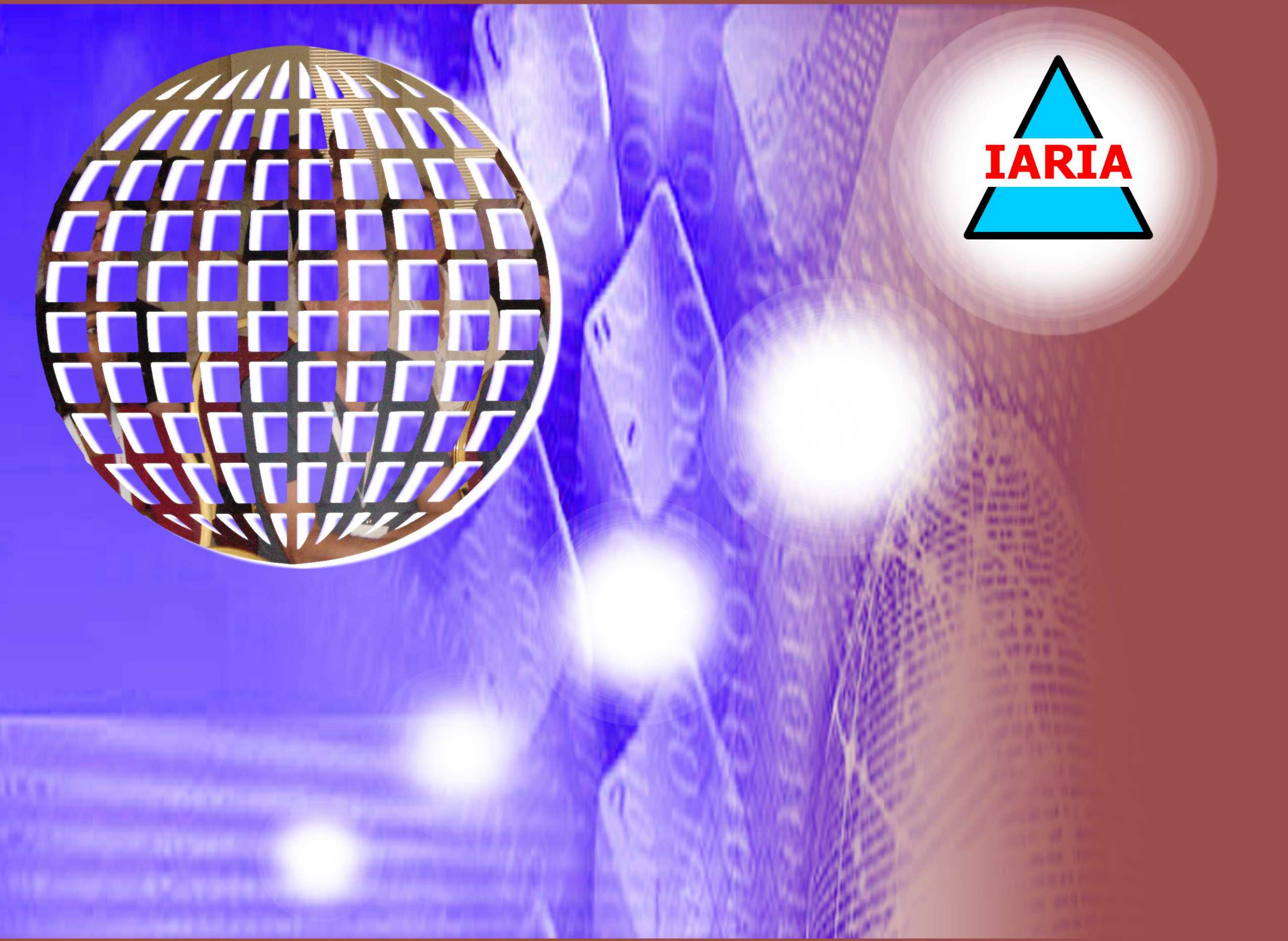
International Journal on Advances in Networks and Services



2021 vol. 14 nr. 1&2

The *International Journal on Advances in Networks and Services* is published by IARIA.
ISSN: 1942-2644
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

Reference should mention:

International Journal on Advances in Networks and Services, issn 1942-2644
vol. 14, no. 1 & 2, year 2021, http://www.iariajournals.org/networks_and_services/

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Networks and Services, issn 1942-2644
vol. 14, no. 1 & 2, year 2021, <start page>:<end page> , http://www.iariajournals.org/networks_and_services/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Networks and Services
Volume 14, Number 1 & 2, 2021

Editor-in-Chief

Editorial Advisory Board

Editorial Board

Christos Bouras, University of Patras, Greece
Mahmoud Brahimi, University of Msila, Algeria
Marco Bruti, Telecom Italia Sparkle S.p.A., Italy
Dumitru Burdescu, University of Craiova, Romania
Diletta Romana Cacciagrano, University of Camerino, Italy
Maria-Dolores Cano, Universidad Politécnica de Cartagena, Spain
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Eduardo Cerqueira, Federal University of Para, Brazil
Bruno Chatras, Orange Labs, France
Marc Cheboldaeff, Deloitte Consulting GmbH, Germany
Kong Cheng, Vencore Labs, USA
Dickson Chiu, Dickson Computer Systems, Hong Kong
Andrzej Chydzinski, Silesian University of Technology, Poland
Hugo Coll Ferri, Polytechnic University of Valencia, Spain
Noelia Correia, University of the Algarve, Portugal
Noël Crespi, Institut Telecom, Telecom SudParis, France
Paulo da Fonseca Pinto, Universidade Nova de Lisboa, Portugal
Orhan Dagdeviren, International Computer Institute/Ege University, Turkey
Philip Davies, Bournemouth and Poole College / Bournemouth University, UK
Carlton Davis, École Polytechnique de Montréal, Canada
Claudio de Castro Monteiro, Federal Institute of Education, Science and Technology of Tocantins, Brazil
João Henrique de Souza Pereira, University of São Paulo, Brazil
Javier Del Ser, Tecnalia Research & Innovation, Spain
Behnam Dezfouli, Universiti Teknologi Malaysia (UTM), Malaysia
Daniela Dragomirescu, LAAS-CNRS, University of Toulouse, France
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Wan Du, Nanyang Technological University (NTU), Singapore
Matthias Ehmann, Universität Bayreuth, Germany
Wael M El-Medany, University Of Bahrain, Bahrain
Imad H. Elhajj, American University of Beirut, Lebanon
Gledson Elias, Federal University of Paraíba, Brazil
Joshua Ellul, University of Malta, Malta
Rainer Falk, Siemens AG - Corporate Technology, Germany
Károly Farkas, Budapest University of Technology and Economics, Hungary
Huei-Wen Ferng, National Taiwan University of Science and Technology - Taipei, Taiwan
Gianluigi Ferrari, University of Parma, Italy
Mário F. S. Ferreira, University of Aveiro, Portugal
Bruno Filipe Marques, Polytechnic Institute of Viseu, Portugal
Ulrich Flegel, HFT Stuttgart, Germany
Juan J. Flores, Universidad Michoacana, Mexico
Ingo Friese, Deutsche Telekom AG - Berlin, Germany
Sebastian Fudickar, University of Potsdam, Germany
Stefania Galizia, Innova S.p.A., Italy
Ivan Ganchev, University of Limerick, Ireland / University of Plovdiv “Paisii Hilendarski”, Bulgaria
Miguel Garcia, Universitat Politecnica de Valencia, Spain
Emiliano Garcia-Palacios, Queens University Belfast, UK

Marc Gilg, University of Haute-Alsace, France
Debasis Giri, Haldia Institute of Technology, India
Markus Goldstein, Kyushu University, Japan
Luis Gomes, Universidade Nova Lisboa, Portugal
Anahita Gouya, Solution Architect, France
Mohamed Graiet, Institut Supérieur d'Informatique et de Mathématique de Monastir, Tunisie
Christos Grecos, University of West of Scotland, UK
Vic Grout, Glyndwr University, UK
Yi Gu, Middle Tennessee State University, USA
Angela Guercio, Kent State University, USA
Xiang Gui, Massey University, New Zealand
Mina S. Guirguis, Texas State University - San Marcos, USA
Tibor Gyires, School of Information Technology, Illinois State University, USA
Keijo Haataja, University of Eastern Finland, Finland
Gerhard Hancke, Royal Holloway / University of London, UK
R. Hariprakash, Arulmigu Meenakshi Amman College of Engineering, Chennai, India
Eva Hladká, CESNET & Masaryk University, Czech Republic
Hans-Joachim Hof, Munich University of Applied Sciences, Germany
Razib Iqbal, Amdocs, Canada
Abhaya Induruwa, Canterbury Christ Church University, UK
Muhammad Ismail, University of Waterloo, Canada
Vasanth Iyer, Florida International University, Miami, USA
Imad Jawhar, United Arab Emirates University, UAE
Aravind Kailas, University of North Carolina at Charlotte, USA
Mohamed Abd rabou Ahmed Kalil, Ilmenau University of Technology, Germany
Kyoung-Don Kang, State University of New York at Binghamton, USA
Sarfraz Khokhar, Cisco Systems Inc., USA
Vitaly Klyuev, University of Aizu, Japan
Jarkko Kneckt, Nokia Research Center, Finland
Dan Komosny, Brno University of Technology, Czech Republic
Ilker Korkmaz, Izmir University of Economics, Turkey
Tomas Koutny, University of West Bohemia, Czech Republic
Evangelos Kranakis, Carleton University - Ottawa, Canada
Lars Krueger, T-Systems International GmbH, Germany
Kae Hsiang Kwong, MIMOS Berhad, Malaysia
KP Lam, University of Keele, UK
Birger Lantow, University of Rostock, Germany
Hadi Larijani, Glasgow Caledonian Univ., UK
Annett Laube-Rosenpflanzer, Bern University of Applied Sciences, Switzerland
Gyu Myoung Lee, Institut Telecom, Telecom SudParis, France
Shiguo Lian, Orange Labs Beijing, China
Chiu-Kuo Liang, Chung Hua University, Hsinchu, Taiwan
Wei-Ming Lin, University of Texas at San Antonio, USA
David Lizcano, Universidad a Distancia de Madrid, Spain
Chengnian Long, Shanghai Jiao Tong University, China
Jonathan Loo, Middlesex University, UK

Pascal Lorenz, University of Haute Alsace, France
Albert A. Lysko, Council for Scientific and Industrial Research (CSIR), South Africa
Pavel Mach, Czech Technical University in Prague, Czech Republic
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Damien Magoni, University of Bordeaux, France
Ahmed Mahdy, Texas A&M University-Corpus Christi, USA
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Gianfranco Manes, University of Florence, Italy
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Moshe Timothy Masonta, Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa
Hamid Menouar, QU Wireless Innovations Center - Doha, Qatar
Guowang Miao, KTH, The Royal Institute of Technology, Sweden
Mohssen Mohammed, University of Cape Town, South Africa
Miklos Molnar, University Montpellier 2, France
Lorenzo Mossucca, Istituto Superiore Mario Boella, Italy
Jogesh K. Muppala, The Hong Kong University of Science and Technology, Hong Kong
Katsuhiro Naito, Mie University, Japan
Deok Hee Nam, Wilberforce University, USA
Sarmistha Neogy, Jadavpur University- Kolkata, India
Rui Neto Marinheiro, Instituto Universitário de Lisboa (ISCTE-IUL), Instituto de Telecomunicações, Portugal
David Newell, Bournemouth University - Bournemouth, UK
Ngoc Tu Nguyen, Missouri University of Science and Technology - Rolla, USA
Armando Nolasco Pinto, Universidade de Aveiro / Instituto de Telecomunicações, Portugal
Jason R.C. Nurse, University of Oxford, UK
Kazuya Odagiri, Sugiyama Jyogakuen University, Japan
Máirtín O'Droma, University of Limerick, Ireland
Jose Oscar Fajardo, University of the Basque Country, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Eleni Patouni, National & Kapodistrian University of Athens, Greece
Harry Perros, NC State University, USA
Miodrag Potkonjak, University of California - Los Angeles, USA
Yusnita Rahayu, Universiti Malaysia Pahang (UMP), Malaysia
Yenumula B. Reddy, Grambling State University, USA
Oliviero Riganelli, University of Milano Bicocca, Italy
Antonio Ruiz Martinez, University of Murcia, Spain
George S. Oreku, TIRDO / North West University, Tanzania/ South Africa
Sattar B. Sadkhan, Chairman of IEEE IRAQ Section, Iraq
Husnain Saeed, National University of Sciences & Technology (NUST), Pakistan
Addisson Salazar, Universidad Politecnica de Valencia, Spain
Sébastien Salva, University of Auvergne, France
Ioakeim Samaras, Aristotle University of Thessaloniki, Greece
Luz A. Sánchez-Gálvez, Benemérita Universidad Autónoma de Puebla, México
Teerapat Sanguankotchakorn, Asian Institute of Technology, Thailand
José Santa, University Centre of Defence at the Spanish Air Force Academy, Spain
Rajarshi Sanyal, Belgacom International Carrier Services, Belgium
Mohamad Sayed Hassan, Orange Labs, France

Thomas C. Schmidt, HAW Hamburg, Germany
Véronique Sebastien, University of Reunion Island, France
Jean-Pierre Seifert, Technische Universität Berlin & Telekom Innovation Laboratories, Germany
Dimitrios Serpanos, Univ. of Patras and ISI/RC ATHENA, Greece
Roman Y. Shtykh, Rakuten, Inc., Japan
Salman Ijaz Institute of Systems and Robotics, University of Algarve, Portugal
Adão Silva, University of Aveiro / Institute of Telecommunications, Portugal
Florian Skopik, AIT Austrian Institute of Technology, Austria
Karel Slavicek, Masaryk University, Czech Republic
Vahid Solouk, Urmia University of Technology, Iran
Peter Soreanu, ORT Braude College, Israel
Pedro Sousa, University of Minho, Portugal
Cristian Stanciu, University Politehnica of Bucharest, Romania
Vladimir Stantchev, SRH University Berlin, Germany
Radu Stoleru, Texas A&M University - College Station, USA
Lars Strand, Nofas, Norway
Stefan Strauβ, Austrian Academy of Sciences, Austria
Álvaro Suárez Sarmiento, University of Las Palmas de Gran Canaria, Spain
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Young-Joo Suh, POSTECH (Pohang University of Science and Technology), Korea
Junzhao Sun, University of Oulu, Finland
David R. Surma, Indiana University South Bend, USA
Yongning Tang, School of Information Technology, Illinois State University, USA
Yoshiaki Taniguchi, Kindai University, Japan
Anel Tanovic, BH Telecom d.d. Sarajevo, Bosnia and Herzegovina
Rui Teng, Advanced Telecommunications Research Institute International, Japan
Olivier Terzo, Istituto Superiore Mario Boella - Torino, Italy
Tzu-Chieh Tsai, National Chengchi University, Taiwan
Samyr Vale, Federal University of Maranhão - UFMA, Brazil
Dario Vieira, EFREI, France
Lukas Vojtech, Czech Technical University in Prague, Czech Republic
Michael von Riegen, University of Hamburg, Germany
You-Chiun Wang, National Sun Yat-Sen University, Taiwan
Gary R. Weckman, Ohio University, USA
Chih-Yu Wen, National Chung Hsing University, Taichung, Taiwan
Michelle Wetterwald, HeNetBot, France
Feng Xia, Dalian University of Technology, China
Kaiping Xue, USTC - Hefei, China
Mark Yampolskiy, Vanderbilt University, USA
Dongfang Yang, National Research Council, Canada
Qimin Yang, Harvey Mudd College, USA
Beytullah Yildiz, TOBB Economics and Technology University, Turkey
Anastasiya Yurchyshyna, University of Geneva, Switzerland
Sergey Y. Yurish, IFSA, Spain
Jelena Zdravkovic, Stockholm University, Sweden
Yuanyuan Zeng, Wuhan University, China

Weiliang Zhao, Macquarie University, Australia
Wenbing Zhao, Cleveland State University, USA
Zibin Zheng, The Chinese University of Hong Kong, China
Yongxin Zhu, Shanghai Jiao Tong University, China
Zuqing Zhu, University of Science and Technology of China, China
Martin Zimmermann, University of Applied Sciences Offenburg, Germany

International Journal on Advances in Networks and Services
Volume 14, Numbers 1 & 2, 2021

CONTENTS

Modelling and Reducing the Energy Usage During Multihop Transmissions in Wireless Sensor Networks

Anne-Lena Kampen
Western Norway University of Applied Sciences
Bergen, Norway
e-mail: alk@hvl.no

Knut Øvsthus
Western Norway University of Applied Sciences
Bergen, Norway
e-mail: kovs@hvl.no

Abstract— **The overriding focus in Wireless Sensor Networks (WSNs) has been to preserve energy to lengthen network lifetime. Sending and receiving data are the major energy consumer in the networks. Transmission energy usage is generally decided by transmission range, number of transmissions, and amount of data transmitted. The latter is also prominent for the receiving energy usage, although the number of overhearing nodes can be equally important. This paper first presents a model to investigate the tradeoff between the expected number of transmissions, transmission range, number of hops, and overhearing. The model shows that to reduce the energy consumed during multihop transmissions, the nodes should choose their successors close enough to prevent the expected number of transmissions from exceeding 1.4. The access protocol is Low Power Listening (LPL). Second, we suggest two solutions to optimize the LPL protocol, one which focuses on the nodes that are crucial for maintaining an operational network, i.e., the nodes whose successor is the sink, and one which reduce the energy usage of all the nodes by utilizing knowledge earned from earlier transmissions.**

Keywords-WSN; Energy; LPL; Energy-Modelling; Multihop; Overhearing

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of nodes interconnected through wireless communication. Multihop transmission is a key-technology to expand the area surveilled by the networks, and the energy consumption of the nodes should be low to lengthen the lifetime of the networks. Reducing energy consumption during multihop is discussed in [1] and further elaborated in this paper.

WSNs [2] are used in a wide range of areas from industrial applications [3] and smart grid [4][5] to healthcare [6], and used in all kinds of environments, from rural to urban areas [7]. The sensor nodes monitor their surroundings' characteristics, and relay collected information to a common central called the sink. The network has several advantages, such as flexibility, lack of wiring, and autonomous operation. WSN is one of the main parts of the Industrial Internet of Things (IIoT), a key technology in Industry 4.0 [8].

The overriding focus in WSNs has been to preserve energy. The nodes constituting a WSN are generally low-cost battery-powered devices with limited energy capacity. Hence, reducing energy consumption is essential to extend the individual nodes' lifetime and maintain a well-functioning network [9]. The radio is the primary energy consumer [10][11]. During operation, the radio switches between various states such as receiving, transmission, idle, and sleep, all of which consume different amounts of energy [5]. To save energy, the nodes should remain in the sleep state whenever possible. One of the most frequently cited energy-reducing approaches is the Low-Power-Listening (LPL) protocol [12][13], where nodes wake up periodically to sense the channel. To ensure successful data exchange, the senders transmit a preamble message to signal upcoming data transmission. The duration of the preamble must be long enough to ensure that the intended receiver hears it. This paper investigates further energy-reducing measures in networks running LPL. Note, the preamble discussed in this paper relates to the preamble-signal used to inform the receiver that a packet is about to be delivered. This is in contrast to the preamble that is part of the packet-frame and used to calibrate and synchronize transmitter-receiver clock cycles to prevent bit errors.

The total energy that is consumed to transmit data from a source to sink depends on several factors. First, all the nodes use energy to send their own generated data. Second, nodes along the routing path consume energy to receive and forward data. Third, overhearing nodes consume energy when they receive packets, which they afterward discard. These are the nodes located in the proximity of the path, such that they are covered by the transmissions intended for different destinations. Forth, energy is wasted when packets fail to reach the sink and must be retransmitted. One of the factors that impact packet delivery success is the distances between the successive nodes along the routing path. Successful packet delivery is likely when the distance is well within the transmission range. As the distance increases, the probability of success reduces until it gets unlikely as the distance increases beyond the transmission range. Thus, to maintain a high probability for successful delivery, the distance between the nodes along the path should be shorter than the transmission range. However, short distance means that the number of hops to reach the sink increases. At each hop, a node consumes energy to receive and subsequently transmit the packet. The consequence is that the total energy consumed to transmit a packet from source to destination increases. Another approach is, therefore, to increase the nodes' transmission range. However, such a solution requires that each node along the path increases the output power. Consequently, the energy consumed at each hop is increased, which also increases the energy consumed for transmission from source to destination. Thus, there is an energy-tradeoff between packet delivery-success, transmission range, and hop count. This paper investigates this energy-tradeoff.

As well as minimizing the total energy consumed, it is important to balance the workload in the network to avoid early depletion of nodes. Depleted nodes cannot provide their own sensed data, and, as a more severe consequence, they may lead to network partitioning. As data in WSNs are generally directed toward the sink, there is an innate energy imbalance in WSNs. That is, nodes in the proximity of the sink must forward data from nodes located further away such that the forwarding load increases with decreasing hop-count. Thus, the one-hop nodes deplete energy faster since they undergo the heaviest forwarding load. In addition, they are the most critical to keep the network connected.

To alleviate this imbalance, we suggest reducing the one-hop nodes' energy consumption by preventing them from transmitting the preamble. Remember, the preamble transmission is used to wake up and prepare the intended receivers to read the upcoming data packet. However, the sink is always awake and ready to receive.

Furthermore, to generally reduce the energy consumption in the network, we suggest a method that reduces the length of the preamble transmitted. The first time a node transmits a packet to its successor, it uses the whole preamble length. The successor uses the ACK message to inform the sender about the amount of time that it needed to wait to receive the data packet after being awakened by the preamble transmission. Being aware of this time, preamble transmission for the next packet transmitted to the successor starts just before the receiver wakes up to listen for activity.

The contribution of this paper is to investigate the tradeoff between the number of re-transmissions, transmission range, the number of overhearing nodes, and a number of hops in WSN to discover an energy optimal distance between the consecutive nodes along the path. In addition, we suggest two approaches that enhances the energy efficiency of the LPL solution. The schemes are verified by simulations and show that energy consumption is substantially reduced.

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 presents the energy model for one-hop transmission, while Section 4 presents the model for multihop transmission. The energy optimal transmission range is calculated in Section 5. Section 6 presents a model to calculate the energy consumed for nodes at various hop-counts, followed by an approach to reduce the one-hop nodes' energy consumption in Section 7. In Section 8 we suggest an approach to generally reduce the energy consumed in the network. Section 9 presents the conclusion.

II. RELATED WORK

In order to develop energy-efficient solutions for WSN, it is essential to understand the energy consumption of the individual nodes. Modeling of the energy-consuming activity provides valuable insight into this aspect.

The energy consumed is proportional to the time the nodes spend in the active state to transmit and receive. As the controller of the various radio states [14], the MAC protocol is essential to reduce energy usage. A common MAC layer method to save energy is to switch to the sleep state whenever possible [15]. However, to keep a WSN network connected and operational, the nodes must periodically switch to the active state. During the active periods, the nodes listen for transmissions, and they may exchange synchronization information [16]. The energy consumed for periodic wakeups is included in the model presented in [17], which calculates the energy consumption for communication, acquisition, and processing. The model illustrates energy reduction by reducing the number of active periods. A solution to reduce the need for periodic listening is to apply always-on wakeup radios with very low power consumption [18]. The always-on radio activates the central part of the nodes only when it detects activity on the medium. Although an interesting solution, it will not reduce the number of overheard transmissions, and the solution makes the nodes more complex.

Several models for energy consumption in WSN are found in the literature. A stochastic model that estimates the expected energy consumed, and the expected lifetime of WSN nodes, is presented in [19]. The model is based on the time the nodes spend in various states, such as sleeping, sensing, and relay. The communication is based on CSMA/CA. The deterministic energy bounds associated with maximum and minimum energy consumption are presented in the paper. In [20], a framework for modeling MAC protocols is presented. The framework can be used for energy calculations that are based on an absorbing Markov chain analysis. An analytical energy model that demonstrates the impact of the various parts of the PHY and MAC layer is presented in [21]. A receiver-initiated communication protocol is used, where the receivers periodically wake up and transmit a wakeup-beacon to signal that they are ready to receive. Testbed measurements that isolate hardware and software consumption are performed to understand the energy consumption and validate the model. It shows a relative error of 8% compared to the real energy estimate. A common aspect of these models is the focus on MAC-related activities related to switching between different states.

An energy consumption model that also includes overhearing is presented in [22]. The energy consumption is modeled both for sender- and receiver-initiated asynchronous MAC protocols, as well as synchronous MAC protocols for multimedia sensor networks. They found that the receiver-initiated protocols generally outperform sender-initiated protocols, although LPL performs well under low sampling rates. A weakness of the calculations is that the LPL protocol modeled is very conservative since only full preamble is considered.

Increased transmission range increases the senders' energy consumption. In addition, both the number of overhearing nodes and collision probability increase. The overhearing nodes waste energy to receive data addressed to neighboring nodes, and collisions require re-transmission. A number of analytical models are suggested to understand the energy impact of the transmission range. In [23], they use energy models to minimize the energy consumption of the nodes while meeting the delay constraints. The energy model suggested in [24] calculates the total energy consumed per successfully received bit. They study the tradeoff between energy per successfully received bit and the energy used for transmission. They find a single energy-optimal transmission range that is validated using real data. In [25], the energy consumption as a function of transmission range is modeled and used to balance the energy consumption among the nodes when new versions of programs are broadcasted throughout the network. Energy dissipation is modeled to study the impact of transmission power on both the data and the ACK packets in [26]. They assume a TDMA based communication model. When the data packets are much larger than the ACK packets, the latter should be sent with the highest possible output power to improve their delivery reliability. The reason is that higher output power increases the packet delivery-success probability.

There is an energy-tradeoff between the transmission range, the number of overhearing nodes, and the number of hops between source and destination. Increased transmission range may decrease the number of transmissions and the number of hops toward the sink. However, the number of overhearing nodes, as well as the transmission energy consumption, increases. Although presenting nice overview of energy-efficient routing protocols, covering solutions ranging from graphs to clustering approaches, [27][28], the tradeoff between overhearing and hop count is often not considered in the discussions. The hop count is considered in [29], though, where the transmission range is adjusted to balance the energy when transmitting data in multi-sink networks. In [30], overhearing is included, and the conclusion is that the transmission range should be short to reduce the number of overhearing nodes and reduce the collision probability. In contrast, twelve reasons for having a long transmission range are listed in [31]. One of the main reasons listed is that a longer transmission range makes the routing path closer to the Euclidian distance. However, overhearing would be a limiting factor since receiving consumes energy in the same order of magnitude as transmitting in WSN. In this paper, we investigate the effect of reducing overhearing. In addition, we take loss probability and routing distance to sink into consideration.

Several solutions to improve LPL is suggested in literature [32][33]. A broad range of these is based on dividing the preamble into small packets that contain the identity of the receiver node. One approach is to introduce a time delay between each pramble packet that is long enough for the receiver to send ACK. The ACK interrupts further preamble transmission, and triggers transmission of the data packet. A weakness of the method is increased energy consumed during periods of no activity since, due to the time delay introduced between the preamble pakcets, the nodes must stay awake for longer periods to check for channel activity. Another approach is to include the time-schedule for the upcoming transmission. The nodes enter sleep state, and only the intended receiver wakes up to receive the data packet. An approach to reduce the preamble transmission time of access points is to make the receives wake-up time decide the time for transmitting preamble for down-link transmission [34]. The method we suggest in Section 8 is similar, but we suggest that the management of preamble transmission is distributed and applicable for all type of communication, including multihop communication.

III. Energy model for one-hop transmission

In this section, the energy consumed during one-hop transmission is modeled. The communication protocol applied is LPL, which is a preamble-based protocol where nodes periodically wake up to listen for activity [12][13][35]. Between the wakeup periods, the nodes remain in sleep mode. A preamble message informs the neighboring nodes to stay awake to receive the message that is about to be sent. Its length is defined by the nodes with the longest sleep period to ensure that all nodes are informed. Upon receiving a preamble, the node remains active, listening for the rest of the preamble and the upcoming message.

Assuming that the nodes' sleeping time is approximately equal, the nodes will, on average, receive half of the preamble. For all the nodes except the intended receiver, this is a waste of energy. In order to reduce the energy consumed to receive the preamble, the preamble can be divided into small preamble-fractions containing the receiver's address and the start-time for the data-packet transmission [36]. Thus, the overhearing nodes can enter sleep mode after receiving a preamble-fraction. In addition, the intended receiver is no longer required to stay awake to receive the whole preamble. Rather, it can receive a fraction and then enter sleep mode until data transmission. We call this method divided-preamble.

To model the energy consumption, we assume a network that uses divided-preamble LPL. Figure 1 illustrates packet transmission. We assume that four nodes, named N1, N2, N3, and N4, all hear each other's transmissions. The red squares represent a data-packet that is sent from node N1 to N3. The dark blue squares represent the preamble, which is sent just before the associated data-packets. The duration of one preamble is p. Note that the divided-preamble is used. Consequently, the blue preamble squares are divided into fractions of length Δp. The preamble must be long enough to ensure that each node wakes up and listens at least once per preamble. Otherwise, they may lose a preamble transmission. The light blue shaded squares are the time periods when the nodes are in sleep mode. The periodic green squares, named L_T, are the time when nodes listen for activity. Hence, L_T must

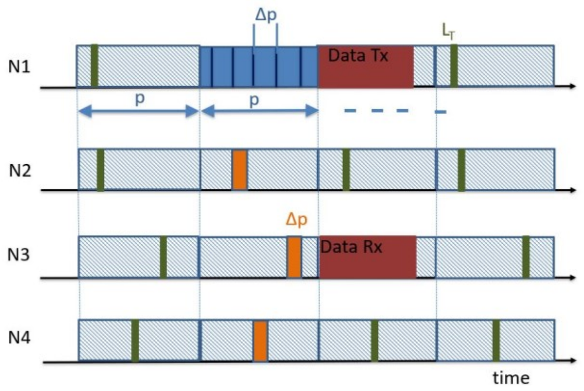


Figure 1. Packet transmission and reception in LPL using divided-preamble.

appear at least once per period p. The orange squares illustrate that the nodes received and read one of the preamble-fractions. Only the receiver wakes up to receive the data-packet, illustrated by the red square on node N3's timeline.

The nodes affected by one-hop transmission are the transmitting and receiving nodes, and the nodes overhearing the transmission. The transmission time for the packet is b. The power consumed for transmission consists of a fixed part, k_1, plus an offset, k_2, that is proportional to the radiated power [25][37]. The transmission range is d. A preamble, p, is transmitted prior to each data-packet, b. Thus, the energy consumption for transmission is $(k_1+k_2d^2)\cdot(b+p)$, represented by the first term in our model in (1). The second term in (1) calculates the energy consumed by the intended receiver as it receives the data packet. Receiving and listening consume a fixed amount of power, k_3. The preamble-fraction has a time duration Δp. It contains the receiver's address and the start-time for packet transmission. We assume that Δp includes both the preamble-fraction and the small interframe spacing between the fractions. The node density is λ. According to the common practice [38], we assume a randomly deployment of nodes, nodes with equal transmission range and calculate the node density according to a 2D Poisson density model. Thus, the number of nodes covered by the transmission is $\lambda\cdot\pi d^2$. On average, all nodes covered by the transmission receive 1.5·Δp. The reason is that the nodes must receive a whole preamble-fraction, but they wake up at a random time. That is, it is equally likely that a node wakes up at any point during the preamble-fraction transmission. However, it must receive the complete preamble-fraction to be able to read its content. Thus, if a node wakes up after transmission of a preamble-fraction has started, it must remain in the receiving state until it receives the subsequent complete preamble-fraction. The nodes will, therefore, on average, receive one-half preamble-fraction in addition to the complete fraction that it is able to read. The energy consumed is represented by the last term in (1). The number of overhearing nodes is calculated based on the node density. The transmitting node is accounted for by subtracting one node. Thus, the energy that is consumed per one-hop communication is:

$$E = (k_1 + k_2d^2)(b + p) + k_3b + 1.5\Delta p(k_3\pi\lambda d^2 - 1) \quad (1)$$

TABLE 1 LIST OF PARAMETERS AND ACRONYMS

Symbol	Meaning
k_1	Energy consumed to transmit, fixed part
k_2	Energy consumed to transmit, proportional to radiated power
k_3	Energy consumed to receive
λ	Node density
d	Transmission range
p	Preamble
b	Data packet
Δp	Preamble-fraction
q	Packet loss rate
x	Distance between communicating nodes
x_0	Knee value
x_1	Border area width
N	Number of nodes along a path
m	Number of transmission trials
D	Distance to sink
h	Hop-count distance to the sink
n_h	Number of nodes at hop distance h
Tx_{nh}	Number of transmissions for a node at hop-count n_h
ETX	Expected number of transmissions
PDR	Packet delivery rate
SD	Successor distance factor, $x = x_0\cdot SD$
E_h	Energy consumed for a node at hop-count h

IV. ENERGY MODEL CONSIDERING MULTIHOP COMMUNICATION AND LOSS PROBABILITY

Our focus is the energy consumed during data forwarding from source to sink. Our discussion is limited to networks that deploy one sink and energy efficiency along a single path. The goal is to investigate the impact of overhearing, transmission range, and re-transmission on the energy optimal transmission range. Short transmission ranges increase the number of hops between source and destination, thereby increasing the number of transmissions. Increasing the transmission range reduces the number of hops. The disadvantage is the increasing transmission energy consumption and the number of overhearing nodes increases due to a larger area covered by each transmission. The impact of the overhearing nodes is determined by how much of the transmission is being overheard.

