud instead of to the connected devices. Many platforms, called IoT Cloud Platforms, exploit Cloud Computing features to provide IoT services. For this purpose, a number of open sources and proprietary IoT platforms have been proposed and implemented to provide many efficient and easy IoT services, such as data collection, storage, analysis, monitoring, control, and management of connected things. Today, more than 300 IoT platforms are available on the market [2]. Mobile and Web applications make the IoT very user-friendly. A mobile application is a software application that is created to run on mobile devices especially those that are small and wireless. A Web application is a software application that is hosted on a server and accessible through a Web browser. Mobile and Web applications allow users to perform a set of specific functions and tasks on the Internet. These functions and tasks are summarized in the connection, monitoring, control, and management of connected objects.

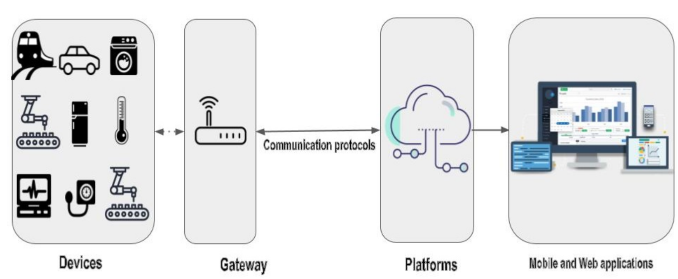

Fig. 1. IoT Basic Architecture.

Communication Protocols are used to allow the connectivity for data exchange between physical or virtual entities, e.g., Devices and the Cloud, by defining rules and constraints where several requirements must be taken into account by these protocols in order to be successful within the IoT systems. Those protocols are divided into three categories. This classification is based on the purposes of the protocols. It is possible to build an IoT system with the typical communication protocols based on classical HTTP Web requests even if they do not have certain requirements. However, they are no longer the right choice for Internet of Things, for this purpose, other messaging protocols (such as Message Queuing Telemetry Transport (MQTT) [3] [4], Constrained Application Protocol (CoAP) [5] [6], Data Distribution Service (DDS) [7] [8] [9] [10]) have been implemented. It is seen that these messaging protocols are the best way to create an interconnection between heterogeneous devices and IoT systems, however, this is not sufficient to resolve all connectivity issues. Indeed, a huge number of connected devices are deployed in remote, hostile and hard to reach places, which makes their configurations and maintenance are difficult. Many solutions are proposed to provide device management necessity. For example, Perumal et al. [11] proposed a lightweight IoT device management framework for smart home services. Mechanisms for discovery are important to use the services of the Internet of Things. Service discovery is a process of automatically locating the appropriate services. Ahmed et al. [12] proposed a secured service discovery technique for the Internet of Things.

To sum up, application protocols are subdivided into three types, namely, (i) for messaging. (ii) for device management, and (iii) for service discovery.

The weaknesses of the current solutions have motivated the improvement of the existing protocols that seek to enhance the Internet of Things ecosystems' performance and avoid faults.

Thus, this work aims at presenting an extensive survey about the existent communications protocols that can be used in IoT applications. Different from the current existing surveys in the literature, this work does not only consider existing and well-known base protocols, but also all relevant solutions that have been introduced during recent years. The main contributions of this paper are summarized as follows:

- Overviews of the initial communication protocols adapted for application.
- Overviews of the initial communication protocols adopted for device management.
- Overviews of the initial communication protocols adopted for service discovery.
- Identifies the problems most studied by the existing protocols in IoT scenarios.
- Reviews of the studied solutions that improve existing protocols.

The remainder of this paper is organized as follows: Section II presents the communication protocols for the Internet of Things applications. Section III lists the communication protocols for device management. Section IV sums up the communication protocols for service discovery. Section V introduces the challenges for the communication protocols and surveys the recent approaches to the protocols enhancement. Section VI concludes the paper.

II. Messaging Protocols

IoT cannot rely on a single protocol for all needs [13]. Consequently, several of available messaging protocols are chosen for various types of requirements of the IoT system [14]. Thus, in the rest of this section, the most relevant protocols are cited with their descriptions.

Message Queuing Telemetry Transport (MQTT) is a lightweight [3] [4] and flexible [15] messaging protocol. MQTT uses different approaches for routing mechanisms, such as one-to-one, one-to-many, or many-to-many, making the connection between IoT and Machine-to-Machine (M2M) to connected devices/applications possible [15]. M2M is used to provide communications between machines without human intervention. MQTT is designed as a publish-subscribe model [15], using TCP as transport layer protocol. The process of message exchange in MQTT consists of multiple clients connected to a central broker which is a server running somewhere in the Internet network [16]. These clients' act as senders that publish a message on a particular topic that could be considered as message subject [15] or as a receiver that is subscribed to a topic registered at the broker. Every time the MQTT Broker gets a new publish message to a specific topic, it broadcasts this message to the entire subscribed receivers. The MQTT protocol is used between devices and Cloud, but it cannot be used between devices. Another disadvantage is that MQTT does support multi-cast.

Constrained Application Protocol (CoAP) [5] [6] is mainly used in a constrained environment with constrained devices and constrained networks. It uses unicast and multicast request-response model for interaction between multiple clients and multiple servers by sending request and response messages using a URI with GET, POST, PUT and DELETE actions over UDP to keep things lightweight. The main merit of the CoAP protocol is that it can be used with constrained devices in interaction device-to-device. This protocol cannot be used in asynchronous communication because it does not support publisher-subscriber architecture. Also, it does not support broadcast. The clients cannot use a topic to send and respond to messages.

Data Distribution Service (DDS) [7] [8] [9] [10] is used for real-time and industrial M2M communications, running over both TCP or UDP. DDS supports broker-less architecture where it uses a publish-subscribe model for interaction between entities without the use of a Broker. The tasks of a broker are handled by Data Writers (DW) and Data Readers (DR). The main advantages of DDS protocol are that the data usage is fundamentally anonymous, since the publishers do not enquire about who consumes their data, and the probability of system failure is limited (system more reliable) because there is no single point (no broker) of failure for the entire system [7]. The most remarkable disadvantage of DDS is that it is designed for Industrial application (IIoT) with considerable hardware resources. This makes the implementation for constrained devices that need a Lightweight protocol even harder. The other disadvantage is related to the increase of the communication workload by the publishing of data even if there are no interested subscribers [7].

Advanced Message Queuing Protocol (AMQP) [7] [8] is designed as a publish-subscribe model which uses TCP as transport layer protocol. Mainly, it has three components, Publishers, Subscribers and, both parts of an AMQP Broker are Exchanges of Message queues. The Publisher creates a bare message and sends it to the Exchanges components that are used to forward the messages to appropriate message queues using the routing keys contained in messages. The latter can be stored into message queues before forwarding them to Subscribers. If there are more subscribers interested in a particular message, the broker can duplicate the messages and send their copies to multiple queues waiting for annotated messages from subscribers. The main advantage of the AMQP protocol is that it could be used in device-to-device, device-to-Cloud, and Cloud-to-Cloud interaction. But its main disadvantage is that the publishers and subscribers cannot publish and subscribe using the topic.

As introduced in [17] [18] [19], the eXtensible Messaging and Presence Protocol (XMPP), also known as Jabber, is a standard initially designed for instant messaging and exchange of messages between applications no matter which operating system they are using in IoT. It is designed to allow users to send messages in real-time and manage the presence of the user. XMPP supports Publish-Subscribe and Request-Response models with TCP transport protocol. To exchange messages between clients and servers XMPP uses streams of stanzas. XMPP is a text-based protocol where XMPP stanzas [17] [18] [19] are Extensible Markup Language (XML) messages exchanged between clients. The main advantage of the AMQP protocol is that could be used in device-to-device, device-to-Cloud interaction. As AMQP, DDS, and CoAP, topics are not used to publish and subscribe with XMPP.

IoT messaging protocols are designed for communication. All the cited protocols lack IoT device management and service discovery procedure. Our depth study allows us to conclude that the messaging protocol with the greatest impact on the IoT is the MQTT. The MQTT protocol is the lightest, the most robust and the one that has the least overhead.

III. Device Management Protocols

A huge amount of heterogeneous devices, which are integrated into IoT, need to be (re)discovered, reconfigured, and maintained to fix security issues, deploy new features, or recover from their failures. It is possible to manage devices with the IoT messaging protocols by inventing new building blocks. It seems that these protocols are no longer the right choice for device management because of the high cost of development, where for every new management feature, a new block should be developed. To solve this problem, new protocols known as Device Management Protocols are proposed. A device management protocol enables the abstraction of an IoT/M2M device as a managed object to make the management of the device much easier [20].

Open Mobile Alliance Device Management standard [21] [22] [23] named as OMA-DM and designed by Open Mobile Alliance for device management, is used for Terminal M2M devices and Mobile terminal devices, e.g., Mobile phones, Smartphones, Tablets, laptops. Mobile network operators and enterprises use OMA-DM to manage mobile devices remotely. The main features of OMA-DM are: read and write configuration or monitoring nodes, read and set parameter keys and values, Firmware Update Management Object (FUMO), software components management object (SCMO) that means install, upgrade, or uninstall software elements. OMA-DM has several demerits. The OMA-DM protocol is designed only for no constrained and fixed devices. Another disadvantage is that OMA-DM cannot be used for industrial applications and cannot be built on top of the MQTT protocol. The disadvantage before the last is that it supports the only XML serialization format and it does not support either Binary, Plain text, or TLV and JSON serialization format. The final disadvantage refers to the no support of interoperability.

Lightweight M2M [24] [25] (LwM2M) is a client-server standard developed by the Open Mobile Alliance (OMA). It is an OMA-DM successor. The LwM2M is a standard device and service management built on top of CoAP to ensure remote management and configuration of constrained and powerful devices. It can beneficiate from efficient communication in M2M and IoT environments over UDP and SMS bearers. So, SMS can be used for waking up the device or any GET, POST, and PUT request. The LwM2M main features are: device monitoring and configuration, server provisioning (bootstrapping) and firmware upgrades. There are numerous advantages of the LwM2M protocol. The most remarkable advantage of LwM2M is that it could be used with fixed and mobile-constrained devices. Another advantage refers to the support of the industrial application and interoperability. The most important disadvantages of LwM2M are: cannot support XML serialization format, cannot be built on top of MQTT and cannot be used in telecommunication applications.

The Broadband Forum defined CPE WAN management protocol (CWMP) that is used for remote management of home and business network devices, such as modems, gateways, routers, and VOIP phones (see Technical report 069 [26] [27] known as TR-069). The main capabilities of this protocol are firmware management, auto-configuration, dynamic service provisioning, software module management, status monitoring, performance monitoring, and diagnostics. The TR-069 uses SOAP/HTTP protocol for communication between network devices called the Customer Premises Equipment (CPE) and central server called the Auto-Configuration Servers (ACS). The CPE and ACS present the main components of this protocol. TR-069 has the same disadvantages as OMA-DM.

IoT devices management protocols are not oriented for communication and service discovery features. Our depth study allows us to conclude that the device management protocol with the greatest impact on the IoT is the LwM2M. The overhead, footprint, and server load of the LwM2M are lighter than TR-065 and OMA-DM protocols, while the response time of LwM2M is faster than TR-065 and OMA-DM protocols.

IV. Services Discovery Protocols

Service Discovery Protocols (SDPs) are communication protocols that provide mechanisms to help clients to discover services available on the network. There are several SDPs for the IoT environment. This section focuses on the most known SDPs by introducing the following protocols: mDNS, SSDP. Multicast Domain Name System (mDNS) [28] [29] [30] is an open protocol defined by IETF which requires minimal configuration, based on the Internet Protocol (IP) and the User Datagram Protocol (UDP). An mDNS client can discover a thing's endpoint by resolving its hostname to an IP address. An mDNS client has to send an IP multi-cast query message over the network. The message calls the host with that name to reply and identify. Once the host receives the message, it replies via a multi-cast message that contains its IP address. All nodes in the network receiving that multi-cast message update their mDNS caches accordingly. This protocol, coupled with DNS-based Service Discovery (DNS-SD), offers the flexibility required by environments where it is necessary to automatically integrate new devices and perform DNS-like operations without the presence of a conventional DNS server.