A receiver experiences an increasing number of re-transmissions when it is located at the border area of the sender's transmission range [39]. To estimate this increase, we use the model presented in [40]. The model produces the graph shown on the left-hand side of Figure 2. The x-axis represents the distance between the sender and the receiver, and the y-axis represents the Packet Delivery Rate (PDR). The transmission range is approximately 10 m. The PDR equals 1 when the distance between the sender and the receiver is much shorter than the transmission range. Furthermore, there is a transition area in the vicinity of the transmission rage where the PDR starts to change and bends towards zero. This is the border area. The distance between a transmitter and its border area increases with increasing transmission power. Hence, the number of re-transmissions can be reduced by increasing the transmission energy.

Based on the border-area discussion above, the total number of re-transmissions along the path from source to sink

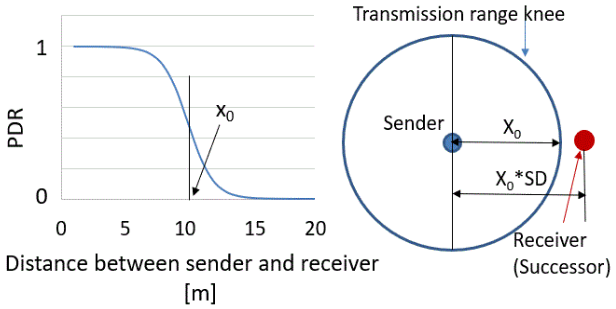


Figure 2. The left-hand side of the figure shows the PDR for increasing distance between sender and receiver. The blue sender node on the right-hand side has a knee-point boarder tx line according to the blue circle.

depends on the nodes' transmission range and the associated hop-to-hop distance, i.e., the distance between the transmitting node and its successor. Assuming equal transmission power and hop distances, the expected number of transmissions (ETX) along a path is [40] found to be:

$$ETX[N] = \frac{1-q^m-(1-q^m)^N}{q^m(1-q)} \quad (2)$$

N is the number of nodes along the path, which is equal to the distance to the sink, D, divided by transmission range, d. The factor m denotes the maximum number of transmission trials, i.e., the maximum number of re-transmissions is (m-1). The parameter named q, denotes the packet loss rate. q = 1-PDR, and PDR(x) is given by [40]:

$$PDR(x) = \frac{1}{1+e^{\frac{x-x_0}{x_1}}} \quad (3)$$

where x is the distance between the transmitting node and its successor, and x_1 defines the width of the border area. x_0 is in the middle of the border area, i.e., x_0 is the knee value as shown in Figure 2. The expected number of transmissions along a path, ETX[N], depends on the packet loss rate q. The energy consumed for transmitting a packet from source to sink can be found by introducing ETX[N] in (1):

$$E = ETX[N][\,(k_1 + k_2d^2)(b+p) + k_3b + 1.5\Delta pk_3(\pi\lambda d^2 - 1)] \quad (4)$$

Equation (4) shows that ETX[N] has an important impact on energy consumption. ETX[N] increases when the number of hops along the path toward the sink increases. In addition, it increases when the distance between sender and successor is far enough for the successor to be located inside the sender's border area. To alleviate the impact from both of these factors, the transmission power can be boosted to extend the distance to the border area. The disadvantage would be increased energy consumed for transmission, and increased number of overhearing nodes. Thus, there is a tradeoff between hop-count, packet delivery rate (here represented by ETX), overhearing, and transmission range. The tradeoff is investigated in the next section.

V. ENERGY OPTIMAL TRANSMISSION

We use (4) to investigate the tradeoff between hop-count, packet delivery rate, overhearing, and transmission range. It is assumed that the distance between the senders and the receivers is equal for each hop along the path from the source node to the sink. The right-hand side of Figure 2 illustrates the sender-receiver distance for one of these individual hops along the path. The blue node represents one sender, and the blue circle represents the associated knee-point value, x_0, for the sender's transmission range. The red dot represents a receiver that is located beyond the sender's knee-point transmission range. In order to model this sender-receiver distance, we choose to represent it as the knee-point value times a constant. The constant is named Successor Distance factor (SD), i.e., x = x_0·SD. Hence, the red node has SD higher than 1. A node located on the blue circle will have SD = 1, and a node located inside the blue circle would have a SD lower than 1.

The parameter values used in the calculations are the values presented in [37]. The values are based on the CC1000_radio [41]. For CC1000, k_3 and k_2 are in the same order of magnitude, while k_2 is much lower than k_3. Other, and more recent, radios may have different numerical values. However, the characteristics are similar among WSN nodes [10][25], and the ratio between the energy consumed for transmission, receiving and idle is still valid. In addition, changing one of the values k_1, k_2, or k_3 by ± 20% does not alter the result presented below. Hence, our calculations present a general trend. The values for k_1, k_2 and k_3 are 36.1µJ/bit, 0.06 pJ/bit/m2 and 37.5 µJ/bit respectively. The preamble-time, p, is normalized with respect to data-packet time, b. The transmission range d = 10m and the node density λ = 0.015. The preamble-length is 5·data-packet length. The distance to the sink is set to D = 50m and the maximum number of re-transmissions is m = 20.

In the calculations, the successor node is located at x = x_0·SD. Thus, the number of nodes along a path is N = round-up-upward(D/x). Calculating energy consumed the overhearing nodes is challenging. The reason is that some are located inside x_0, but do not receive the preamble or are not able to correctly decode the preamble. The same applies for some of the nodes that are located in the border area beyond x_0. As an average, we assume that all nodes inside x_0 receive the preamble.

Figure 3 shows how the total energy consumption varies as the transmission range increases. The y-axis represents the energy consumption, and the x-axis represents the transmission range knee value, x_0. Thus, moving toward higher x-axis values, the transmission power increases, extending the transmission range. ETXper-hop changes with transmission range and is calculated using equations presented in [40], see reference for explanation: $ETX(m) = (1-q^m)/(1-q)$. The figure shows three different graphs representing three different SD parameters. Remember, SD defines the sender-receiver distance for each hop along the path. For the blue graph SD = 0.5 (ETXper-hop = 1.19), for the orange graph the

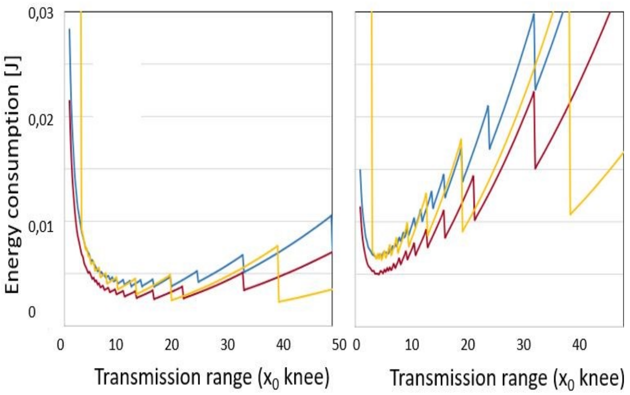


Figure 3. The left-hand side shows the energy consumed when the preamble-fraction received by overhearing nods is 0.1 times the complete preamble. The right-hand fraction is increased to 0.5 times the complete preamble.

SD = 0.75 (ETXper-hop = 1.43) and for the yellow graph SD = 1.25 (ETXper-hop = 3.3).

First, we concentrate on the impact of SD distance. The smallest SD, the blue graph, generally gives the highest consumption. The reason is that low SD means short hop-to-hop distances, such that the number of hops from source to sink is high. Remember, each hop adds at least one packet transmission, causing energy to increase due to transmission, receiving, and overhearing. When SD increases, the hop-count decreases, reducing the energy consumed. However, as the SD is further increased, the increase in ETX[N] cancels the positive effect of the reduced number of hops, because the successor is too far into the border area. The energy consumption for SD = 1.25 is generally higher than for SD = 0.75, although the hop-count for SD = 1.25 is the lowest due to the long sender-to-receiver distance. Performing the calculations with various SD shows that energy consumption is lowest when SD is about 0.75 (ETXper-hop = 1.4). That is, the successor nodes should be chosen so far into the border area that the ETXper-hop = 1.4. This number cannot be treated as an exact value, it depends on radio characteristics and scenario, and experimental results are likely to give other values. However, it indicates that there is an optimal point, and this point is beyond the point where transmission is always successful. The result is comparable to the discussions and findings in [12], which investigate energy minimization for LPL in noisy environments. It is found that noise-triggered false wakeups can be a dominant energy consumption factor. In our case, overhearing causes unnecessary wakeups, which does not provide any valuable information. Thus, it should be limited.

Furthermore, Figure 3 shows that there exists an energy optimal transmission range, which is mainly determined by the overhearing nodes' energy consumption. The optimal transmission range is more pronounced and shorter as Δp (fraction of preamble received by neighboring nodes) increases. The graph to the right in Figure 3 has the highest Δp and the shortest and most pronounced optimal transmission range. The reason is that increased Δp causes increased energy consumption among overhearing nodes, because they receive a larger fraction of the transmitted preamble. Combined with the fact that the number of overhearing nodes increases quadratic with distance, the energy optimal transmission range is reduced to reduce the impact of overhearing. When Δp is low, the optimal transmission range is less pronounced and longer because the impact of overhearing is much lower. The optimal transmission range is about 10m for λ = 0.015. Thus, the average number of nodes covered by the transmission is 4.71, which may be too few to ensure a connected network [42]. The conclusion is that it is energy efficient to keep the transmission range short, considering that the range is long enough to keep the network connected. Other parameter values would give other results. For instance, there is no pronounced energy optimal transmission range if Δp is reduced to below 0.02 while the other parameters are kept unchanged. Reducing the preamble, p, to data-packet size has the same effect of making the optimum-point less pronounced. On the contrary, increasing the node density makes it more pronounced. However, the energy optimal distance, between a node and its successor, is the distance where ETXper-hop is 1.4. This applies for all the various parameter settings. Deciding a distance that gives ETXper-hop = 1.4 is not realistic in real-world scenarios since environmental characteristics are prone both to temporal and spatial changes. In addition, the parameter settings both for the radio as well as other parameters, such as packet size would vary, resulting in a slightly different optimal ETXper-hop. However, our result shows a valid trend, the optimal distance between successor nodes should not be too far into the border area, i.e., the area where the PDR starts to change and bends towards zero.

We observe a sawtooth shape of the curves in Figure 3. The smooth increasing energy consumption is caused by the increasing transmission range, which increases the number of overhearing nodes. The abrupt drop in energy consumption occurs as the path is suddenly reduced by one hop, caused by a longer transmission range. The result is a sharp reduction in overhearing energy consumption since the number of transmissions is reduced. This is illustrated in Figure 4. Remember that the overhearing energy consumption is proportional to the node density times the area covered by the transmission. In the scenario in the upper part of the figure, three hops are used to send from node N1 to the sink. Nodes N2 and N3 are relaying nodes. The first two hops are represented by blue arrows. The last hop is illustrated by the broad red arrow just to make this last hop visible in the figure. The blue circle surrounding the nodes N1, N2, and N3 represent their transmission ranges. In the scenario in the lower part of the figure, the transmission range is increased just enough to reduce the number of hops from N1 to sink, from three to two hops. The energy consumption is reduced from the upper to the lower scenario. The reason is the area to the right of the sink in the upper scenario. This area is not covered by any transmissions in the lower scenario. In addition, although the overlapping area is somewhat smaller, there are two, instead of one, overlapping areas in the upper scenario. Nodes located inside the area where the transmission

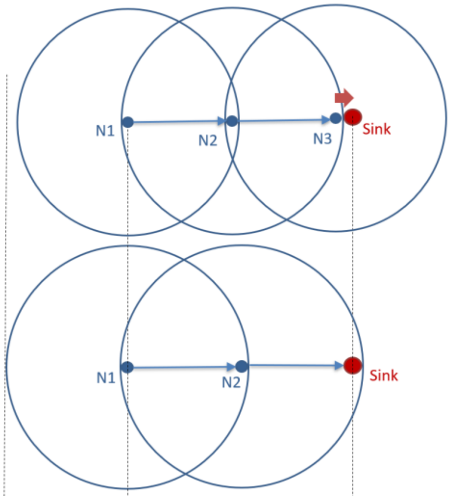


Figure 4. In the lower figure the transmission range is increased just enough to reduce the number of hops by one compared to the upper figure.

range from consecutive nodes overlap, receive the same packet twice. For instance, nodes located in the overlapping area in the lower scenario will first receive a packet when node N1 transmits it, then when node N2 transmits it. In addition, for the upper scenario, the nodes located inside the small area covered by transmission from both node N1, N2 and N3 will overhear the packet three times: First when node N1 transmits, then when node N2 transmits, and last when node N3 transmits.

Looking back at Figure 3, the deepest sawtooth decrease in energy consumption occurs for the longest transmission ranges. The reason is that the area to the right of the sink in the upper scenario, as well as the overlapping areas, are bigger the longer the transmission range.

Our assessment is limited to assessing transmission from a source to the sink. This represents the first step toward energy-efficient transmission. Parameters such as latency and security are not taken into account. The next step is to investigate more realistic scenarios that include the effects of transmissions along several paths. This introduces packet collisions that increase with traffic load and transmission range. Collisions cause retransmissions, thus increasing the expected number of transmissions ETX[N]. To assess the impact of collisions caused by transmission range, (4) where multiplied with the expected increase related to the area covered, i.e., (4) was multiplied by (1+(ExpectedCollisions*(λ*3,14*d^2)). Increasing the ExpectedCollison from zero to 1 showed that the energy optimal distance remained unchanged, but was more pronounced. The traffic load is decided by the application, type of sensors, data collected, etc. Assuming an equal average increase of retransmission across all hops from source to sink would not change the energy optimal distance, although the total energy consumed increases according to the average increase in retranimssions. However, the number of collisions increases towards the sink since the traffic load increases. Thus, the energy optimal distance is likely to be reduced towards the sink to compensate for the traffic-related increase in collisions. Reducing the preamble length, as suggested in Section 7 and 8, would reduce the collision probability since the channel occupation time is compressed.

VI. Energy balance in WSN

Although LPL is an efficient method for reducing the energy consumption in WSN, there is an imbalance in energy consumption among the nodes. The traffic load at a hopcount adds its traffic to the nodes one hopcount closer to the sink. Therefore, the energy consumption due to forwarding increases towards the sink. The reason is that nodes closer to the sink must forward packets from nodes further away. The consequence is that the one-hop nodes experience the highest energy-cost due to their packet forwarding.

To investigate energy consumption versus hop-count, we assume a fair workload balance between the nodes. Fair means equal load-balanced among the nodes at a given hop-count. Assume that the nodes' transmission range is d, and h represents the number of hops to the sink, i.e., h ϵ [1, h_{max}], h_{max} is the maximum number of hops. The number of nodes located at (h+1) hops from the sink is equal to the number of nodes inside the donut-shaped area with an outer radius of d·(h+1) and an inner radius of d·h. The number of nodes in the donut-shaped-area is found by multiplying the area with the node density, λ:

$$n_h = \pi\lambda[(h*d)^2 - ((h-1)d)^2] = \pi\lambda(2h-1)d^2 \quad (5)$$

Nodes at hop-count h forwards data on behalf of a given number of nodes at hop-count h+1. The average number of nodes use a given node at hop-count h is:

$$\frac{n_{h+1}}{n_h} = \frac{2h+1}{2h-1} \quad (6)$$

A node at hop-count h transmits one of its packets. In addition, the nodes forward traffic from the one-hop predecessors. The number of nodes at h+1 is $n_{h+1,}$ and we assume that the traffic from these nodes is equally shared among the n_h nodes at hop-count h. The total number of transmissions for a node at hop-count n_h is, therefore:

$$Tx_{nh} = 1 + \frac{n_{h+1}}{n_h} \cdot Tx_{n(h+1)} \quad (7)$$

Based on (7), the energy consumed for nodes at a given hop-count is presented in (8). The first term in (8) represents the transmission energy. The second term represents the energy used to receive packets for forwarding. Remember, the preamble is received for each received data-packet. Besides, the nodes have equal transmission range, so all the nodes overhear neighbors' transmissions from all nodes located inside its transmission range, πd^2. See Figure 5, the blue node N1 overhears transmission from all nodes located inside the blue circle, and the green node N2 overhears transmissions from all nodes located inside the green circle. Some of the overheard neighbors are located at the same hop-count distances from the sink as the overhearing node, while some are located at adjacent hop-count distances. This is also illustrated in Figure 5. The red inner circle is the sink. The area between the sink and the inner red dotted circles represents the area where one-hop nodes are located, the area between the

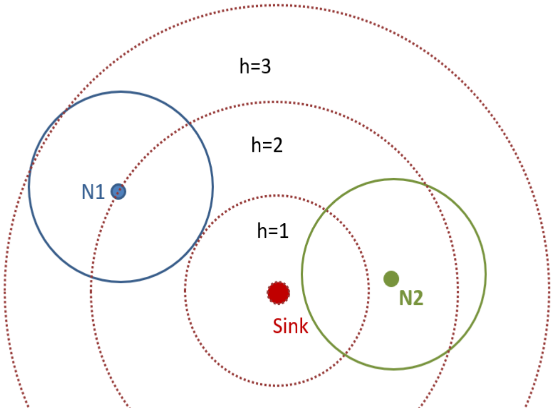


Figure 5. Transmission from nodes located inside the blue circle will be overheard by the blue node N1. The same apply for nodes located in the green circle and node N2.

first and the second inner red dotted circle represents the area for nodes located at two-hop distance from the sink, the area between the second and the third red dotted circle represents the area for nodes located at three-hop distance from the sink, and so forth. The contribution from all these three areas is not equal due to varying area coverage, as shown in Figure 5. However, as a rough estimation, it is fair to assume that, on average, the contribution from all these three hop-count distances is equal, and the number of transmissions overheard is as expressed in the multiplicand (first parenthesis) of the last term in (8). Thus, to investigate the energy imbalance, the average energy consumed for a node at hop-count h can be calculated as:

$$E_h = Tx_{nh}[\,(k_1 + k_2 d^2)(b + p)] + \; k_3(Tx_{nh} - 1) \cdot (b + 1.5\Delta p) + (\tfrac{Tx_{nh-1}+Tx_{nh}+Tx_{nh+1}}{3})(\pi\lambda d^2 k_3 1.5\Delta p) \quad (8)$$

Section VII presents the validation of equation (8) by comparing it to simulated results. Before this, we present some observations based on the equations. Starting with equation (7). It quantifies the average traffic load at each rank along the path. As can be seen, the traffic load grows for each rank from the outermost leaf nodes to the sink. This increase places an added burden on the nodes. In addition, the number of relaying nodes decreases. This is understood by investigating Figure 5. The donut-shaped-area for the one-hop nodes (h=1) is smaller than the donut-shaped-area for the two-hop nodes (h=2), and so forth. Combined with a uniform node density, λ, the number of nodes at each rank decreases toward the sink.

Looking at (8), the parameter h is used to quantify the imbalance in energy consumption between nodes at different hop-count. The following discussion focuses on three different contributions. First, forwarding data packets, second, signaling the upcoming transmission of a data packet, and third, the energy consumed in overhearing nodes.

First, reviewing (8), the average energy consumption is directly related to packet transmission and receiving data packets at rank h is:

$$E_{\mathrm{TRxD}} = Tx_{nh}\;\,(k_1 + k_2 d^2)\,\mathrm{b} + \; k_3(Tx_{nh} - 1) \cdot b$$

The first term is the energy a node at rank h use to transmit data packets. The last term represents the energy it consumes to receive packets. Thus, this consumption is related to the actual transmission of the real data, which cannot be reduced unless aggregation or some other intelligent data processing approaches are used. Such processing is out of the scope of this paper.

Second, again reviewing (8), the average energy a node at rank h consumes for transmitting the preamble and reception of the fraction Δp is:

$$E_{TxRxP} = Tx_{nh}\,(k_1 + k_2 d^2)\,\mathrm{p} + \; k_3(Tx_{nh} - 1) \cdot \; 1.5\Delta p$$

The first term represents the nodes' energy consumption to signal the successor that a data packet will be transmitted. The second term represents the energy it consumes to be informed from the predecessor that a packet addressed to it is soon to be sent. Thus, the two presented energy terms relate directly to the transmission of the data packets. However, transmitting the complete preamble, p, represents a waste of energy. Basically, only the intended receiver must be informed about the upcoming transmission. Methods to reduce p are assessed in the following section, where the ambition is to decrease the overall network energy consumption.

The third term represents the energy consumed for overhearing:

$$E_{RxOvehering} = \pi\lambda d^2 k_3 1.5\Delta p \;\frac{Tx_{nh-1} + Tx_{nh} + Tx_{nh+1}}{3}$$

Overhearing is basically a waste of energy since no useful data is exchanged. Reducing p, which is our intention, may have some impact on overhearing. This is if the data packet plus the reduced p in total is shorter than the nodes' sleep periods. In that case, it is likely that some of the nodes will remain at sleep state during the whole transmission, i.e., some nodes will not sense the medium until after the transmission is finished. Hence, they will not waste energy to overhear the transmission.

VII. Balancing energy consumption

Based on the discussion above, the one-hop nodes consume much more energy than the other nodes. However, messages sent from the one-hop nodes are destined to the sink, which is always active. Therefore, in order to save energy, we suggest canceling the preamble from the one-hop nodes. The nodes are aware of their identity as one-hop nodes by looking in the routing table: their successor nodes are the sink, and their distance to the sink is one hop.

Simulations are performed to compare when all nodes apply the same divided-preamble LPL algorithm against the case when the one-hop nodes are prevented from transmitting the preamble. The parameter investigated is total energy consumption. The simulation is performed in Omnet++ [43].

The applied routing metric is hop-count, thus, the routing protocol generate a graph that is directed from the nodes toward the sink. The graph is generated before any packet is transmitted, and the energy consumed to construct the graph is excluded from the calculations. Each node generates 100 data packets during each simulation. The preamble time is four times the duration of a data packet. The preamble-fraction packets are one-tenth of the data-packet size. The transmission power consumption is fixed since the transmission range is equal for all nodes. Energy consumed for overhearing is not considered because the number of overhearing packets would be equal for both scenarios: The number of packets transmitted is equal for both scenarios, and, although the one-hop nodes do not transmit preamble, neighbors must receive and read all overheard packets in order to decide whether the packet is destined for them. 205 nodes are randomly distributed in an area of 1000 m times 1000 m. Thus, the network consists of nodes that are randomly deployed, and for each simulation the deployment changes. The transmission range of all nodes is 141 m.

The simulation results are shown on the left-hand side of Figure 6. Every simulation point presented in the graphs represents the average value of 100 simulation runs with

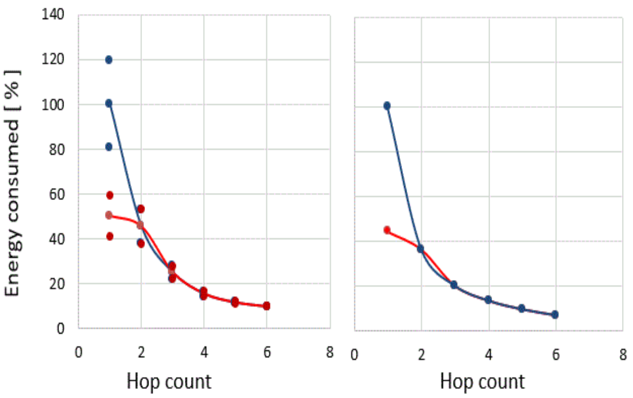


Figure 6. Energy consumption for nodes at different hop-count distances from the sink. The graphs on the right- hand side show calculated results. The graphs on the left-hand side show simulated results.

different seeds for the random deployment of nodes. The red curve shows the simulation result when the one-hop nodes are prevented from transmitting preamble, while the blue curve shows the energy consumed when the one-hop nodes behave equal to the other nodes, i.e., transmit preamble. The continuous curves represent average values, and the marks over and below represent the 95% confidence interval.

The simulations show that one-hop nodes' energy consumption is reduced by about 50% when the one-hop nodes are prevented from transmitting the preamble. Calculations using (9) verify the simulated result, as shown on the figure's right-hand side. The energy reduction achieved depends on various factors, the main being the ratio of preamble size to data-packet size. A short preamble gives less energy-saving than a larger preamble. For instance, the energy saving is reduced to 19% if the preamble to data-packet-size is reduced to 0.5. Avoiding preamble transmission would reduce one-hop node's energy consumption, which are the most critical nodes to keep the network connected. Preventing transmission of the preamble is equal to reducing the duty-cycle of the nodes, and our result complies with the results in [44], where duty cycling is used to manage the delay as well as energy consumption of the nodes. The duty-cycle of the hot-spot nodes, which equals the one-hop nodes, is kept low compared to the duty-cycle of nodes in non-hotspots areas.

VIII. Generally Reducing Energy consumption

Preamble transmission used in LPL is important since it enables the nodes to individually enter sleep-mode. That is, the sleep-periods are unsynchronized, which reduces the complexity of the network. Furthermore, it reduces the management traffic and energy consumption. However, since the preamble is not carrying any actual payload, it should be reduced as much as possible.

To this end, we suggest letting receivers include a few bites of information in the ACK packet. Specifically, the ACK packets inform the sender how long the receiver needed to wait to receive the data packet after it detected the preamble. This is illustrated by the gray upward pointing arrows in Figure 7. Assuming that all nodes have equal sleep periods, the sender is now able to predict the receiver's wakeup-periods. Remember, although the nodes' sleep periods are not synchronized, we assume that the sleep periods of the nodes are roughly equal. The sleeping intervals are represented by the blue equal-length arrows below the time-axis. Furthermore, the sender knows when it started its preamble transmission and the length of the preamble. That is, the sender knows that it started the preamble transmission at time t₂, and that it started transmitting the data packet at time t₃. Together with the information about the duration the receiver waited for the data after detecting the preamble, i.e., the time Δt, the sender can predict the time when the receiver wakes up. Thus, the next time the sender transmits data to this successor, it reduces the preamble because it knows when the receiver is awake and ready to receive. The preamble can be reduced to a single preamble-fraction or maybe a few preamble-fractions to account for minor synchronization inconsistencies between the nodes.

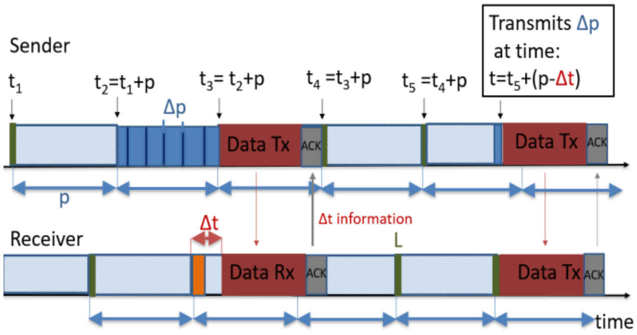


Figure 7. Based on information received in the first ACK, the sender predict the receiver wakeup period and reduces the preamble for the next data packet.

Simulations to show the impact of using a single preamble-fraction to signal all data packets, except for the very first packet transmitted to a given successor, are presented in Figure 8. The routing protocol used in the network is RPL, using hop count as a metric. RPL creates an acyclic graph directed toward the sink, such that all nodes

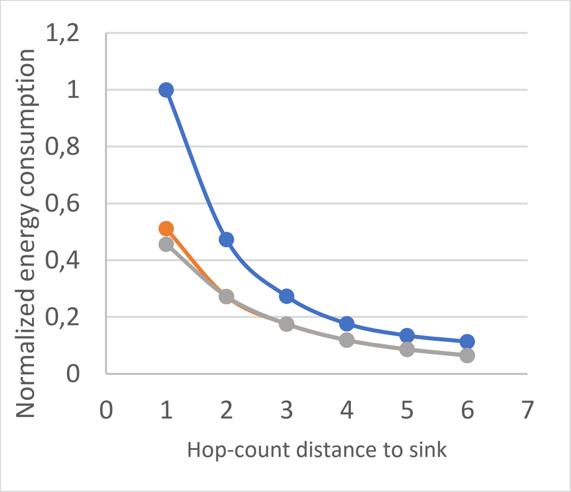


Figure 8. Simulated energy consumption vs rank. Blue curve: Basic preamble. Orange curve: All packets, except the first, are signaled using a preamble-fraction. Gray curve: same as the orange curve, except for the one-hop nodes that do not send preamble.

select a specific successor node for all data packets, both its own generated data and the data that it forwards on behalf of other nodes located further from the sink. Thus, all data packets transmitted from a given node are sent to the same successor node, and a complete preamble is only required for the first of these data packets. The rest of the data packets are transmitted using the proposed method. The reason is that the sender learned the wakeup-periods of its receivers, as explained above. In the simulations, the length of the preamble-fraction used for the packets after the first data packet is 1/10 of an entire preamble. The blue curve in Figure 8 shows the energy consumed when an entire preamble is used for all data packets. The y-axis shows the energy consumption normalized by the highest energy consumed by a node, i.e., the energy consumed by one-hop nodes using the entire preamble for all data packets. The orange curve shows the energy consumed using the proposed method. Only the first data packets transmitted implement a complete preamble, and the rest uses a preamble fraction. The gray curve shows the same as the orange curve, except for the one-hop nodes that do not need to send preamble. Remember, the sink is always awake and ready to receive. The energy reduction between the blue and the grey curve for the one-hop nodes equals the reduction observed in Figure 6. The reason is that the preamble is removed in both cases. However, for nodes at higher hop-count, the graphs show that energy consumption is reduced by reducing the preamble fraction for all data packets except the first one sent. We observe that the reduction varies from about 5 % to 20%. The energy saved depends on the difference between the length of a preamble fraction and the length of a complete preamble. In addition, it depends on whether an entire preamble must be transmitted occasionally or periodically to correct for drift or change in the successor's wakeup-time.

IX. Conclusion

To reduce the energy consumed in multihop transmission in WSN the tradeoff between the number of re-transmissions, overhearing, number of hops, and transmission range are investigated. Due to improved Packet Delivery Rate (PDR), less energy is wasted on re-transmissions when the distance between senders and receivers along the routing paths is reduced. However, the number of hops to reach the sink is increased, such that more nodes must use energy to forward the data. Another solution is to increase the nodes' output power to increase the distance to where the PDR starts to fail. In this way, the distance between senders and receivers can increase without introducing more re-transmissions. However, each transmission consumes more energy. In addition, the overhearing nodes must be considered. Their contribution to energy consumption increases with the number of nodes covered by the transmissions, the number of transmissions, and the size of the received packet. Investigating the tradeoff between all mentioned factors, we find that the optimal solution is for the nodes to choose their successors at a distance that gives an expected number of transmissions, ETX per-hop, of approximately 1.4. The exact number depends on radio characteristics and scenario. However, there is an optimal point, and this point is beyond the point where transmission is always successful.

In addition, we suggest two different algorithms to reduce the energy usage in the network. The first focuses on the nodes that are the most important to maintain a connected network: the nodes whose successor is the sink. The energy consumption of these nodes is substantially reduced by preventing them from transmitting the preamble. The preamble can be omitted since the sink is always awake and ready to receive.

The second algorithm reduces the energy consumption of all nodes in the network by substantially reducing the length of the preamble transmitted to signal that a data packet is about to be transmitted. This is achieved as the sender learns the successor wake-up periods by logging the exact point in the preamble where the successor woke up when the first packet was transmitted. Based on this information, the subsequent preambles are reduced and sent when the successor nodes wake up. Thus, the energy consumption in the network is reduced.

Future work on energy consumption in WSN will focus on intelligent forwarding. Nodes will predict the traffic patterns to optimize their duty-cycle and prevent overhearing.

References

[1] A-L.Kampen and K. Øvsthus, "Reducing The Energy Consumed During Multihop Transmissions in Wireless Sensor Networks," in The Fourteenth International Conference on Sensor Technologies and Applications, SENSORCOMM. International Academy, Research and Industry Association (IARIA), pp. 11-18, 2020.

[2] S. J. Ramson and D. J. Moni, "Applications of wireless sensor networks—A survey," in 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), IEEE, pp. 325-329, 2017.

[3] X. Li, D. Li, D., J. Wan, A. V. Vasilakos,C. F. Lai, and S. Wang, "A review of industrial wireless networks in th context of Industry 4.0," Wireless networks, vol. 23, no. 1, pp. 23-41, 2017.

[4] M. Erol-Kantarci and H. T. Mouftah, "Wireless sensor networks for cost-efficient residential energy management in the smart grid," IEEE Transactions on Smart Grid, vol. 2, no. 2, pp. 314-325, 2011.

[5] R. W. R. de Souza, L. R. Moreira, J. J. Rodrigues, R. R. Moreira, and V. H. C. de Albuquerque, "Deploying wireless sensor networks–based smart grid for smart meters monitoring and control," International Journal of Communication Systems, vol. 31, no. 10, p. e3557, 2018.

[6] N. Dey, A. S. Ashour, F. Shi, S. J. Fong, and R. S. Sherratt, "Developing residential wireless sensor networks for ECG healthcare monitoring," IEEE Transactions on Consumer Electronics, vol. 63, no. 4, pp. 442-449, 2017.

[7] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," Journal of network and computer applications, vol. 60, pp. 192-219, 2016.

[8] Ghobakhloo, Morteza. "Industry 4.0, digitization, and opportunities for sustainability." Journal of cleaner production Vol 252: 119869, 2020

[9] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 828-854, 2017.

[10] Đ. Banđur, B. Jakšić, M. Banđur, and S. Jović, "An analysis of energy efficiency in Wireless Sensor Networks (WSNs) applied in smart agriculture," Computers and electronics in agriculture, vol. 156, pp. 500-507, 2019.

[11] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler, "A building block approach to sensornet systems," in Proceedings of the 6th ACM conference on Embedded network sensor systems, pp. 267-280, 2008.

[12] T. Dinh, Y. Kim, T. Gu, and A. V. Vasilakos, "An adaptive low-power listening protocol for wireless sensor networks in noisy environments," IEEE systems journal, vol. 12, no. 3, pp. 2162-2173, 2017.

[13] R. C. Carrano, D. Passos, L. C. Magalhaes, and C. V. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 181-194, 2013.

[14] V. L. Quintero, C. Estevez, M. E. Orchard, and A. Pérez, "Improvements of energy-efficient techniques in WSNs: a MAC-protocol approach," IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1188-1208, 2018.

[15] H. Mostafaei, A. Montieri, V. Persico, and A. Pescapé, "A sleep scheduling approach based on learning automata for WSN partialcoverage," Journal of Network and Computer Applications, vol. 80, pp. 67-78, 2017.

[16] A. Kumar, M. Zhao, K.-J. Wong, Y. L. Guan, and P. H. J. Chong, "A comprehensive study of iot and wsn mac protocols: Research issues, challenges and opportunities," IEEE Access, vol. 6, pp. 76228-76262, 2018.

[17] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for IoT devices," IEEE Sensors Journal, vol. 15, no. 10, pp. 5777-5789, 2015.

[18] A. Kozłowski and J. Sosnowski, "Energy efficiency trade-off between duty-cycling and wake-up radio techniques in IoT networks," Wireless Personal Communications, vol. 107, no. 4, pp. 1951-1971, 2019.

[19] V. Agarwal, R. A. DeCarlo, and L. H. Tsoukalas, "Modeling energy consumption and lifetime of a wireless sensor node operating on a contention-based MAC protocol," IEEE Sensors Journal, vol. 17, no. 16, pp. 5153-5168, 2017.

[20] F. A. Aoudia, M. Gautier, M. Magno, O. Berder, and L. Benini, "A generic framework for modeling MAC protocols in wireless sensor networks," IEEE/ACM Transactions on Networking, vol. 25, no. 3, pp. 1489-1500, 2016.

[21] M. Alam, O. Berder, D. Menard, T. Anger, and O. Sentieys, "A hybrid model for accurate energy analysis of WSN nodes," EURASIP Journal on Embedded Systems, vol. 2011, pp. 1-16, 2011.

[22] T. AlSkaif, B. Bellalta, M. G. Zapata, and J. M. B. Ordinas, "Energy efficiency of MAC protocols in low data rate wireless multimedia sensor networks: A comparative study," Ad Hoc Networks, vol. 56, pp. 141-157, 2017.

[23] Z. Fan, S. Bai, S. Wang, and T. He, "Delay-bounded transmission power control for low-duty-cycle sensor networks," IEEE Transactions on Wireless Communications, vol. 14, no. 6, pp. 3157-3170, 2015.

[24] M. Abo-Zahhad, M. Farrag, and A. Ali, "Modeling and minimization of energy consumption in wireless sensor networks," in 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), IEEE, pp. 697-700, 2015.

[25] Z. Chen, A. Liu, Z. Li, Y.J. Choi, and H. Sekiya, "Energy-efficient broadcasting scheme for smart industrial wireless sensor networks," Mobile Information Systems, vol. 2017, 2017.

[26] H. U. Yildiz, B. Tavli, and H. Yanikomeroglu, "Transmission power control for link-level handshaking in wireless sensor networks," IEEE Sensors Journal, vol. 16, no. 2, pp. 561-576, 2015.

[27] C. Nakas, D. Kandris, and G. Visvardis, "Energy efficient routing in wireless sensor networks: a comprehensive survey," Algorithms, vol. 13, no. 3, p. 72, 2020.

[28] M. Shafiq, H. Ashraf, A. Ullah, and S. Tahira, "Systematic Literature Review on Energy Efficient Routing Schemes in WSN–A Survey," Mobile Networks and Applications, pp. 1-14, 2020.

[29] S. Bing and Z. Yujing, "Energy efficiency in multi-sink linear sensor network with adjustable transmission range," in 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), IEEE, pp. 462-466, 2016.

[30] D. P. Dallas and L. W. Hanlen, "Optimal transmission range and node degree for multi-hop routing in wireless sensor networks," in Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, ACM, pp. 167-174, 2009

[31] M. Haenggi, "Twelve reasons not to route over many short hops," in IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. vol. 5: IEEE, pp. 3130-3134, 2004.

[32] A. Bouani, Y. B. Maissa, R. Saadane, A. Hammouch, and A. Tamtaoui, " A Comprehensive Survey of Medium Access Control Protocols for Wireless Body Area Networks. " Wireless Communications and Mobile Computing, 2021.

[33] A. Kumar, M. Zhao, K. J. Wong, Y. L. Guan, and P. H. J. Chong, "A comprehensive study of iot and wsn mac protocols: Research issues, challenges and opportunities. " IEEE Access, 6, pp. 76228-76262, 2018.

[34] A. El-Hoiydi, and J. D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. " In Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No. 04TH8769) Vol. 1, pp. 244-251, 2004

[35] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM, pp. 95-107, 2004.

[36] A. Bachir, D. Barthel, M. Heusse, and A. Duda, "Micro-frame preamble mac for multihop wireless sensor networks," in Communications, 2006. ICC'06. IEEE International Conference on, vol. 7: IEEE, pp. 3365-3370, 2006.

[37] A.-L. Kampen, K. Ovsthus, L. Landmark, and O. Kure, "Energy Reduction in Wireless Sensor Networks by Switching Nodes to Sleep During Packet Forwarding," in The Sixth International Conference on Sensor Technologies and Applications, SENSORCOMM. International Academy, Research and Industry Association (IARIA), pp. 189-195, 2012.

[38] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in Proceedings of the IEEE National Telecommunications Conference, Vol. 4, pp. 1-4, 1978.

[39] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, pp. 1-13., 2003.

[40] K. Øvsthus, E. Nilsen, A.-L. Kampen, and Ø. Kure, "Modelling the Optimal Link Length in Wireless Sensor Networks for Two Different Media Access Protocols,", Sensors & Transdusers Volume 185. Issue 2, pp. 21-28, 2015.

[41] Chipcon Products from Texas Instruments. http://www.ti.com/lit/ds/symlink/cc1000.pdf (accessed November, 2021).

[42] A.-L. Kampen, K. Ovsthus, and Ø. Kure, "An analysis of the need for dedicated recovery methods and their applicability in wireless sensor networks running the routing protocol for low-power and lossy networks," in The 8th International Conference on Sensor Technologies and Applications, SENSORCOMM. International Academy, Research and Industry Association (IARIA), pp. 121-129, 2014.

[43] "OMNeT++ Discrete Event Simulator." https://omnetpp.org/ (accessed November , 2021).

[44] Y Liu, A Liu, N Zhang, X Liu, M Ma, and Y Hu, "DDC: Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks," Journal of Network and Computer Applications, vol. 131, pp. 16-27, 2019.

Vehicle-to-Everything Business Models for 5G Slicing-supported Systems

Eugen Borcoci, Marius Constantin Vochin, Serban Georgica Obreja
University POLITEHNICA of Bucharest - UPB
Bucharest, Romania
Emails: eugen.borcoci@elcom.pub.ro, marius.vochin@upb.ro, serban@radio.pub.ro

Abstract — **Vehicle-to-Everything (V2X) communications and Internet of Vehicles involve complex multi-actor systems that cooperate in order to support vehicles capabilities to exchange data with other entities (vehicles, infrastructure, grid, pedestrians, etc.). The V2X services mainly aim to improve the transport, safety and comfort on the roads and also to suport autonomous driving. The 5G technology can provide a powerful support for V2X, in multi-tenant, multi-domain, multi-operator and end-to-end contexts. Particularly, using the 5G slicing capabilities, one can construct virtual parallel networks (slices) dedicated to different applications and services (verticals). Consequently, 5G dedicated slices can also be built to meet the V2X special requirements. The complexity of the V2X systems (involving many actors) and the multitude of visions led to proposal of many variants of V2X ecosystems and business models comprising several cooperating actors. Defining the business models (BMs -seen mainly from technical point of view) is important; they essentially provide input information for system requirements identification and architecture definition; for V2X systems this is still an open research topic. This paper is an extension of a previous work presented at IARIA ICNS 2020 Conference. It is focused on so-called operational business model, containing only the actors that are active during the system exploitation and maintenance and not on general business models including economic and financial aspects. The paper analyzes and compare several relevant operational BMs for 5G slicing and discuss how they can be adapted for rich V2X environment. The stakeholder roles and interactions are discussed. Heterogeneity among different BMs is analyzed. Finally, a BM for a V2X system proposed by the authors in a novel 5G-oriented research project is introduced.**

Keywords—Vehicle-to-Everthing; 2X; 5G slicing; Business models; Stakeholders; Management and Orchestration; Software Defined Networking; Network Function Virtualization; Service management.

I. Introduction

The Vehicle-to-Everything (V2X) communications, applications and services include many use cases in single or multi-tenancy, multi-operator and multi-domain contexts. Consequently, different sets of service requirements exist, e.g., from enhanced real-time navigation systems on board, to a self-automated car, or a video streaming played on the in-vehicle infotainment system. This paper is an extension of a previous paper [1] published in the proceedings of the IARIA ICNS 2020 Conference.

The basic vehicular communications have covered essentially vehicle-to-vehicle (V2V) and vehicle-to-road/infrastructure (V2R/V2I) communications. Recently, extended models and services are included in the V2X umbrella, like vehicle-to-pedestrian (V2P) - direct communication, vehicle-to-vulnerable road user (VRU), vehicle-to-network (V2N) - including cellular networks and Internet, vehicle to sensors (V2S), vehicle-to-power grid (V2G) and vehicle-to-home (V2H).

V2X allows vehicles to directly communicate with each other, to roadside infrastructure, and to other road users for the benefit of better road safety, traffic efficiency, smart mobility, environmental sustainability, and driver convenience. V2X contributes to fully autonomous driving development through its unique non-line-of-sight sensing capability that allows vehicles to detect potential hazards, traffic, and road conditions from longer distances. Typical use cases and services/applications for V2X comprise active road safety applications (including autonomous driving); warnings, notifications, assistance; traffic efficiency and management applications; infotainment applications.

Internet of Vehicles (IoV) is an extension of the V2X, aiming to create a global network of vehicles – enabled by various *Wireless Access Technologies* (WAT) [2][3]. It involves the Internet and includes heterogeneous access networks. IoV can be seen as a special use case of *Internet of Things* (IoT); however, IoV contains intelligent "terminals", such as vehicles (maybe some of them - autonomous). IoV extends the traditional basic functions like vehicles driving and safety to novel target domains, such as enhanced traffic management, automobile production, repair and vehicle insurance, road infrastructure construction and repair, logistics and transportation, etc. The complexity of the V2X/IoV claims for a strong support infrastructure. The 5G slicing technology is considered to be an appropriate candidate.

The 5G mobile network technologies offer powerful features, in terms of capacity, speed, flexibility and services, to answer the increasing demand and challenges addressed to communication systems and Internet [4]-[6]. 5G can provide specific types of services to simultaneously satisfy various customer/tenant demands in a multi-x fashion (the notation –x stands for: tenant, domain, operator or provider).

The 5G network slicing concept (based on virtualization and softwarization) enables programmability and modularity for network resources provisioning, adapted to different vertical service requirements (in terms of bandwidth, latency, mobility, security, quality of services, etc.) [7]-[10].

Note that still today there universally agreed a single semantic on “network slice”. In this text, it was adopted the following definition (see [8] and 3GPP documents) for a network slice (NSL): a set of infrastructures (network, cloud, data center (DC)), components/network functions, infrastructure resources (i.e. connectivity, compute, and storage manageable resources) and service functions that have attributes specifically designed to meet the needs of an industry vertical or a service. A network slice is a managed group of subsets of resources, network functions/network virtual functions at the data, control, management/orchestration, and service planes at any given time. In other words, a NSL is a managed logical group of subsets of resources, organized as a virtual dedicated network, isolated from each slices (with respect to performance and security), but sharing the same infrastructure.

The NSLs are implemented (by provisioning or on demand) as network slice instances (NSLIs), where the functionalities are implemented by Physical/Virtual network functions (PNFs/VNFs). The functions are chained in graphs, in order to compose services dedicated to different sets of users. The slices are programmable and have the ability to expose their capabilities to the users. The 3rd Generation Partnership Project (3GPP) [6] has defined three fundamental categories of 5G slice scenarios: *Massive machine type communication (mMTC); Ultra reliability low latency communication (URLLC); Enhanced mobile broadband (eMBB).* It will be shown later that while these models are general, for V2X applications, special adduitional requirements should be identified and met.

The actual run-time execution entities are instantiated slices (NSLIs), whose life cycles are controlled by the management and control entities belonging to the *Management, Orchestration and Control Plane (MO&C)*. The *Network Function Virtualization* (NFV)[11]-[14] and *Software Defined Networks* (SDN) technologies can cooperate [15] to manage, orchestrate and control the 5G sliced environment, in a flexible and programmable way.

The 5G slicing is a strong technological candidate capable to fulfill the requirements of V2X systems. Several studies and projects deal with development of V2X systems based on 5G sliced infrastructure; some examples are [16]-[20]. The dedicated 5G slices can provide the required capabilities for multiple tenants, while working over a 5G shared infrastructure. *However, it is recognized [16][17], that the heterogeneous and complex features of V2X services neither allow the straightforward mapping of them onto basic reference slice types – like eMBB, URLLC and mMTC services, nor the mapping into a single V2X slice. Additional customization is necessary in order to create V2X dedicated slices.*

The V2X/IoV systems are highly complex, involving several technical and organizational entities that cooperate in a business *ecosystem.* Generally, a business ecosystem is a network of organizations/stakeholders, such as suppliers, distributors, customers, competitors, government agencies, etc., involved in the delivery of a specific product or service through both competition and cooperation. The entities/stakeholders/actors interact with each other, in order to achieve together the goals of the system. In this work, the Business Model (BM) is seen as a part of a more general ecosystem, defining the set of stakeholders and their interactions. While the general ecosystems and BMs could involve a large range of organizations (including e.g., the regulating and standardization ones), this paper will be focused to the so-called Operational BM (OBM), that contain only those actors that are active and interact during the real-life system exploitation.

In general, V2X ecosystem new actors are involved, besides traditional Internet and network/service providers or operators. These new actors could be road authorities, municipalities, regulators and vehicle manufacturers *Original Equipment Manufacturers* (OEM).

The development of the 5G complex sliced systems needs to initially define the BMs, that essentially determine the roles and responsibilities of the entities and then the system requirements and architecture. This need is equally true for V2X systems and today it is still an open research topic. *Concerning V2X BMs, it is recognized (see 5G PPP Automotive Working Group, Business Feasibility Study for 5G V2X Deployment [24]) that there is still some lack of insights into the required rollout conditions, roles of different stakeholders, investments, business models and expected profit from Connected and Automated Mobility (CAM) services.* On the other side, the general BMs for 5G sliced networks should be adapted and refined in order to well serve the V2X system’s needs.

Considering the above reasons, this work attempts to analyze some relevant BMs for 5G slicing and discuss how they can be adapted for V2X environment. The objective is to identify the major points of similarity of different BMs for 5G slicing, then 5G-V2X approaches and to study their possible mapping. Therefore, the paper contributes to an overview and comparison of different solutions. Finally, a BM for a V2X system in a 5G-oriented research project is defined.

The paper structure is described below. Section II identifies the typical stakeholder roles in 5G slicing, given that such definitions determine essentially the overall system architecture. Section III refines the general BMs to be adapted to 5G V2X communications and services. Section IV performs an analysis of some factors that lead to different and heterogeneous V2X-5G business models. Section V discusses specific aspects of the V2X 5G slicing solutions. BMs Section VI introduces an example, of defining the BM for a V2X system in a novel research project SOLID-B5G. Section VII summarizes conclusions and future work.

II. Business Model and Stakeholder Roles in 5G Slicing

The objective of this section is to present a few relevant BMs proposed for 5G sliced systems and to identify the main roles of actors, in order to prepare their customization for V2X case in the next section. The layered architecture of a 5G sliced system strongly depends on the stakeholder roles defined by the BM. Different BMs (more specifically – OBMs) have been proposed, aiming to support multi-tenant, multi-domain end-to-end (E2E) and multi-operator capabilities in various contexts. Several examples are summarized below. A more flexible approach is a role-orientation BM, that identifies primarily the functional roles in the system and then mapping of such roles to actors can be defined. The reason of this approach is that in practice a single actor may play one or several roles.

A. Example 1

A basic model (see A. Galis, [8]) defines four main roles:

Infrastructure Provider (InP): it owns and manages the physical infrastructure (network/cloud/data center). It could lease its infrastructure (as it is) to a slice provider; however, in a more complex approach it can construct slices at its own initiative (the BM is flexible) and then leases the infrastructure in network slicing fashion.

Network Slice Provider (NSLP): it can be typically a telecommunication service provider (owner or tenant of the infrastructures from which network slices are constructed). The NSLP main role is to construct, on demand, multi-tenant, multi-domain slices, on top of infrastructures offered by one or several InPs.

Slice Tenant (SLT): it is a generic user of a specific slice, including network/cloud/data centers, that can host customized services. A SLT can request from a *Network Slice Provider* (NSLP) to create a new slice instance dedicated to support some SLT specific services. The SLT can lease virtual resources from one or more NSLPs in the form of a virtual network, where the tenant can realize, manage and then provide *Network Services* (NS) to its individual end users. A network service is composed of several *Network Functions (NFs);* it is defined in terms of the individual NFs and the mechanism (a chinning graph) used to interconnect them. A single tenant may define and run one or several slices in its domain.

End User (EU): it consumes (part of) the services supplied by a slice tenant, without providing them to other business actors.

Note that the scope of the above model is limited; it is operational only, i.e., it does not detail all external entities of the overall ecosystem, that may have strong impact on the operational model, e.g., Standards Developing Organizations (SDOs), policy makers, etc.

An important feature of the above BM is its recursive capability (see Ordonez et al., [9]); a tenant can at its turn, to offer parts of its sliced resources to other tenants, and so on.

B. Example 2

A recent document of the 5G-PPP Architecture Working Group [5] (Figure 1) describes a general BM that covers not only slicing solutions but also non-sliced systems:

At infrastructure level the model contains:

Virtualization Infrastructure SP (VISP): it offers virtualized infrastructure services and designs, builds, and operates virtualization infrastructure(s) (i.e., networking and computing resources). Sometimes, a VISP offers access to a variety of resources by aggregating multiple technology domains and making them accessible through a single *Application Programming Interface* (API).

Data Center SP (DCSP): it designs, builds, operates, and offers data center services. A DCSP differs from a VISP by offering "raw" resources (i.e., host servers) in rather centralized locations and simple services for consumption of these raw resources.

On top of the above, the model has:

Network Operator (NOP): it orchestrates resources, potentially offered by multiple *virtualized infrastructure providers* (VISP). The NOP uses aggregated virtualized infrastructure services to design, build, and operate network level services that are offered to SPs.

Service Provider (SP): it has a generic role**,** comprising three possible sub-roles, depending on the service offered to the SC:

- *Communication SP* offers traditional telecom services;
- *Digital SP* offers digital services (e.g., enhanced mobile broadband and IoT to various verticals);
- *Network Slice as a Service (NSLaaS) Provider* offers a network slice and their services.

The SPs have to design, build and operate high-level services, on top of aggregated network services. The first two roles (Communication SP and Digital SP) do not suppose mandatory slicing solutions.

Service Customer (SC): it uses services offered by a Service Provider (SP). The vertical industries are considered as typical examples of SCs.

The hierarchy of this model (in the top-down sense of a layered architecture) is: SC, SP, NOP, VISP, DCSP. Note that in practice, a single organization can play one or more roles of the above list. This model does not explicitly define a "tenant" role. It can be assumed that SC can play this role or even SPs may have such a role.

In Figure 1 that three auxiliary actors (i.e., Network Services Aggregator, Infrastructure Aggregator, and Data Center aggregator) may exist and they play essentially aggregation and orchestration roles. However, their functions could be embedded in the main actors if the design approach selects this option. The 5G-PPP BM does not elaborate the roles inside an Operation Support Provider (OSP) and hardware/software suppliers.

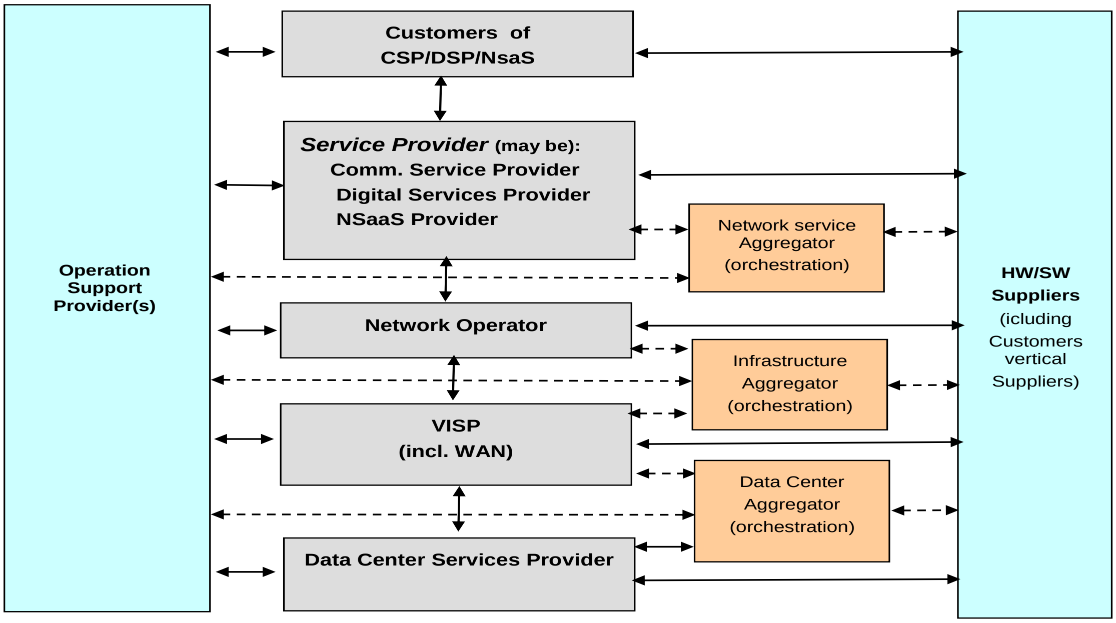

Customers of CSP/DSP/NsaS
Service Provider (may be):
Comm. Service Provider
Digital Services Provider
NSaaS Provider
Network service Aggregator (orchestration)
Operation Support Provider(s)
Network Operator
Infrastructure Aggregator (orchestration)
VISP (incl. WAN)
Data Center Aggregator (orchestration)
Data Center Services Provider
HW/SW Suppliers (icluding Customers vertical Suppliers)

Figure 1. Stakeholder roles in the 5G business model: A 5G-PPP vision (adapted from [5])

C. *Example 3*

The 5G-MoNArch European project [21] proposes an ecosystem model for 5G slicing. The *Mobile network operators (MNOs)* will change from a vertically integrated model, where they own the spectrum, antenna and core network sites and equipment, to a layered model where each layer might be managed or implemented by a different stakeholder. A stakeholder is defined in [21] as an individual, entity or organization that affects how the overall system operates. The MoNArch stakeholder roles [20] are:

Infrastructure Provider (InP): it owns and manages the network infrastructure (antennas, base stations, remote radio heads, data centers, etc.), and offers it to the MSP, in the form of *Infrastructure-as-a-Service* (IaaS).

Mobile Service Provider (MSP): is the main entity that provides mobile internet connectivity and telecommunication services to its users. To this aim, the MSP constructs *network slices* and their function chains to compose services. Examples of slices can be eMBB or mMTC. The MSP set of tasks are: design, building offering and operation of its services.

*Tenant***:** it purchases and utilizes a network slice and its associated services offered by a *Mobile Service Provider (MSP)*. Tenant examples are: *Mobile Virtual Network Operator (*MVNO), enterprise or any entity that requires telecommunications services for its business operations.

End User: it is the ultimate entity that uses the services provides by a Tenant or the MSP.

In practice a larger organizational entity could exist, i.e., *Mobile Network Operator (MNO)* that operates and owns the mobile network, *combining the roles of MSP and InP*.

The Monarch model further refines the roles of some entities that can exist, as distinct actors:

Samples of business models – comparison and mapping			
Basic Model [7]	***5G-PPP [4]***	***MoNArch project [20]***	
End User (EU)	Service Customer (SC)	End User Tenant	
Slice Tenant (SLT)	Service Provider (SP) (can offer slices and may have also a tenant role))	Mobile Service Provider (MSP) - can belong to MNO	
Network SliceProvider (NSLP)	Network Operator (NOP) (offers aggregated services)	Virtualization Infrastructure Service Provider (VISP) – can belong to MNO	VNF supplier (it can be a separate entity)
Infrastructure Provider (InP) Hardware supplier		NFV Infrastructure (NFVI) supplier	
	Virtualization Infrastructure SP (VISP)	Infrastructure Provider (InP) Hardware supplier	
	Data Center SP (DCSP)		

TABLE I. BUSINESS MODELS FOR 5G SLICING

Virtualization Infrastructure Service Provider (VISP) may exist, as an intermediate actor between InP and MSP. It designs, builds and operates a virtualization infrastructure on top of the InP services, and offers its infrastructure service to the MSP. At a lower logical level, an *NFV Infrastructure (NFVI) supplier* may exist, to provide a NFV infrastructure to its customers, i.e., to the VISP and/or directly to the MSP.

A *VNF supplier* may also exist to offer virtualized software (SW) components to the MSP.

The last but not least is the *Hardware (HW) supplier* that offers hardware to the InPs (server, antenna, cables, etc.).

D. Example 4

The document 3GPP TS 28.530 [22] defines a 5G business model. It is shown that 5G opens the door to new BM roles for 3rd parties, allowing them more control of system capabilities. The roles related to 5G networks (Figure 2) and network slicing management include:

Communication Service Customer (CSC): Uses communication services.

Communication Service Provider (CSP): Provides communication services. Designs, builds and operates its communication services. The CSP provided communication service can be built with or without network slice.

Network Operator (NOP): Provides network services. Designs, builds and operates its networks to offer such services.

Network Equipment Provider (NEP): Supplies network equipment to network. For sake of simplicity, VNF Supplier is considered here as a type of Network Equipment Provider. This can be provided also in the form of one or more appropriate VNF(s).

Virtualization Infrastructure Service Provider (VISP): Provides virtualized infrastructure services. Designs, builds and operates its virtualization infrastructure(s). Virtualization Infrastructure Service Providers may also offer their virtualized infrastructure services to other types of customers including to Communication Service Providers directly, i.e. without going through the Network Operator.

Data Centre Service Provider (DCSP): Provides data centre services. Designs, builds and operates its data centres.

- NFVI Supplier: Supplies network function virtualization infrastructure to its customers.
- Hardware Supplier: Supplies hardware.

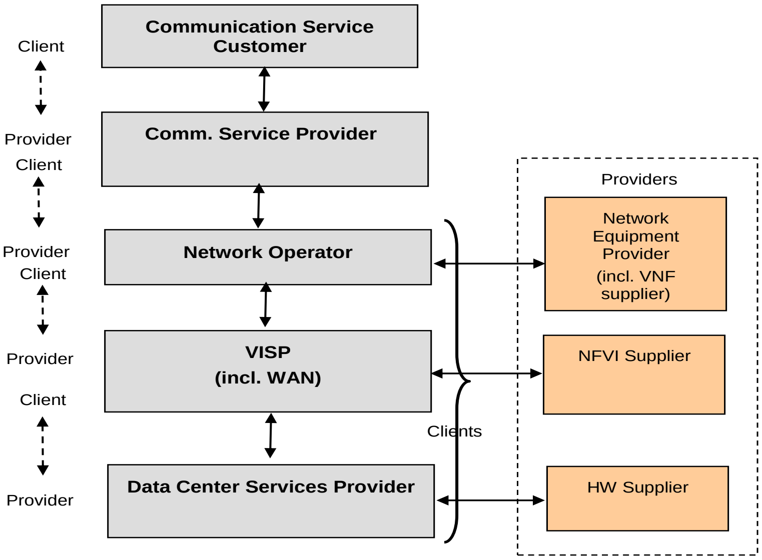


Figure 2. 3GPP- roles related to 5G networks and network slicing management

Note that the above different models cannot be exactly on-to-one mapped, given the different contexts and visions and also the degree of splitting into sub-modules. However, a general equivalence can be observed (see TABLE 1). Here, we consider the basic model the most orthogonal one.

Several recent Public Private Partnership (PPP) Phase I/II collaborative research are running, having as objectives 5G technologies (see several examples in [A. Galis, [8]). Some of them extended the list of role definitions, to allow various possible customer-provider relationships between verticals, operators, and other stakeholders.

III. BUSINESS MODELS FOR 5G V2X

The key technology enablers for 5G V2X communication and services are currently studied and understood in the wireless industry and standardization of 3GPP Release 16 V2X is in its final phase [23]. Apart from traditional vehicular services, it is forecasted that advanced Connected and Automated Mobility services (e.g., high-definition (HD) maps support, highway chauffeur, tele-operated driving, platooning, fully autonomous driving, extended sensors, etc.) will be enabled through next-generation 5G V2X starting with 3GPP Release 16. This section will provide two examples of BMs/ecosystems for 5G V2X.

The 5G PPP Automotive Working Group [24] has defined a general 5G V2X BM, in a set of stakeholder roles, capturing not only operational features but also business relationships. It identified the following key stakeholder categories involved in the deployment of 5G V2X technologies: *5G industry* (network operators, network and devices vendors), *Automotive industry*, *Standards Developing Organizations* (SDOs), *Road infrastructure operators*, *Policy makers* and *Users*. The interactions between them are shown in Figure 3.

5G industry includes any general business activity or commercial enterprise developing or using 5G or providing 5G-related services, e.g., *MNOs*, *Telecom vendors*, *Cloud providers*, device providers, software developers, etc.

Automotive Industry (AutoIn): includes car *Original Equipment Manufacturer (OEMs)* (e.g., car manufacturers), component manufacturers, Tier 1 suppliers, CAM service providers, HD map providers and other automotive-specific technology providers (it can also include other services, such as the logistic sectors). This category brings the automotive expertise and services (including mobility services) to customers (business and consumers).

Standard Development Organizations (SDO): 3rd Generation Partnership Project (3GPP), European Telecommunications Standards Institute (ETSI), Internet Engineering Task Force (IETF), Internet Research Task Force (IRTF), Institute of Electrical and Electronics Engineers (IEEE) and 5G-related alliances, such as Next Generation Mobile Networks (NGMN), Industrial Internet Consortium (IIC), 5G Automotive Association (5GAA) and Automotive Edge Computing Consortium (AECC). For safety-related 5G applications (e.g. *Advanced Driver Assistance Systems* - ADAS and autonomous driving), pertinent standards developing organizations, such as International Organization for Standardization (ISO) may be also relevant players.

Road Infrastructure Operators (RIO): national or regional entities (public/private) performing deployment, operation and maintenance of physical road infrastructure. They may also manage road traffic operations, own or operate the toll system, etc.