The Simple Service Discovery Protocol (SSDP) [28] [29] [30] is an open protocol, based on IP, UDP, and SOAP (Simple Object Access Protocol) [28] [29] [30]. An SSDP client discovers SSDP services by multi-casting a discovery request to the SSDP multicast channel and port. SSDP services listen on that channel until they receive a discovery request that matches the service they offer, then they respond using a unicast response. This protocol—included in the Universal Plug-and-Play (UPnP) architecture—makes it possible to transparently plug and play devices without the need for any manual configuration.

V. Challenges and Enhancements of Communication Protocols

Throughout the latest studies, it has been shown that protocols limitations and drawbacks do exist. Among these, we highlight communication protocols challenges:

- Real-time and industrial communication issues.
- Not suitable for constrained devices.
- Interoperability issues.
- Security issues.
- Quality of Service (QoS) issues.

Motivated by the presented issues, several new solutions have emerged recently. In this section, an overview of studies focusing on the improvement of existing and well-known base protocols are divided and presented according to their proposals. Table I summarizes the existing studies of some widely efficient and recently enhanced approaches for application layer protocols in IoT environment.

Several applications in IoT fields, such as medical, factory, and transportation are time-sensitive applications. Mostly, the delays of communications between the different parts of the IoT systems are in-bounded. Therefore, the real-time requirement is one of the challenges of communication protocols. Most IoT solutions involve time constraints to gather and process information, make decisions, and deliver actions that system components must perform. When time restrictions are present, the system is said to be real-time if at least one of the tasks is performed but it must be executed before a certain deadline. XMPP and DDS protocols are designed for real-time communication. Even though the other protocols, such as MQTT and CoAP, have received a lot of attention due to their simplicity and scalability, none of them support real-time interactions. To address this, many approaches are proposed to add enhancement to applications protocols without changing their simplicity and scalability. Kim et al. [31] propose to integrate MMS and MQTT protocol for Internet of Things industrial applications. Konieczek et al. [32] presented a lightweight Java implementation of the Constrained Application Protocol called jCoAP that enables CoAP-based communication for embedded devices with comparably small latencies (real-time interaction).

IoT devices are constrained. They have limited capabilities, memory, and energy. And the use of heavy communication protocols on these devices reduces the performance of IoT communication. i.e shut down the devices quickly, increase the delay of communication. Power consumption is one of the most constrained aspects of IoT devices, "which makes the most powerful applications protocols not suitable for Internet of Things ecosystems. Although MQTT is a lightweight protocol, it has its drawbacks for extreme environments. MQTT clients must support TCP and would normally keep an open connection to the broker at all times where packets loss and connection drop rates are high or computing resources are scarce. Moreover, topic names are often long ones which make the header bigger and use significant bandwidth and power as well. To address this, many variations and enhancements are proposed. First, Query Telemetry Transport for Sensor Networks (MQTT-SN) was created [33], which runs over UDP. UDP is mainly used for sensor nodes and devices with low computing performance. MQTT-SN requires additional gateways to connect the clients to the MQTT broker over UDP which can be suitable for devices with multicast support. A modification to MQTT-SN with additional security elements adopted from DTLS is proposed to replace the DTLS protocol to enable shorter lightweight packet headers [7]. Akintade et al. [34] proposed another architecture to facilitate the development of energy-efficient and low-cost IoT solutions, namely, the aMQTT architecture. The architecture is based on the existing MQTT architecture and the low cost ESP8266 IoT hardware platform. Second, many enhancement solutions were added to the MQTT-SN protocol to increase its performance especially in extremely lossy channels where re-transmission creates a huge overhead in terms of power consumption, delay, and processing. Alshantout et al. [35] created MQTT-SN with LT (MQTT-SN-LT). They aim to use Luby Transform Codes (LT) with the MQTT-SN-QoS1 protocol without changing the protocol itself. The authors [36] proposed to add Network Coding to an MQTT-SN network.

Application protocols were not designed with security in mind [37]. They are based on common security solutions, such as DTLS and TLS which are not sufficient for optimal security as they reduce the performance of IoT systems. To go further, these solutions are very heavy for constrained devices. Added to that, certain attacks are no longer covered by these solutions which require the development of new standards to improve the security levels of each protocol. In the rest of this section, several attacks and problems are cited as well as their solutions. The flow of the distributed messages between the users of application protocol based on Publisher-Subscriber models is insecure. Wherein authentication layer authenticating credentials are sent in plain text and some form of encryption should be used. In authorization layer all users connected to the broker are listening to a Topic and receiving all the information. For authentication layer, ChaCha20- Poly1305 AEAD solution is proposed as a lightweight security scheme for MQTT/MQTT-SN communication in [38]. Since, in a MQTT environment, a user in the broker's access is authorized to access all information, after their connection to the broker the user is listening to a Topic and receiving all the information. A new solution of certified authority is opted for in [39] to generate two kinds of certificates, the first one for the client and the second one for the Topics. Denial-of-sleep attacks aim at depriving victims of devices entering low-power sleep mode. Since the CoAP protocol suffers from this type of attack where Internet-located attackers can force IoT devices that run CoAP servers to expend much energy by sending lots of CoAP messages to them, a new solution is proposed by adding a block to filter the CoAP messages en route before entering the network [40].

Interoperability is meant to make communication among heterogeneous devices and software applications from different vendors possible. Interoperability has four dimensions: technical, syntactical, semantic, and organizational interoperability [41]. There is no compatibility in inter-communication between application protocols. Messages are not supposed to be exchanged. Thus, we need new standards to convert communication protocols and to enlarge the protocol's capabilities for larger interoperability. A new efficient application layer gateway that converts MQTT messages into HTTP is proposed in [42]. To address the problems of the interconnection of embedded systems in networks, the authors of [43] aim to dynamically model and create links between MQTT brokers based on multi-agent systems to establish the highest level of connectivity for brokers to ensure maximum transmission of messages to subscribing clients. Since there is no compatibility between the sensors, where each sensor for example has its own data display units, there is a need for common semantics for these sensors. To solve this problem, several standards have been developed to ensure that the precise meaning of exchanged information can be understood by any other application that was not initially developed for that purpose. A semantic data extraction implementation over MQTT for Internet of Things centric wireless sensor networks was introduced [44].

The Quality of Service (QoS) characterizes the quality of communication links between nodes. Generally, it is the capacity to carry the traffic between nodes in the best condition, such as in terms of availability, packet loss rate, and throughput. So, to ensure good communication it is recommended to define clearly the quality metrics and to enhance the communication protocols accordingly. Quality of Service is the strength of application layer protocol, that represents the ability to configure the performance and reliability of the network. Some protocols do not define any QoS level which reduces their performance, while others, such as CoAP, MQTT, and DDS define different levels of QoS which address different requirements, such as message delivery, timing, loose coupling, and fault tolerance. As MQTT provides only three levels of QoS for different classes of traffic, so many drawbacks arise. Firstly, the traffic flow between subscribers and publishers is not controlled since publishers send data to broker and broker forwards it to subscribers which could increase the number of packet losses and delays. A new flow control mechanism is designed to overcome the flow control problem of MQTT where the publisher can overwhelm the subscriber [45], [46]. Secondly, MQTT does not support the urgency of the message. Hence, normal and urgent messages are processed with the same priority. Many approaches are designed for this purpose. Hwang et al. [47] proposed a new method to expand the functions of the MQTT to transmit urgent messages first by creating a U-Mosquitto broker capable of processing urgent messages. However, MQTT protocol has vulnerability to maintain order between messages, which is very important in some home automation, such as controlling gas valve. Hwang et al. [48] designed and implemented a reliable message transmission system using MQTT protocol to maintain messages order. The absence of a standard for controlling the number of messages received is such a serious problem where the subscriber devices are forced to receive all messages even if they do not need to receive them frequently. To solve this problem, reducing the delivery of unnecessary messages is the best solution . Hwang et al. [49] focused on the MQTT protocol that is currently used to deliver messages between IoT devices and proposed the concept of Reception Frequency Control (RFC), which is designed to control the frequency at which subscribers receive messages.

VI. Conclusions

Application communication protocols in IoT ecosystems are used to successfully interact between IoT devices and servers / Clouds that process the information collected. Application protocols specific to IoT have been developed to meet the requirements of devices with limited resources, and those of networks with low bandwidth and high latency. However, es-

 ISBN: 978-1-61208-886-0

TABLE I
SURVEYS ON COMMUNICATION PROTOCOLS CHALLENGES AND ENHANCEMENT.

Challenge	Focus	Protocol	References
Real-time communication	Industrial application	MQTT	[31]
	IoT based system	MQTT	[50]
	Embedded devices	CoAP	[32]
	Prototype Medical Instruments Applied to Neurodegenerative Disease Diagnosis	MQTT/AMQP	[51]
Constrained devices	Power saving	MQTT	[33], [34]
	Power saving	MQTT-SN	[35], [36]
	Power saving	CoAP	[52]
	Decrease the computational complexity of the clients	MQTT	[53]
Interoperability	Technical Interoperability	MQTT/HTTP	[42]
	Technical Interoperability	MQTT	[43]
	Syntactical interoperability	All protocols	[54]
	Semantic interoperability	MQTT	[44]
Security	Authentication	MQTT/MQTT-SN	[38], [55]
	User authority to information access	MQTT	[39]
	User Registration	MQTT	[56]
	Denial-of-sleep attacks	CoAP	[40]
Quality of services	Control the traffic flow between the subscribers and publishers	MQTT	[45], [46]
	Maintain message order	MQTT	[48]
	Transit urgent message first	MQTT	[47]
	Reduce the delivery of unnecessary messages	MQTT	[49]
	Data Delivery in Mobile Scenarios	MQTT	[57]
	Network Congestion Control	CoAP	[58]
	Object Discovery	CoAP	[58]

tablishing low-cost communications is not enough. These protocols must allow data to be exchanged and this data must be understood by the entities of different types which receive them. The interoperability of distributed applications is defined as the ability of success for the IoT thanks to a set of application protocols for users to communicate and exchange data and services, wherever they are in the world regardless of the origin of the equipment they use.

The application protocols are still quite young and have yet to demonstrate their reliability. However, several challenges remain to be taken up to further broaden their acceptances. These challenges are related to the drawbacks of application protocols. The limits of these protocols have an important size. They can be summarized in the following points: not suitable for real-time and industrial application, not suitable for constrained devices and lack of interoperability, security mechanisms and Quality of Service (QoS).

In this paper, we surveyed the most suitable communication protocols for the Internet of Things and related challenges of IoT issues by introducing relevant and recent approaches for improving the performance of application layer IoT systems.

The studied application protocols, in general, are based on MQTT, CoAP applications protocols. This is justified due to MQTT and CoAP being already the most suitable solutions in IoT since they are initially better attending to the application layer criteria: message size, overhead, power consumption, resource requirement, bandwidth, and reliability.

REFERENCES

[1] IRENA, "Internet of things innovation landscape brief," International Renewable Energy Agency, Abu Dhabi, 2019.

[2] M. Atiquzzaman, M. Noura, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, pp. 796–809, 07 2018.

[3] G. Salazar Chacón, C. Venegas, M. Baca, I. Rodriguez, and L. Marrone, "Open middleware proposal for iot focused on industry 4.0," 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), pp. 1–6, 11 2018.

[4] S. Nuratch, "Applying the mqtt protocol on embedded system for smart sensors/actuators and iot applications," 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, no. 1, 2018.

[5] M. Iglesias Urkia, A. Orive, and A. Urbieta, "Analysis of coap implementations for industrial internet of things: A survey," Procedia Computer Science, pp. 188–195, 12 2017.

[6] M. Mónica, G.-R. Carlos, and C. Celeste, "Performance evaluation of coap and mqtt-sn in an iot environment," *Proceedings*, vol. 31, p. 49, 11 2019.

[7] S. Jaikar and R. Kamatchi, "A survey of messaging protocols for iot systems," International Journal of Advanced in Management, Technology and Engineering Sciences, 02.

[8] J. Dizdarevic, F. Carpio, A. Jukan, and X. Masip, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," ACM Computing Surveys, 04 2018.

[9] T. Sultana, "Choice of application layer protocols for next generation video surveillance using internet of video things," IEEE Access, pp. 41607 – 41624, 2019.

[10] Z. B. Babovic, J. Protic, and V. Milutinovic, "Web performance evaluation for internet of things applications," IEEE Access, 2016.

[11] T. Perumal, S. K. Datta, and C. Bonnet, “Iot device management framework for smart home scenarios,” in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 54–55, 2015.
[12] A. Ahmed, “Secured service discovery technique in iot,” Journal of Communications, vol. 14, pp. 40–46, 01 2019.
[13] N. Naik and P. Jenkins, “Web protocols and challenges of web latency in the web of things,” 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 845–850, aug 2016.
[14] N.Naik, “Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http,” 2017 IEEE International Systems Engineering Symposium(ISSE), pp. 1–7, 2017.
[15] A. Esfahani and et al., “A lightweight authentication mechanism for m2m communications in industrial iot environment,” IEEE Internet of Things Journal, pp. 1–1, 08 2017.
[16] D. Soni and A. Makwana, “A survey on mqtt: A protocol of internet of things(iot).,” International Conference on Telecommunication, Power Analysis And Computing Technique (ICTPACT), 2017.
[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” IEEE Communications Surveys Tutorials, no. 4, pp. 2347–2376, 2015.
[18] A. Hornsby and R. Walsh, “From instant messaging to cloud computing, an xmpp review,” Proceedings of the International Symposium on Consumer Electronics, ISCE, pp. 1 – 6, 07 2010.
[19] Z. B. Babovic, J. Protic, and V. Milutinovic, “Web performance evaluation for internet of things applications,” IEEE Access, pp. 6974–6992, 2016.
[20] C. A. L. Putera and F. J. Lin, “Incorporating oma lightweight m2m protocol in iot/m2m standard architecture,” 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 559–564, 2015.
[21] OMA, “Oma device management protocol approved version 1.2.1,” 06 2008.
[22] OMA, “Oma device management protocol approved version 1.3,” 05 2016.
[23] OMA, “Device management architecture candidate version 1.3,” 03 2012.
[24] O. M. Alliance, “Lightweight machine to machine technical specification,” 02 2018.
[25] G. Association, “Iot standard diagnostic logging version 1.0,” officiel document,” 01 2018.
[26] F. Broadband, “Tr-069 cpe wan management protocol,” 03 2018.
[27] F. Broadband, “Tr-069 cpe wan management protocol v1.1,” 09 2007.
[28] G. Nebbione and M. Calzarossa, “Security of iot application layer protocols: Challenges and findings,” Future Internet, p. 55, 03 2020.
[29] C. Cabrera, A. Palade, and S. Clarke, “An evaluation of service discovery protocols in the internet of things,” Proceedings of the Symposium on Applied Computing, p. 469–476, 2017.
[30] A. Bröring, S. K. Datta, and C. Bonnet, “A categorization of discovery technologies for the internet of things,” IoT’16, 2016.
[31] D. Kim, H. Lee, and D. Kim, “Enhanced industrial message protocol for real-time iot platform,” 2018 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–2, 2018.
[32] B. Konieczek, M. Rethfeldt, F. Golatowski, and D. Timmermann, “Real-time communication for the internet of things using jcoap,” 2015 IEEE 18th International Symposium on Real-Time Distributed Computing, pp. 134–141, 2015.
[33] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “Mqtt-s — a publish/subscribe protocol for wireless sensor networks,” 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE ’08), pp. 791–798, 2008.
[34] O. Akintade, T. Yesufu, and L. Kehinde, “Development of an mqtt-based iot architecture for energy-efficient and low-cost applications,” International Journal of Internet of Things, pp. 27–35, 06 2019.
[35] A. Alshantout and L. Al-Awami, “Enhancing mqtt-sn performance via fountain codes in extreme conditions,” 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), pp. 1184–1189, 2019.
[36] B. Schütz, J. Bauer, and N. Aschenbruck, “Improving energy efficiency of mqtt-sn in lossy environments using seed-based network coding,” 2017 IEEE 42nd Conference on Local Computer Networks (LCN), pp. 286–293, 2017.
[37] A. Oak and R. D. Daruwala, “Assessment of message queue telemetry and transport (mqtt) protocol with symmetric encryption,” 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 5–8, 2018.
[38] O. Sadio, I. Ngom, and C. Lishou, “Lightweight security scheme for mqtt/ mqtt-sn protocol,” 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pp. 119–123, 2019.
[39] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, “New approach for securing communication over mqtt protocol a comparaison between rsa and elliptic curve,” 2016 Third International Conference on Systems of Collaboration (SysCo), pp. 1–6, 2016.
[40] F. Seidel, K.-F. Krentz, and C. Meinel, “Deep en-route filtering of constrained application protocol (coap) messages on 6lowpan border routers,” 04 2019.
[41] T. Vadluri, “A research on interoperability issues in internet of things at application layer,” Mobile Netw Appl, 03 2020.
[42] M. A. A. da Cruz, J. J. P. C. Rodrigues, E. S. Paradello, P. Lorenz, P. Solic, and V. H. C. Albuquerque, “A proposal for bridging the message queuing telemetry transport protocol to http on iot solutions,” 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1–5, 2018.
[43] S. Alexandre, F. Carlier, and V. Renault, “Dynamic bridge generation for iot data exchange via the mqtt protocol,” Procedia Computer Science, pp. 90–97, 01 2018.
[44] S. Wagle, “Semantic data extraction over mqtt for iotcentric wireless sensor networks,” 2016 International Conference on Internet of Things and Applications (IOTA), pp. 227–232, 2016.
[45] A. S. Sadeq, R. Hassan, S. S. Al-rawi, A. M. Jubair, and A. H. M. Aman, “A qos approach for internet of things (iot) environment using mqtt protocol,” 2019 International Conference on Cybersecurity (ICoCSec), pp. 59–63, 2019.
[46] A. Sadeq, R. Hassan, and A. Jubair, “Enhanced mqtt for providing qos in internet of things (iot): A study,” Advanced Science Letters, pp. 5199–5203, 07 2018.
[47] K. Hwang, J. M. Lee, I. H. Jung, and D. Lee, “Modification of mosquitto broker for delivery of urgent mqtt message,” 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 166–167, 2019.
[48] H. Hwang, J. Park, B. Lee, and J. Shon, “An enhanced reliable message transmission system based on mqtt protocol in iot environment,” Lecture Notes in Electrical Engineering, pp. 982–987, 01 2017.
[49] K. Hwang, J. Moon Lee, and I. Hwan Jung, “Extension of reception frequency control to mqtt protocol,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), pp. 2278–3075, 06 2019.
[50] J. Finochietto, G. Eggly, R. Santos, J. Orozco, S. Ochoa, and R. Meseguer, “A role-based software architecture to support mobile service computing in iot scenarios,” Sensors, p. 4801, 11 2019.
[51] A. Depari, D. Carvalho, P. Bellagente, P. Ferrari, E. Sisinni, and A. Padovani, “An iot based architecture for enhancing the effectiveness of prototype medical instruments applied to neurodegenerative disease diagnosis,” Sensors, vol. 19, p. 1564, 03 2019.
[52] W. Mardini, M. Bani Yassein, M. N. Alrashdan, A. Alsmadi, and A. Bani Amer, “Application-based power saving approach for iot coap protocol,” Proceedings of the First International Conference on Data Science, E-Learning and Information Systems, 10 2018.
[53] R. Giambona, A. Redondi, and M. Cesana, “Mqtt+: Enhanced syntax and broker functionalities for data filtering, processing and aggregation,” ArXiv, 10 2018.
[54] M. Al-Osta, B. Ahmed, and A. Gherbi, “A lightweight semantic web-based approach for data annotation on iot gateways,” Procedia Computer Science, pp. 186–193, 12 2017.
[55] C. Fetzer, G. Pfeifer, and T. Jim, “Enhancing dns security using the ssl trust infrastructure,” pp. 21– 27, 03 2005.
[56] S. Jakeer Hussain and G. Ramyasri, “Security enhancement for devices using restapi, middleware iot gateway,” International Journal of Engineering and Advanced Technology (IJEAT), p. 2249 – 8958, 09 2019.
[57] J. Luzuriaga, M. Perez, P. Boronat, J.-C. Cano, C. Calafate, and P. Manzoni, “Improving mqtt data delivery in mobile scenarios: Results from a realistic testbed,” Mobile Information Systems, pp. 1–11, 07 2016.
[58] M. Swarna and T. Godhavari, “Enhancement of coap based congestion control in iot network - a novel approach,” Materials Today: Proceedings, 2020.

Sounds Real: Using Hardware Accelerated Real-time Ray-Tracing for Augmenting Location Dependent Audio Samples

Alexander Madzar
TU Kaiserslautern
Kaiserslautern, Germany
madzar@rhrk.uni-kl.de

Linfeng Li
TU Kaiserslautern
Kaiserslautern, Germany
lli@rhrk.uni-kl.de

Hymalai Bello
DFKI
Kaiserslautern, Germany
hymalai.bello@dfki.de

Bo Zhou
TU Kaiserslautern and DFKI
Kaiserslautern, Germany
bo.zhou@dfki.de

Paul Lukowicz
TU Kaiserslautern and DFKI
Kaiserslautern, Germany
paul.lukowicz@dfki.de

***Abstract*—We present a data augmentation technique for generating location variant audio samples using ray-traced audio in virtual recreations of the real world. Hardware Audio-Based Location-Aware Systems are capable of locating audio sources in relation to mobile devices. This is a relevant technique in the context of location-based and person tracking in ubiquitous environments. However, this solution is limited in collecting vast data to train the machine learning model reliably. To overcome this problem, we constructed a virtual environment using the audio ray-tracing solution, NVidia VRWorks Audio in Unreal Engine 4, to simulate a real-world setting. The environmental sounds in the real-world scenario were imported into the virtual environment. This strategy could augment data for training Hardware Audio-Based Location-Aware Systems machine learning models with the necessary calibration of the unreal and real data sets. Our results show the audio ray-tracing framework could simulate real-world sound in the virtual environment to a certain extent.**

***Index Terms*—audio synthesis; ray-tracing; Unreal Engine; VRWorks audio.**

I. Introduction

Virtual Environments (VE) have steadily increased in popularity among researchers over the last decade [1][2]. VE can give an immersive simulation experience that is suitable to a variety of use cases, including gaming, automobile, construction, and education [3][4][5]. Apart from graphics, audio processing is a vital component of the virtual environment's immersive experience. Numerous ray-tracing audio frameworks, such as VRWorks Audio (NVIDIA) [6], or Steam Audio [7], can enhance and increase the realism and immersion impact in virtual reality technologies.

Sound is an interesting tool to localize users and mobile devices. For example, the location of a user walking within a building can be determined based on ambient sounds [8]. In this scope, we define such systems as Hardware Audio-Based Location-Aware Systems (HABLAS). In ubiquitous computing, sound can be a helpful sensing modality to classify location [9][10].

However, hardware-based location solutions have limitations in terms of data collection. For example, in a particular scene and combination of environmental sounds, the data can only be collected where the devices are located. If the opportunity has passed, one cannot try another location and recollect the data with that exact scene and environment sounds. This is on top of the already well-known general difficulty in gathering annotated data.

In this work, we proposed to use virtual environments created by game design engines via audio path tracing frameworks to generate location-dependent sound data for training HABLAS machine learning models. A critical remark of our proposal is that the quality of the augmentation of virtually generated sound will have a strong dependency in the calibration with the real-sound and the data augmentation algorithm selected for the task.