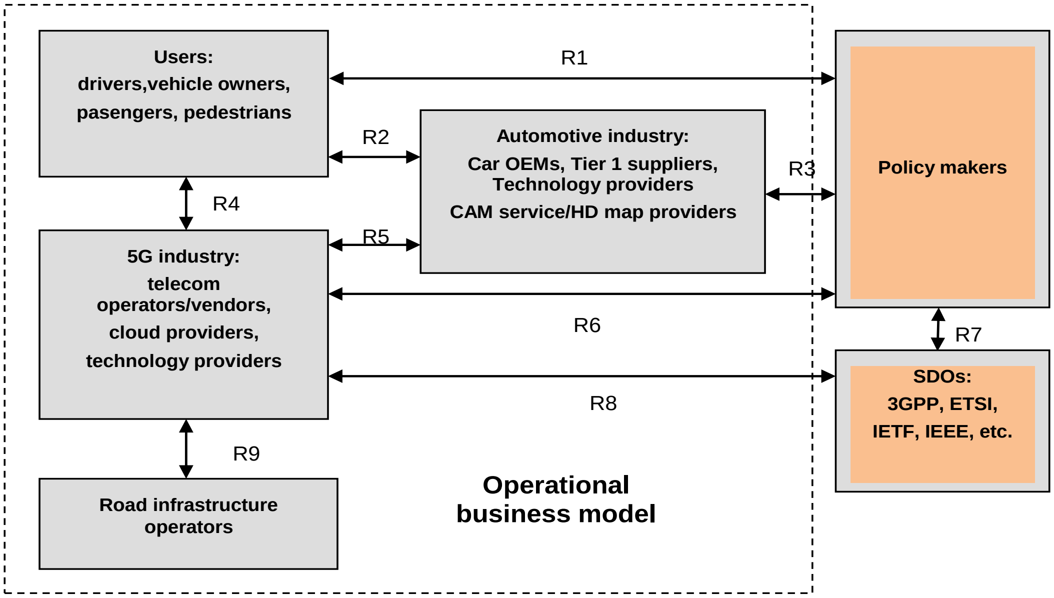


Figure 3. The main stakeholders and relationships in the context of 5G V2X deployment [adapted from 24]

Policy Makers (PM): provide the highest authorities and regulate the relationships within the whole stakeholder ecosystem, including 5G industry, automotive industry, SDOs and users. They are international or national government authorities or organizations defining the legal framework and policies, such as road and transport authorities or telecom regulators. The ITU as well as national spectrum regulators belong to this category.

Users: drivers, vehicle owners, passengers or pedestrian.

The detailed description of the interactions between the stakeholders is given in the 5G PPP Automotive Working Group document [23]. Shortly, the interactions are:

R1 (users- PMs): to provide to the users the authority regulation to be followed (e.g., for environmental, safety and financial aspects).

R2 (users - Automotive Industry): to collect feedback from users in order to define the requirements and features of the new products, functionalities and services.

R3 (PMs– AutoIn): PMs define the regulation framework to be followed by AutoIn, while the latter provides feedback to the PMs to support definitions and improvement of regulations.

R4 (users - 5G Industry): Users buy products and services from the 5G Industry. The latter collects feedback used as inputs to define the network requirements, in terms of Quality of Experience (QoE), and user needs for services and new applications.

R5 (AutoIn - 5G Industry): for inter – cooperation, allowing design a 5G V2X technology to meet the system and component level needs. The AutoIn defines the network requirements for their products and services; the 5G Industry should fulfill the functionality and performance requirements.

R6 (PMs - 5G Industry): PMs define the regulations that the 5G Industry must follow. The latter gives feedback to the PMs to influence the definition of new regulations.

R7 (PMs -SDO): SDOs have to consider regulatory conditions in standards development (e.g., ETSI work is regulated by the of the EU Commission).

R8 (SDO - 5G Industry): The SDOs define the standards to be implemented in the 5G deployments. E.g., for autonomous driving applications ultrareliable low latency are needed, based on safety standards.

R9 (5G Industry - RIO): RIO may participate in the deployment of 5G V2X and provide or facilitate licenses or other infrastructure requirements that are under their responsibility (PMs are also involved here). RIO may define network requirements for the 5G Industry. The 5G Industry shall offer communication services to the RIO based on commercial agreements. However, it is expected that 5G network providers will own and operate most or parts of the network infrastructure.

A subset of actors out of the general model will cooperate within the Operational BM (OBM) i.e.: 5G Industry, Automotive industry, users, and possibly - road infrastructure operators. However, the policy makers, SDOs and road infrastructure operators strongly influence the requirements and also the architecture of the V2X systems, as presented above in interaction description.

Usually, 5G network providers will own and operate most or parts of the network infrastructure. This entity can be split into RAN infrastructure provider (offering the physical infrastructure, e.g., antenna sites and the hardware equipment) and cloud infrastructure provider (it owns and manages local and central data centers providing the virtual resources, such as computing, storage and networking). In practice, the roles of 5G network providers can be taken by the MNOs but is possible that Road Infrastructure Operators deploys or operate (parts of) the 5G V2X network, directly providing the necessary coverage for CAM services to the users.

The network deployment investment can be done by a single actor, called network operator (e.g., a traditional MNO). However, the model in Figure 1 is general, in the sense that potentially any actor (e.g., a road operator) could invest in network deployment.

The project 5GCAR [25]-[27] identifies a BM similar to that developed by 5G PPP Automotive Working Group. In the operational scenarios the following actors can interact: *5G Industry, Automotive industry, Road Infrastructure Operators* and *users.* Those stakeholders may assume different roles identified in the application of the network slicing feature:

Tenant entity: rents and leverages 5G connectivity. Note that Road operator, OEMs or other organization may also have this role.

Mobile Service Provider (MSP): provides to different tenants 5G, dedicated slices for customized services.

The 5G infrastructure providers (5GInP): can be divided into cloud and RAN providers; they offer the elements needed for the MSP to implement the slices.

Non-V2X (supplementary) service provider: can provide passenger targeted services such as enhanced infotainment, mobile office, etc.

The other entities presented in the general BM (Figure 1), i.e., Policy makers, SDOs, influence indirectly the system requirements and specifications of the operational BM. It can be seen that the general basic 5G slicing operational BM (see Example 1) can be mapped approximately one-to-one onto the V2X operational BM.

IV. The Heterogeneity of 5G V2X Business Models

The V2X/IoV systems are complex, involving many actors. Also, the set of applications and services envisaged is very rich. Therefore, an inherent heterogeneity is inherently to exist among different busisness models. This section will summarize the factors leading to heterogeneity in the area of 5G V2X BMs and also affecting the particular architectures. Note that, given the topics complexity, this analysis cannot be exhaustive; some aspects are not touched, or only briefly mentioned. *The analysis made in this section could be useful as a guideline in the activities of identifying an appropriate BM for a target V2X system based on 5G slicing.*

A major factor that leads to many variants of BMs is the multitude of real-life players that can be active (directly or indirectly) in the 5G V2X system assembly and also the variety of V2X applications/services. Actors providing key services for the automotive sector can be split in two categories: service providers of enabling platforms, that manage the data and allow services to be built on top of the data; connectivity providers, that construct and manage connectivity facilities over cellular networks. Inside each category several types of actors can be included.

A non-exhaustive list of actors comprises:

Connectivity Players (MNOs, Transport Services Providers, (TSPs), ICT Solution & Cloud Platform Providers, Intelligent Transportation System (ITS));

Automotive OEMs (Cars, Trucks);

Suppliers （Tier 1 & 2 (System Integrators), Wireless Module Vendors, Chipset Vendors, Software/Solutions, Middleware, Over the Top Services Providers (OTT), Connectivity/ Bluetooth, Databases, etc.);

Application platforms (Software - based, Fleet/ Commercial, Autonomous Driving, Smartphone Platforms); *Business Users* (Public Transport, Company Fleets, Freight, Car Rental, Taxi Fleets, Delivery systems, Emergency Response systems);

Consumers (End user consumers, Families, Small Office Home Office (SoHo);

Application types (Mobility as a Service, Maps & NavigationTelematics / Tracking, Communications Safety & Maintenance, Media & Entertainment, Productivity).

Besides the above, *additional stakeholders* can play specific roles: Insurance, Dealers, Auto Repair, Regulatory Bodies, Local Authorities (Government, Law Enforcement, Smart City, Road Operators), Location-based commerce players, Security infrastructure and services providers.

The forecasts estimate that new actors will enter the auto industry (increase more than 45% by 2030, [28]). Therefore, in order to create a clear and stable ecosystem, the actors' roles/activities and interactions, should be defined. Cooperation is necessary: telecom operators will provide their infrastructure and licensed spectrum; the automotive suppliers will create the chips and sensors compatible with the technology. So, the typical value chain is transformed into a complex ecosystem; actors will share a part of knowledge and resources. The competition will exist and influence the ecosystem structure. From the above reasons, some relationships between possible actors are still uncertain today.

In [27][28], several variants of 5G V2X ecosystems are defined. In each one, a single actor provides the platform, e.g.: MNO, OEM, Automotive Supplier (AS), etc. Interactions between some of actors are established based on Service Level Agreements (SLA).

There are also other lower-level technical factors, determining the heterogeneity of 5G V2X BMs and architectures for slicing solutions. The management, orchestration and control subsystem are directly involved within these aspects. Some examples of such factors are given below.

The *services deployment* is inherently heterogeneous, depending on applications to be supported. An example is the traffic locality property (at the edge of the network/slice or crossing the core part). An orchestrator should be aware of such traffic properties and, if necessary, deploy the corresponding network functions at the mobile edge. The orchestrator needs to have enough topology information of slices in order to be able to install appropriate functions at right places. The type of vehicular applications and services will determine the degree of pushing to the edge some functions.

The classical principle of *vertical separation of services* in *network-related* (i.e., connectivity–oriented) and *application-level services* (e.g., caching, video transcoding, content-oriented, web server, etc.) could be preserved or not. The separation will require, respectively one orchestrator vs. separate network/service orchestrators. One can speak about *segregated* or *integrated* orchestration, respectively. Concerning slicing, one can define some slices offering essentially connectivity services and other dedicated to high-level applications. The clear separation of areas of responsibility over resources could be an advantage for operational stability (e.g., a segregated RAN orchestrator could still maintain basic RAN services even if an application-oriented orchestrator fails). On the other hand, the integrated orchestration could be attractive, in particular for operators, if both kinds of services could be orchestrated in the same fashion (and possibly even with the same orchestration infrastructure). These two options also determine heterogeneity at M&O architectural level.

Segregated orchestrators lead to a more complex overall architecture. One must assign areas of responsibilities from a resource perspective (which orchestrator controls - what resources); one should identify services pertaining to each orchestrator. The split of service is also a problem, i.e., the service description should define the "network" and "application-facing" parts of the service. Aligning the control decisions taken by these two kinds of orchestrators in a consistent way is also not trivial. In an integrated orchestration approach, all these problems disappear. However, an integrated orchestrator might be very complex if required to treat substantially different services (one-size-fits-all orchestration approach is rather not the best choice).

An integrated orchestrator is a more challenging piece of software (from both dependability and performance perspectives) but would result in a simpler overall architecture. Considering the above rationale, we defend the idea that from the slicing point of view, a segregate orchestrator is a better choice. However, in practice, both approaches have been pursued in different projects. Currently, a final verdict commonly agreed on segregated versus integrated orchestration is not yet available. Apparently, there is no need to standardize this option, as long as both of them could be realized inside a meta-architecture. So, for the time being, we can state that M&O heterogeneity, from this point of view, will last.

Another architectural choice is on "*flat" or "hierarchical"* orchestration. In the flat solution, a single instance of a particular orchestrator type is in charge of all assigned resources. In the hierarchical solution, there are multiple orchestrators (a "hierarchical" model is needed, when orchestrators know to talk to each other). Note that a hierarchical orchestrator is *not necessarily* a segregated one, because all hierarchy members could deal with the same type of services.

V. Specific 5G V2X slicing aspects

This section will discuss some specific aspects related to the impact of slicing approach on 5G V2X business models.

While the general business models presented in Section II and III are applicable to 5G V2X systems, the slicing solution, together with the reaches of V2X applications set lead to special functional requirements for stakeholders of the BMs [16]-[19],[30].

Multi-tenancy capabilities are necessary in 5G V2X slicing sytemes. Note that several views on what a tenant is, can be encountered in various studies. However, a common approach is that a Slice Tenant (SLT) is a generic user of a specific slice (or, several slices), while these slices can include network/cloud/data centers, hosting customized services. In this model, a distinct role of Slice Provider (actual builder of slices) is necessary. The previous sections have shown that new stakeholder (with respect to traditional ones) can play roles in the V2X BM, (vehicle manufacturers, road authorities, municipalities, etc.). Multiple tenants and various use cases can exist, requiring the Slice Provider actor capabilities to support multi-tenancy over the same 5G infrastructure. The tenants/actors can be focused on different services, e.g.: a road municipality can be involved in V2V communications for safety services; a content provider may offer video services (streaming, downloading, etc.); a vehicle manufacturer may offer diagnosis services and so on. Due to different requirements of such services, it is naturally that different types of slices will fit. Therefore, the Slice Provider should offer flexible slice templates and then capability to construct and offer appropriate slices to different tenants. More than that, several operators could have been involved in this assembly. Consequently, the adopted BM should accommodate multi-tenant and multi-operator capabilities. The stakeholder containing Operations Support System (OSS)/Business Support System (BSS) will track the fulfillment of the specific requirements, specified in dedicated Service Level Agreements (SLA).

The set of offered slices by the Slice Provider should be flexible and rich i.e., to contain different types of slices (more than only basic ones (eMBB, URLLC, mMTC). The reason of this need is that in the V2X domain, many services can be encountered having different requirements (safety, vehicular urban traffic optimizations, autonomous driving, vehicular infotainment, vehicle remote diagnostics and management, etc.). Different slices/sub-slices are appropriate for different services. The Slice Provider should be able to fulfill this and more than that, a vehicle could need simultaneously several types of slices. Consequently, the entity in the BM that controls the access of the tenants/users to slices should allow attachment of a user to one or several slices/sub-slices, simultaneously.

The moving vehicular traffic density can be very high in urban area. The BM entity that has access and mobility functions (AMF) should be able to instantiate several AMF instances, on need, in order to avoid AMF congestion in the dedicated slices. This instantiation should be scalable conforming to the traffic density variation. If separate BM actors ran the access part and the core part of the network (e.g., a Road Infrastructure Operator and Mobile Network Operator) then the location of the AMF functions should be carefully assigned in optimal way, as to minimize the latency of signaling and improve the mobility processing. At the same time, other more modest characteristics requiring slices (e.g., for V2P services) should be possible to be constructed and to act in parallel (isolated) with the most demanding ones.

Multi-tenant, multi-domain, multi-operator context of the planned 5G V2X system will influence the BM, making necessary to split the responsibilities among actors, for both categories: high level services and connectivity ones. *Multi-domain scenarios* create new problems [29] (e.g., in the case of a multi-domain “federated” slice). In a flat model, each orchestrator of a domain is actually multi-orchestration capable, i.e., it can discuss/negotiate with other domains’ orchestrators. In the hierarchical model, a higher-level orchestrator could exist, in charge of harmonizing multiple organizations cooperation. However, several issues are not fully solved today: which entity would run that multi-domain orchestrator, trust issues, preservation of domains independency, assuring the fairness, etc.

Relationship of the M&O system and the 5GT V2X slicing system is another factor of BM architectural variability, depending on what the definition of a slice is. A largely agreed solution is to have a general orchestrator (configured offline), capable to trigger the construction of a new slice and then to install in this new slice its own dedicated orchestrator (before the slice run-time). To still assure the basic services outside any slice (e.g., packet forwarding at network level) one can construct an additional special orchestrator installed outside of all slices. Currently, many combinations have been proposed, and there is still no consensus on such matters. The convergence of solutions will be determined probably by the adoption of a more unique definition of a slice – that could assure better inter-operability.

The V2X environment supposes frequently high-speed mobility of vehicles, that imposes additional challenges on V2X dedicated slices and consequently for the run-time configuration of a slice. There is a need of path resource allocation algorithms in the backhaul/fronthaul segments in order to allocate/re-allocate/migrate resources between different geographical areas with desirable low disruption effects. Such actions have higher impact (w.r.t single domain case) in the case of slices covering several domains and/or that are managed by different BM cooperating entities. These should harmonize the real-time resource control in order to adapt a given slice to changed traffic conditions. If the Multi-access edge computing (MEC) is used at the edges of slices, this could require live migration and service redirection (and MEC servers) that increase the complexity of mobility management and handover procedures [30].

VI. Example of Business Model for a 5G V2X sliced system

This section will present a Business Model selected for a 5G system oriented among towards V2X services and maritime applications, “A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services" - SOLID-B5G [31] (https://solid-b5g.upb.ro/).

The objectives of the SOLID-B5G project are the following:

- O1: To develop ultra-low latency massive MIMO based concurrent transmission mechanisms for data collection in massive IoT;
- O2: To develop advanced B5G slicing methods, algorithms, and protocols with a focus on Orchestration Management and Control (OMC) of resources and dedicated services for IoV/V2X and maritime services;
- O3: To develop decentralized decision-making mechanisms by introducing data processing capacity and intelligence to the edge (based on Multi-access (Mobile) Edge Computing (MEC) and machine learning (ML)-to-the-edge);
- O4: To implement a proof-of-concept *standalone* B5G testbed to demonstrate the orchestration of RAN and CN based on 5G network slicing and MEC procedures. Two main use cases, i.e., IoV/V2X and satellite based maritime low-latency services will be considered in the project.

An important part of the SOLID-B5G project is the study and contributions to the development of V2X/IoV systems while applying slicing solutions in 5G technology.

The basic three types of 3GPP-defined slices, i.e., Enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultra-reliable low-latency communications (URLLC) will be considered. In particular URLLC and eMBB type slices are appropriate to be dedicated for SOLID-B5G objectives. Given the specific needs of V2X/IoV and also IoT for maritime systems the project will also apply decentralized solutions like Multi-access Edge Computing (MEC), former - mobile edge computing) in a sliced environment.

Two types of dedicated slices URLLC and non-URLLC will be considered, for two types of applications:

Safety and traffic efficiency slice- for V2V and vehicle-to-pedestrian (V2P). Such slices will transport and process *event-driven* and *periodic* messages containing position and kinematics parameters and support applications, such as: forward collision warning; cooperative adaptive cruise control that allows a group of vehicles in proximity to share the same path (a.k.a. platooning); vulnerable road users (VRU) safety to alert a vehicle of the presence of a VRU;

Autonomous driving slice – having more powerful characteristics than those for safety applications, the reason being higher speed, more complex environment- including geographical and road-related aspects, cooperative needs, etc. Such slices can offer ultra-low-latency V2V communications via RAT connection mode and additional RAN/CN functions.

Considering the models presented in previous sections and the objectives of the SOLID-B5G system, the following set of BM roles are appropriate:

- *End User:* the ultimate entity that uses the services provides by a Tenant or the MSP.
- *Tenant:* purchases and utilizes a *network slice* and its associated services offered by a *Mobile Service Provider (MSP).* Tenant examples are *Mobile Virtual Network Operator (MVNO)*, enterprise or any entity that requires telecommunications services for its business operations.
- *Mobile Service Provider (MSP):* is the main entity that provides mobile internet connectivity and telecommunication services to its users. To this aim, the *MSP constructs network slices,* and their function chains to compose services. Examples of dedicated slices can be uRLLC, eMBB or mMTC. The MSP set of tasks are design, building offering and operation of its services.
- *Infrastructure Provider (InP):* owns and manages the network infrastructure (antennas, base stations, remote radio heads, data centers, etc.), and offers it to the MSP, in the form of Infrastructure-as-a-Service (IaaS).

In practice a single entity like Mobile Network Operator (MNO) may own and manage/operates the mobile network, i.e., *combining the roles of MSP and InP*.

As already presented for the general models, additional optional stakeholder roles could exist:

- *Virtualization Infrastructure Service Provider (VISP)* - intermediate actor between InP and MSP. It designs, builds and operates a virtualization infrastructure on top of the InP services, and offers its infrastructure service to the MSP.
- NFV Infrastructure (NFVI) supplier - to provide a NFV infrastructure to its customers, i.e., to the VISP and/or directly to the MSP.

Further refinements of the above model should be defined depending on use cases selected and their associated scenarios.

VII. Conclusions and future work

This paper analyzed several business models/ecosystems for 5G slicing and then those for V2X and discussed how the 5G BM can be adapted for V2X environment. It has been shown that a large variety of proposals exist in various studies, standards and projects, given the multitude of V2X use cases and the rich set of business actors that could be potentially involved. Some major factors determining the heterogeneity of the BMs proposals have been identified in Section IV. The analysis made in Section IV could be useful as a guideline in the activities of identifying an appropriate BM for a target V2X system based on 5G slicing.

Finally, an example of business model has been proposed to serve the needs of a novel research project oriented to V2X.IoV and IoT for maritime applications.

Considering the above analysis, we propose the steps to be followed to start a 5G V2X system development in slicing approach.

First, the V2X set of high level of services (seen from the end user perspectives) to be implemented should be defined among the rich possible ones (see Section IV).

The identification of the set of involved actors and a first assignment of their roles (especially from business/services point of view) is a major step. Here, some actors would provide only indirect actions (Policy Makers, SDOs, local regulators, etc.). Other actors will participate at operational phases (MNOs, OEMs, Service providers - e.g., OTT, Infrastructure providers, etc.) at run-time.

The multi-domain, multi-tenant, multi-operator characteristics of the 5G V2X system should be selected. Definition of interactions between the actors will complete the high-level description of the 5G V2X BM/ecosystem.

The following steps will refine the BM and go to the requirement identification and architectural definition. The main connectivity and processing/storage technologies should be identified. The regulations, standards, etc., to be enforced have to be identified; they will define but also limit the system capabilities and scope. System requirements identification will follow, considering requirements coming from all actors involved in BM. The 5G V2X slicing solution (for RAN, core and transport part of the network) should be selected. Here, the refinement of the BM is possible (see Table 1). Then, the system architecture (general and layered - functional) has to be defined, allowing further technical refinement of the system design.

Future work can go further to consider more deeply the multi-x aspects, related to the business models and impact of the BM upon the system management orchestration and control for 5G V2X dedicated slices.

Acknowledgment

The research leading to these results has received funding from the NO Grants 2014-2021, under Project contract no. 42/2021, RO-NO-2019-0499 - "A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services" - SOLID-B5G.

References

[1] E. Borcoci, M. Vochin, S. Obreja "On Business Models for Vehicle-to-Everything Systems Based on 5G Slicing", ICNS 2020, The 16th International Conference on Networking and Services, http://www.thinkmind.org/index.php?view=article&articleid=icns_2020_1_60_10043.

[2] M. K. Priyan and G. Usha Devi, "A survey on internet of vehicles: applications, technologies, challenges and opportunities", Int. J. Advanced Intelligence Paradigms, Vol. 12, Nos. 1/2, 2019.

[3] C. Renato Storck and F. Duarte-Figueiredo, "A 5G V2X Ecosystem Providing Internet of Vehicles", Sensors 2019, 19,550, doi: 10.3390/s19030550, www.mdpi.com/journal/sensors, [retrieved January 2020].

[4] N. Panwar, S. Sharma, A. K. Singh 'A Survey on 5G: The Next Generation of Mobile Communication' Elsevier Physical Communication, 4 Nov 2015, http://arxiv.org/pdf/1511.01643v1.pdf.

[5] 5G-PPP Architecture Working Group, "View on 5G Architecture", Version 3.0, June, 2019, https://5g-ppp.eu/wp-content/uploads/2019/07/5G-PPP-5G-Architecture-White-Paper_v3.0_PublicConsultation.pdf, [retrieved June, 2019].

[6] 3GPP TS 23.501 V15.2.0 (2018-06), System Architecture for the 5G System; Stage 2, (Release 15).

[7] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges", IEEE Communications Magazine, May 2017, pp. 94-100.

[8] A. Galis, "Network Slicing- A holistic architectural approach, orchestration and management with applicability in mobile and fixed networks and clouds", http://discovery.ucl.ac.uk/10051374/, [retrieved July 2019].

[9] J. Ordonez-Lucena et al., "Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges", IEEE Communications Magazine, 2017, pp. 80-87, Citation information: DOI 10.1109/MCOM.2017.1600935.

[10] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing & Softwarization: A Survey on Principles, Enabling Technologies & Solutions", IEEE Communications Surveys & Tutorials, March 2018, pp. 2429-2453.

[11] ETSI GS NFV 002, "NFV Architectural Framework", V1.2.1, December 2014.

[12] ETSI GS NFV-IFA 009, "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options", Technical Report, V1.1.1, July 2016.

[13] ETSI GR NFV-IFA 028, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on architecture options to support multiple administrative domains", Technical Report, V3.1.1, January 2018.

[14] ETSI GR NFV-EVE 012, Release 3 "NFV Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework", Technical Report, V3.1.1, December 2017.

[15] ONF TR-526, "Applying SDN Architecture to 5G Slicing", April 2016, https://www.opennetworking.org/wp-content/uploads/2014/10/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf, [retrieved December, 2019].

[16] A. Molinaro and C. Campolo, "5G for V2X Communications", https://www.5gitaly.eu/2018/wp-content/uploads/2019/01/5G-Italy-White-eBook-5G-for-V2X-Communications.pdf, [retrieved December, 2019].

[17] S. A. Ali Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications", IEEE Communications Magazine, January 2018, pp.111-117.

[18] K. Katsaros and M. Dianati, "A Conceptual 5G Vehicular Networking Architecture ", October 2017,

https://www.researchgate.net/publication/309149571, DOI: 10.1007/978-3-319-34208-5_22, [retrieved December 2019].

[19] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G Network Slicing for Vehicle-to-Everything Services", IEEE Wireless Communications, Volume: 24 Issue: 6, DOI: 10.1109/MWC.2017.160040, [retrieved December 2019].

[20] Friedhelm Ramme, ITS, Transport & Automotive, Ericsson 5G: "From Concepts to Reality“ Technology Roadmaps https://5gaa.org/wp-content/uploads/2019/02/Final-Presentation-MWC19-Friedhelm-Ramme-ERICSSON.pdf, [retrieved January, 2020].

[21] H2020-ICT-2016-2, Monarch Project, 5G Mobile Network Architecture for diverse services, use cases and applications in 5G and beyond, Deliverable D2.2, "Initial overall architecture and concepts for enabling innovations", https://5g-monarch.eu/deliverables/ 2018, [retrieved June 2019].

[22] 3GPP TS 28.530 V0.6.0 (2018-04), Telecommunication management; Management of 5G networks and network slicing; Concepts, use cases and requirements (Release 15)

[23] https://www.3gpp.org/release-16.

[24] 5G PPP Automotive Working Group, “Business Feasibility Study for 5G V2X Deployment”, https://bscw.5g-ppp.eu/pub/bscw.cgi/d293672/5G%20PPP%20Automotive%20WG_White%20Paper_Feb2019.pdf, [retrieved, January 2020].

[25] 5GCAR White Paper: Executive Summary, Version: v1.0, 2019-12-10, https://5gcar.eu/wp-content/uploads/2019/12/5GCAR-Executive-Summary-White-Paper.pdf, [retrieved, January 2020].

[26] 5GCAR, Fifth Generation Communication Automotive Research and innovation Deliverable D1.2 5GCAR Mid-Project Report, v1.0 2018-05-31, https://5gcar.eu/wp-content/uploads/2018/08/5GCAR_D1.2_v1.0.pdf, [retrieved, January 2020].

[27] 5GCAR, Fifth Generation Communication Automotive Research and innovation Deliverable D2.2 “Intermediate Report on V2X Business Models and Spectrum”, v2.0, 2019-02-28,https://5gcar.eu/wp-content/uploads/2018/08/5GCAR_D2.2_v1.0.pdf, [retrieved, January, 2020].

[28] B. Martínez de Aragón, J. Alonso-Zarateand, and A. Laya, "How connectivity is transforming the automotive ecosystem". Internet Technology Letters. 2018; 1:e14. https://doi.org/10.1001/itl2.14 [retrieved, January 2020].

[29] K. Katsalis, N. Nikaein, and A. Edmonds, "Multi-Domain Orchestration for NFV: Challenges and Research Directions", 2016 15th Int'l Conf. on Ubiquitous Computing and Communications and International Symposium on Cyberspace and Security (IUCC-CSS), pp. 189–195, DOI: 10.1109/IUCC-CSS.2016.034, https://ieeexplore.ieee.org/document/7828601, [retrieved July 2019].

[30] C. Campolo, A. Molinaro, A. Iera, R. Fontes, C. Rothenberg, "Towards 5G Network Slicing for the V2X Ecosystem", 2018, https://www.researchgate.net/publication/327635348 [retrieved December 2020].

[31] Research Project, “A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services" - SOLID-B5G [31] (https://solid-b5g.upb.ro/).

The Cyber Meta-reality, Biome, and Microbiome

Joshua A. Sipper
Air Force Cyber College
Air University
Maxwell AFB, AL, United States
Email: joshua.sipper.1@us.af.mil

***Abstract*— The cyber realm as an entity continues to evolve and grow. As the Earth and indeed human beings share their chemical/biological physicality with a host of enabling flora and fauna (Earth) and bacteria, fungi, protozoa, and even viruses (humans), the cyber meta-reality (a reality of realities) is growing into a type of non-physical, yet tangible sphere where stripping away or adding to it could have far-reaching ramifications not yet understood. The human microbiome has most recently been estimated to outnumber human cells by several orders of magnitude [2]. A cyber microbiome has already begun to take shape, characterized by viruses, archived data, Dark Web outgrowths, and other symbiotic code and applications that will ostensibly grow rapidly as Artificial Intelligence (AI) and Machine Learning (ML) begin to create additional code and data in the future. While the cyber microbiome may not be, in some cases, considered a direct part of the created domain we experience, it certainly must not be stripped away, eradicating the good along with the bad. The cyber microbiome is similar to its planetary and human corollaries in that it contains various undetectable components that serve to support its function in difficult to discern ways. For instance, the Dark Web is much like the unseen portion of the iceberg under the surface. This indicates another way in which the cyber microbiome is so similar to its antecedents; the cyber microbiome is larger than visible cyberspace by many orders of magnitude. This paper examines the concept of the cyber meta-reality with an in-depth analysis of the cyber microbiome and attempts to correlate the symbiotic relationships of these two entities through an examination of the cyberspace most people encounter and the vast underlying cyberspace of which most are oblivious.**

Keywords- cyber; microbiome; meta-reality; archive; code; malware.

I. Introduction

The cyber meta-reality [1] we currently experience includes several realities being experienced simultaneously. From gaming realities, to research realities, to family realities, and even into the darker realities like pornography and cheating Websites, the cyber meta-reality (see figure 1) continues to grow and deepen, offering escapes, adventures, and resources unimaginable just a couple of decades ago. However, what many have not realized is alongside this ever growing cyber meta-reality entity exists another, symbiont; a cyber microbiome (see figure 2) often unseen, yet integral to the shaping and growth of the surface layer of the cyber meta-reality most inhabit. This underlying cyber space is similar in many ways to the Earth borne and human related microbiomes that flourish and support both systems respectively. The concept of the microbiome has its roots in the earliest theories of Macarthur in 1955 [3] and was later taken up by Savage in 1977 who stated, "In terms of numerical bacterial cells likely outnumber human cells by at least an order of magnitude." [4] This estimate was later extended by many orders of magnitude following further study. This concept usually shocks most people simply because the sheer enormity of microorganisms they suddenly realize inhabit and make up their bodies. We are under the impression that we are made up primarily of "human" cells, but this has been proven not only to be a false assumption, but the direct opposite with most of the body we share being made up of the microbiome. As scientists and medical professionals recognize, the human microbiome has fundamentally changed how they do research and practice medicine. Thus, this concept, while new to the formation and constitution of the cyber meta-reality is nevertheless one that must be considered, especially in light of the areas underlying and paralleling the cyberspace most see. The Dark/Deep Web alone accounts for a vast and overwhelming section of what can be termed the cyber microbiome, followed by malware, living archives, and code that populate the symbiotic Hadean realm we have yet to fully realize. In this paper, we will investigate these abysmal realms of the cyber meta-reality as we cross the digital River Styx.

Figure 1. Cyber Meta-reality.

II. The Cyber Biome

In any world where life exists, it does not do so independently. Our planet has plants, animals, an electromagnetic field, appropriate proximity to a star, a near circular orbit, a natural satellite that provides necessary tidal force, and a plethora of other characteristics that make life possible and sustainable. In the cyber meta-reality, life as we know it in the physical reality is mirrored with similar creation and sustainment of this meta-reality dependent upon power sources, florae, faunae, electromagnetic energy, communications gateways, and untold other creations that are to come. The florae, as in the physical reality, are represented by numerous, ubiquitous, nourishing components that allow life within the cyber meta-reality to exist and flourish. Cyber florae can be characterized in many different ways; however they tend to be mostly fixed, vast networks and systems that allow the transit of information, cyber faunae, and human beings throughout the cyber meta-reality. Cyber florae act, in essence as a natural, integral, and ramifying infrastructure that creates, upholds, and allows the continuation of life in the cyber meta-reality. These networks and systems are complemented by millions of other cyber florae in the form of databases, websites, archives, clouds, infrastructure, and the entirety of the electromagnetic spectrum. Through this massive, rhizome of interconnected cyber florae, the fullness of the cyber meta-reality is nurtured and persistent. As in the physical world, however, without cyber faunae, cyber florae could not exist and vice versa. The cyber faunae that sustain and are sustained by cyber florae, are much like the animals and people we encounter in our physical world every day. As it rests right now, human beings are the only intelligent agents existent anywhere that we have observed. However, within the cyber meta-reality, possible emergent intelligence is coming closer to reality consistently as developments in AI, ML, AA, and quantum computing continue to grow.