Our paper structure is as follows; Section II presents related work in the areas of Virtual Environment (VE) reconstruction and Real-Time ray-tracing solutions. Next, Section III provides a detailed description of the proposed method, including details of the employed framework, and the interface between the real-world sound recording with the VE. Then, in Section IV a spectral comparison between the recorded real-world sounds with the virtually generated sounds is discussed. Finally, in Section V, we conclude our work and discuss further ideas.

II. Related Work

With the growing popularity of Virtual Reality (VR), NVidia has released a software suite VRWorks for VR and game developers to utilize graphics processing units (Audacity Team) acceleration with existing 3D design environments [11]. Such tools have also received attention from researchers, especially in the visual scene reconstruction discipline, in works, such as [12][13][14].

However, traditional graphical rendering struggles to achieve the real-world effect, mainly since, in reality, what we perceive results from light and sound waves bouncing off different surfaces and reaching our eyes and ears. Ray-tracing solves such a problem as the graphics or audio can be rendered not only by geometry, but also by considering the reflection and refraction of simulated rays closer to real life. Real-time

 ISBN: 978-1-61208-886-0

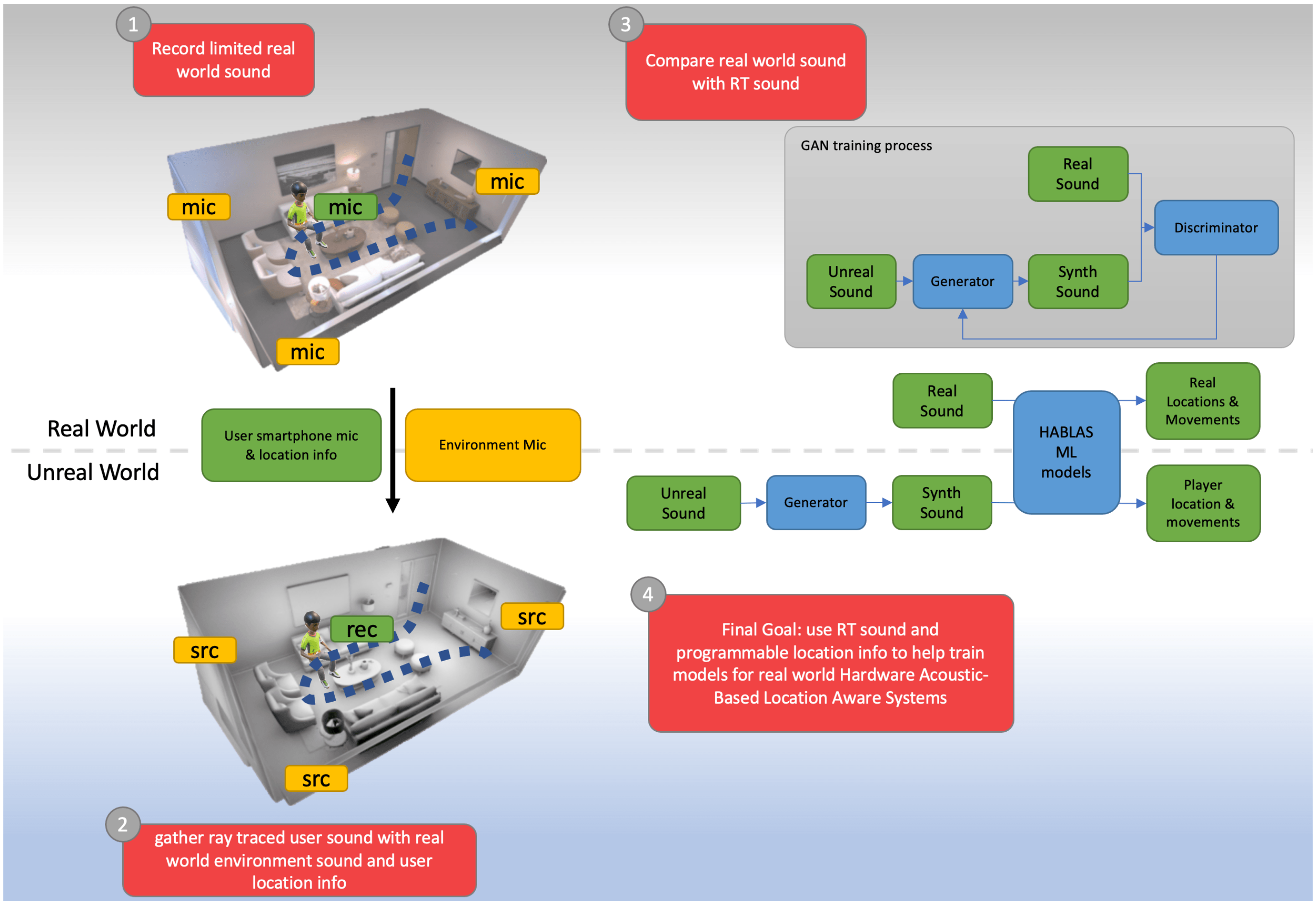

Figure 1. Illustration of our approach to generate location dependent sound samples from virtual recreations of real-world scenes.

ray-tracing has been difficult due to the extreme computation demand [15]. Most recently, real-time ray-tracing has also become possible thanks to the latest GPUs with dedicated accelerators. VRWorks has also been updated with accelerated ray-tracing. GPU acceleration has been investigated for ray-traced sound propagation in underwater environments [16].

Additionally, the idea of using virtual scenes to generate synthetic visual training data as input to machine learning methods has been explored in [17], where Unreal Engine 4 [18] was used to generate and automatically annotate ground truth data of robot agents interacting with objects and between each other. Moreover, a commercial solution for simulation and training of artificial intelligence (AI) robotics was introduced by NVIDIA®Isaac™[19], in which synthetic and virtual data generation techniques can improve the behavior of the robot.

Therefore, we could argue that VE are becoming more similar to the real-world, at least to some degree. The above led us to benefit from virtual scenes recreation for the generation of synthetic data based on ray-traced sound for data augmentation in machine learning solutions, which to the best of our knowledge, has not been done before.

III. Method Proposal

Ray-tracing audio solutions, such as VRWorks Audio (NVidia) and Steam Audio can merge path tracing effects, such as sound propagation, reflection, and constructive/destructive collisions between different sound sources. As a result, and in conjunction with the influence of the various materials' sound properties (e.g., absorption, reflection, transmission, etc.), those solutions augment the immersion experience to the user. For example, a hallway with carpets and wooden walls would sound differently from the same hallway with surrounding marble materials. These frameworks are compatible with game development engines like Unreal Engine 4. This research aims to prove the feasibility of audio data augmentation using the VRWorks Audio virtual reality ray-tracing audio framework. To test our approach, we have created a virtual environment in Unreal Engine 4 modeled after a physical, real-world location (meeting room), as shown in Figure 2, to generate a baseline design with the potential to mimic several users (avatars) in different social circumstances. Our particular focus was to record sound data in the real scene and to compare it with captured sound in the virtual scene designed in the Unreal Engine. Therefore, after calibrating the unreal and real data

 ISBN: 978-1-61208-886-0

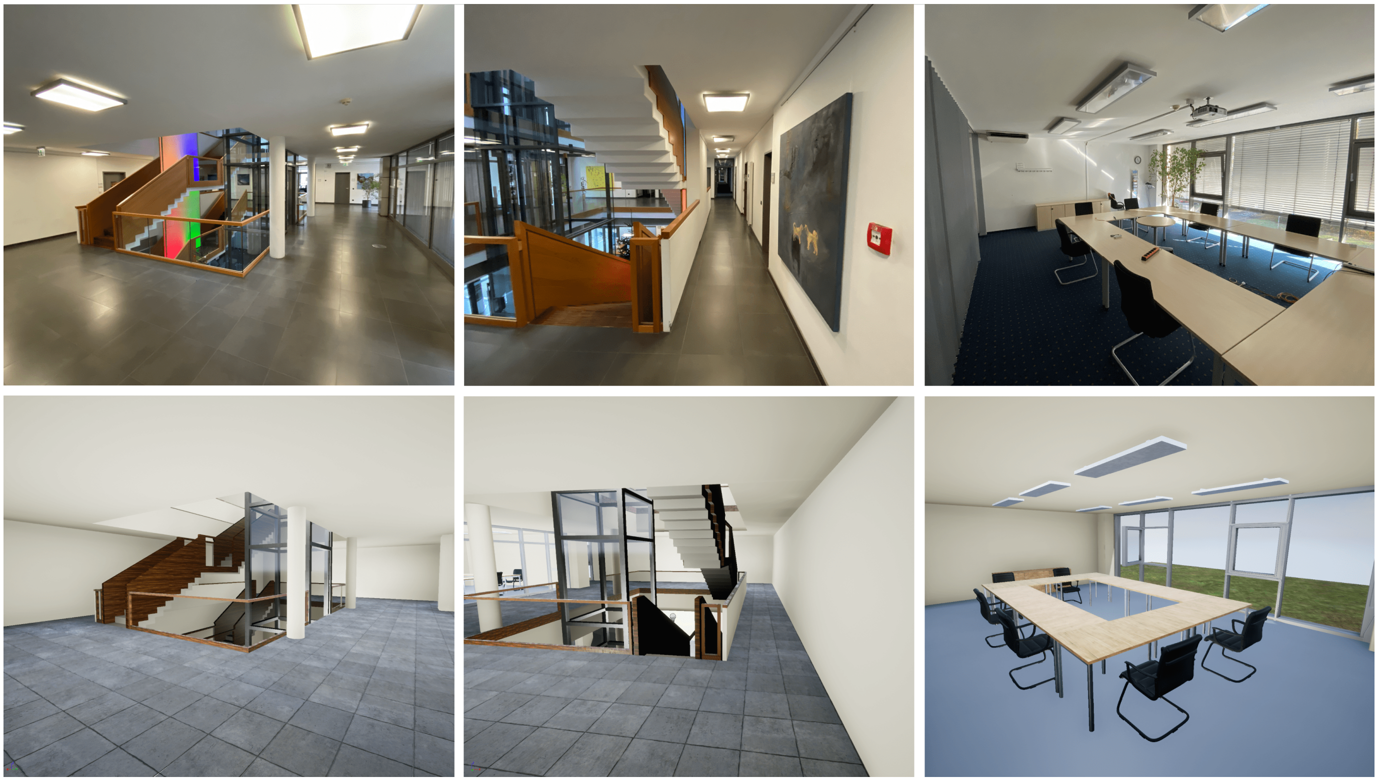

Figure 2. The real scene in the top row and the reconstructed (Unreal Engine) scene in the bottom row.

sets, this strategy could be employed to generate synthetic data for machine learning models to be used in HABLAS solutions, as depicted in the research project's process in Figure 1.

Figure 1 shows a block diagram of the main idea; first, audio recordings from a real scenario will be collected in defined positions. Secondly, the real sound sources are played inside the VE, and the listener is set in the predefined positions in the first step. Next, a comparison between the real sound and the virtually generated sound is made, and a Generative Adversarial Network (GAN) specifically designed for audio [20][21][22] could be used as the data augmentation technique. Finally, these augmented location-aware sounds are used to tackle the lack of training data in models for HABLAS.

A. Frameworks and Methodology

We used Unreal Engine 4 (UE4) to model the scene setup in combination with Nvidia VRWorks Audio. More precisely, Unreal Engine 4.15 is the version for which VRWorks Audio is available as a plugin [23]. The plugin is a private repository; therefore, an Unreal developer with a personal Github account is needed to enable the link. UE4 is originally a game engine for game development but has now been adopted by various other industries. With UE4, a desired scene can be created relatively quickly in a modular fashion. Furthermore, VRWorks Audio enables immersive audio through ray-traced sound in 3D space in real-time. The key features of VRWorks Audio [24] include effects, such as sound propagation, occlusion for direct and indirect paths, attenuation, material reflection, absorption, and transmission, which are needed for ray-traced sound. In the following, we will present how we created the scene setup in Unreal and VRWorks. The software suite was tested on a Dell XPS15 laptop with an NVidia GTX 1050 GPU, as well as two workstations with an NVidia GTX 1080 and RTX A6000 GPUS. All systems can render the scene with real-time performance.