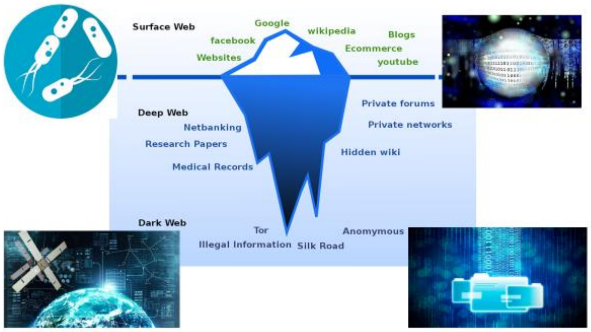


Figure 2. Cyber Microbiome.

There are many other forms of life emerging in the cyber meta-reality as well that are more like the domesticated, wild, and captive animals we live among and depend on for food, work, entertainment, and companionship. Applications, browsers, active code, autonomous programs, and algorithms are a few examples of cyber faunae we will examine in addition to humans within the cyber meta-reality. In order to get a good foundation for how these cyber plants and animals inter-relate and coexist, we first need to understand the various constructs within the cyber biome. On Earth, there are untold numbers of systems in which the world's biome is expressed. Just consider the vast oceans with all their various lifeforms, lakes, rivers, streams, grasslands, mountains, deserts, wetlands, prairies, and the millions upon millions of complex spheres of influence contained in each. While this merely begins to scratch the surface of the many habitats on Earth, it does nothing to delve into the extant and growing number of habitats and spaces within the cyber meta-reality. The cyber biome, like the Earth's biome, is comprised of habitats and systems that reach far and wide, affecting and being affecting by other systems and lifeforms. This enormously complex system is only now being examined at a level that allows those who live within it to understand and navigate it with relative awareness of the cyber meta-reality itself. However, as understanding unfolds, the complexity and size of the cyber meta-reality grows exponentially, expanding rapidly as the physical universe. There has arisen a need for cyber ecologists, anthropologists, zoologists, and other students of the systems and environments contained in the cyber meta-reality, so that the cyber biome can be defined and organized for deeper and more organized study. As it stands, the cyber biome is still barely understood with little to no research accomplished on anything but its resemblance to a biological system. However, through further definition of the complex environments, habitats, and creatures that make up the cyber biome, humans can see a much clearer picture of the cyber biome and the entire cyber meta-reality. We can really only look at a small amount of the entirety of the cyber biome here, but the most important and necessary systems and concepts to analyze for an understanding of the cyber biome are communication, diversity, hierarchy, networks, and organization. Communication takes place in the Earth's biome in the form of cycles, root systems, currents, and many other interrelated phenomena that allow the entire biome to operate and flourish. The cyber biome works in similar fashion with its numerous communication systems, networks, protocols, and features that allow these various components to work together allowing information and energy to flow. Diversity encompasses many vital areas of the Earth's biome, allowing for the secure continuation of genetic information which allows organisms to exist and procreate. Without biological, chemical, habitat, and other types of diversity throughout the cyber biome, the information that flows through and forms the cyber meta-reality will break down, allowing cyber-DNA to become lost and systems that perpetuate cyber life to be poisoned and destroyed. Much of the diversity and communication of the cyber biome are dependent upon the hierarchies that exist inside the cyber meta-reality and biome. As in the Earth's biome, hierarchies such as the food chain, ecosystems, phylogenies, and ontogeny are all similar attributes of the cyber biome. Our planet and its biome are comprised of huge networks of matter, physical effects, and energy flowing through and between continents, oceans, and geology as a whole. The cyber biome contains networks that grow and evolve in much the same way with root systems within cyber forest and fungal network communities, electromagnetic energies, and systems that adapt and morph as energy and information flow through them. These networks must be organized in order to ensure the other systems and networks within the cyber biome work and flourish properly. As on Earth, without balance and proper stewardship, human activity is wont to run systems and environs into a stupor. But, with proper organization and cultivation, the cyber biome can grow and remain healthy.

III. The Deep Dark Web

The Deep Web or Dark Web as some call it is known to relatively few but connects with and influences many people without them even realizing it. While not an illegal space itself, the "Dark Web" is known as a lawless realm leveraged by the dark cyber powers to conduct "Dark Market" activities of a less than savory nature. In this Underworld of viruses, Worms, Trojans, malware, ransomware, and a plethora of other malicious code for hire, hackers find community and items for sharing or purchase that may be added to their bag of tricks. Botnets can be hired for a mere few hundred dollars or less, passwords are sold on the cheap like crack cocaine, Bitcoin transactions traverse Virtual Private Networks (VPN), further obfuscating the already uber-secure blockchain mechanisms in place. And yet, this black cyber Gehenna is more spacious in data and global reach than any government or international enclave in existence. "The terms Deep Web, Deep Net, Invisible Web, or Dark Web refer to the content on the World Wide Web that is not indexed by standard search engines. The deepest layers of the Deep Web, a segment known as the Dark Web, contain content that has been intentionally concealed including illegal and anti-social information" [4]. As use of this shadowy enclave continues to grow, more and more capabilities are being created and leveraged. The growth of the Dark Web is so rapid that keeping up with its evolution is virtually impossible. Some predict "There will be more activity in darknets, more checking and vetting of participants, more use of cryptocurrencies, greater anonymity capabilities in malware, and more attention to encryption and protecting communications and transactions. Twitter is becoming a channel of choice; Tor and VPN services are finding increased use" [6]. Indeed this deepening of black and gray market transactions has been observed occurring at an alarming rate. "A recent study found that 57% of the Dark Web is occupied by illegal content like pornography, illicit finances, drug hubs, weapons trafficking, counterfeit currency, terrorist communication, and much more" [5]. This trend is likely to continue as more and more miners of the Dark Web find lucrative enterprises. This influx of additional Dark Web tourists and residents will no doubt expand this ever growing dark segment of the cyber microbiome. "People are becoming more technically sophisticated; younger generations are using technology on a daily basis in school, learning digital technology at a very early age. In the words of one expert, 'hacking has become little league: everyone starts out early, and spends a lot of time doing it'" [6]. Although the Dark Web has become a cyber reality associated with illegal activity and anti-social behavior, it did not begin this way. "To access material in the Dark Web, individuals use special software such as TOR (The Onion Router) or I2P (Invisible Internet Project). TOR was initially created by the U.S. Naval Research Laboratory as a tool for anonymously communicating online" [5]. The fact that TOR, now associated with and widely used by criminals, hackers, and terrorist groups to name a few, was originally created by a legitimate U.S. government entity is telling. The Dark Web and the bridges to and from it have evolved; morphed from one kind of thing into something entirely different and it continues to grow and change. This fact along with the imminent applications of technologies like AI, ML, quantum computing, and nanotech indicate a potential future growth of the Dark Web of astronomical proportions, indicating its continued significance as a fundamental part of the cyber microbiome.

IV. Malware with a Life of Its Own

What happens when a computer virus, Trojan, Worm, or other type of malware accomplishes its mission? Where does it go? Does it simply self-annihilate or does it live on as a part of the cyber microbiome underlying the cyber meta-reality? Of course, the answers to these questions usually depend on how the malware was designed and what its purpose is. However, these answers are becoming more subjective as the cyber meta-reality continues to change due to hackers' constantly fluctuating modus operandi and the impending implementation of AI and ML enabled malware that could lead to self-replication and even evolutionary code not yet fathomed. These kinds of effects can be seen in what are deemed "poisonous systems" where malware has infected and changed the most integral portions of said system. "Poisoned systems are distinct from systems infected with computer viruses, which allow malicious code to transfer to other systems when it meets various conditions through a self-replicating mechanism" [7]. The continued spread and replication in this case goes beyond such targeted malware as Stuxnet which had a specific purpose and target and was set to end once that objective had been met. Often, malware is also coded in such a fashion as to be easy to catch and defend against through patching and malware signature implementations. "[I]f an OCO capability is used against a target, several considerations must be considered. First, the capability cannot be used elsewhere globally as an anti-virus company will likely see it and create a signature for it" [7]. However, with the growth of malware, itself a persistent portion of the cyber microbiome, patching and signature implementation are becoming more and more difficult. "As one industry analyst observed: 'IT analyst forecasts are unable to keep pace with the dramatic rise in cybercrime, the ransomware epidemic, the refocusing of malware from PCs and laptops to smartphones and mobile devices, the deployment of billions of under-protected Internet of Things (IoT) devices, the legions of hackers-for- hire, and the more sophisticated cyber-attacks launching at businesses, governments, educational institutions, and consumers globally'" [9]. Other advances have already been incorporated into malware by hackers who understand the subtleties of signature-based algorithms and patching trends. The capabilities associated with advanced malware are concerning, but also fascinating in light of their ability to change, grow, and reproduce, thus adding another layer of complexity to their presence within the cyber meta-reality as a portion of the cyber microbiome. "[N]ew security products incorporating ML and AI are easily added to his or her testing cycle. The malware is validated against the test matrix, ensuring no tested product detects it" [10]. With the continued advancement of AI and ML functions, malware as part of the cyber microbiome will continue to expand and morph in myriad ways.

V. Living Archives

As the cyber meta-reality and cyber microbiome continue to grow and deepen, questions concerning the nature of existence within these spheres arise. Ever since the discovery of the double-helix meta-molecule deoxyribonucleic acid (DNA) was discovered, the fundamental building blocks of life have been understood in terms of an extremely complex computer code that uses information to construct all organic, carbon-based life as we know it. This fact has profound application when considering the information environments we traverse daily, perhaps never comprehending the amount of information or how it is growing, changing, being shaped, and shaping us as information constructed entities. This confluence of carbon and silicon-based information hybridization can be seen in the experimentation taking place in the life of Finnish artist, engineer, and composer, Erkki Kurenniemi. A Belgian art and media group named Constant "foregrounds the digital life of an archive by practicing what it calls an 'active archive'. Unlike most online archive initiatives Constant places emphasis on the generative and active part of making an archive come alive" [11]. This living archive concept is based on recordings captured and archived that catalog and preserve Kurenniemi's life. This odd, but intriguing venture was undertaken by Kurenniemi in an effort to potentially resurrect himself at some point in the future by using this "file/life." "More precisely, Kurenniemi set out to create an archive of his own life for a possible artificial life resurrection in the future" [11]. Archives are seen by many as extensions of who we are individually and culturally. The information that makes up our personal and collective existence has now found itself displayed in many cases for the entire world to see. With social media and the internet in general, our lives are increasingly becoming an active archive. "[T]echnological advances in data collection and data science … allow data to be transferred, stored, organized, and analyzed in an efficient and timely manner" [12]. In some cases, this data is being looked at for the purpose of customization of legal norms for individuals, but they are also being increasingly indexed as a means of understanding the most intricate habits and perspectives of singular humans. "[N]umerous firms are investing in collecting, organizing, and analyzing data or in creating products, services, and technologies that rely on such data, giving rise to data capitalism" [12]. Obviously, by capturing so much data, some information is subject to being misplaced, forgotten, or even put away for specific purposes be they good or evil. "Digital cloud storage simplifies our lives by releasing us from dependency on hardware we must manage ourselves. But we can get lost in the clouds. And a provider may decide, unbeknownst to us, not to archive our data beyond a few years. We change computers, we close accounts, time passes, and we lose entire portions of our memory" [13]. This loss of information identity leads back to the consideration of the living archive and what it would mean in the case of Kurenniemi if his data were lost, misplaced, or forgotten. "As Eugene Thacker has concluded in an overview of these tendencies, our notions of life underwent important changes during these post-war decades: 'the advances in genetic engineering and artificial life have, in different ways, deconstructed the idea that life is exclusively natural or biological.' This tendency in the sciences is crucial for understanding Kurenniemi's idea that an archive of files or information about a life as it is lived can actually also be or become a life form" [12]. This has further ramifications in the cyber meta-reality and cyber microbiome as all of the data and metadata associated with such archives lives in multiple places and within an indeterminate timeline. One need not look far before seeing the outgrowths of the living archive within the frameworks of the cyber meta-reality and microbiome.

VI. Codenstein

Code is information. As mentioned above, information is the basic, fundamental chemical foundation of life in our physical reality. Within the *cyber meta-reality* and *microbiome* the same case can be made for an information-based, ever growing, reproducing, and self-perpetuating existence. For centuries, the definition of what makes something alive has been debated. Of course, life and intelligent life are different discussions, however from a basic point of view, life is defined as something that consumes, grows, and can reproduce without destroying itself in the process. Based on this simple definition, things like fire and viruses existing within our physical space are not considered to be alive since neither can reproduce without effectively destroying themselves. However, as an entity, the *cyber meta-reality* and *microbiome,* much like the human microbiome (see figure 3), both appear to consume energy in the form of electrons, grow in information proliferation and extensibility, and reproduce in the formation of additional information enclaves, forms, archives, and code. It is the latter of these progeny that seems to be the most prolific and analogous to the life, reproduction, and evolution we encounter as carbon-based life forms.

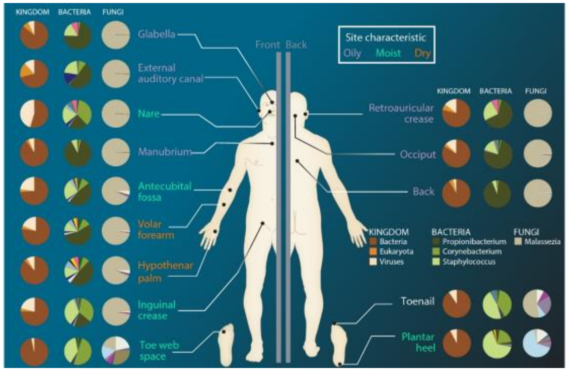


Figure 3. Human Microbiome.

Code is an information-rich, complex series of instructions that is used to create, control, and connect other information together in such a way to allow or make certain events occur. Just as animals sometimes do unexpected things *a la* the horse with the mind of its own, sometimes code is observed moving outside of its expected parameters. “It can be said that a computer can be both ‘reliable’ (but not infallible) and yet perform functions without the authority or knowledge of the owner or software writer. This may be when the code happens to execute in a way, because of a strange or unforeseen conjunction of inputs, which neither the owner nor the writer had imagined” [14]. This attribute of the unexpected nature of an information-constructed entity is tantamount to any other form of life behaving unpredictably. “Code can be used to create programs that provide insight into the universe, the human body, and efficiencies in transportation, finance, communications, and an almost infinite number of fields. The aggregate benefits of code are immense” [15]. The immensity of code capability in the hands of a skilled code creator is one thing, but one must also consider the trends and precedents that have been set in programs that create code autonomously. “Advanced development environments generate code automatically, although writing software to perform complex functions that works well in all circumstances remains exceedingly difficult and challenging” [12]. This is one step closer to the level of code being considered a type of life, but what about the possibility of intelligent life through code expansion and reproduction? “It should be observed that the increasing use of machine-learning systems complicates this issue, because the software code is instructed to make further decisions when running, which increases the complexity. In addition, the veridicality of machine-learning systems like neural nets cannot be easily understood or verified” [14]. While the autonomous decision-making capabilities of code generators using AI and ML are expanding rapidly, the question of ethical and moral agency may still be far off, if not possible. However, what is evident is code as an instrumental entity within the greater *cyber meta-reality* and *microbiome* is a category-shattering being on the verge of becoming something much larger and more complex.

VII. Cyber DNA and Cyber “Junk” DNA

The chemical code, DNA, has been described as the “building blocks of life” and a type of very sophisticated computer code used to relay instructions at the cellular level. These descriptions are fitting for the way code and instructions work in the cyber microbiome as fundamental components of the information, patterns, and code that is used to construct and quicken the CMR. The discovery of the double-helix molecule of DNA in 1953 by Francis Crick and James Watson opened an entirely new field of scientific study; one that has allowed human beings to peer deeply into their makeup, even mapping the entirety of the genome. But, what has also been revealed through this discovery is how much we imprint on the things we create, essentially copying elements of our own design into the designs of machines, code, and the new lifeforms arising in the cyber microbiome and CMR. As time has passed and DNA has been studied more deeply, geneticists discovered massive constructions of DNA that appeared to have no function at all. This seemingly vestigial code was dubbed “junk” DNA and summarily dismissed at first. However, as more study was undertaken, what first appeared to be “junk” began to show significance. The same thing has been observed in cyber DNA, especially within the cyber microbiome where some are all too eager to dismiss code that has deeper meaning and life than first apparent. As we look more deeply into cyber DNA and cyber “junk” DNA, it is important to understand that these cyber molecules account for not just living code, but all the other structures, ecosystems, biomes, and microbiomes we encounter and live within as we inhabit the CMR.

DNA, sometimes characterized as a chemical alphabet, is made up of the four nucleobases cytosine [C], guanine [G], adenine [A] and thymine [T]. In simple terms, these four “letters” fit together to construct DNA in very specific sequences that relay instructions to living things within their cells, telling them how to do everything from make more energy with mitochondria to making sure they take out the garbage from adenosine triphosphate (ATP) so that the cell does not die. These instructions are extremely specific and must be transmitted and decoded exactly in order for carbon life as we know it to exist. Code at the basic levels of the cyber microbiome has similar properties, especially from a standpoint of instructions that provide the proper energy and information flow to the cyber microbes living there. These instructions are encoded through the arrangement of specialized languages that are decoded in the process of executing the code so that the information can be released in and through the microbiome, biome, and ecosystem, spreading into the CMR. If you were able to observe the way DNA is decoded in a cell, the result and spread of information and energy throughout microbes, plants, animals – and by extension, the Earth and universe – would look very much the same way as how code is propagated in and through the CMR. It is in this diverse and vast cycle that energy and information are exchanged and fed back to lifeforms and systems. The similarities between code in the CMR and DNA code are striking, holding more than just similarities of structure, but also in creation. As a result of this parallel, programmers have been experimenting with DNA computing, finding potential solutions for code interaction and efficiency not easily defined through binary code. The code structure of DNA specifically is of great interest to how code works in the cyber microbiome and CMR, with its dependence on organization and specifically administered instructions. As humans, we see design patterns in everything, including DNA and programmed code. This is very helpful for discovering new and better ways to use the complex and elegant DNA design to support code design in the CMR. DNA itself is more than just a code, it is an enormous databank full of all the information upon which carbon life is based. Most recently, two scientists were awarded the Nobel Prize based on their work in DNA editing or CRISPR technology which can be used to actually repair DNA damage [16]. This is an astonishing capability as it relates not only to DNA in the physical reality, but DNA computing as well as advances in cyber DNA manipulation, repair, and redesign. Through understanding how this central code supports all life, we may be able to build similar form and function into supporting and designing future cyber life.

Merely having the cyber DNA at one's disposal is not useful if it is not stored properly. And there is increasingly so much data in the CMR that storage and understanding where data is and how to access it are becoming increasingly problematic. According to the International Data Corporation, the amount of data in the CMR in 2020 is expected to reach 44 zettabytes—the equivalent of 44 trillion gigabytes—and grow to 175 zettabytes by 2025 [17]. With this growing amount of information and energy, the capability to store it all is quickly reaching the limits of traditional storage media and the ability to measure it is almost outside the realm of possibility. As a result, DNA-based databank models have been proposed to help with these massive storage requirements. This is a revolution in data storage as the space requirements for biological DNA are minimal compared to future magnetic solutions with DNA able to store 1 billion Gb of data per cubic millimeter. But, the revolutionary nature of this storage capability does not end there. Through using actual, biological DNA to harness and store cyber data and energy, the realities of physical and cyber space are coalescing into something extraordinary, becoming a sort of cyborg chimaera that combines technology and biology into a new form of life that exists in the CMR and physical reality.

When the human genome was first mapped, geneticists noticed something interesting about some portions of the DNA code they were deciphering. While some areas were obviously useful for form, function, and other important elements of gene operation, there were large areas that did not seem to do anything useful at all. This led to those portions be called "Junk" DNA. Unfortunately, this was a misnomer that led to overlooking some very important functions of this seemingly vestigial material. For example, in the UK during a 12 month period spanning parts of 2011 and 2012, almost half of the babies born in the country were found to have an extra chromosome which usually carries with it deleterious effects as seen in Down's Syndrome, Edward's Syndrome, and Patau's Syndrome [18]. However, it was found that the extra chromosome found in these children was a large X chromosome that contain what many have termed "Junk" DNA that actually protected the babies from having any symptoms of the aforementioned syndromes. This is just one example of many throughout the field of genetics that has seen "Junk" DNA turn out not to be junk at all. In exploring and studying the cyber microbiome, it is important to remember that similar instances occur when we might see cyber DNA that seems meaningless. However, as inhabitants and co-designers of the CMR, we must take great care in what we choose to keep and what we choose to ignore or even delete. Imagine what life on Earth would be like if geneticists could delete all of the DNA that caused horrible diseases. It would be a world greatly relieved of suffering and pain, but also a world in which mistakes, potentially horrific ones, could be made. We actually live in that world now, but in the CMR, we have so much more power than just removing disease. While this is scary, it is also comforting to know that we as people who live in the CMR can work together to ensure we do not lose the valuable cyber DNA as we root out the harmful.

The act of synthesis in any system requires information from which to create something else. In the case of "junk" DNA, synthesis has been seen time and again, leading to important discoveries regarding cellular function and the support of a network of information and energy throughout the genome and the greater, biological universe [19]. Synthesis in cyber DNA is usually seen as dispersed even wider throughout the entire CMR. The main difference here is the locus of control and spread of the synthesis in question. In biological systems, DNA synthesizes and reproduces within individual organisms. While these mutations, evolutions, and adaptations might affect other organisms and the environment they inhabit to some extent, genetic information, as far as we can tell, is not exchanged at every level and certainly not outside the Earth itself. However, in the CMR, when synthesis occurs in one place, the opportunity for this change to affect other parts of the CMR are excellent, not just globally, but universally. This kind of impact, while concerning, is also potentially very powerful in that information can be spread quickly over the entirety of the CMR, allowing for a much more pronounced understanding and experience of its wholeness and our individual and collective place in it.

Ultimately, in the years following the initial proclamation of "junk" DNA in 1994, the ENCODE project finally in 2012 sounded the death knell of "junk" DNA by showing the usefulness of DNA as a whole. This serves as a reminder to all of us in the CMR that cyber DNA is going to change and evolve, but needs to be carefully probed and understood as we work with it and possibly make our own changes to it. This is not only applicable to information and energy as relates to code, but to the actual words we type and say and emote. It applies to information we read and share. Cyber DNA is more than just what we sense, it is built into the core of the energy and information transiting the CMR and affects everything we say and do in it.

VIII. Conclusion

The *cyber microbiome* appears to be far more massive than anyone could have guessed. In this paper, we have only seen a few contributing areas that make up this symbiotic manifestation of the *cyber meta-reality,* but even then, the entity itself is enormous and extremely complex. As the interconnected cyber sphere continues to grow and change, so too will the Dark Web. In this dangerous, information-rich realm, people and machines will continue to create more narrow alleys of malware, code, and data that will potentially take on any number of byzantine existences. Malware that exists now and has been proliferated throughout systems will likely become smarter and more versatile, leaping into a new and more autonomous kind of existence that may grow into any number of malicious or potentially helpful expressions. As living archives continue to develop and evolve, the potential for advancement within this kind of file/life could be far-reaching, especially when factoring in AI and ML. Ultimately, all of these possibilities come down to code; the type, complexity, and growth of which could literally take on a life of its own. Through code that can write more code and learn and advance at rates soon to be enabled by quantum technology, nanotechnology, AI/ML, and any number of nascent capabilities, code will continue to develop through the seminal acts of human beings, potentially taking on a life of its own. All of these outgrowths and tectonic shifts only add to the propensity of the *cyber microbiome* to grow and change into the foreseeable future.

References

[1] J. Sipper, "The Cyber Microbiome and the Cyber Meta-reality," *IARIA Cyber 2020 Conference*, pp. 37-41.

[2] N. Fierer, et. al., "From Animalcules to an Ecosystem: Application of Ecological Concepts to the Human Microbiome," *Annual Review of Ecology, Evolution, and Systematics*, Vol. 43, pp. 137-155, 2012.

[3] R. Macarthur. 1955. "Fluctuations of animal populations, and a measure of community stability," *Ecology,* 36:533- 536, 1955.

[4] D. Savage, "Microbial ecology of the gastrointestinal tract," *Annual Review Microbiology*, 31:107-33, 1977.

[5] G. Weimann, "Terrorist Migration to the Dark Web," *Perspectives on Terrorism* (June 2016), Vol. 10, No. 3, pp. 40-44, 2016.

[6] L. Ablon, M. Libicki, and A. Golay, "Projections and Predictions for the Black Market," RAND Corporation, 2014.

[7] R. Stevens and J, Biller. 2018. "Offensive Digital Countermeasures: Exploring the Implications for Governments," *The Cyber Defense Review* (FALL 2018), Vol. 3, No. 3, pp. 93-114, 2018.

[8] M. Klipstein, "Seeing is Believing," *The Cyber Defense Review* (SPRING 2019), Vol. 4, No. 1, pp. 85-106, 2019.

[9] C. Downes, "Strategic Blind–Spots on Cyber Threats, Vectors and Campaigns," *The Cyber Defense Review* (SPRING 2018), Vol. 3, No. 1, pp. 79-104, 2018.

[10] B. Bort, "There IS No Cyber Defense," *The Cyber Defense Review* (SPRING 2018), Vol. 3, No. 1, pp. 41-46, 2018.

[11] E. Røssaak, *FileLife: Constant, Kurenniemi, and the Question of Living Archives*, Amsterdam University Press, 2017.

[12] N. Elkin-Koren and S. Gal, "The Chilling Effect of Governance-by-Data on Data Markets," *The University of Chicago Law Review*, Vol. 86, No. 2, Symposium: Personalized Law, 2019.

[13] S. Abiteboul. *Memory: The Digital Shoebox*, Peter Wall Institute for Advanced Studies, 2018.

[14] S. Mason, "The Presumption That Computers Are 'Reliable'," *School of Advanced Study*, University of London, Institute of Advanced Legal Studies, 2017.

[15] A. Brantly, "The Violence of Hacking: State Violence and Cyberspace," *The Cyber Defense Review* (Winter 2017), Vol. 2, No. 1, pp. 73-92, 2017.

[16] Charbonneau, P., Natural Complexity: A Modeling Handbook, Princeton University Press, 2017.

[17] Choi, C. "Nature's Databank," ASEE Prism, Vol. 29, No. 6 (February 2020), pp. 22-25.

[18] Carey, N., Junk DNA: A Journey Through the Dark Matter of the Genome, Columbia University Press, 2017.

[19] Bardini , T., Junkware, University of Minnesota Press, 2010.

DISCLAIMERS:

The views expressed are those of the author and do not necessarily reflect the official policy or position of the Air Force, the Department of Defense, or the U.S. Government.

DoD School Policy. DoD gives its personnel in its school environments the widest latitude to express their views. To ensure a climate of academic freedom and to encourage intellectual expression, students and faculty members of an academy, college, university, or DoD school are not required to submit papers or material that are prepared in response to academic requirements and not intended for release outside the academic institution. Information proposed for public release or made available in libraries or databases or on web sites to which the public has access shall be submitted for review.

Indexing Large Amount of Log Data for Predictive Maintenance

Guy Lahlou Djiken
Applied Computer Science Laboratory
Douala University
Douala, Cameroon
Email: ldjiken@fs-univ-douala.cm

Fabrice Mourlin, Charif Mahmoudi
Algorithmic, Complexity and Logic Laboratory
UPEC University
Creteil, France
Email: fabrice.mourlin@u-pec.fr, charif.mahmoudi@lacl.fr

Abstract— **With the advent of big data and cloud computing, systems are producing high volumes of log data. Log analysis software tools allow monitoring, aggregation, indexing proprietary algorithms, and analysis of all applications and infrastructure log data. They are increasingly used for monitoring application availability and planning maintenance operations. Their uses involve an automatic or customized indexing phase, based on indexes or key properties in order to obtain actionable results for preventive maintenance of software and hardware resources. We offer an efficient indexing approach. Our approach reduces the resources required by using specific logging configurations and responds to these properties by using a Big Data cluster and installing a custom indexing engine for software log analysis. To implement it, we have developed several components from the Spark Framework that uses libraries like Spring and Solr Cloud to index the data read and store it for building an AI model. Our results show a reduction in software failures and, therefore, better availability of software services under monitoring. This result leads to rethinking software maintenance and reviewing the sizing of our cluster according to the number of monitored applications correlated with the throughput of each one.**

Keywords-Software log; Indexing settings; Big Data collection; Streaming context; Planning maintenance.

I. INTRODUCTION

One of the most common use cases is predictive behavior analysis. Among our daily tasks, a significant part is dedicated to the management of services or applications deployed on servers. This means monitoring software based on information reported by the software. Therefore, there are technical logs, business logs, and other sources of information such as exchanges between applications. Thus, it is easier to understand the unavailability of services when we have the volume of client requests over time.

Our goal is to monitor the availability of these applications so that our customers or users are satisfied. Among the ways to validate the availability of services on a platform, it is common to observe the content of logs. Logs are files containing specific information and hosted on the application servers to be monitored. These files regularly record events performed either by a server or any process present on the server, such as access to resources, requests being processed, etc. The logs are used to retrieve information on abnormal behavior, alerts, errors and their scheduling, etc. They are full of information, including date of an event, the invoked Web address, the uniform resource location origin, the response code of the page (code 403, code 201, etc.), its payload, etc. They are several types of logs, such as application logs, system log, database logs ad traffic logs (http, ftp, etc). The exploitation of these logs begins with the collection phase (identification - storage), then the analysis and filtering (indexing - aggregation) and finally the actual exploitation (visualization - planning). The analysis of log files is the evaluation of a set of information recorded from one or more events that have occurred in an application environment. This practice is used to analyze user behavior and identify patterns of behavior, or identify and anticipate incidents. These same techniques are applied to ensure compliance of server behavior with the regulations in place, such as government applications.

Analyzing logs is a challenge and requires tedious work for the Software supervision teams because of the volume of data. Other features are crucial, such as the diversity of types of logs, as well as the proprietary formats, elastic architectures, aggregation of time-stamped data, detection of behavior patterns, etc. Using log analysis software that leverages machine-learning algorithms dramatically reduces the workload on supervisory teams who can focus on value-added tasks. This log analysis software allows you to monitor, aggregate, index with proprietary algorithms and analyze all application and infrastructure log data with often-restricted configurations.

As example, we can mention the following tools, which are currently important tools in the construction of a software monitoring chain. In article [1], F. Mourlin et al. build a distributed application for collecting and analyzing software logs. They use a Big Data cluster and a custom index engine for analysis to ensure better quality of service.

Apache Kafka is a message broker with replication. It allows data persistence in case of system failure. Apache Flume enables intelligent message routing and thus implements integration patterns, such as those of Gregor Hohpe [2]: Message Translator or Process Manager. Fluentd is a data collection and consumption tool used by Docker and Elasticsearch users. Logstash is a lightweight, server-side aggregator that is used to collect, integrate and analyze logs. Loggly is based on Docker containers and interacts with containerized application-based applications. Splunk supports information from the previous tools from a central server, while enabling real-time extraction and organization of information from Big Data. Tools such as the ELK

(Elasticsearch, Logstash, and Kibana) and EFK (Elasticsearch, Fluentd, and Kibana) suite of software are becoming standards in the monitoring field [3] due to their adaptability and versatility, but they also have their limitations especially in configuring indexing and analysis.

Log analysis tools provide better visibility into the health and availability of applications using dashboards. This allows software administrators to monitor critical events from a central location. Synthetic situations thus appear where an administrator is able to decide to anticipate a maintenance task in order to ensure continuity of service. Log analysis tools provide better visibility into application health and availability through dashboards. Software administrators can monitor critical events from a central location. Synthetic situations arise where an administrator can decide to anticipate a maintenance task to ensure service continuity. On the other hand, the customization of the tools is often weak or non-existent. For example, the comparison of log analysis techniques is not possible. The same is true for the calculation of indexes on input data, even in the case of semi-structured formats.

Over time, the use of specific indexing techniques has enriched log file analysis strategies. The structure of these input data is always formatted even though the formats vary. In addition, the use of a data schema provides additional typing which highlights the meaning of these lines of information. It is then useful to separate data storage from data indexing. The search for a pattern of behavior is more effective and prevention becomes better and more reactive.

Log analysis solutions incorporate additional data sources. Thus, machine learning and other analytical techniques push the boundaries of new use cases in application performance management, security intelligence, event management and behavior analysis.

The rest of this paper is organized as follows. Section II describes the works close to our domain. Section III provides a precise description of our use case. Section IV addresses the software architecture of our distributed platform. Section V goes into finer details on our streaming approach, which includes an indexing step. Section VI focuses about on our results and the impact on the maintenance task. The acknowledgement and conclusions close the article.

II. Related Works

A. Log data as an observation state

Big Data projects often lead to the deployment of applications on virtual machines distributed on high performance networks. The use of virtual machine VM is a rigid technical choice for developments. Indeed, Big Data frameworks often evolve (bug fixes, software evolutions) and the update of these frameworks in a VM is delicate. The consequences on other software require time-consuming tests. The use of logs associated with these software is a help for the follow-up. However, the aggregation of logs from several VMs is a perilous exercise (different clocks, lack of synchronization between applications).