B. Real World Recordings

A real-world experiment was designed to record ambient sound at predetermined places and simulate the static positions of the avatar inside Unreal Engine. We recorded the meeting room's ambient sound using eleven iOS devices (iPhone and iPad). As shown in Figure 3, devices 1, 2, 3, 4, 6, 7, and 11 were arranged on the tables. The microphone symbols were used for gathering environment sound assets to play during real-time rendering, and the numerical symbols are used to distinguish sound differences at different locations. Device 5 was located on the cabinet on the left side of the meeting room. To create some distinctions for sound recording, we left the left door closed and placed devices 8 and 10 in front of the left and right doors, respectively. Device 9 was placed in front of the elevator in the hallway. The entire recording took almost two hours. During the time, various activities were induced at random places in the scene, such as moving trolleys in the corridor, multiple people walking, powered drills, hammering metal, conversations, music, etc. The windows face a busy road, and there is always traffic

 ISBN: 978-1-61208-886-0

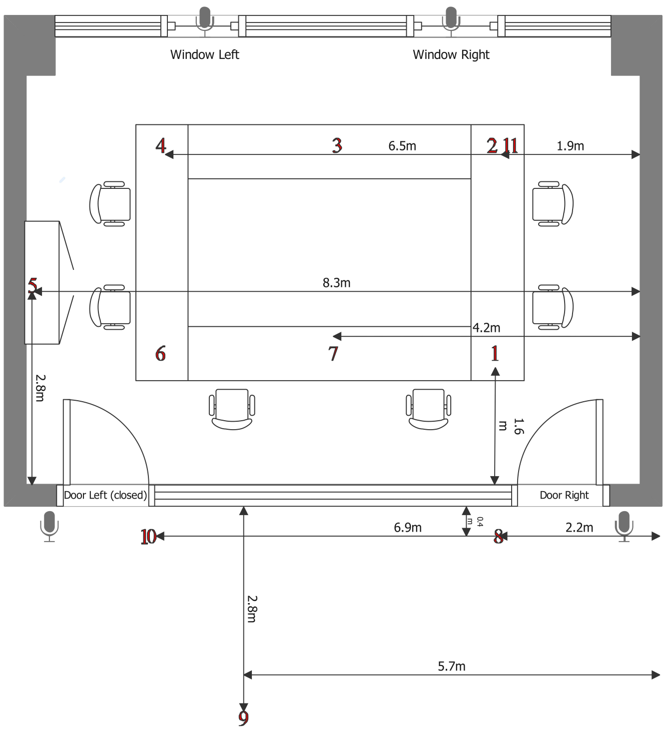

Figure 3. Floor plan: placement of mobile recording devices inside the real/virtual meeting room.

sound captured at the two microphones at the windows. The four microphones depicted in Figure 3 were used as sound sources inside the real-world. They were placed on the doors and windows, and all of them possess unidirectional cardioid polar patterns, which means that most audio with an incident angle outside 90-270 degrees will be attenuated [25]. Android devices recorded such microphones at 48Khz with 256kbps. To reduce the effect of the wind, they were covered with a fur windshield.

C. Sound Recordings of Unreal World

In the Unreal World, the static locations described in Section III-B were used to perform the augmentation of the sound recordings coming from the iOS devices. First, the recordings of the four ambient sound microphones at the windows and doors were imported into Unreal Engine. They were then used as audio sources at the exact corresponding locations. Next, the sound sources needed to be calibrated according to the following considerations:

- **Attenuate:** Enables the sound attenuation. If false is specified, the sound will play at maximum volume regardless of the distance between the sound source and the listener. The setting must be set to true.
- **Spatialize:** This property enables spatialization, which denotes the projection and localization of sound sources inside the virtual environment. If set to false, the sound will be non-spatialized, and no panning will be applied when the listener moves around.
- **Distance Algorithm:** Five different distance algorithms are available, which determine the attenuation rate over distance. We experimented with all the functions and, by ad-hoc method, concluded that the *Inverse*, *Natural Sound* and *Logarithmic* distance algorithms realistically simulate our experimental sound.
- **Attenuation Shape:** This property specifies the shape used to establish the sound's minimum and maximum attenuation points. There are four alternative shapes available, with *Sphere* being the default setting and producing a spherical attenuation shape. The spherical form is the most accurate representation of how sound propagates in the real-world [26].
- **Radius:** The radius describes the distance from the location of the sound at which the falloff begins. Our radius was set to 50 cm.
- **Falloff Distance:** This describes the distance over which the falloff ends. We experimented with several values and got the most realistic results with a falloff distance of 2500 cm.
- **Occlusion:** Occlusion is disabled as ray-tracing is the more realistic alternative.
- **Direct Path Gain:** We use the default setting from the VRWorks Audio Tutorial [6]. Set the value to 5.0.
- **Indirect Path Gain:** Here, we also take the default setting, set the value to 5.0.
- **Effect Strength:** We also use the same preset as in the tutorial: *High*.

To conduct the recordings in the Unreal World, the avatar was placed at the eleven static locations that are presented in Figure 3, so that we can compare them with the real-world recordings from iOS devices. This means that we placed the game character (the listener) at locations one to eleven. The complete recorded sounds from the sound sources (4 microphones, doors, and windows) were reproduced every time the avatar changed location. The sound from the audio sources inside the virtual environment was recorded for about two hours. The recordings were made using the audio software *Audacity*® [27]. Audacity® software is copyright © 1999-2021 Audacity Team. The name Audacity® is a registered trademark. The operating system's digital output was looped back to the recording software.

IV. Comparison Real and Unreal Recordings

After recording in Unreal Engine 4 to determine the performance of Nvidia's ray-tracing technology, a user perception assessment was performed. Our initial test was to have four persons as the audience to only listen to the audio, while one developer navigates the avatar in the real-time rendered scene in various locations. The audience would then describe the change of location. Two audiences tested with the laptop's speaker, and the other two audiences tested via Zoom meetings. The initial test result has revealed that it is evident whether the avatar is in the meeting room, close to the window, or the hallway outside the meeting room. They describe their major clues were the traffic sound outside the window and

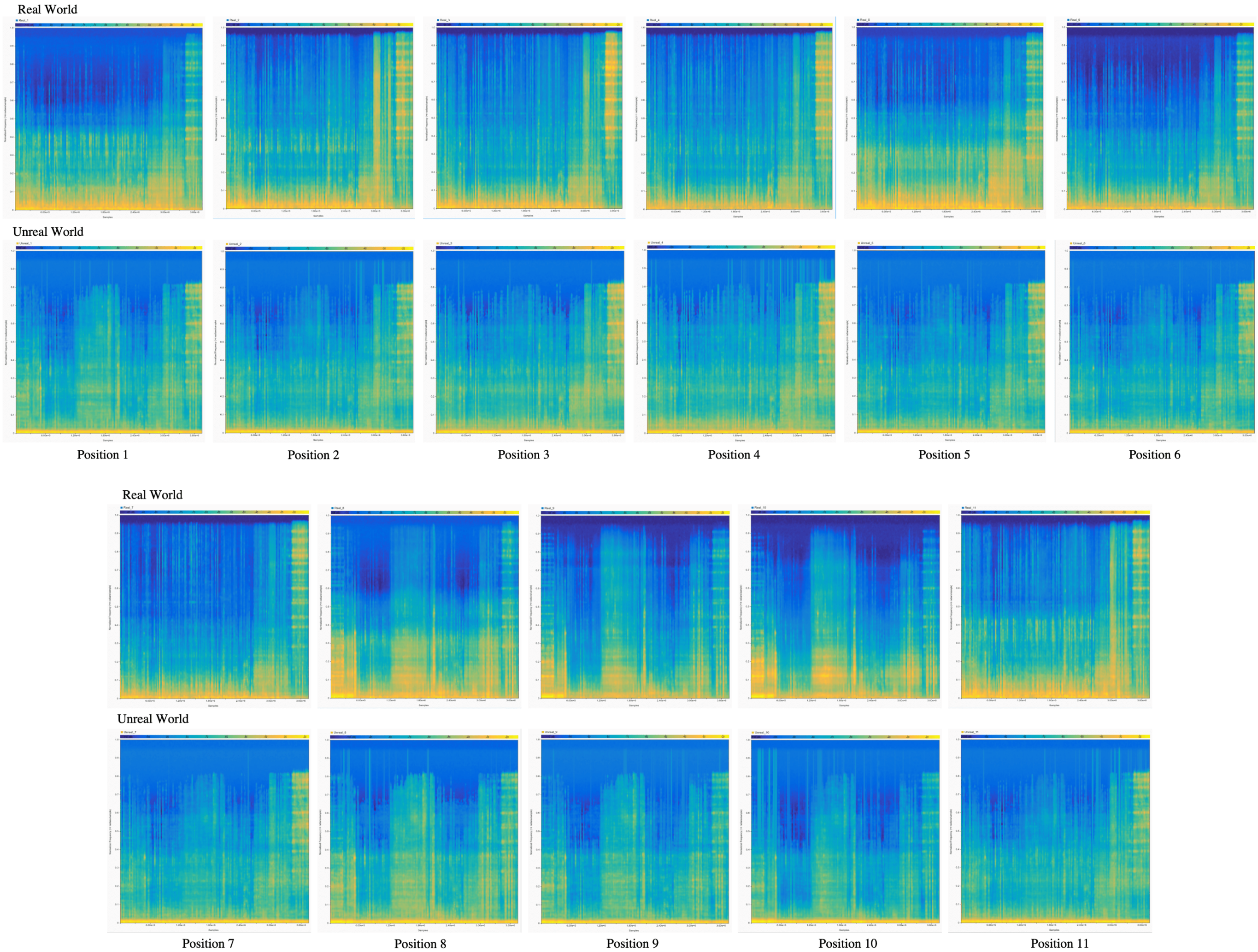

Figure 4. Spectrogram comparison between real scene and the reconstructed (Unreal) scene setup at various locations, during exactly the same time.

the ambient echo characteristic. The hallway with ceramic tile floors and emptier space sounds significantly different than the smaller furnished meeting room with carpet floors.

Next, we present the differences between the real and unreal worlds at different predefined locations with the same time moments using the spectrograms in Figure 4. First, we cropped identical 90-second sections from the 2-hour recordings for each position. We then converted these sections into spectrograms using the Matlab ®Signal Analyzer App [28] to evaluate the differences or similarities between the real and unreal recordings.

When examining the spectrograms of the different positions, it is noticeable that each position has a distinct appearance, although the sound played was the same in all 11 positions, as expected. It is indicating that the localization by the ray-tracing is valid. Depending on the position in the room, the sound is influenced by the arrangement of the sound sources, surrounding objects like chairs and tables, and the materials used. By comparing the spectrograms of the real and virtual environment worlds, it is clear that the virtual environment world's spectrograms are less detailed in the high-frequency range compared to the real-world. This cut-off of high frequencies is most likely due to the simulation environment, i.e., software restrictions, resolution of the digital sound card on the computer.

Additionally, certain events in the real-world (sections with intense yellow colors) can be noticed on the spectrograms of the virtual environment world in Figure 4. For instance, three events stand out at positions 8-10. These are construction activities that occurred in the hallway adjacent to these positions. However, these events are not discernible from the remaining locations, as construction work was barely audible within the meeting room.

Although in our work the virtual scene is manually reconstructed based on real-world measurements, there are already trends towards automatic scene reconstruction, such as [29].