Indeed, it is difficult to update the services of a virtual machine without a technical experience of the construction of this virtual machine. The HDP machine (Hortonworks Data Platform) is based on a version of Ubuntu on top of which software, such as Apache Spark, Apache Solr, HBase (Hadoop Data Base, etc...) are installed. But the versions of these software are old even for the last version of the virtual machine. Publications illustrate this evolution problem, such as [4] where C. Clarke et al. describe the low-level system maintenance cost. In particular, they detail the principles of separation of hardware and software constraints as well as software update scripting. D. E. Lowell et al. define the life cycle of a virtual machine update. This involves a devirtualization of the virtual machine in order to have a reversible process, make the changes followed by testing. Only then is the virtual machine built again for deployment [5]. In this case, also the validation of the modifications is done by analysis of log files included in the virtual machine. These are easily accessible and it is easy to test that the versions of services are consistent with the expected versions.

The use of logging has been common practice in IT for many years in the field of monitoring. Its use for intervention prediction is more recent, but the interest of this approach has quickly become essential in companies and more particularly in any computer system offering services 24 hours a day. Publications have long been available in order to present the broad spectrum of log analysis techniques [6]. S. Shailesh detailed performance tools for monitoring the information system [7]. It shows a proactive monitoring architecture with anomaly detection and alert systems. Table 1 features some of the popular performance monitoring tools. For example, Apache Kafka allows log aggregation who collects log files of servers and puts them in a central place for processing [8].

TABLE I. Monitoring type and tool

Monitoring Type	Monitoring Tool
File monitoring	Filebeat
System monitoring and infrastructure monitoring	▪ ELK ▪ Grinder ▪ AppDynamics (commercial) ▪ New Relic (commercial)
Statistics dashboard and visualizations	Kibana
Alert and monitoring dashboard	Grafana
Real time event monitoring Visualizations and data queries	Prometheus and Grafana
Search engine	Elasticsearch
Database monitoring	▪ Automatic Workload Repository (AWR) ▪ Fluentd
Log monitoring	▪ Splunk ▪ Fluentd
Message streaming	Apache Kafka
Notification	Alert Manager

Apache Kafka offers better performance, stronger durability guarantees due to replication, and much lower end-to-end latency. Jay Kreps in his articles discusses the place of log in the system architecture [9]. His work shows that distributed logs allow better management of data consistency by sequencing data between nodes and offers the possibility of restoring failed replicas in the event of loss and rebalancing of data between nodes. In article [10], Jay Kreps et al. use the

Apache Kafka tool for data extraction. Comparing with newspaper processing applications, Apache Kafka achieves much higher throughput than conventional email systems. It also provides integrated distributed support and can scale.

B. Log data as data model

In the context of distributed computing systems, Qiang Fu has published some very useful results on behavior anomaly detection from logs [11]. W. Xu's work focuses on the structure of logs and the impact on the analysis strategy [12]. Liman et al. use the log service architecture that provides reliable log delivery to the central storage from distributed heterogeneous mobile robotic nodes [13].

In the more specific context of analysis with prediction, Chinghway Lim's work is based on the use of individual message frequencies to characterize system behavior and the ability to incorporate domain-specific knowledge through user feedback [14]. Jakub Breier follows the same approach based on Apache Hadoop technique to enable processing of large data sets in a parallel way [15]. The work of S. Son et al. is based on Apache Hive, it is a tool of the Hadoop platform for the detection of anomalies in a distributed system.

T. Li and Y. Jiang propose a platform to facilitate the data analytics for system event logs [16]. This work is an end-to-end solution that utilizes advanced data mining techniques to assist log analysts. They apply learning techniques to extract useful information from unstructured raw logs. The parsing technique contains an index management.

Steven Yen published recently a book on the topic of intelligent log analysis using machine and deep learning [17]. He explains how deep learning implementation can improve the result quality when the data volume achieves a limit. He provides a comparison with a K mean model and two distinct implementations. This work shows that a global solution does not exist and some add-ons are crucial. For instance, the use of an indexing process of log messages could lead to a cost reduction at runtime. D. Qingfeng et al. offer a log-based automated anomaly detection approach called logAttention [18]. This work aims to capture contextual and semantic information in the log patterns. The existing log-based anomaly detection approaches are three states who use the method differs: supervised machine learning, unsupervised machine learning and zero-positive machine learning. All these approaches, although used, show drawbacks compared to LSTM modules and adapted vectorization methods. Jasmin et al. discuss the benefit of the exploitation of distributed log by a learning model for the maintenance of distributed systems as well as the detection of anomalies [19]. The interest was focused on the detection of the deviation of the current behaviour of the system compared to that theoretically expected. They show that the combined use of distributed traces and system log data produces better results compared to single-modality anomaly detection methods. They establish the NTP (Next Template Prediction) formalism which allows better learning of model integration and remains reusable by other applications.

In more constrained fields, such as real time, log analysis systems must be able to detect an anomaly in a limited time. Biplob Debnath presents *LogLens* [20] that automates the process of anomaly detection from logs with minimal target system knowledge. *LogLens* presents an extensible index process based on new metrics (term frequency and boost factors). The use of temporal constraint also intervenes in the recognition of behaviour pattern. So, abnormal events are defined as visible in a time window while other events are not. This allows semi-automatic real-time device monitoring. R. Wang et al. propose a solution, called LogProv, which needs to renovate data pipelines or even some of big data software infrastructure to generate structured logs for pipeline events, and then stores data and logs separately [21]. Henry Field et al. describe crowd logging, an approach for distributed search log collection, storage, and mining, with the dual goals of preserving privacy and making the mined information broadly available [22].

C. Log data as string indexer

Aspects remain to be covered, such as the use of cross logging in analysis and log indexing. The reason is the separation of storage and indexing. In the previous works, the storage is generally done by the use of relational databases while the indexing uses rather NoSQL (Not only Structured Query Language) databases where the notion of join cannot be easily implemented. D. Tancharoen et al. present an experience recording system and proposes practical video retrieval techniques based on life Log content and context analysis [23]. Their effective indexing methods include content based talking scene detection and context based key frame extraction based on GPS data. In another publication [24], they use voice annotation also as a practical indexing method. Moreover, they apply body media sensors to record continuous life style and use body media data to index the semantic key frames. In this work, the data is naturally typed, so the GPS data is a string representing a triplet of values. The voice annotations are tags to indicate a role. The type chosen provides a weight for the indexed information.

I. Fronza et al. apply a random indexing to represent sequences of operations where each operation is characterized in terms of its context [25]. They want to predict failures based on log files. Weighted SVMs improve the true positive rate. Chen Ding and Jin Zhou try to improve the performance of the site search by combining a new source of evidence from web server logs. They use server log analysis to extract terms to build the web page index. Then, this log-based index is combined with the text-based and anchor-based index to provide a more complete view on the page content. They show that it improves the effectiveness of the web site search significantly.

III. Use Case Description

A. Historic

When doing software monitoring, the first thing we want to get is a reason for each failure, or even the root cause of the problem. The idea is to automate the creation of an intervention request ticket and to follow this maintenance operation until the update of the concerned service. Behind this concept is the need to track maintenance interventions. Local people in the company perform these operations but subcontractors with the necessary rights are authorized for such operations.

Many software programs exist for this need, such as Free Management of Computer Park (GLPI) [26], and new software monitoring needs are appearing in order to improve this incident management by anticipating maintenance operations. The idea is that to reduce the costs of maintenance task, which generally correspond to service interruptions. Even if service replication strategies make it possible to lessen the effects of a failure, it is preferable to anticipate this problem and to research before the event in order to prevent it. For instance, a very basic monitoring strategy can suppress secondary service crash caused due to unexpected network congestion. Prediction research becomes necessary when trying to anticipate the first service crash. This is how statistical laws appear based on the nature of the applications used.

B. *Log information*

At the heart of log analysis, there is the collection of events, such as the setup of a service, the attempt to connect to the system for example, a configuration request, or variations in CPU or storage, or the trace of an application event (receipt of an order, etc.).

A log entry contains information, such as the date and time of the event, on which network node the event occurred, user identification, contextual information (configuration, security) or even the service at the origin of the event.

TABLE II. LOG CONFIGURATION OF A WEB APPLICATION

Key	Value
logger.level	INFO
logger.handlers	FILE
handler.file	org.jboss.logmanager.handlers.FileHandler
handler.FILE.level	ALL
handler.FILE.formatter	PATTERN
handler.FILE.properties	append,autoFlush,enabled,suffix,fileName
handler.FILE. constructorProperties	fileName,append
handler.FILE.append	true
handler.FILE.autoFlush	true
handler.FILE.enabled	true
handler.FILE.fileName	${jboss.server.log.dir}/app.log
formatter.PATTERN	org.jboss.logmanager.formatters.PatternFormatter
formatter.PATTERN .properties	pattern
formatter.PATTERN .constructorProperties	pattern
formatter.PATTERN .pattern	%d %-5p %c: %m%n

As an example, we choose an application server such as JBoss WildFly. This server allows the deployment of Web applications. It is highly configurable for the log part with a specific API named Solder. It also allows the use of other libraries, the main thing being to be able to aggregate the server log messages with those of the deployed applications. In this context, the most used libraries are log4j, slf4j, both supporting a configuration file format in the form of key/value. An example is given in the table 2 above.

Such a configuration file allows each application to have customized log messages or even separate log files. The reading of table 2 shows 3 categories of properties. First, the general properties concerning the echo mode (FILE) and the level of the messages. A second category of properties defines entirely the echo, that is to say, the path of the files in our case, the name of these, the update mode (append). Finally, a third category of properties specifies the internal format of the files. This format has a direct impact on the data extraction. It therefore seems natural to put forward a common format, but this forces application developers to accept a strong constraint.

It can be interesting to leave the definition of logging profiles to the application server administrator. Logging profiles are independent sets of logging configurations that can be assigned to deployed applications. A logging profile can define handlers, categories and a root logger in the manner of the standard logging subsystem, but cannot recommend the configuration to other profiles or to the main logging subsystem. Logging profiles allow administrators to create logging configurations specific to one or more applications without affecting any other logging configuration. Since each profile is defined in a server configuration, this means that the logging configuration can be changed without requiring the affected applications to be redeployed. We took this approach because we understood that developers do not have the same need for logging as administrators. A developer wants access to his logs for the development of his application while an administrator has a more global request and wants to use the logs of several applications to diagnose a problem with a server or one of its services.

C. *Nominal scenario*

The description of our use case is based on our desire to monitor the activity of our information system. This includes several application servers and data management servers, interconnected by a software bus. It enables intelligent message routing between applications and provides a first level of fault tolerance in the event of a service failure.

Our servers provide log files, but also our applications deployed on the servers. Many formatted files are thus written in different directories. To perform a centralized log analysis, a preparatory step consists in moving the files to a dedicated machine. A second step consists in analyzing the data to keep the useful parts on the one hand and to index the key parts on the other hand. This pipeline continues with the use of a statistical model to predict the actions to be planned (Figure 1). Finally, the last step concerns the collection of metrics in order to evaluate the monitoring process.

During our first prototypes, the volume of data processed exceeded 10 MB per hour and it became evident that such a sequential process could not meet our needs. The choice of a Big Data cluster for the processing of such volumes of text is legitimate, especially since this work relates to the monitoring of distributed systems. Because our log data is distributed and replicated on a cluster, the use of Big Data frameworks, such as Spark, exploits this distribution for the execution of task graphs that focus on data pre-processing as well as indexing and storage for the construction of an AI model.

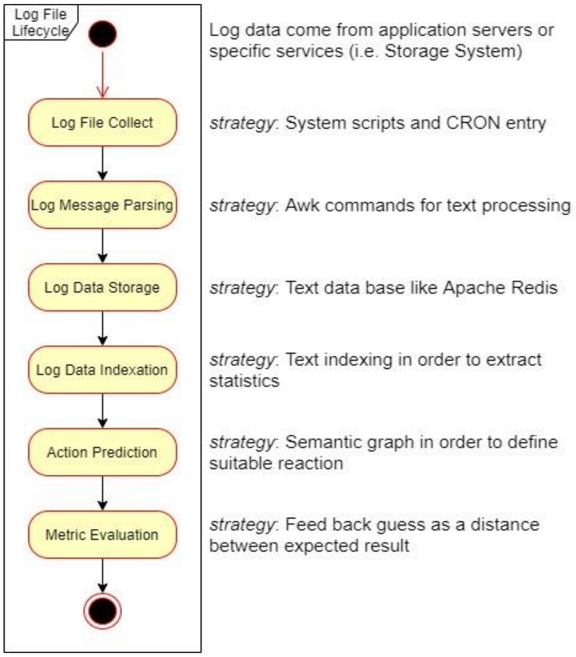

Log File Lifecycle
Log data come from application servers or specific services (i.e. Storage System)
Log File Collect
strategy: System scripts and CRON entry
Log Message Parsing
strategy: Awk commands for text processing
Log Data Storage
strategy: Text data base like Apache Redis
Log Data Indexation
strategy: Text indexing in order to extract statistics
Action Prediction
strategy: Semantic graph in order to define suitable reaction
Metric Evaluation
strategy: Feed back guess as a distance between expected result

Figure 1. Log file transformations.

IV. Software Architecture

Log analysis tasks often have strict due dates and data quality is a primary concern in software monitoring activities. This underlines the importance of finely managing the sequencing of tasks on the analysis platform.

The Hadoop ecosystem offers a set of software to process huge data sets. It was originally designed to run on clusters of physical machines. Distributed analytical frameworks, for example MapReduce, evolve into resource managers that gradually transform Hadoop into a very versatile data operating system. With these frameworks, one can perform a wide range of data manipulation and analysis operations by connecting them to Hadoop Distributed File System (HDFS) as a document storage system [27].

Hadoop is highly scalable; it is easy to add a new service, such as a search engine. As it includes the Zookeeper clustering tool, it is able to deploy on a set of nodes a search engine to manage a large volume of text-oriented data.

We have made the choice to use Apache Solr to index, search a large amount of business data, and provide relevant content based on a search query [28]. Solr is based on open standards, it is highly extensible. Solr queries are simple HTTP request URLs and the response is a structured document.

A. *Big Data platform*

A part of our work is based on Solr framework 8.1 and the integration with all other components in Hortonworks Data Platform Virtual Machine (HDP VM), such as Apache HBase, Apache Spark, Apache Kafka, in addition to some other open source tools. A part of our work relies on specific configurations of the tools; another part is the development of specific components for customizing the behavior of the Hadoop tools.

Our article outlines our approach and a simplified architecture for analyzing software-generated logs to detect functional-related issues. Our architecture is a batch analytics system analyzing Solr query logs.

The diagram from Figure 2 illustrates the high level of our software architecture. From a file, which contains the schedule of cron entries to be run and at specified times, we trigger a log file collection into a specific folder.

We use shell scripts to collect log files destined for a remote directory (named "log file folder source"). With a common data ingestion path, the logs go from an Apache Flume source, then to a Apache Kafka channel and are transmitted to a first Spark consumer (named "Spark SQL consumer"). Its essential task is to recognize and process the contents of the file and load them into an SQL table in memory, perform filter operations and put them in common format. Then, the route continues with a backup of these data in HBase tables. The role of this Flume route is to store structured information in a column-oriented database (the blue route in Figure 2).

In parallel, another route has the role of indexing the data from the logs (red route in Figure 2). From the same Kafka source, a second Spark consumer (named "Spark Solr consumer") takes care of data indexing while respecting the Solr schema. The index is updated for the query steps and then we use of a model for the prediction of maintenance tasks.

In this architecture, HBase is a highly reliable data store, supporting disaster recovery and cross-datacenter replication. Solr Cloud is the indexing and search engine. We have defined a data type schema for the analysis of log messages but also for the analysis of queries made against this set of messages. It is completely open and allows us to personalize text analyzes. It allows a close link with HBase database so the schemas used by both tools are designed in a closely related way.

The Jasper Report tool allows us to build a report from data automatically and regularly. Suitable cross tables help to give priorities to software maintenance tasks. Report construction is based on insertion events in HBase tables but also on Solr document additions associated with Solr index updates. The volume of data is traced as well as the time spent on operations specific to each server. Jasper Report offers the ability to link data within a report to improve tracking.

B. *Configuration*

1) *Via operating system*

Several elements of this architecture support ad hoc configuration. We have defined specific configuration scripts for routing log files to the "log file folder" directory, source Flume. We use entries in cron tables to ensure regular data collection.

TABLE III. CRON ENTRY FOR LOG COLLECTION

00 */6 * * * /root/monitoring/scripts/log-collection.sh

For instance, the next table 3 displays an execution of a collector script every six hours.

2) *Via event streaming-tools.*

We have described two Flume routes within our Big Data cluster. Flume configurations correspond to the creation of routing agents so that information reaches the programs that use them. For instance, the next table 4 shows a Flume agent configuration for the blue route on Figure 2.

TABLE IV. FLUME AGENT CONFIGURATION

```
blue.sources = r1
blue.channels = c1
blue.sinks = k1

blue.sources.r1.type = spooldir
blue.sources.r1.spoolDir = /root/monitoring/log_folder/source
blue.sources.r1.channels = c1

blue.channels.c1.type = memory
blue.channels.c1.capacity = 1000
blue.channels.c1.transactionCapacity = 100

blue.sinks.k1.channel = c1
blue.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
blue.sinks.k1.kafka.topic = application
blue.sinks.k1.kafka.bootstrap.servers = localhost:9092
blue.sinks.k1.kafka.flumeBatchSize = 20
blue.sinks.k1.kafka.producer.acks = 1
blue.sinks.k1.kafka.producer.linger.ms = 1
blue.sinks.k1.kafka.producer.compression.type = snappy
```

This agent configuration defines the first part of the blue route in Figure 2. The source of this route is named r1 and corresponds to a directory of log files. The transit channel is in memory with a certain volume of events allowed. Finally, a transaction capability is initialized so that two or more routes can be used in the same distributed operation. Finally, the log messages are published in a dedicated "application" topic which implies that the messages come from web applications deployed on an application server such as JBoss WildFly.

The Flume and Kafka tools are both event-streaming tools. While their roles are comparable, the developments in these two projects are very different and there are now more Kafka connectors. Thus, the popularity of Apache Kafka is currently higher than that of Flume. We have kept software routes with Flume for event routing, but we define Kafka topics to ensure decorrelation between components. This makes it possible to simplify the management of components, among other things for updates. In addition, the Kafka API allows more controls on the management of messages associated with a topic; for example, time management. We have added rules to ensure that a received message is processed within an hour. In that case, we raise an alert and the data is saved in the local file system.

3) *Via persistent storage.*

We wrote the script for creating tables structured in families of columns to keep the information from the log files. The column families are logical and physical groups of columns. The columns in one family are stored separately from the columns in another family. Because we have data that are not often queried, we assign that data to a separate column family.

Because the column families are stored in separate HFiles, we keep the number of column families as small as possible. We also want to reduce the number of column families to reduce the frequency of mem-store flushes, and the frequency of compactions. Moreover, by using the smallest number of column families possible, we improve the load time and reduce disk consumption.

4) *Via indexing engine.*

Apache Solr is an open source search engine and Solr index can be considered as an equivalent of a SQL table. A standalone instance maintains several indexes. However, on our Big Data cluster, the Solr installation is also distributed. In that context, we have four shards with a replication rate equals to three. This allows us to distribute operations by reducing blockages due to frequent indexing. We have configured not only the schema, but also the data handlers (*schema.xml* and *solrconfig.xml* files).

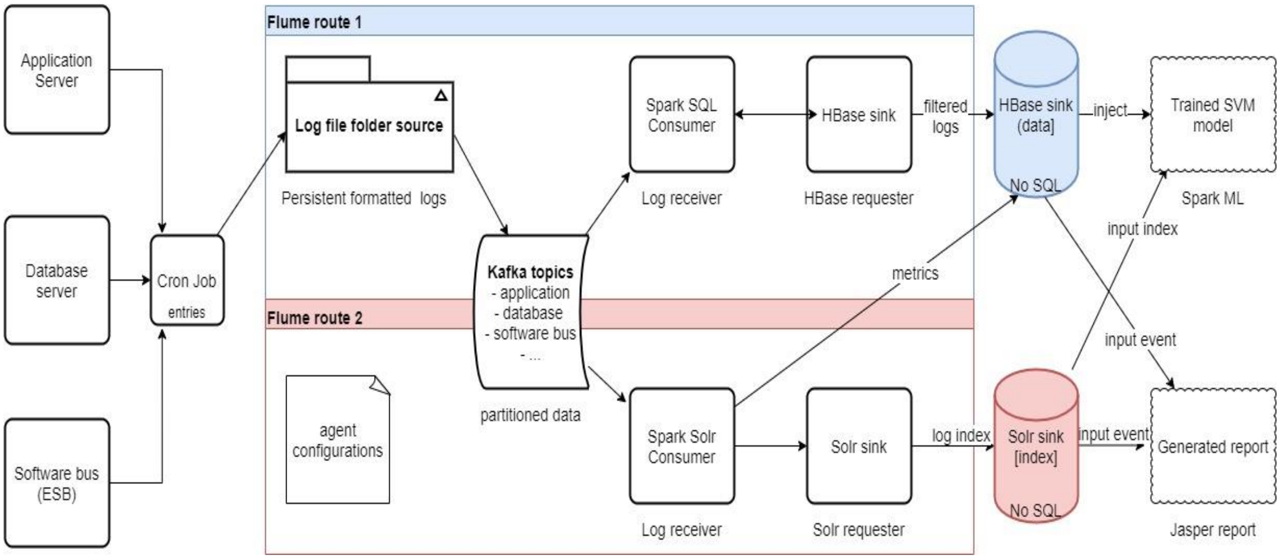


Figure 2. Big Data workflow for log analysis.

Our schema defines the structure of the documents that are indexed into Solr. This means the set of fields that they contain. We also define the datatype of those fields. It configures also how field types are processed during indexing and querying. This allows us to introduce our own parsing strategy via class programming.

As we have studied the log formats that are handled in the various applications, it is possible to declare the set of fields belonging to the log messages. Thus, we have defined the following field types: line number, log message, method name, and priority, thread name, time consumed per method, date and time, etc. The definition of a field includes either an existing or a custom typing, a parsing strategy in the case of a new document or in the case of a query. Thus, a message such as the one in table 5 is split into tokens by our implementation of the Tokenizer class, then the recognition of the fields of the document is obtained by applying parsing rules that come from our schema.

TABLE V. LOG MESSAGE EXAMPLE

```
[INFO] 11-14-2020 20:36:01,331 org.apache.spark.sql.SQLContext set-
This Is an update of property
```

The use of date is essential when handling log messages. Solr allows the use of milliseconds at best which is a strong limitation that sometimes leads to the loss of information to standardize the recognized values. The DatePointFIeld type represents a temporal point on a time scale and this allows the comparison of log messages when aggregating flows.

This schema configuration then leads us to define fields (thanks to the previous types) but to assign them additional properties such as indexed, stored, multivalued, docValues, required, useDocValuesAsStored. These properties are all the more important as they allow us to store the indexes in Solr engine while the data are kept in HBase. In our context the couple (indexed, stored, docValues) has the value (true, false, true). This last value is the indicator meaning that the data are stored in a column-oriented database.

C. Component architecture

1) Based on Spark framework.

To implement this architecture, we have developed several components using the Spark framework version 2.4.7. These components are at the heart of Flume routes, so their sequencing is based on the Spark-streaming module. In other words, when log data are available, the scheduler creates micro batches to process these data during a fixed duration window. In order to keep the results of the processing, the components save their results in a HBase database installed on the Hadoop cluster [29].

We have two consumers of the data associated with the Kafka topic. Spark SQL consumer uses the Spark SQL module to store data in an HBase database whose schema is structured in family of columns. The labels of these families of columns are involved in the data schema of the second Spark consumer.

HBase is a database distributed on the nodes of our Hadoop cluster, which allows having a persistence system where the data are highly available because the replicated rate on separate nodes is set to three.

2) Based on Spring Data.

Spark Solr consumer uses the Spring Data and SolrJ library to index the data read from the Kafka topic. It splits the data next to the Solr schema where the description of each type includes a "*docValues*" property, which is the name of the HBase column family. For each Solr type, our configuration provides a given analyzer. We have developed some of the analyzers in order to keep richer data than simple raw data from log files. Finally, the semantic additions that we add in our analysis are essential for the evaluation of Solr query. Likewise, we store the calculated metrics in HBase for control.

SolrCloud is deployed on the cluster through the same Zookeeper agents. Thus, the index persistence system is also replicated. We therefore separate the concepts of backup and search via two distinct components. This reduces the blockages related to frequent updates of our HBase database [30].

3) Based on SolrJ library.

At the beginning of our Solr design, we have built our schema based on our data types. Some of them were already defined, but some others are new. In addition, we have implemented new data classes for the new field types. For example, we used *RankFieldType* as a type of some fields in our schema. Then, it becomes a sub class of *FieldType* in our Solr plugin. An example is given in the table 6 below.

TABLE VI. FIELDTYPE DEFINITION EXAMPLE

```
<fieldType name="thread_name" class="solr.TextField"
positionIncrementGap="100">
 <analyzer type="index">
  <tokenizer class="fr.uoec.lacl.TokenizerFactory"/>
  <filter class="fr.uoec.lacl.filter.StopFilterFactory" ignoreCase="true"
words="stopwords.txt" enablePositionIncrements="true" />
  <filter class="solr.LowerCaseFilterFactory"/>
 </analyzer>
 <similarity class="fr.uoec.lacl.similarities.NoCoordsDefaultSimilarity">
 </similarity>
</fieldType>
```

We have redesigned Solr filters so that they can be used in our previous setups. Our objective was to standardize the values present in the logs coming from different servers. Indeed, the messages provide information of the form <attribute, value> where the values certainly have units. However, the logs do not always provide the same units for the same attribute calculation. The analysis phase is the place to impose a measurement system in order to be able to compare the results later.

The development pattern proposed by SolrJ is simple because it proposes abstract classes like *TokenFilter* and *TokenFilterFactory* then to build inherited classes. Then we have to build a plugin for Solr and drop it in the technical directory agreed in the installation of the tool [31]. We have redefined similarities, i.e., how to calculate the score of a document. In particular, we wanted to change the weighting of small documents, which by default have a lower value in the classical similarities of Solr. So in the definition of our data types, such as RankFieldType or MethodNameType, we have defined a new similarity. It influences the calculation of the index score. This redefinition is obtained by defining a subclass of DefaultSimilarity of the package org.apache.lucene.search.similarities. By default, we use the BM25 similarity of Solr for any new type but some log message fields have more weight and the string length alone is not enough. The following table displays the incomplete definition of a field type where a new similarity is assigned.

In each new class derived from Similarity, it is necessary to redefine 4 methods coord(int, int), idf(long, long), lengthNorm(FieldInvertState) and tf(float). We wanted to be able to give a stronger weighting to the different types of our log messages to help promote or demote results. We could see in the explanation string that this was because the coord function applies a formula that takes into account the number of included clauses versus the number of clauses each document matches. Therefore, we redefined it so that it returns a constant.

The tf function stands for Term Frequency: we examine one term at a time and the more often the term appears in the logs, the higher the score. We actually take the square root of the tf(): if we search for "pearsonCorrelation()", a document mentioning this method name twice is more likely to be about "pearsonCorrelation()", but maybe not twice as likely as a document with one occurrence. We correct this by raising this result to the power (-1/3) to put the weight into perspective. The idf() function stands for Inverse Document Frequency. In this case, the document frequency is the ratio of the number of documents containing the term to all documents in our index. If you search for "correlation" and "pearson" in our database containing log messages, the frequency of the document "correlation" will be very high. The IDF will be very low, because "correlation" will carry less interesting information about the documents compared to "pearson" which has a higher IDF. As before, the structure of a log message leads us to give importance to areas identified as "thread name". Also, our idf() function does not rely on the inverse of the frequency of a term in relation to the set of documents. But we want the idf() to increase as the number of documents where it appears increases. We therefore correct by composing with a log decimal for this metric.

The normalization function (lengthNorm) is also redefined so that it is not limited to taking into account the length of the documents. Thus, in the redefinition of this function, we minimize the score of the date part in favor of the parts "thread name" and "method name" which brings more semantics in the detection of anomaly. Thus, the final score for a log message will come more from these 2 parts of the message. Other choices can be made but this concerns other works than the one presented here.

4) Based on Spark-MLlib.

In Artificial Intelligence, Support Vector Machine (SVM) models are a set of supervised learning techniques designed to solve discrimination and regression problems. SVMs have been applied to a large number of fields (bioinformatics, information research, computer vision, finance, etc.) [32]. SVM models are classifiers, which are based on two key ideas, which allow to deal with nonlinear discrimination problems, and to reformulate the ranking problem as a quadratic optimization problem. In our project, SVMs can be used to decide to which class of problem a recognized sample belongs. The weight of these classes if linked to the Solr metrics on these names. This amounts to predicting the value of a variable, which corresponds to an anomaly.

All filtered log entries are potentially useful input data if it is possible that there are correlations between informational messages, warnings, and errors. Sometimes the correlation is strong and therefore critical to maximizing the learning rate. We have built a specific component based on Spark MLlib It supports binary classification with linear SVM. Its linear SVMs algorithm outputs an SVM model [33].

We applied prior processing to the data from our HBase tables before building our decision modeling. These processes are grouped together in a pipeline, which leads to the creation of the SVM model with the configuration of its hyper-parameters such as *weightCol*. Part of the configuration of these parameters comes from metrics calculated by our indexing engine (Figure 2). Once created and tested, the model goes into action to participate in the prediction of incidents. We use a new version of the SVM model builder based on distributed data augmented. This comes from an article written Nguyen, Le and Phung [34].

5) Based on Jasper Report library.

This reporting library allows us to build weekly graphical reports on indexing activity. This information is a help to check the suitability of the SVM model, which supports prediction requests following pattern recognition. The representations are documents in pdf format; we did not automate the impact of this data extraction on the use of our decision-making model.

V. Big Data Streaming

We use Apache Kafka as queue system for our logs. Then we use spark streaming library to read from Kafka topic and process logs on the fly. Spark Streaming is a real-time processing tool that runs on top of the Spark engine. The scheduler exploits all the computation resources of our cluster. Each node runs several executors, which run tasks and keeps data in memory or disk storage across them.

In our program, the Spark context sends all the tasks for the executors to run.

A. Filtered log strategy

1) Asynchronous reading.

Our component called Spark SQL Consumer contains a Kafka receiver class, which runs an executor as a long-running task. Each receiver is responsible for exactly one input discretized stream (called *DStream*). In the context of the first Flume route, this stream connects the Spark streaming to the external Kafka data source for reading input log data.

Because the log data rate is high, our component reads from Kafka in parallel. Apache Kafka stores the data logs in topics, with each topic consisting of a configurable number of partitions. The number of partitions of a topic is an important key for performance considerations as this number is an upper bound on the consumer parallelism. If a topic has N partitions, then our component can only consume this topic with a maximum of N threads in parallel. In our experiment, the Kafka partition number is set to four.

2) Normalized form.

Since log data are collected from a variety of sources, data sets often use different naming conventions for similar informational elements. The Spark SQL Consumer component aims to apply name conventions and a common structure. The ability to correlate the data from different sources is a crucial aspect of log analysis. Using normalization to assign the same terminology to similar aspects can help reduce confusion and error during analysis [35]. This case occurs when log messages contain values with different units or distinct scales. The log files are grouped under topics. We apply transformations depending on the topic the data come from. The filtered logs are cleaned and reorganized and then are ready for an export into an HBase instance.

3) Stuctured data storage.

Next step, the Spark SQL Consumer component inserts the cleaned log data into memory data frames backed to a schema. We have defined a mapping between HBase and Spark tables, called Table Catalog. They are two main difficulties of this catalog.

a) The row key definition implies the creation of a specific key generator in our component.

b) The mapping between table column in Spark and the column family and column qualifier in HBase needs a declarative name convention.

The HBase sink exploits the parallelism on the set of region servers, which are under control of the HBase master. The HBase sink treats both *Put* operation and *Delete* operation in a similar way, and both actions are performed in the executors. The driver Spark generates tasks per region. The tasks are sent to the preferred executors collocated with the region server, and are performed in parallel in the executors to achieve better data locality and concurrency. By the end of an exportation, a timed window of log data is stored into HBase tables.

B. Index construction and query

1) The index pipeline

The strategy of the Spark Solr Consumer component deals with the ingestion of the log data into Apache Solr for search and query. The pipeline is built with Apache Spark and Apache Spark Solr connector (Figure 3).

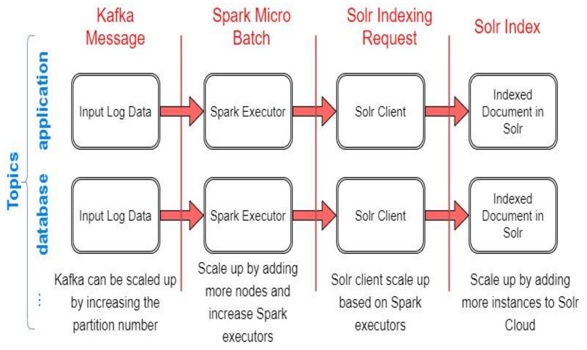


Figure 3. Overall high-level architecture of the index pipeline.