V. Conclusion and Future Work

Overall, we have investigated sound sample augmentation using existing software tools meant for 3D game and VR experience developers for data augmentation, specifically in location-dependent sound samples. This can be used further for training Hardware Audio-Based Location-Aware Systems, bypassing difficulties in data collection. Modern tools utilize hardware-accelerated ray-tracing for the audio rendering, thus making it more realistic than traditional propagation-only synthesis.

A significant advantage of our method is that the scene can be replayed, and the listener can be placed at different locations for the same period of surrounding activities, which is not possible in real-world data collection for this purpose. In this work, although we only presented the locations that were also recorded in the real-world in Figure 4, our approach can generate a new soundtrack from any location other than those inside the scene. It is impossible for actual data collection procedures to gather data at every centimeter inside the scene.

In our future work, we would continue to implement our method with SteamVR, which also has ray-traced sound. Furthermore, we will also improve the generated sound samples using generative adversarial networks as a post-processing step.

Acknowledgment

This work has been supported by BMBF (German Federal Ministry of Education and Research) in the project SocialWear.

References

[1] Y. Lang, L. Wei, F. Xu, Y. Zhao, and L.-F. Yu, “Synthesizing personalized training programs for improving driving habits via virtual reality,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 297–304.

[2] H. Zhang, “Head-mounted display-based intuitive virtual reality training system for the mining industry,” *International Journal of Mining Science and Technology*, vol. 27, no. 4, pp. 717–722, 2017.

[3] M. Thees, S. Kapp, M. P. Strzys, F. Beil, P. Lukowicz, and J. Kuhn, “Effects of augmented reality on learning and cognitive load in university physics laboratory courses,” *Computers in Human Behavior*, vol. 108, p. 106316, 2020.

[4] S. Kapp *et al.*, “Augmenting kirchhoff’s laws: Using augmented reality and smartglasses to enhance conceptual electrical experiments for high school students,” *The Physics Teacher*, vol. 57, no. 1, pp. 52–53, 2019.

[5] M. P. Strzys *et al.*, “Physics holo. lab learning experience: using smartglasses for augmented reality labwork to foster the concepts of heat conduction,” *European Journal of Physics*, vol. 39, no. 3, p. 035703, 2018.

[6] A. Dantrey and T. Scudiero. (2018) Creating immersive audio effects in games and applications using vrworks audio. NVIDIA. [Online]. Available: https://on-demand.gputechconf.com/gtc/2018/presentation/s8680-creating-immersive-audio-effects-in-games-and-applications-using-vrworks-audio.pdf

[7] Steam Audio. (2019) Steam audio unreal engine 4 plugin. Steam Audio. [Online]. Available: https://valvesoftware.github.io/steam-audio

[8] Y. Teshima, J.-y. Takayama, S. Ohyama, and K. Oshima, “Person localization using tdoa of non-speech sound signal based on multiplexed csp analysis,” in *Proceedings of SICE Annual Conference 2010*. IEEE, 2010, pp. 1207–1213.

[9] C. E. Galván-Tejada *et al.*, “A generalized model for indoor location estimation using environmental sound from human activity recognition,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, p. 81, 2018.

[10] S. Park, S. Mun, Y. Lee, and H. Ko, “Acoustic scene classification based on convolutional neural network using double image features,” in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017, pp. 98–102.

[11] M. Kraemer. (2018) Accelerating your vr games with vrworks. [Online]. Available: https://www.gdcvault.com/play/1024356/Accelerating-Your-VR-Games-with

[12] O. Zia, J.-H. Kim, K. Han, and J. W. Lee, “360 panorama generation using drone mounted fisheye cameras,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–3.

[13] I. Baek, A. Kanda, T. C. Tai, A. Saxena, and R. Rajkumar, “Thin-plate spline-based adaptive 3d surround view,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 586–593.

[14] D. Pohl, N. Choudhury, and M. Achtelik, “Concept for rendering optimizations for full human field of view hmds,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 663–664.

[15] J. Gunther, S. Popov, H.-P. Seidel, and P. Slusallek, “Realtime ray tracing on gpu with bvh-based packet traversal,” in *2007 IEEE Symposium on Interactive Ray Tracing*. IEEE, 2007, pp. 113–118.

[16] M. Ulmstedt and J. Stålberg. (2019) Gpu accelerated ray-tracing for simulating sound propagation in water. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1352170/FULLTEXT01.pdf

[17] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, “Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation,” *Virtual Reality*, pp. 1–18, 2019.

[18] Epic Games. (2019) Unreal engine. Epic Games. [Online]. Available: https://www.unrealengine.com

[19] NVIDIA. (2019) Nvidia isaac platform for robotics. NVIDIA. [Online]. Available: https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/

[20] I. Goodfellow *et al.*, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[21] K. Kumar *et al.*, “Melgan: Generative adversarial networks for conditional waveform synthesis,” *arXiv preprint arXiv:1910.06711*, 2019.

[22] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis,” *arXiv preprint arXiv:1802.04208*, 2018.

[23] NvPhysX. (2017) Vrworks audio sample project. NVIDIA. [Online]. Available: https://github.com/NvPhysX/UnrealEngine/tree/VRWorks-Audio-4.15.1

[24] NVIDIA DEVELOPER. (2017) Vrworks - audio. NVIDIA DEVELOPER. [Online]. Available: https://developer.nvidia.com/vrworks/vrworks-audio

[25] A. Kuntz and R. Rabenstein, “Cardioid pattern optimization for a virtual circular microphone array,” in *EAA Symposium on Auralization*. Citeseer, 2009, pp. 1–4.

[26] Epic Games. (2019) Sound attenuation. Epic Games. [Online]. Available: https://docs.unrealengine.com/4.26/en-US/WorkingWithMedia/Audio/DistanceModelAttenuation/

[27] Audacity Team . (1999-2021) Audacity(r): Free audio editor and recorder. Audacity Team. [Online]. Available: https://audacityteam.org/

[28] MathWorks, *Matlab Signal Analyzer*, Natick, Massachusetts, United State, 2021. [Online]. Available: https://de.mathworks.com/help/signal/signal-analyzer.html

[29] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, “Atlas: End-to-end 3d scene reconstruction from posed images,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020, pp. 414–431.

Blockchain and Distributed Ledger Technologies for Intelligent Transportation Systems: a Survey

Giuseppe Loseto, Filippo Gramegna, Saverio Ieva, Agnese Pinto, Floriano Scioscia, Michele Ruta
Department of Electrical and Information Engineering (DEI)
Polytechnic University of Bari – Bari, Italy
name.surname@poliba.it

***Abstract*—Intelligent Transportation Systems (ITSs) aim to provide novel services to drivers and passengers, increasing safety, efficiency end environmental sustainability. Trustworthiness, reliability and auditability of distributed software and data are increasingly necessary from both a technical and a regulatory perspective, particularly as ITSs evolve towards autonomous driving. Unfortunately, heterogeneity of virtual counterparts of users and vehicles along with their intrinsic volatility make coordination and trust management difficult for cooperation. To solve these issues, blockchain and Distributed Ledger Technologies are increasingly adopted in ITSs. This paper surveys key aspects of blockchain research and usage in ITS and automotive sectors, comprising the main technological trends and open issues, the most significant application scenarios and an analysis of relevant DLT platforms.**

***Keywords*—Intelligent Transportation Systems, Blockchain, Distributed Ledger Technologies**

ACRONYMS

CVIM	Common Vehicle Information Model
DAG	Direct Acyclic Graph
DAO	Decentralized Autonomous Organization
DBMS	Data Base Management System
DLT	Distributed Ledger Technology
DSRC	Dedicated Short-Range Communication
EOV	Execute-Order-Validate
EVM	Ethereum Virtual Machine
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent Transportation System
MEC	Mobile Edge Computing
PBFT	Practical Byzantine Fault Tolerance
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
RSU	Road-Side Unit
SC	Smart Contract
SGX	Software Guard eXtensions
SOA	Service-Oriented Architecture
TEE	Trusted Execution Environment
UTXO	Unspent Transaction Outputs
VANET	Vehicular Ad-hoc NETwork
VM	Virtual Machine
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

I. INTRODUCTION

ITSs aim to provide novel and improved services to drivers, riders and passengers, increasing safety, efficiency end environmental sustainability of transportation [1]. ITS platforms integrate four main technological layers: (i) *sensing* internal and environmental data of vehicles and the road network; (ii) *processing* in distributed and multi-core architectures by means of artificial intelligence techniques for decision support and autonomous control; (iii) *communications* in Vehicular Ad-hoc NETworks (VANETs) comprising Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) links through Dedicated Short-Range Communication (DSRC) or cellular infrastructures; (iv) *Service-Oriented Architectures* (SOA) to discover and exchange situational awareness information, multimedia resources and application services among vehicles, in addition to basic emergency data; (v) *human-machine interfaces* designed to present information effectively and properly for drivers.

Due to the high-speed mobility of vehicles, VANETs suffer from high device churn, with consequent unpredictable volatility of host and resource availability. This makes coordination and trust management difficult for device cooperation. Anyway, reliability, trustworthiness and auditability of distributed software and data are increasingly necessary from both a technical and a regulatory perspective, particularly as ITSs evolve towards autonomous driving [2]. Hence, even more research solutions tend to adopt *blockchain* technology in ITS pilot projects [3]. Blockchain denotes a data structure and protocol for peer-to-peer *trustless* distributed transactional systems. In traditional distributed databases, a trusted intermediary is needed to prevent *censorship* (*i.e.*, all valid transactions are committed) and ensure *irreversibility* (*i.e.*, no committed transaction can be reverted or altered). Blockchain systems avoid intermediaries by approving transactions through a distributed *consensus* approach, which guarantees no single host or small group of colluding hosts can force addition, removal or modification of data. Transactions approved in a given time period are grouped in *blocks*, which are appended sequentially. The blockchain works as a *distributed ledger* of transactions, and research on *Distributed Ledger Technologies* (DLT) is investigating several types of data structures, consensus protocols and architectural variations for DLT platforms.

The blockchain potential as general-purpose distributed database was understood soon after the introduction of the technology with the *Bitcoin* open source platform for digital currency. In particular, DLTs enable practical implementations of the *Smart Contract* (SC) idea [4], *i.e.*, programs encoding and enforcing cooperative processes like the terms of a contract among two or more parties. Consensus about SCs on

 ISBN: 978-1-61208-886-0

a blockchain is reached through a parallel execution in the network, effectively making every SC-enabled DLT a general-purpose application platform based on a distributed Virtual Machine (VM).

Due to the above features, ITS applications and projects are increasingly adopting DLT platforms. This survey provides a compact but comprehensive overview of the state of the art about blockchain and DLTs for ITSs.

The remainder of the paper is as follows: Section II briefly recalls the most important features of blockchain and DLT technologies, while Section III discusses on relevant applications of them in the ITS field. An analysis of four DLT platforms among the most adopted and technologically significant ones for ITS follows in Section IV, before conclusion.

II. Blockchain and Distributed Ledger Technologies

Blockchain and DLT systems are increasingly different in architecture, technologies and applications. Useful surveys exist, focusing on their integration in the Internet of Things (IoT) [5]–[7]. DLT types can be classified with respect to key design policies [8]:

Network access - *Permissionless* blockchains allow any host to join –even anonymously– at any time. Conversely, hosts are uniquely identified in *permissioned* platforms and only authorized ones can connect. This choice affects the blockchain project: permission-less chains usually have to reward participants for their computational effort, *e.g.*, Bitcoin allows hosts to generate (*mine*) and keep new currency for the validation of transaction blocks. Permissioned chains are instead adopted in more controlled collaboration contexts, where access itself is a reward, as it enables selling and buying services or resources.

Consensus protocol - Permissionless systems require stricter consensus methods, such as *Proof-of-Work* (PoW), which guarantees data security unless hosts collectively possessing the majority of computational power in the network are colluding to subvert the blockchain: *Byzantine fault tolerance* [9] is, in fact, typically required. For adding the next block to the chain, PoW elects a leader among all hosts having candidate blocks. Election requires solving a cryptographic challenge, and implies high computational and energy cost with commodity hardware, low transaction throughput and the need for *mining* (a reward to transaction validators participating in the consensus by means of small digital currency amounts). *Proof-of-Stake* (PoS) consensus, instead, elects the host with the highest value of *coin-age*, that is the product of the amount of owned currency and the time since the host has been holding it without spending. This mechanism avoids high computational costs and mining. Permissioned systems –where each host is accountable– may relax consensus constraints to *crash fault tolerance* guarantees [9]. Permissioned blockchains can follow either the *private* or the *consortium* model: in the former consensus is managed by hosts from a single organization, giving up some of the benefits of decentralization in exchange for higher transaction throughput; the latter adopts a trade off where a subset of all hosts, belonging to multiple organizations, can participate in the consensus.

Transaction model - *Assets* can be registered or transferred by means of transactions on a blockchain. In the *unspent transaction outputs* (UTXO) model, an $A \rightarrow B$ transfer implies *consuming* (*i.e.*, deleting) records for A's spent assets and *producing* (*i.e.*, adding) new ones for B's received assets. In the *account-based* model, instead, every host has an account reporting all its assets, which is updated by transactions. The former is simpler to manage and fits the digital currency use cases, but it is not general-purpose; the latter is required to support SCs [8].

SC language - Blockchains can adopt any formalism for SC specification and execution, such as procedural programming languages, logic programming or automata [10]. Industry proposals mostly adopt computationally complete programming languages, either existing (*e.g.*, Java in the *Iroha* framework) or created for the purpose (*e.g.*, Ethereum's *Solidity*).

Centralized information management models are clearly not scalable enough for the ever-growing IoT and the Internet of Vehicles (IoV) [11]. They pose issues with respect to cost and performance, as well as security and trust. The viability of blockchain and DLT technologies for the IoT is analyzed in [7] and strategies are outlined to combine security and scalability. Running IoV resource/service marketplaces with minimal or no human intervention [8] requires a cross-application peer-to-peer middleware layer comprising several building blocks. Recent proposals include the *Inter-Planetary File System* (IPFS) distributed storage protocol [12], intelligent service discovery [13] and billing services [14].

Research on blockchain scalability is very active, mainly by optimizing performance of consensus protocols [13], [15] and by introducting parallelism in a blockchain through *sidechains* and/or *sharding* [16]. Basically, the use of sidechains transforms the chain structure in a Direct Acyclic Graph (DAG). On the other hand, sharding is a parallelization technique borrowed from Database Management Systems, consisting in splitting data elements (*e.g.*, rows in relational databases) horizontally across host subsets in a cluster. Research results, however, are not mature enough [17] for building efficient, robust, large-scale IoT-oriented blockchains.

III. DLT solutions for Intelligent Transportation Systems

Emerging blockchains increasingly refer to novel transparent and trustless models, particularly fitting needs and requirements of sectors like transportation [18].

A reference ITS-oriented blockchain model has been proposed in [19], with seven conceptual layers characterizing and standardizing the typical architecture of blockchain systems:

1) *physical*: concerning devices, vehicles and physical assets;
2) *data*: core data structures and cryptographic primitives of the digital ledger;
3) *network*: peer-to-peer networking primitives;

 ISBN: 978-1-61208-886-0

4) *consensus*: implementing the supported consensus protocol(s);
5) *incentive*: policies for the issuance and allocation of incentives to miners;
6) *contract*: SC execution environment and SC instances;
7) *application*: ITS applications and services implemented on top of the previous layers.

According to the proposed framework, a real-time decentralized ride-sharing service has been also implemented to prove its applicability. In [20], the authors have extended previous modeling approaches to combine IoT and blockchain technologies for smart logistics and transportation in a general-purpose and reliable architecture fitting different ITS scenarios. Anyway, the layered architectural model is useful to guide a systematic analysis of the state of the art.

Physical layer. *Collaborative Vehicular Edge Computing* [21] maps the typical VANET architecture to the Edge Computing paradigm: vehicles and other mobile hosts belong to the infrastructure layer, where local computation and direct communication among nearby devices occur. Analogously, Road-Side Units (RSUs) enable the Edge Computing layer, which interconnects clusters of local devices and supports both vertical and horizontal collaboration through software-defined networking for dynamic resource provisioning and management. In [21] this general model has been specialized to include the three most popular approaches to Edge Computing: Mobile Edge Computing (MEC), Fog Computing, and Cloudlets.

Data layer. In VANET-based blockchains, security and privacy still have several relevant open research questions. Blockchain networks are not immune to cyberattacks and frauds. Attackers could exploit vulnerabilities in blockchain infrastructure to penetrate protected systems, compromise data, overload networks, and cause potentially severe risks to users. In [22], a consortium blockchain and SCs improve security in data exchange and storage within VANETs, while a reputation model enhances the quality of shared data. Information interoperability among heterogeneous hosts is a further largely open problem, and semantic-based structured representations of blockchain assets exploiting Semantic Web technologies have been proposed for that [13]. Defining interoperable methods to assess vehicle reputation and level of trustworthiness, based on both its prior actions and nearby vehicles information, is one of the main goals.

Network layer. As highlighted in [23], current vehicles basically integrate wireless communication and sensing devices providing high speed connectivity and a huge amount of gathered data. Information can be stored and manipulated by a distributed computing platform to create innovative smart applications. 5G is seen as a key enabling technology to overcome bandwidth, reliability and security problems of DSRC. The combination of both technologies simplify process automation in several ITS scenarios, including transmission of tracking information in fleet management applications, monitoring of resource flows and the administration of logistics processes. In fact, 5G communication improves the connectivity of IoT devices by maximizing channel transfer capacity, reducing network latency and increasing the density of interconnected devices, whereas blockchain ensures secure, verifiable and auditable storage of transaction data [24].

Consensus layer. From this perspective, scalability is the main open issue, as PoW and SCs still have a significant impact in terms of transaction throughput [8], [25] and cost of energy. For this reason the MEC blockchain architecture proposed in [26] offloads PoW computation to nearby Edge Computing hosts. Furthermore, the amount of computing resources required by a SC cannot be predicted, as it may recursively invoke further SCs. Approaches to solve this problem include associating currency costs to computation or preventing recursive SC calls [13]. Decentralized energy and charging service marketplaces for electric vehicles and Smart Grid integration are among the scenarios where the benefits of blockchain are most evident, as they need supporting secure and verifiable commercial transactions and providing facilities of service discovery, negotiation, selection and resource allocation. The Cloud-Edge architecture proposed by Liu *et al.* [27] aims to manage interactions concerning vehicular information and energy flows simultaneously. Following Cloud-Edge and information-energy interactions, they identify four categories of context-aware applications and propose a PoS consensus protocol based on *data coins* and *energy coins*.

Incentive layer. Blockchain can facilitate interoperable vehicle data exchanges among car makers, but the hardest obstacle to such cooperation is the historically competitive and secretive nature of the automotive industry. Several initiatives have been launched to overcome this limitation: most recently, the *AutoMat* [28] Horizon 2020 project has defined a Common Vehicle Information Model (CVIM) to represent and share hierarchically organized information about vehicles in an interoperable way, as well as an architecture for a cloud-based Big Data marketplace. Despite the adoption of blockchain and SCs is not in the AutoMat proposal, it is easy to see it would be a natural fit *e.g.*, with the European Blockchain Services Infrastructure (EBSI) [29], possibly granting stronger security, traceability, verifiability and flexibility to the data marketplace.

Contract layer. The adoption of DLTs and SCs can improve the automation of complex logistic procedures by introducing the following benefits [30], [31]: component traceability with real-time data transmission and identification of new resources; persistent and reliable storage of complex data, usually pre-processed by means of data mining and machine learning algorithms; user and data privacy, exploiting different encryption methods, which is particularly useful in industrial e-procurement scenarios where strict regulations must be applied; definition of simple testing procedures required to optimize business processes and automation procedures. All the above capabilities are very important for ITS-based supply chain and logistics applications, where DLTs can be exploited to reduce wait times and management costs and to improve (i) timely delivery of goods, (ii) use of connected devices according to current regulations, (iii) accuracy and efficiency of customer services, and (iv) monitoring of goods

 ISBN: 978-1-61208-886-0

while transiting.

Application layer. In latest years, several automotive companies have been proposing interesting solutions for ITS, combining novel communication technologies and blockchain. BMW Group has been particularly active, developing pilot projects for several real-world ITS scenarios. The *VerifyCar* project, based on the *VeChainThor* [32] blockchain platform, has introduced a digital car registration document to store information about mileage, accident history, inspections, maintenance procedures and other useful information related to the lifecycle of a vehicle. Each car results equipped with an up-to-date and certified data log which can be properly accessed by authorized parties at any time and cannot be tampered with. This constitutes a very important benefit not only for vehicle servicing, but also for insurance and for the overall car market. Since 2019 BMW Group has also been working on a blockchain-based system called *PartChain* [33], aiming to improve the supply chain management in the automotive industry. The project has defined an industrial solution to share data of production tasks and simplify all the procedures for tracing the origin of each vehicle component. The company is planning to use the system also for the management of raw materials, focusing on the traceability of the most critical resources involved in vehicle manufacturing. Moreover, BMW is one of the founding members of *MOBI* (Mobility Open Blockchain Initiative) [34], a nonprofit consortium including over 100 companies in the automotive and information technology sectors. MOBI aims to define reference standards and control models for developing new platforms for ITS. Dedicated working group have been established for: vehicle identity, aiming at an extension of VerifyCar towards a vehicle digital twin; usage-based mobility and insurance; electric vehicle grid integration; connected mobility data marketplace; finance, securitization, and SCs; supply chain. The latter working group would extend the PartChain project to all companies interested and involved in the initiative, in order to facilitate cross-industry data exchange and speed up the integration of blockchain-based platforms in different automotive and transportation scenarios.

IV. Analysis of relevant platforms

Blockchains and DLTs can be classified and compared according to several criteria, including data structures, consensus protocols and SCs, allowed data security and privacy [8]. Four of the most relevant platforms are discussed in what follows, while Table I summarizes benefits and potential limitations from an ITS perspective.

Ethereum [37] is a permissionless blockchain; it allows integrating SCs for developing Decentralized Autonomous Organizations (DAOs) interacting without the intervention of a central authority [38]. The platform core is the *Ethereum Virtual Machine* (EVM) [39], *i.e.*, a *quasi-Turing-complete* execution environment for general-purpose transactions and SCs, replicated on each participating host for validation. Since Turing-completeness would open the platform to abuses and security risks, the EVM associates a cost in *gas* units to code execution, which must be paid in Ethereum's currency, *Ether*. When a host invokes a transaction or a SC, it must pay gas for that in advance; if execution does not complete before running out of the prepaid amount of gas, it is rolled back completely, otherwise it is committed and the possibly remaining gas is refunded. Ethereum is currently the largest SC platform. Several languages can be used: the most popular and mature one is *Solidity* [40]. Ethereum adopts a PoW consensus algorithm called *Ethash*. It aims to be less computationally intensive and more memory intensive than Bitcoin's PoW, in order to limit the recourse to specialized mining hardware and to concentrating computational power in few large mining pools. However, it does not solve the high energy consumption and limited scalability problems. With this motivation, the Ethereum Foundation and community are currently transitioning to Ethereum 2.0, which will adopt a PoS consensus algorithm to increase transaction throughput, reduce computational costs and make the platform fairer and more accessible. Chain sharding will be also employed to partition the validation load among validators and therefore increase platform scalability.