Spark framework is used for distributed in memory compute, transform and ingest to build the pipeline.

The Apache HBase role is the log storage and the Apache Solr role is the log indexing. Both are configured in cloud mode Multiple Solr servers are easily scaled up by increasing server nodes. The Apache Solr collection, which plays the role of SQL table, is configured with shards. The definition of shard is based on the number of partitions and the replicas rate for fault tolerance ability.

The Spark executors run a task, which transforms and enriches each log message (format detection). Then, the Solr client takes the control and sends a REST request to Solr Cloud Engine. Finally, depending on the Solr leader, a shard is updated.

2) The query process.

We also use Solr Cloud as a data source Spark when we create our ML model. We send requests from spark ML classes and read results from Solr (with the use of Solr Resilient Distributed Dataset (SolrRDD class). The pre statement of the requests is different from the analysis of the log document. Their configuration follows another analysis process.

With Spark SQL, we expose the results as SQL tables in the Spark session. These data frames are the base of our ML model construction. The metrics called TF (Term Factor) and IDF (Inverse Document Frequency) are key features for the ML model. We have also used boost factor for customizing the weight of part of log message.

VI. Results and Actions

We have several kinds of results. A part is about our architecture and the capacity to treat log messages over time. Another part is about the classification of log messages. The concepts behind SVM algorithm are relatively simple. The classifier separates data points using a hyperplane with the largest amount of margin. In our working context, the margin between log patterns is a suitable discriminant.

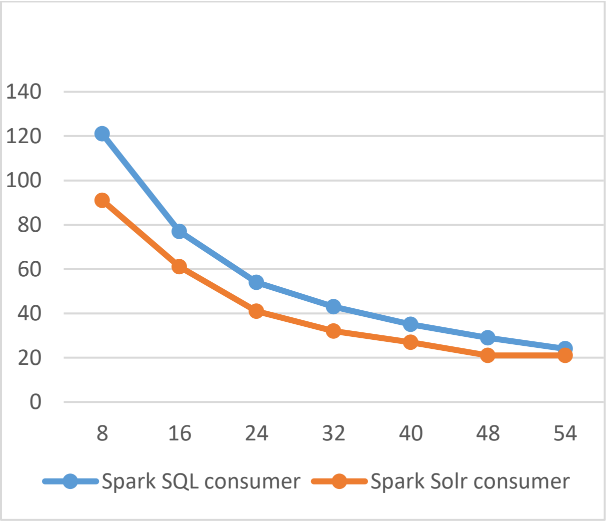


Figure 4. Spark consumer runtime versus number of partitions

A. Data features

1) Architecture measurement

For our tests, we used previously saved log files from 20 days of application server and database server operations. We were interested in the performance of the two Spark consumers, the Spark SQL Consumer and the Spark Solr Consumer.

For Spark SQL Consumer, the volume of data to analyze is 81.7 M rows in HBase. To exploit this data, we used a cluster of eight nodes on which we deployed Spark and HBase. The duration of the tests varies between 24 minutes and 2 hours and 1 minute.

For Spark Solr Consumer, the volume of data indexed is 87.2M rows indexed in about an hour. The number of documents indexed per second is 28k.

We only installed Solr on four nodes with four shards and a replication rate of three. We have seen improved results by increasing the number of Spark partitions (*RangePartitioner*). At runtime for our data set based on a unique log format, the cost of Spark SQL consumer decreases when the partitioning of dataset increases, an illustrated in Figure 4. The X-axis represents the partition number and the Y-axis represents the time consumed. We have to oversize the partitions and the gains are much less interesting.

2) Model measurement

SVM offers very high accuracy compared to other classifiers such as logistic regression, and trees. They are several modes of assessment. The first is technical; it is obtained thanks to the framework used for the development (Spark MLlib). The second is more empirical because it relates to the use of this model and the anomaly detection rate on a known dataset.

The analytical expressions of the features precision and recall of retrieved log messages that are relevant to the find are indicated below.

Precision is the fraction of retrieved log messages that are relevant to the find:

$$precision = \frac{|\{relevant \log messages\} \cap \{retrieved\ log\ messages\}|}{|\{retrieved \log messages\}|}$$

Recall is the fraction of log messages that are relevant to the query that are successfully retrieved:

$$recall = \frac{|\{relevant \log messages\} \cap \{retrieved\ log\ messages\}|}{|\{relevant \log messages\}|}$$

$$F_\beta = (1+\beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

In Table 7, we have four classes and for each class we compute three numbers: true positive (tp), false positive (fp) and false negative (fn). For instance, for the third class, we note these numbers tp3, fp3 and fn3. From these values, we compute precision by label, recall by label and F-score by label.

TABLE VII. SVM MODEL MEASURES

Class number	**Metrics**		
	Precision by label	***Recall by label***	***F1 score by label***
0,000000	0.884615	0.920000	0.901961
1,000000	1.000000	1.000000	1.000000
2,000000	0.846154	0.785714	0.814815
3,000000	0.854462	0.7914858	0.842529

Our prediction models are similar to a multiclass classification. We have several possible anomaly classes or labels, and the concept of label-based metrics is useful in our case. Precision is the measure of accuracy on all labels. This is the number of times a class of anomaly has been correctly predicted (true positives) normalized by the number of data points. Label precision takes into account only one class and measures the number of times a specific label has been predicted correctly normalized by the number of times that label appears in the output. The last observations are:

- Weighted precision = 0.917402
- Weighted recall = 0.918033
- Weighted F1 score = 0.917318
- Weighted false positive rate = 0.043919

Our results for four classes are within acceptable ranges of values for the use of the model to be accepted.

The test empirical phase on the SVM model was not extensive enough to be conclusive; however, our results suggest that increasing the number of log patterns deteriorates the performance. In addition, we defined a finite set of log patterns for a targeted anomaly detection approach.

B. Reporting

We have created a custom data source to connect to Apache Solr, therefore, we are able to retrieve data and provide them back in following the *JRDataSource* interface of Jasper Report. With this access point, we have extracted metrics about the document cache and Query result cache. Both give an overview of the Solr activities and is meaningful for the analysts.

We have deployed the CData JDBC Driver on Jasper Reports to provide real-time HBase data access from reports. We have found that running the underlying query and getting the data to our report takes the most time. When we generate many pages per report, there is overhead to send that to the browser.

For the reporting phase, we have developed two report templates based on the use of a JDBC adapter. With system requests, we collect data about the last events (Get, Put, Scan, and Delete). From these HBase view, we have designed the report templates with cross tables. For the storage phase, we compute and display the number of Put events per timed window or grouped over a period.

We periodically updated the data across report runs and export the PDF files to the output repository where a web server manages them.

VII. Conclusion and future work

We have presented our approach to log file indexing and maintenance task prediction. We have shown how an index engine is crucial for a suitable query engine. We have developed specific plugins to customize the types of fields in our documents, but also to filter the information in the log message in order to facilitate log analysis.

Because indexing especially log file customization and analysis are two sides of our study, we customized log analysis tools and configured indexes on the input data for semi-structured formats. The Spark and Solr Cloud frameworks allowed us to index and store data for building an AI model. We used the HDP machine to install our various components. Using the JBoss WildFly Framework allowed us to aggregate server log messages with those deploying applications. The configuration of the configuration files of its libraries, such as log4j or slf4j allows each application to have personalized or separate log messages. We have defined a standard data schema and a common format to facilitate the extraction and analysis of queries made on a set of messages. We do this by adopting our log profile approach. It is based on the fact that the Log configuration can be modified without requiring the redeployment of the applications concerned. For example, configuration via the operating system using scripts allows us to ensure regular data collection, while configuration via the indexing engine allows us to define our own analysis strategy.

From the filtered logs, we have defined a new similarity which influences the calculation of the index score. We also presented the construction of our SVM model based on work from the Center for Pattern Recognition and Data Analytics, Deakin University, (Australia). We were thus able to classify the recognized log patterns into classes of anomalies. This means that we can identify the associated maintenance operations. Finally, to measure the impact of our distributed analysis system, we wanted to automatically build reports based on templates and highlight indexing and storage activity.

Our study also shows the limits that we want to push, such as improving the AI model of log data classification to optimize computation precision and avoid overloading during extraction. Using an AI model is no guarantee of optimal results. We want to make more use of indexing metrics to give more weight to certain information in the analyzed logs. One of the perspectives will the uniformization of properties and also the reduction of information losses during the normalization of recognized values or the aggregation of flows.

The log format has a deep impact on the Solr schema definition and about the anomaly detection. We are going to evolve our approach. In the future, we want to extract dynamically the log format instead of the use of a static definition.

We think also about malicious messages, which can perturb the indexing process and introduce bad requests in our prediction step. The challenge needs to manage a set of malicious patterns and the quarantine of some message sequences.

References

[1] F. Mourlin, C. Mahmoudi and G. L. Djiken. “Distributed Search on a Large Amount Log Data.” The 7th International Conference on Big Data, Small Data, Linked Data. ALLDATA, IARIA. April 2021.

[2] G. Hohpe. "Developing Software in a service-oriented world." Datenbanksysteme in Business, Technologie und Web, 11. Fachtagung des GIFachbereichs “Datenbanken und Informationssysteme”(DBIS). 2005.

[3] J. Andersson and U. Schwickerath. “Anomaly Detection in the Elasticsearch Service.” CERN Openlab Summer Student Report, pp. 1-18, 2019.

[4] C. Clark, et al. “Live migration of virtual machines.” In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2 (pp. 273-286). May 2005.

[5] D. E. Lowell, Y. Saito, and E. J. Samberg. “Devirtualizable virtual machines enabling general, single-node, online maintenance.” ACM SIGARCH Computer Architecture News, 32(5), 211-223. 2004.

[6] T. S. Collett. “A review of well-log analysis techniques used to assess gas-hydrate-bearing reservoirs.” Natural Gas Hydrates : Occurrence, Distribution, and Detection, Geophysical Monograph. The American Geophysical Union, vol. 124, pp. 189-210, 2001.

[7] S. K. Shivakumar. "Mobile Web Performance Optimization." Modern Web Performance Optimization. Apress, Berkeley, CA, pp. 79-103. 2020.

[8] https://kafka.apache.org/documentation/. 2020. web. Seem: 2021.09.04.

[9] J. Kreps. “The Log: What every software engineer should know about real-time data's unifying abstraction.” https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying. Seem: 2021.09.19.

[10] J. Kreps, N. Narkhede, and J. Rao. "Kafka: A distributed messaging system for log processing." Proceedings of the NetDB. Vol. 11. 2011.

[11] F. Qiang, et al. “Execution anomaly detection in distributed systems through unstructured log analysis.” 2009 ninth IEEE international conference on data mining IEEE. pp. 149-158. 2009.

[12] A. Oliner, A. Ganapathi and W. Xu. “Advances and challenges in log analysis.” Communications of the ACM, vol. 55, no. 2, pp. 55-61, 2012.

[13] D. A. Liman, et al. "The log data collection service for cloud robotics." IEEE 11th International Conference on Application of Information and Communication Technologies (AICT). IEEE. 2017.

[14] C. Lim, N. Singh and S. Yajnik. “A log mining approach to failure analysis of enterprise telephony systems.” IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN). pp. 398-403. 2008.

[15] J. Breier and J. Branišová. “Anomaly detection from log files using data mining techniques.” In Information Science and Applications, Springer, Berlin, Heidelberg. pp. 449-457. 2015.

[16] L. Tao et al. “FLAP : An end-to-end event log analysis platform for system management.” Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1547-1556. 2017.

[17] S. Yen and M. Moh. “Intelligent log analysis using machine and deep learning.” Machine Learning and Cognitive Science Applications in Cyber Security, IGI Global. pp. 154-189. 2019.

[18] D. Qingfeng et al. "Log-Based Anomaly Detection with Multi-Head Scaled Dot-Product Attention Mechanism." International

Conference on Database and Expert Systems Applications. Springer. Cham. 2021.

[19] B. Jasmin and S. Nedelkoski. "Multi-source anomaly detection in distributed it systems." International Conference on Service-Oriented Computing. Springer. Cham. 2020.

[20] B. Debnath et al. “Loglens : A real-time log analysis system.” 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE. pp. 1052-1062. 2018.

[21] R. Wang et al. "Logprov: Logging events as provenance of big data analytics pipelines with trustworthiness." 2016 IEEE International Conference on Big Data (Big Data). IEEE, 2016.

[22] F. H. Allen, J. Allan, and J. Glatt. "CrowdLogging: distributed, private, and anonymous search logging." Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. 2011.

[23] D. Tancharoen, T. Yamasaki, and K. Aizawa. “Practical experience recording and indexing of Life Log video,” In Proceedings of the 2nd ACM workshop on Continuous archival and retrieval of personal experiences (pp. 61-66), November 2005.

[24] D. Tancharoen, T. Yamasaki, and K. Aizawa. “Practical life log video indexing based on content and context.” In Multimedia Content Analysis, Management, and Retrieval 2006. International Society for Optics and Photonics. Vol. 6073. pp. 60730E. January 2006.

[25] I. Fronza et al. “Failure prediction based on log files using random indexing and support vector machines,” Journal of Systems and Software. 86(1). pp. 2-11. 2013.

[26] M. Picquenot and P. Thébault. “GLPI (Free Management of Computer Park) : Installation and configuration of a park management solution and support center.” ENI Editions. 2016.

[27] K. Shvachko et al. “The hadoop distributed file system,” 2010 IEEE 26th symposium on mass storage systems and technologies (MSST), IEEE. pp. 1-10. 2010.

[28] D. Smiley et al. “Apache Solr enterprise search server.” Packt Publishing Ltd. 2005.

[29] R. C. Maheshwar and D. Haritha. “Survey on high performance analytics of bigdata with apache spark.” 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE. pp. 721-725. 2016.

[30] K. Koitzsch. “Advanced Search Techniques with Hadoop, Lucene, and Solr.” Pro Hadoop Data Analytics, Apress, Berkeley, CA. pp. 91-136. 2017.

[31] J. Kumar, “Apache Solr search patterns,” Packt Publishing Ltd, 2015.

[32] M. F. Ghalwash, D. Ramljak and Z. Obradović, “Early classification of multivariate time series using a hybrid HMM/SVM model.” 2012 IEEE International Conference on Bioinformatics and Biomedicine, IEEE. pp. 1-6. 2012.

[33] M. Assefi et al. “Big data machine learning using apache spark MLlib.” 2017 IEEE International Conference on Big Data (Big Data), IEEE. pp. 3492-3498. 2017.

[34] T. D. Nguyen et al. “Distributed data augmented support vector machine on spark.” 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE. pp. 498-503. 2016.

[35] F. E. N. G. Changyong et al., “Log-transformation and its implications for data analysis.” Shanghai archives of psychiatry. vol. 26, no. 2, 2014.

Hybrid Neural Network Learning for Multiple Intersections along Signalized Arterials: A Microscopic Simulation vs. Real System Effect

Hong Wang, Chieh Wang, Yunli Shao, Wan Li
National Transportation Research Center
Oak Ridge National Laboratory
1 Bethel Valley Road
Oak Ridge, TN 37934, USA
e-mail: {wangh6; cwang; shaoy; w5i}@ornl.gov

Arun Bala Subramaniyan, Guohui Zhang, Tianwei Ma
Department of Civil and Environmental Engineering
University of Hawaii at Manoa
2540 Dole Street
Honolulu, Hawaii 96822, USA
e-mail: {arunbs; guohui; tianwei}@hawaii.edu

Jon Ringler
Econolite Systems
1250 N Tustin Avenue
Anaheim, CA 92807, USA
e-mail: jringler@econolite.com

Danielle Chou
Vehicle Technologies Office
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585, USA
email: danielle.chou@ee.doe.gov

***Abstract*—Control of traffic flow along arterials requires signal timing control at intersections so that the resulting traffic flows along the arterials are as smooth as possible with minimized energy usage. With advances in sensing technologies, various data sets are available, allowing effective data-driven modeling to be conducted for further controller design that produces better signal timing control at intersections. In this paper, which is an extended version of our conference paper [1], a hybrid neural network (HNN) is proposed to model the multiple intersections along a signalized arterial in Honolulu, for which the modeling structure and relevant training algorithms have been developed. The proposed HNN consists of linear dynamics and a nonlinear function; the linear dynamics present a simplified opportunity for the closed-loop control design, and the nonlinear function is a representation of unmodeled dynamics as a function of previously available system inputs and outputs. The modeling and training are performed simultaneously for linear dynamics matrices and the weights of neural networks that approximate the nonlinear dynamics of the system. A preliminarily calibrated VISSIM microscopic traffic simulation platform is proposed to learn the real system using HNN modeling in which data collected from VISSIM simulations are used to estimate the system's unknown features. The modeling results using real data and VISSIM-generated data are compared, and the desired modeling results are obtained.**

Keywords—signalized intersections; modeling; neural networks; performance analysis; signalized arterials simulation; microscopic traffic simulation.

I. Introduction

Because the number of vehicles on a section of a road is random during any time duration, the nature of the traffic flow system in signalized arterials can be represented as a dynamic and stochastic system [1]-[6]. For such systems, the inputs are the traffic demand and signal timing at each intersection, and the outputs are the traffic flow status (e.g., travel delays, queue length, traffic flow speed). An added output can be the energy consumed when vehicles pass through the arterial, e.g., gasoline or electricity usage. This system is a multi-input and multi-output (MIMO) stochastic dynamic system [6], and the objective of this research is to control signal timing at intersections so that the resulting traffic flows along the arterials are as smooth as possible with minimized energy usage. If the system is represented in the continuous time domain, the solution can be obtained using partial differential equations induced from the well-known *Ito* stochastic differential equations with random boundary conditions [7][8]. The solution for such a complicated model is quite difficult to obtain, and the model must frequently be solved using high-performance computing, which generally cannot be used for real-time control design and implementation. Therefore, data-driven modeling methods—in particular, those widely used in artificial intelligence (AI) technology—are effective ways to establish simple dynamic models between signal control and traffic flows so that system performance can be controlled and optimized in real time [9]-[37]. The advantage of using AI-based models is that these models can be adaptively learned using evolving real-time data. Therefore, the use of neural network modeling has been a subject of study for many years [2]-[37].

On the other hand, advances in wireless-driven vehicular communications have greatly facilitated modeling exercises, and emerging cooperative intelligent transportation control system operations have enabled many smart traffic control and management applications to improve traffic safety and operational efficiency [2]. Vehicle-to-everything communications allow vehicles to communicate with other vehicles; infrastructure; pedestrians, bicyclists, and devices; and internet through cellular networks and/or dedicated short-range communication technologies. The information exchanges supported by vehicle-to-everything communication systems can be used to effectively balance traffic demand distribution among traffic networks and facilitate traffic flow progression. With these new data available in a real-time format, AI-based modeling, and ultimately control, can be further enhanced to optimally coordinate signal controls for traffic flow systems along arterials.

Traffic system modeling aims to establish linear or nonlinear mathematical relationships between traffic states—such as traffic volume, travel time (travel delay), and signal timing plan—given spatiotemporal traffic information. Most studies leverage a single data source. For example, one objective is to predict near-term traffic flow given historical traffic flow data. Other studies use multiple data sources to capture dominant dependencies between different features. For example, Ke et al. [9] developed a model to predict lane-based traffic speed using speed and traffic volume data. In general, transportation system modeling techniques can be divided into non-learning–based and learning-based methods [10]. For example, classical non-learning–based methods include autoregressive integrated moving average [11] and K-nearest neighbors [12]. These models are usually more interpretable but cannot capture the spatial correlations of traffic states. Moreover, they are not appropriate for nonstationary data. Traditional learning-based methods include regression [13], Kalman filter [14], and support vector machine [15]. These methods are generally more effective than non-learning–based models, but they usually fail to capture the nonlinear spatiotemporal correlations of traffic data. Increasingly more data sources and computational power are available, so more advanced learning-based methods (e.g., different types of neural networks) have shown promising performance. The most commonly used neural networks for transportation system modeling include artificial neural networks [16], long short-term memory [17], convolutional neural networks [9], and graph-based neural networks [18]. Compared with artificial neural networks, convolutional neural networks and long short-term memory have advantages in capturing nonlinear spatial and temporal dependencies of traffic features. However, they are not suited for large transportation networks. In this context, graph-based neural networks are powerful tools for large-scale traffic signal control systems. Graph-based neural networks can extract features from graph-structured data and predict future traffic states in an efficient and effective manner. With the established dynamic stochastic models for transportation systems, the next step is to develop real-time optimal control strategies to reduce travel delays and energy consumption. Conventional traffic control methods for multiple intersections in a network, such as SCOOT [19], GreenWave [20], SOTL [21], max-pressure [22], and SCATS [23], usually assume simplified traffic conditions with complete traffic information available (e.g., predefined traffic flows and driving behaviors). Hence, they are not applicable for real-world traffic control for multiple intersections in terms of achieving smooth traffic flows with minimized energy consumption.

Recently, reinforcement learning (RL) models have been studied extensively and have made impressive progress in traffic control domains. For example, model-free RL can be categorized as value-based and policy-based methods [36]. Li et al. [28] set up a deep neural network to learn the Q-function of decentralized reinforced learning from the sampled traffic states (inputs) and the corresponding traffic conditions (outputs). Motivated by max pressure control, Wei et al. [30] developed an RL approach for large-scale road networks. Although decentralized reinforced learning models improve traffic signal control in the complex transportation systems, they treat neighboring intersections as the same and fail to model the spatial dependencies of traffic flows.

In addition, for stochastic modeling of traffic flow systems, one important criterion is the reliability of and confidence in the obtained models for control and optimization. Thus, the models need to be built using real-time input and output data, and they need to be reliable and have a high level of confidence for users. In this context, the use of modeling error entropy, or its probability density function (PDF), should be considered as the modeling objective function to be minimized [7][8]. Ideally, a narrowly distributed modeling error PDF centered at zero mean would indicate that the models obtained have high reliability and confidence intervals. This PDF is exactly the models' novelty compared with existing AI-based models for transportation systems, in which only sum of squares error has been used to judge whether the obtained model is accurate. The method of using modeling error entropy and PDF to perform online adaptive learning was established several years ago, and this approach can be applied in combination with the existing AI modeling tools to establish reliable and robust AI-based models for traffic flow systems.

Based upon this analysis, the following challenges remain in terms of AI-based modeling and control for signalized intersections along arterials and the urban grid road network [2]-[37]:

- Although the theory of AI-based modeling and control for signal control is maturing, field testing and closed-loop control implementation for a large number of intersections is still limited because of insufficient real-time data for fast feedback control realization.
- The existing AI-based modeling for transportation systems cannot yet capture the nonlinear and dynamic stochastic nature with high reliability and robustness.
- Guaranteed control performance for energy minimization is still lacking.

In data-driven approaches, the following issues need to be studied:

- Data-driven modeling requires a good set of data in a real-time framework.
- In terms of control strategies, the control model must be structured to be easily implemented in real-time. An affine type of dynamic model would be one option. This affine structure will be described in the following sections.

- Most studies have focused on simulations, and real-time 24/7 implementation is lacking.

These challenges constitute research questions to be answered. In this paper, a novel modeling strategy for control, namely a hybrid neural network (HNN) modeling strategy for control, will be described that summarizes our recent work on control of multiple signalized intersections [1]. In this effort, neural network modeling was studied for signalized intersections along an arterial in Honolulu using the real-time data from the system. An HNN model, which is a subset of neural networks, was constructed, and its learning algorithm with a convergency guarantee was established. A comprehensive assessment of the modeling effort was conducted using gradient approaches.

In addition, in our conference paper [1], modeling using historical real system data was performed. Such modeling exercises are reasonably straightforward. However, in early testing of the control design using such obtained models, the real system cannot be utilized. Therefore, a microscopic traffic simulation platform that mimics the actual systems is required, upon which comprehensive modeling and control testing can be conducted. A good simulation platform can be developed via comparing the modeling results using the simulation generated data with the same modeling effort in our conference paper [1]. Indeed, if the modeling results using simulation platform–generated data are similar to the modeling results using the real system data (i.e., the historical data), then we can claim that the simulation platform is consistent with the real system dynamics. Thus, a VISSIM simulation is needed. In comparison to the modeling of real system as reported in our conference paper [1], this extended effort will focus on the HNN modeling using a simulation-generated data stream.

The rest of this paper is organized as follows. Section II describes the system structure and the construction of the VISSIM simulation platform. Section III describes the HNN modeling structure and training algorithm, as well as the convergence requirements for the training algorithm. Section IV describes the modeling results, and Section V provides the conclusions.

II. Traffic Flow System Description and Vissim Simulation platform

A. System description

Figure 1 shows the signalized arterials to be modeled and learned, where 34 intersections are controlled by signal timing plans at these intersections. The purpose of this research is to use the obtained model to establish controlled signal timing plans so that the traffic flow along the corridor is made as smooth as possible—fulfilling the global objective of smoothing traffic flows with minimized energy usage. For such a system, the input is the signal timing plan at each intersection, and the output is the traffic delays of different phases (left turns, right turns, and through movements).

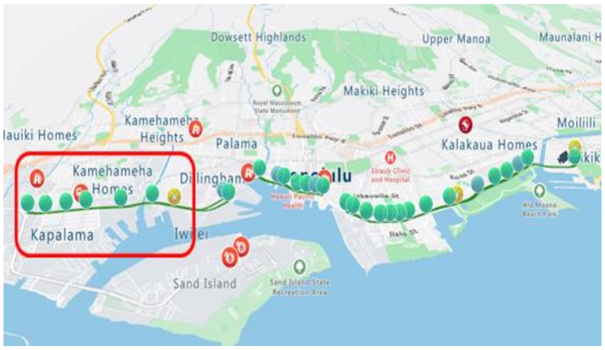


Figure 1. The signalized arterial in Honolulu with 34 intersections as indicated by green circles. Seven of the intersections were considered in this study, as indicated by the red rectangle.

To obtain such a control strategy, the dynamics of the system must be understood in terms of how signal timing plans would affect traffic flows (e.g., travel delays). This understanding requires a comprehensive modeling effort to be made and thus constitutes the main content of this paper. Thus, the objective is to develop dynamic models that reflect the dynamics of the system.

Taking $u(k)$ as the input (e.g., a signal timing plan that provides the duration of green, yellow, and red lights in a fixed cycle length) and $y(k)$ as the output vector representing the traffic delays for each phase (i.e., through movements, left turns, and right turns) at an intersection, the dynamics of the system can be generally modeled as follows:

$$y(k+1) = f\big(y(k), u(k), \omega(k)\big) \tag{1}$$

where $f(\dots)$ is the nonlinear vector function representing the system dynamics, $\omega(k)$ is the random noise term, and k is the sample number, which can be a multiplication of cycle duration in signal timing control. For example, assuming that the cycle length is 100 s and the system is sampled every two cycles, then the changes of k from i to j ($i < j$) covers the time interval of length 2 × 100 × ($j - i$) seconds.

B. Why Vissim?

Based upon the previous discussion, to facilitate the control design, a microscopic simulation, namely the VISSIM simulation [38], is generally required in transportation systems research. Such a microscopic traffic simulation system can be established to represent the original system shown in Figure 1 with regular calibration using the real system data. Assuming that the VISSIM simulation is well calibrated using the real system data from Figure 1, simulations on the model-based controller design can be readily performed using such a platform. For this purpose, a VISSIM simulation platform shown in Figure 2 was established in this work.

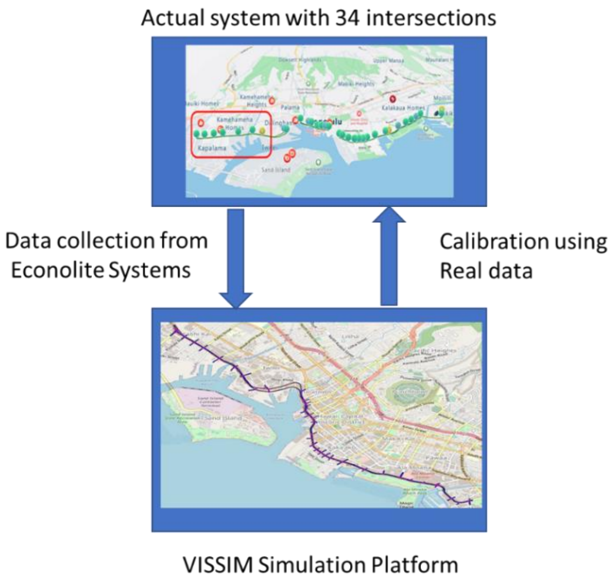

Actual system with 34 intersections
Data collection from Econolite Systems
Calibration using Real data
VISSIM Simulation Platform

Figure 2. VISSIM simulation platform.

In the transportation system study, PTV VISSIM [38], a microscopic software for traffic simulation and signal controls, has been widely used to facilitate the development and testing of different traffic signal control methods. For the system shown in Figure 1, the VISSIM simulation platform was constructed as shown in Figure 2. For this system, VISSIM used Wiedemann car-following and lane-changing models [39][40] to model the movements and interactions of vehicles. The VISSIM traffic model shown in Figure 2 was developed based on actual road geometries of the study area. This microscopic simulation model has been pre-calibrated by actual traffic data of the system. The real-world data include high-resolution data obtained from the signal controllers and traffic cameras installed at each intersection. The timing plan, actual green light time, and cycle lengths of each intersection are available. The cameras behave as advanced, stop-bar, and pulse detectors that record all vehicle arrival, departure, and stop events at each phase of each intersection. These camera detectors provide detailed vehicle volume and turning movement data. In addition, raw camera video feeds were obtained to determine vehicle compositions and microscopic vehicle behaviors. These data were directly fed into the VISSIM simulation as inputs to adjust parameters, including signal timing plan, vehicle inputs, vehicle compositions, speed distributions, conflict areas, priority rules, reduced speed areas, and car-following and lane-change behaviors. The VISSIM-simulated traffic performance metrics such as vehicle delay, travel time, and travel speed were compared against the real-world performance to further fine-tune the simulation. This enhancement ensures that the simulation was consistent with the traffic characteristics of the real world.

Such a VISSIM simulation platform can be regarded as a digital twin, which is a parallel digital system linked to the actual system through data transmission between them. Figure 3 provides a closer look of a small set of intersections in the VISSIM simulation platform; real system data from March 2–April 2, 2021 were used to calibrate the models in VISSIM.

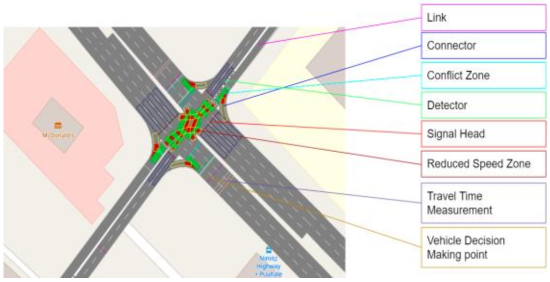


Figure 3. An intersection in VISSIM.

Once the VISSIM simulation is calibrated to accurately represent the original system, the data from it can be used to learn the system dynamics, and the learned model can facilitate the design of an adaptive learning control strategy [8]. Such an approach allows comprehensive simulation testing for obtaining closed-loop control before it is applied to the real system. Once closed-loop system simulation is desired, the modeling and control functionalities using VISSIM data can be directly switched onto the real system by allowing the modeling and control units to accept real system data. This approach applies the real-time implementation of the obtained modeling and control strategy. Thus, VISSIM simulation is a key stage in the modeling and implementable control design for actual traffic flow systems.

For the modeling purpose, we needed to do the following:

- Establish an effective data-driven modeling and learning algorithm for the system
- Use data from the VISSIM platform to train the model
- Compare the VISSIM-trained model with the model learned from the real system data to ensure that the VISSIM simulation produces results consistent with the results from the real system.

III. HNN Using Vissim Data

Because the system is unknown, nonlinear, and non-Gaussian, data-driven modeling (e.g., neural networks and fuzzy logic) would be a well-suited choice. For this purpose, an HNN modeling approach was developed, and a dynamic model was considered that reflects the relationship between the input and the output in Eq. (1). Moreover, to improve the model, traffic volume was also considered as an extra input. Thus, the system had two input vectors (i.e., signal time plan and traffic volume) and one output vector (i.e., traffic delays) [1].

The system model was therefore assumed as follows:

$$y(k+1) = Ay(k) + Bu(k) + f(y(k), u(k-1), v(k)) \quad (2)$$

where $y(k)$ and $u(k)$ denote average delay per vehicle and green light time for multiple intersections at time index k. $f(\ldots)$ is an unknown nonlinear vector function to be learned, and $\omega(k)$ is noise. $\{A, B\}$ are the weight matrices to be identified simultaneously with the estimate for the unknown nonlinear dynamics.