IOTA is a DLT designed specifically for Internet of Things scenarios. It is based on a Direct Acyclic Graph (DAG) data structure, the *tangle*, where each node stores a transaction [41]. Starting from a genesis node n_0, in order to accept a new node n_j the issuer must validate 2 or more transactions $n_{i_k}, k \geq 2$ already in the tangle; then n_j will be added to the tangle with edges from each of the nodes n_{i_k}. This consensus mechanism creates tamper-proof records of network participants' transactions, as older transactions are validated by newer ones either directly or indirectly (through a DAG path). Further key tangle properties are: scalability, because the validation load is spread across all participants; throughput, since a low number of validations is required to accept a new transaction; robustness, in case of network partitioning the tangle may fork temporarily, but it will merge again when connectivity is restored; no reward mechanism for validators is required, avoiding the potential distortions of hosts' behavior related with mining; a snapshot mechanism allowing the periodic removal of very old transactions from the DAG safely, so reducing ledger storage requirements for each peer. IOTA claims the protocol is as secure as PoW, however no formal proof has been produced yet. Based on the IOTA DLT, the *IOTA Streams* protocol (formerly known as *Masked Authenticated Messaging*) has been designed for secure and metered access to IoT data streams. It adopts a publish/subscribe model, integrating both cryptography for controlled data sharing and payment in the IOTA cryptocurrency. Its features make IOTA Streams particularly suitable to vehicular applications [42]. A IOTA based solution for ITS system has been proposed by the ORCHESTRA consortium [35].

Hyperledger Sawtooth is an open source project within the *Hyperledger* initiative for a business-oriented blockchain platform with support for SCs. By design, Sawtooth allows applications to dynamically select transaction rules, authorizations, and consensus protocol, based on business requirements.

TABLE I
COMPARISON OF BDLT PLATFORMS

BDLT	Benefits	Limitations	ITS Implementations
Ethereum	• SC support • DAO development	• PoW-related limitations • Low throughput, high resource consumption	• Xiong *et al.* [25]
IOTA	• Fast, inexpensive consensus • No transaction fee • Low energy consumption • High scalability when transactions increase • DAG snapshot mechanism • IoT-oriented data stream access services	• Limited SC support • Unproven consensus security	• ORCHESTRA [35]
Hyperledger Sawtooth	• SC support in several programming languages • Permissioned and permissionless blockchain support • Relatively high transaction throughput • Low energy consumption of PoET	• Specific hardware required for trusted execution environments	• Salesforce blockchain [36]
Hyperledger Fabric	• Permissioned blockchain support • High transaction throughput and low latency • SC support in several programming languages • Private data areas with privacy-preserving authentication	• Relatively higher architectural complexity	• PartChain [33]

Its architecture [43] clearly separates the application layer from the main platform layer, and in particular it isolates consensus from transaction semantics. The consensus mechanism is selected during the network configuration and can also be modified later on a running blockchain, by means of specific transactions. Sawtooth currently supports four types of consensus protocols: Practical Byzantine Fault Tolerance (PBFT) [44]; Proof of Elapsed Time (PoET) [45], implementing a fair lottery-based leader election system without the power consumption disadvantages of PoW; *Raft* [46], a consensus strategy optimized for small networks. In particular, a Byzantine fault tolerant PoET variant is enabled if a Trusted Execution Environment (TEE) is provided by the hosts' platform, such as the Software Guard Extensions (SGX) instructions of Intel CPUs, otherwise a less robust crash fault tolerant variant is implemented purely in software (essentially used for development purposes but not recommended in production environments). Similar flexibility is achieved for SC support, as an interface abstraction mechanism allows developers to write contract logic in multiple programming languages, including Python, JavaScript, Go, C++, Java, and Rust. Finally, Sawtooth includes a scheduler that splits transactions to parallel flows, mutually isolating their execution. Whenever possible, transactions run in parallel, allowing for a significant increase in performance over sequential execution. Salesforce introduced a blockchain solution for supply chain management based on Sawtooth [36].

Hyperledger Fabric [47] adopts an *Execute-Order-Validate* (EOV) transaction processing model: peers *endorse* transactions by checking their correctness, then *order* them by consensus and finally *validate* them against a particular *endorsement policy*, before committing them to the ledger. Conversely, the majority of DLT platforms adopt an *Order-Execute* model, where transactions are individually validated, then ordered and propagated to all peer hosts, which must execute them sequentially to commit the updated system state in the ledger. The EOV model grants parallel transaction execution and pluggable consensus protocols. Further Fabric peculiarities include: (i) *Zero-Knowledge-Proofs* for privacy-preserving authentication, which is particularly valuable in promiscuous environments like vehicular networks [48]; (ii) high modularity, since responsibilities within the EOV model are divided among *clients*, *endorsing peers*, *orderers* and *committing peers*, which may run on independent hosts.

V. PERSPECTIVES AND CONCLUSION

This paper has explored most relevant aspects of blockchain and Distributed Ledger Technologies for Intelligent Transportation Systems. The proposed outline of the main technological trends, results and open problems has shown the increasing adoption and importance of DLTs for both the automotive and transportation industries. While research must continue on technical issues, it is clear from this survey that, more and more, challenges for the success of DLTs in ITSs go beyond purely technological aspects, intertwining with complex societal, legal and business matters.

Blockchain, DLTs and SCs have been conceived to impact how people and organizations transact business, coordinate, and cooperate. Since their inception, they have been continuously debated with growing interest from technological, financial, regulatory, social and environmental viewpoints. This is a feature they have in common with transportation technologies, and in particular with Intelligent Transportation Systems; perhaps this is at the core of why each one of them is so relevant for the future of the other.

REFERENCES

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.
[2] M. Obaidat, M. Khodjaeva, J. Holst, and M. B. Zid, "Security and privacy challenges in vehicular ad hoc networks," in *Connected Vehicles in the Internet of Things*. Springer, 2020, pp. 223–251.
[3] G. Falco and J. E. Siegel, "A Distributed "Black Box" Audit Trail Design Specification for Connected and Automated Vehicle Data and Software Assurance," *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 3, no. 2, pp. 97–111, 2020.

 ISBN: 978-1-61208-886-0

[4] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[5] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 179–190, 2018.

[6] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and IoT Integration: A Systematic Survey," *Sensors*, vol. 18, no. 8, p. 2575, 2018.

[7] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's Adoption in IoT: The Challenges, and a Way Forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, 2019.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] S. Liu, P. Viotti, C. Cachin, V. Quéma, and M. Vukolić, "XFT: Practical fault tolerance beyond crashes," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 485–500.

[10] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*. Springer, 2016, pp. 167–183.

[11] W. Xu *et al.*, "Internet of Vehicles in Big Data Era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, 2017.

[12] M. Naz *et al.*, "A Secure Data Sharing Platform Using Blockchain and InterPlanetary File System," *Sustainability*, vol. 11, no. 24, p. 7054, 2019.

[13] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, and E. Di Sciascio, "Semantic blockchain to improve scalability in the Internet of Things," *Open Journal of Internet Of Things (OJIOT)*, vol. 3, no. 1, pp. 46–61, 2017.

[14] R. Yaqub, S. Ahmad, H. Ali, and A. ul Asar, "AI and Blockchain Integrated Billing Architecture for Charging the Roaming Electric Vehicles," *IoT*, vol. 1, no. 2, pp. 382–397, 2020.

[15] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 112–125.

[16] K. Croman *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125.

[17] H. Wang, K. Chen, and D. Xu, "A maturity model for blockchain adoption," *Financial Innovation*, vol. 2, no. 12, pp. 1–5, 2016.

[18] V. Pureswaran and P. Brody, "Device democracy: Saving the future of the Internet of Things," IBM Institute for Business Value, Tech. Rep., September 2014, available: http://www-935.ibm.com/services/us/gbs/thoughtleadership/internetofthings, retrieved: August 2021.

[19] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2663–2668.

[20] M. Humayun, N. Jhanjhi, B. Hamid, and G. Ahmed, "Emerging smart logistics and transportation using IoT and blockchain," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 58–62, 2020.

[21] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling Collaborative Edge Computing for Software Defined Vehicular Networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[22] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[23] M. Zichichi, S. Ferretti, and G. D'Angelo, "Are Distributed Ledger Technologies Ready for Intelligent Transportation Systems?" in *3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, ser. CryBlock '20. ACM, 2020, p. 12–17.

[24] I. Jovović, S. Husnjak, I. Forenbacher, and S. Maček, "Innovative Application of 5G and Blockchain Technology in Industry 4.0," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 6, no. 18, pp. 1–6, 2019.

[25] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When Mobile Blockchain Meets Edge Computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.

[26] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 008–11 021, 2018.

[27] H. Liu, Y. Zhang, and T. Yang, "Blockchain-Enabled Security in Electric Vehicles Cloud and Edge Computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[28] J. Pillmann, C. Wietfeld, A. Zarcula, T. Raugust, and D. C. Alonso, "Novel common vehicle information model (CVIM) for future automotive vehicle big data marketplaces," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1910–1915.

[29] I. Williams, "Cross-chain blockchain networks, compatibility standards, and interoperability standards: The case of European Blockchain Services Infrastructure," in *Cross-Industry Use of Blockchain Technology and Opportunities for the Future*. IGI global, 2020, pp. 150–165.

[30] K. Valtanen, J. Backman, and S. Yrjölä, "Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case," in *2018 IEEE Wireless Communications and Networking Conference Workshops*. IEEE, 2018, pp. 185–190.

[31] M. B. Mollah *et al.*, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2021.

[32] VeChain Foundation, "VeChain Whitepaper 2.0 – #Creating valuable TXs," 12 2019, www.vechain.org/qfy-content/uploads/2020/01/VeChainWhitepaper_2.0_en.pdf, retrieved: August 2021.

[33] D. Miehle, D. Henze, A. Seitz, A. Luckow, and B. Bruegge, "PartChain: a decentralized traceability application for multi-tier supply chain networks in the automotive industry," in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. IEEE, 2019, pp. 140–145.

[34] L. M. Powell, J. Schwartz, and M. Hendon, "The Mobility Open Blockchain Initiative: Identity, Members, Technologies, and Future Trends," in *Revolutionary Applications of Blockchain-Enabled Privacy and Access Control*. IGI Global, 2021, pp. 99–118.

[35] European Commission, "Coordinating and synchronising multimodal transport improving road, rail, water and air transport through increased automation and user involvement," 2021, https://cordis.europa.eu/project/id/953618, retrieved: August 2021.

[36] D. Harrison, R. Bohm, T. Flynn, and V. Nadi, "Blockchain: the building block of trust," Salesforce, Tech. Rep., 2018, https://www.salesforce.com/content/dam/web/en_us/www/documents/white-papers/finserv-blockchain.pdf, retrieved: August 2021.

[37] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum Foundation, Tech. Rep., 2013, https://github.com/ethereum/wiki/wiki/White-Paper, retrieved: August 2021.

[38] K. Wüst and A. Gervais, "Do you need a blockchain?" in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.

[39] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Foundation, Tech. Rep., 2021, https://ethereum.github.io/yellowpaper/paper.pdf, retrieved: August 2021.

[40] C. Dannen, *Introducing Ethereum and Solidity*. APress, 2017, vol. 318.

[41] S. Popov, "The Tangle," IOTA Foundation, Tech. Rep. 1.8.3, 2018, https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf, retrieved: August 2021.

[42] P. C. Bartolomeu, E. Vieira, and J. Ferreira, "IOTA feasibility and perspectives for enabling vehicular applications," in *2018 IEEE Globecom Workshops*. IEEE, 2018, pp. 1–7.

[43] K. Olson *et al.*, "Sawtooth: an introduction," Hyperledger – The Linux Foundation, Tech. Rep., 2018, https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf, retrieved: August 2021.

[44] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[45] A. Corso, "Performance Analysis of Proof-of-Elapsed-Time (PoET) Consensus in the Sawtooth Blockchain Framework," Master's thesis, University of Oregon, 2019.

[46] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the Raft consensus algorithm for private blockchains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2019.

[47] E. Androulaki *et al.*, "Hyperledger Fabric: a distributed operating system for permissioned blockchains," in *Thirteenth EuroSys Conference*, 2018, pp. 1–15.

[48] D. Gabay, K. Akkaya, and M. Cebe, "Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowledge proofs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5760–5772, 2020.