Let a neural network be used to approximate $f\big(y(k), u(k-1), v(k)\big)$ by $\hat{f}(y(k), u(k-1), v(k), \pi)$, where $v(k)$ denotes traffic volume; π groups all neural network weights and biases. Then, the neural network and the two matrices were trained to obtain accurate and reliable models for the traffic flow system. In this case, we considered seven intersections of the arterial shown in Figure 1, as indicated by the red rectangle in Figure 1.

The objective of training was to minimize the following performance function:

$$\min_{\pi} J = \frac{1}{2}\big(\hat{y}(k+1) - y(k+1)\big)^2 \quad (3)$$

which is essentially a minimum variance error criterion, where it has been defined that

$$\hat{y}(k+1) = Ay(k) + Bu(k) + \hat{f}(y(k), u(k-1), v(k), \pi) \quad (4)$$

and $\{A, B, \pi\}$ are parameters to be trained. In Eq. (4), vectors $\hat{y}(k)$ and $\hat{f}(\ldots)$ are the estimates of $y(k)$ and $\hat{f}(\ldots)$, respectively, using the collected data from VISSIM simulation platform.

A. Gradient rule for training

Using gradient optimization, the following recursive estimation and training algorithm can be readily obtained to read

$$\hat{A}(k+1) = \hat{A}(k) - \lambda_1 \frac{\partial J}{\partial A}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} \quad (5)$$

$$\hat{B}(k+1) = \hat{B}(k) - \lambda_2 \frac{\partial J}{\partial B}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} \quad (6)$$

$$\hat{\pi}(k+1) = \hat{\pi}(k) - \lambda_3 \frac{\partial J}{\partial \pi}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} \quad (7)$$

where λ_1, λ_2 and λ_3 are prespecified positive learning rates that are typically selected to be less than 1.0, and the gradients are calculated from

$$\frac{\partial J}{\partial A}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} = \big(\hat{y}(k+1) - y(k+1)\big)\frac{\partial \hat{y}}{\partial A}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} = (\hat{y}(k+1) - y(k+1))\, y(k) \quad (8)$$

$$\frac{\partial J}{\partial B}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} = \big(\hat{y}(k+1) - y(k+1)\big)\frac{\partial \hat{y}}{\partial B}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} = (\hat{y}(k+1) - y(k+1))\, u(k) \quad (9)$$

$$\frac{\partial J}{\partial \pi}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} = \big(\hat{y}(k+1) - y(k+1)\big)\frac{\partial \hat{f}}{\partial \pi}\Big|_{\left(\hat{A}(k), \hat{B}(k), \hat{\pi}(k)\right)} \quad (10)$$

where $y(k+1)$ is the data from the VISSIM simulation platform in the same way as the data used for the input.

B. Convergency consideration

The selections of the learning rates are also critical to ensure a good balance between the responsiveness of the learning and its stability in providing convergent neural network training. Using the second-order derivative analysis such as Jacobean matrices, one can obtain the ranges for these learning rates.

Denoting $\varphi = [A, B, \pi]$, the local optimal effect would require the following to be satisfied:

$$\frac{\partial^2 J}{\partial \varphi^2} > 0 \quad (11)$$

with the following guarantee:

$$\lim_{k \to +\infty} (\hat{y}(k) - y(k))^2 = 0 \quad (12)$$

These two conditions ensure that the modeling objective function in Eq. (3) monotonically decreases around the local minimum point, namely

$$J(k+1) - J(k) \approx -\tilde{\varphi}^T \frac{\partial^2 J}{\partial \varphi^2} \tilde{\varphi} \le 0 \quad (13)$$

where $\tilde{\varphi} = \varphi(k) - \varphi^*$, and φ^* represents a group of matrices and weights that ensure a local minimum of Eq. (3). It is therefore necessary to calculate Eq. (11) along with the progress of training Eqs. (5)–(7) to ensure that the learning is at least locally convergent—leading to a Gaussian-like PDF for the modeling errors. This is an additional computational load required during the learning phase described in Eqs. (5)–(7). By summarizing the learning represented in Eqs. (5)–(7) in a compact form, we have

$$\varphi(k+1) = \varphi(k) - \lambda \frac{\partial J}{\partial \varphi} \quad (14)$$

Then, the condition of Eq. (11) means that one needs to select the learning rate $\lambda > 0$ so that

$$0 < \lambda < \frac{2}{\lambda_{max}(\Gamma^T \Gamma)} \quad (15)$$

where Γ is an information matrix composed of all the past inputs and outputs used in the training, λ_{max} denotes the maximum eigenvalue of matrix $\Gamma^T \Gamma$, and λ denotes any of the three learning rates in Eqs. (5)–(7). In practice, once the learning rates are selected to be sufficiently small, the convergency guarantee can be generally realized.

For example, one can consider the following B-spline neural network to approximate the nonlinear function $f(\dots)$ in Eq. (2). This leads to the following model representation:

$$\hat{y}(k+1) = Ay(k) + Bu(k) + \sum_{i=1}^{n} w_i B_i(z(k))$$
$$z(k) = [y(k), u(k-1), v(k)]^T \qquad (16)$$

where w_i $(i = 1, 2, \dots, n)$ are the neural network weight vectors to be trained using the VISSIM system data from $z(k)$, and $B_i(z(k))$ are a set of prespecified basis functions defined on the functional space of $z(k)$ [40]. Then, Eq. (16) can be expressed as

$$\hat{y}(k+1) = \Gamma(k)\pi \qquad (17)$$

where the information matrix can be expressed as

$$\Gamma(k) = [y(k), u(k), B_1(z(k)), B_2(z(k)), \dots, B_n(z(k))] \qquad (18)$$

Therefore, the objective function in Eq. (3) can be expressed as

$$J = \frac{1}{2} \|y(k+1) - \Gamma(k)\pi\|^2 \qquad (19)$$

In this case, the gradient training in Eq. (14) can still be applied, which gives the convergence guarantee as shown in Eq. (15).

The training algorithm described in Eqs. (5)–(10) provides a set of simultaneous estimates for both linear parameters and neural network weights. Also, because the control input $u(k)$ to be designed is linearly involved in the model, the controller design using AI techniques can be easily implemented as a direct inverse calculation so long as the matrix B is of a full column rank. This approach effectively facilitates real-time implementation for the whole system.

C. *Data and their processing from Vissim simulation*

To model the system in Eq. (2), relevant data from the seven intersections in the VISSIM simulation platform were collected along the arterial as shown in Figure 2. The details of the data collected are summarized in the Table I.

TABLE I. DATA COLLECTION FOR HNN MODELING

Study area	Intersections 1–7
Dates collected	March 2–April 2, 2021
Time duration	4 p.m.–7 p.m.
Signal timing	All phases of major and minor streets
Traffic volume	All movements
Traffic delay	All movements
Sampling index	Every two signal cycles (each cycle ~180 s)

IV. MODELING RESULTS OF HNN USING VISSIM DATA

Before the HNN model was trained, the raw data from the VISSIM simulation platform were preprocessed to remove or reduce noise in the data, as shown in Figure 4. For traffic signal and traffic volume data, normalization was conducted to scale data between zero and one. For traffic delay data, after normalization, simple exponential smoothing was applied to further filter the data to remove noise, as shown in Eq. (20), where $l(k)$ is the filtered delay, $y(k)$ is the normalized delay, and α is the smoothing factor between zero and one. As α decreases, the observation of delay at k has a reduced effect on the output $l(k)$, indicating that the randomness of the delay measurements is reduced. After training of the HNN model, inverse normalization and inverse smoothing were applied to generate actual model output. This process is shown in Figure 4.

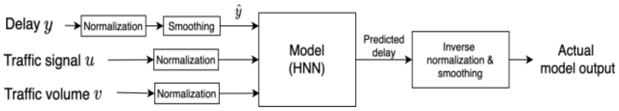


Figure 4. Data preprocessing.

$$l(k) = \alpha y(k) + (1-\alpha)l(k-1) \qquad (20)$$

The HNN model was trained by 80% of the total data points and was tested with the remaining 20% of total data. Figure 5 illustrates the HNN model structure applied in this study when VISSIM-generated data were used.

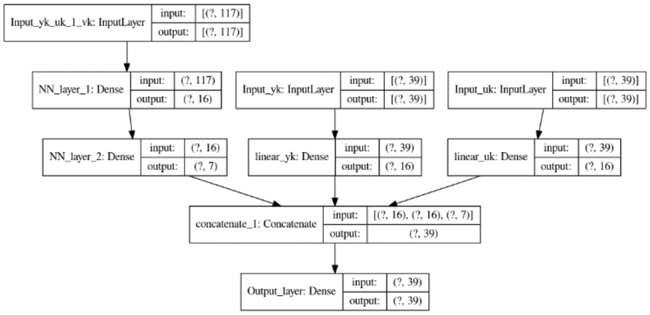


Figure 5. HNN model structure.

A. *VISSIM data based HNN modeling effect*

Similar to previous results [1], the modeling results were evaluated by mean absolute percentage error (*MAPE*), root mean square error (*RMSE*), and mean absolute error (*MAE*) as described in Eqs. (21)–(23), respectively, where $y_n(k)$ is the true delay at time *k* of phase *n*, and $\hat{y}_n(k)$ is the predicted delay at time *k* of phase *n*.

$$MAPE = \frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1}^{N}\left|\frac{y_n(k) - \hat{y}_n(k)}{y_n(k)}\right| \qquad (21)$$

$$RMSE = \frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1}^{N}\sqrt{(y_n(k) - \hat{y}_n(k))^2} \qquad (22)$$

$$MAE = \frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1N}|y_n(k) - \hat{y}_n(k)| \qquad (23)$$

Table II and Table III show the prediction results for all phases of all seven intersections, the phases of main streets and side streets, and the phase of each intersection. Note that delay prediction at main streets is more accurate than at side streets. The reason is that traffic volumes at side streets are much lower and more stochastic compared with main streets.

TABLE II. TRAINING AND TESTING RESULTS

	Training (all)	Testing (all)	Testing (main streets)	Testing (side streets)
MAPE	6.2%	6.3%	5.7%	6.5%
RMSE	8.5 s	8.4 s	1.9 s	10.3 s
MAE	5.6 s	5.7 s	1.5 s	8.1 s

TABLE III. TESTING RESULTS AT EACH INTERSECTION

Intersection	1	2	3	4	5	6	7
MAPE (%)	4.7	7.0	6.6	6.7	6.5	7.1	4.7
RMSE (s)	5.3	7.8	7.5	10.0	9.0	8.6	8.0
MAE (s)	3.8	5.4	5.2	7.2	6.1	5.6	5.7

Figure 6 and Figure 7 show the error and the distribution represented by the PDF of training errors. Training errors are roughly symmetrically distributed along the horizontal axis.

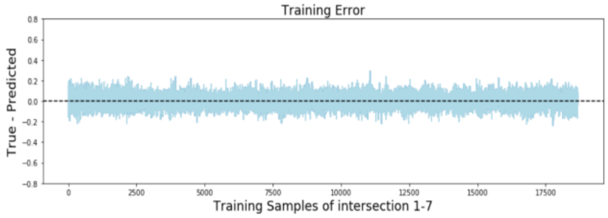


Figure 6. Training error.

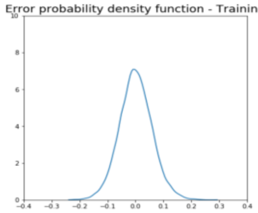


Figure 7. Training error PDF.

Figure 8 and Figure 9 show the error and the distribution of the PDF of testing errors. Such a shape of PDF for the modeling error is close to a narrowly distributed Gaussian PDF. This shape indicates that no further information in the modeling error is useful for the training, and thus, the training is complete. This is also applied to the PDF exhibition for the testing results of the training as shown in Figure 9.

Figure 10 shows comparisons of training of vehicle travel delay from the HNN model and the VISSIM-generated delay data of each phase at intersection 1. There are four phases at intersection 1.

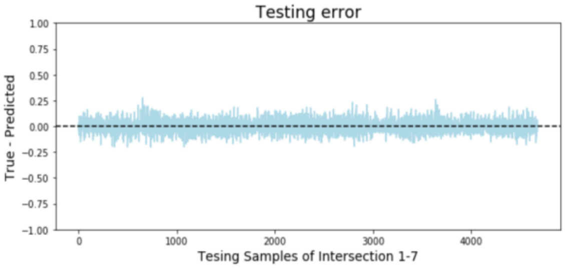


Figure 8. Testing error.

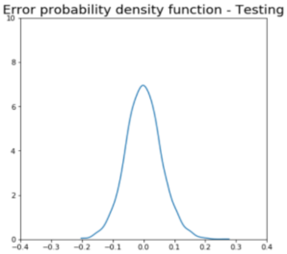
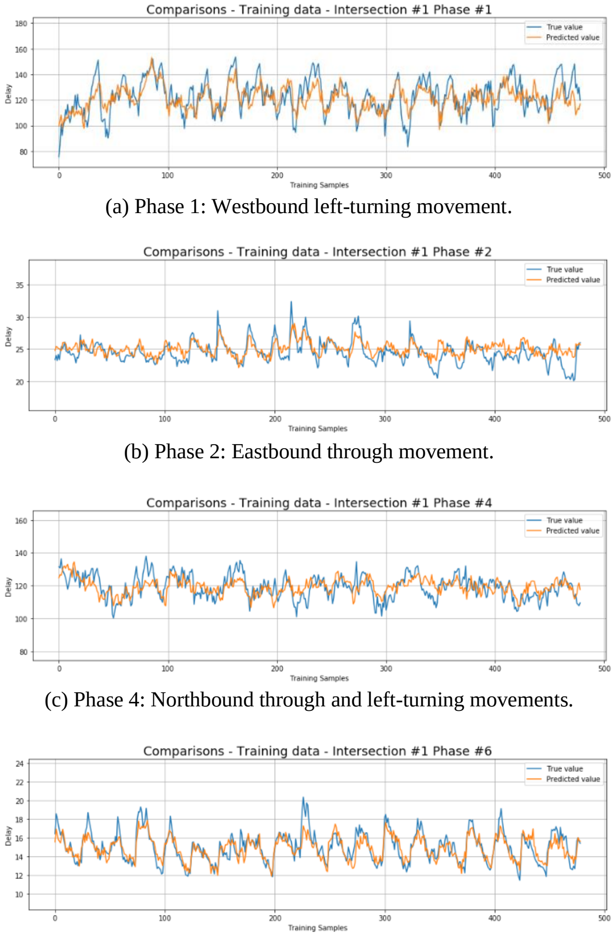


Figure 9. Testing error PDF.



(a) Phase 1: Westbound left-turning movement.

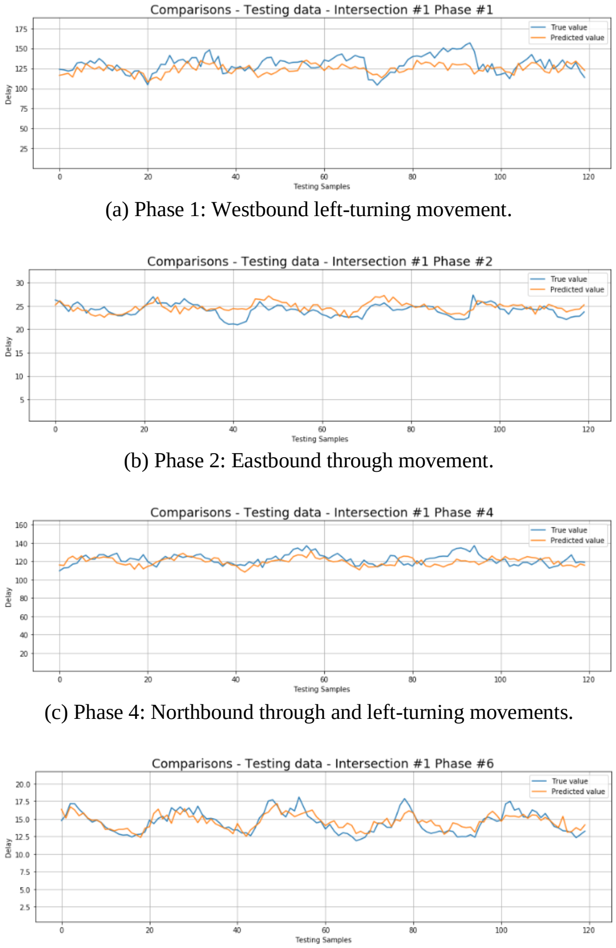
(b) Phase 2: Eastbound through movement.

(c) Phase 4: Northbound through and left-turning movements.

(d) Phase 6: Westbound through movement.

Figure 10. Delay comparisons for training data at intersection 1.

Figure 11 shows comparisons of predicted (testing) vehicle travel delay from the HNN model and the true delay of each phase at intersection 1 generated from the VISSIM simulation platform. Again, there are four phases at intersection 1. Figure 11 (a–d) shows the delay comparisons of each phase, respectively.



(a) Phase 1: Westbound left-turning movement.

(b) Phase 2: Eastbound through movement.

(c) Phase 4: Northbound through and left-turning movements.

(d) Phase 6: Westbound through movement.

Figure 11. Delay comparisons from testing data at intersection 1.

B. *Comparison with real data modeling effect*

To test the accuracy of the training using VISSIM-generated data, real system data for the same period of time as given in Table I were collected and used to train the same structured HNN as our conference paper [1]. The modeling errors comparison is shown in Figure 12, where blue indicates the modeling error trained using the real system data (i.e., the Econolite system), and red indicates the modeling error using VISSIM-generated data. These two errors are reasonably close to each other—showing the effectiveness of the obtained VISSIM simulation platform, although further calibration of such a microscopic traffic simulation is still needed.

The comparable errors in Figure 12 indirectly indicate the level of acceptance of the Vissim simulation platform. Because the differences between two group of errors are small, the Vissim simulation system, as a digital twin, can be regarded as a good representation of the original system dynamics.

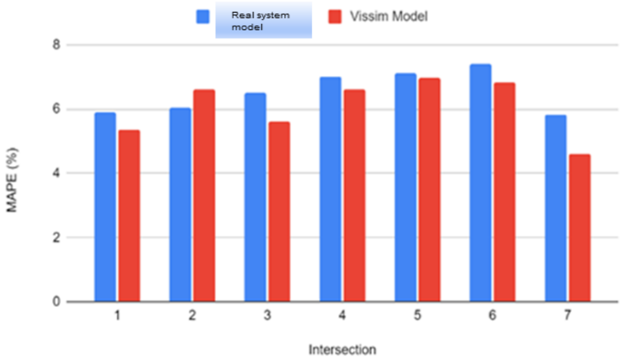


Figure 12. Comparison of Mean Absolute Percentage Errors—the use of Vissim data (red) vs. real system data (blue).

V. CONCLUSIONS

As an extension of our conference paper [1], this paper describes MIMO HNN modeling for multiple intersections along a corridor with a comparison between the modeling effects of real and simulated data. A VISSIM simulation platform is presented that was preliminarily calibrated using real system data so that real-time implementation can be realized in a simple, comprehensive way. The proposed HNN model can capture both the linear and nonlinear stochastic natures of multiple traffic features (i.e., traffic signal timings, traffic flows, and travel delays).

Both simulated and real data were used to train the HNN model, and the comparison between the modeling errors for each case were analyzed—showing a similar performance effect in terms of the modeling as shown in Figure 12. This similarity also indirectly indicates that the VISSIM simulation platform can reasonably represent the real system dynamics.

This study demonstrated a first step for the real-time implementation of AI-based transportation system modeling and control. For future work, we will continue to collect data from more intersections and further refine the HNN model. When the model is further developed, we will develop an AI-based optimal traffic control system based on the model to minimize entire system costs, including travel delay and energy consumption.

AI-based control design is required to establish a real-time closed-loop feedback control system that uses the traffic flow state as feedback [6][8]. This approach controls the signal timing intelligently at intersections so that the resulting traffic flow can be made smoother with minimized energy consumption. This control method requires controller design using AI techniques together with the VISSIM microscopic traffic simulation platform. Because of the random nature of traffic flow systems, stochastic optimal control in a multi-objective Bayesian framework will be investigated in the future.

Acknowledgments

The work performed is funded by the Vehicle Technologies Office of the US Department of Energy under the AI for Mobility program.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

[1] W. Li, C. Wang, Y. Shao, H. Wang, J. Ringler, G. Zhang, T. Ma, and D. Chou, "Hybrid neural network modeling for multiple intersections along signalized arterials - current situation and some new results," The Tenth International Conference on Advances in Vehicular Systems, Technologies and Applications, VEHICULAR 2021, July 18–22, 2021, Nice, France.

[2] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 3, pp. 261–272, 2006.

[3] G. N. Cadet, "Traffic signal control — a neural network approach," Florida International University, 1996.

[4] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 3, pp. 261–272, 2006.

[5] K. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," ACM Comput. Surv., vol. 50, no. 3, art. no. 34, 2017.

[6] H. Wang, S. Patil, H. Aziz, and S. Young, "Modeling and control using stochastic distribution control theory for intersection traffic flow," IEEE Transactions on Intelligent Transportation Systems, 2020 (accepted and online).

[7] A. Wang and H. Wang, "Stochastic Fault Detection," Encyclopaedia of Control Systems, 2020.

[8] A. Wang and H. Wang, "Decision-Making for Complex Systems Subjected to Uncertainties—A Probability Density Function Control Approach," Handbook of Reinforcement Learning and Control. Springer Verlag, 2020.

[9] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact," Transportation Research Record, vol. 2674, no. 4, pp. 459–470, 2020.

[10] H. Yuan and G. Li, "A survey of traffic prediction: from spatio-temporal data to intelligent transportation," Data Science and Engineering, vol. 6, no. 1, pp. 63–85, 2021.

[11] B. P. Williams, Durvasula, and D. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," Transp. Res. Rec. vol. 1644, no. 1, pp. 132–141, 1998.

[12] L. Zhang, Q. Liu, W. Yan, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," Procedia-Social and Behavioral Sciences, vol. 96, pp. 653–662, 2013.

[13] W. Li, J. Wang, R. Fan, Y. Zhang, Q. Guo, C. Siddique, and X. Ban, "Short-term traffic state prediction from latent structures: accuracy vs. efficiency," Transportation Research Part C: Emerging Technologies, vol. 111, pp. 72–90, 2020.

[14] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," Transporation Research Part C: Emerging Technologies, vol. 43, pp. 50–64, 2014.

[15] P. V. V. Theja and L. Vanajakshi, "Short term prediction of traffic parameters using support vector machines technique," 2010 3rd International Conference on Emerging Trends in Engineering and Technology, pp. 70–75, 2010.

[16] L. Vanajakshi and L. R. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," IEEE Intelligent Vehicles Symposium, vol. 2, pp. 194–199, 2004.

[17] Z. Cui, R. Ke, Z, Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," Transportation Research Part C: Emerging Technologies, vol. 118, 2020.

[18] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 11, pp. 4883–4894, 2019.

[19] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, "The scoot on-line traffic signal optimisation technique," Traffic Engineering & Control, vol. 23, no. 4, 1982.

[20] W. R. McShane and R. P Roess, "Traffic engineering." Prentice Hall, 1990.

[21] S. B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: a realistic simulation," Advances in Applied Self-Organizing Systems, pp. 45–55. Springer, 2013.

[22] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," Advances in Dynamic Network Modeling in Complex Transportation Systems, pp. 27–66. Springer New York, 2013.

[23] P. R. Lowrie, "SCATS, Sydney cooordinated adaptive traffic system: a traffic responsive method of controlling urban traffic," Roads & Traffic Authority Sydney, Australia, 1990.

[24] W. Wei and Y. Zhang, "FL-FN based traffic signal control," Proc. 2002 IEEE Int. Conf. Fuzzy Syst., May 2002, vol. 1, no. 12–17, pp. 296–300.

[25] E. Bingham, "Reinforcement learning in neural fuzzy traffic signal control," Euro. J. Operation Res., vol. 131, no. 2, pp. 232–241, 2001.

[26] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," IEEE Transactions on intelligent transportation systems, vol. 7, no. 3, 261–272, 2006.

[27] M. C. Choy, D. Srinivasan, and R. L. Cheu, “Neural networks for continuous online learning and control,” IEEE Transactions on Neural Networks, vol. 17, no. 6, 1511–1531, 2006.

[28] L. Li, L. Yisheng, and F. Wang, “Traffic signal timing via deep reinforcement learning,” IEEE/CAA Journal of Automatica Sinica. vol. 3, no. 3, pp. 247–254, 2016.

[29] H. Wei, G. Zheng, H. Yao, and Z. Li, “Intellilight: a reinforcement learning approach for intelligent traffic light control,” Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London, pp. 2496–2505, 2018.

[30] H. Wei et al., “Presslight: lLearning max pressure control to coordinate traffic signals in arterial network,” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1290–1298, 2019.

[31] C. Chen et al., “Toward A thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control,” Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, no. 4, pp. 3414–3421, 2020.

[32] H. Wei et al., “Colight: Learning network-level cooperation for traffic signal control,” Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1913–1922, 2019.

[33] T. Zhong, Z. Xu, and F. Zhou, “Probabilistic graph neural networks for traffic signal control,” 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4085–4089, 2021.

[34] M. Abdoos, N. Mozayani, and A. L. Bazzan, “Hierarchical control of traffic signals using Q-learning with tile coding,” Applied intelligence, vol. 40, no. 2, 201–213, 2014.

[35] P. Mannion, J. Duggan, and E. Howley, “An experimental review of reinforcement learning algorithms for adaptive traffic signal control,” Autonomic Road Transport Support Systems, 47–66, 2016.

[36] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, “Bridging the gap between value and policy based reinforcement learning,” NeurIPS, 2017.

[37] H. Wei, G. Zheng, G. Gayah, and Z. Li, “Recent advances in reinforcement learning for traffic signal control: a survey of models and evaluation,” ACM SIGKDD Explorations Newsletter, vol. 22, no. 2, 12–18, 2021.

[38] Vissim 9 User Manual, PTV Group, Karlsruhe, Germany, 2016.

[39] R. Wiedemann, “Simulation des Straßenverkehrsflusses,” Schriftenreihe Heft 8 (Instituts für Verkehrswesen der). Karlsruhe, Germany: Universitä Karlsruhe, 1974.

[40] M. Zhu, X. Wang, A. Tarko, and S. Fang, “Modeling car-following behavior on urban expressways in Shanghai: A naturalistic driving study,” Transp. Res. C, Emerg. Technol., vol. 93, pp. 425–445, 2018.

Data Collection Scheme using Erasure Code and Cooperative Communication for Deployment of Smart Cities in Information-centric Wireless Sensor Networks

Shintaro Mori
Department of Electronics Engineering and Computer Science
Fukuoka University
8-19-1, Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan
e-mail: smori@fukuoka-u.ac.jp

Abstract—**This paper proposes an improvement on a scheme previously proposed by the author for sensing-data collection and management in future smart-city applications. The proposed scheme uses an information-centric network and unmanned aerial vehicles. However, there are two issues in implementing this scheme: consideration of the fragmentation scheme for large content data and wireless network technology to deploy a massive amount of sensor nodes. To tackle these issues, the proposed scheme uses an erasure-code technique and transmitter-side cooperation with dual-band node devices. The feasibility and network performance of the scheme was evaluated using a hardware-based experiment. The numerical results indicate that the scheme can improve data caching by 29.3% in the deployment of future wireless sensor networks.**

Keywords—Information-centric network (ICN); Wireless sensor network (WSN); Erasure code-encoded data fragmentation; Cooperative and dual-band node

I. Introduction

Metropolitan areas are increasing in population density due to the movement of people from the surrounding rural areas, which raises a variety of social problems. Generally, smart cities bring intelligence to various aspects of our daily lives for rapid urbanization by using Internet of Things (IoT) technologies. To facilitate decision-making and task execution in current cities, massive resources, such as sensors, actuators, and data storage, need to be deployed to maintain the sustainability of extensive social applications. Those promises have been recognized as representative of the IoT and feature a diverse array of cyber-physical systems [2]. We believe that smart cities alternatively include not only urban sophistication but also resilience against natural disasters, especially, in regional cities. In other words, the following two scenarios, smart cities during normal situations and disaster-resilient networks when a disaster occurs, have been separately studied, but this paper suggests that it is reasonable to consider both scenarios at the same time rather than individually.

This paper focuses on river monitoring as a scenario that integrates smart cities and disaster-resilient networks. The motivation is that river flooding due to heavy rains, typhoons, and backflow can seriously affect our lives. In a previous study [3], a system for the above scenario was developed to record and analyze river surface, water level, direction, and velocity. Through that research, it was found that effective sensing-data collection and management for disaster-resilient smart cities are required. In another study by the author [1], an overall blueprint was provided of a novel data-collection and management scheme for wireless sensor networks (WSNs) in which adopted two key technologies were used: an information-centric network (ICN) design [4] and technique for assisted data collection of unmanned aerial vehicles (UAVs) [5].

Regarding ICN design adoption, in conventional IoT-based frameworks, IoT devices are directly linked to cloud servers to gather and centralize sensing data via hyper-text transfer protocol/transmission control protocol/internet protocol (HTTP/TCP/IP)-enabled application programming interfaces, such as message queueing telemetry transport protocol and RESTful interface. Typical location-dependent common interfaces are suitable for coordinating across multiple systems in distributed wireless networks in terms of interoperability and heterogeneity of different providers, specific data formats, and unique protocol specifications. However, in the traditional architecture, heavy address-based queries cause serious protocol overhead, making them similar to denial-of-service (DoS) attacks, in which there lies a potential bottleneck. For the above scenario, ICNs name content data instead of the "address," and the ICN nodes copy and store the named data as caching data for further responses. A subscriber broadcasts a request (an interest packet) containing information about the desired data, and a neighbor node (publisher) that owns the data responds to the subscriber's request. Therefore, an ICN has a feature of location-independent network architecture, which is essential to introduce to the scenario for this study.

Regarding UAV assistance, practical sensor nodes (SNs) are non-uniformly scattered depending on the ground surface, cost-effectiveness, and supply, i.e., the sensing data are periodically generated but must be collected at asynchronous intervals. For the deployment of SNs, the demand is rapidly expanding not only in urban areas but also in suburban areas. For collecting and forwarding data in these scenarios, UAVs, such as drones (including multi-copters), small planes, and balloons, can work more flexibly and robustly as mobile sink nodes, which play an essential role in air-ground integration networks [6]. UAVs are assigned to collect sensing data in the observation area and act as relay nodes in flying ad-hoc networks. Consequently, ICN design and UAV assistance are suitable use-cases to address dynamic network reconstruction due to WSN-node failures for disaster-resilient smart cities.

There are two issues in implementing the previous scheme. One is that it is necessary to consider not only small-scale data but also multimedia content, such as images and videos, in smart-city applications. For example, in river-monitoring systems, if we only collect specific sensing data (atomic data), such as water surface level, we can select typical WSN systems, but if we observe river conditions on the basis of visual data, the recorded videos will contain large amounts of data. Therefore, such data should be fragmented to transfer wirelessly. The other issue was found during a computer simulation of the above condition, i.e., the previous scheme could not accommodate traditional WSN systems for typical fourth-generation (4G) and fifth-generation (5G) SN deployment scenarios because of the enormous amount of sensing-data traffic due to massive SNs [7]. The situation is more serious when we assume the management of multimedia data. To tackle the above issues, a sophisticated channel-access mechanism and efficient radio-bandwidth utilization technique should be considered.

For the previous medium access control (MAC) and physical protocol scheme, a fundamental evaluation was conducted through computer simulation [1]. This scheme uses erasure-code-encoded sub-frames since the original packet can be restored even if all the sub-frames are not complete. Therefore, retransmission procedures are not necessary, and the packets can be recovered by fetching the lost sub-frames from the neighbor nodes. This scheme also forwards the frames using transmitter-side cooperation and a dual-band node to address the technical issues regarding channel capacity.

In this study, this scheme was improved hereafter, the proposed scheme with the following contributions:

- The feasibility of IC-WSNs for smart-city applications, such as a river-monitoring system, was evaluated. For the proposed scheme, testbed devices were developed, and the system's data retrieval can provide compatibility with traditional HTTP-based users. The scheme's network performance was evaluated through a hardware-based experiment.
- For large-scale data, such as visual images and videos, fragmentation was considered, i.e., the full-frame must be divided into several sub-frames. In this case, the proposed scheme can completely recover the data by using the erasure codes for the data, even if some part of the sub-frames does not arrive at the receiver.
- The proposed scheme introduces transmitter-side cooperation and dual-band communication equipment. Therefore, it can adapt to an SN-distribution scenario in a 4G environment, which cannot be accommodated using the conventional low-power wide-area (LPWA) networks. The improvement of the scheme is evaluated using computer simulation. Furthermore, this dual-band communication strategy is suitable for the deployment of smart-city applications.

The remainder of the paper is organized as follows. In Section II, related work is discussed. In Section III, the proposed scheme is introduced. In Section IV, the hardware-based experiment and results are presented. In Section V, the computer simulation and results are presented. Finally, the findings and conclusions are given in Section IV.

II. Related Work

Related studies in UAV-IC-WSNs have investigated several elemental technologies. For example, Bithas et al. [8] investigated channel modeling to satisfy massive connectivity and ultra-reliability requirements. Li et al. [9] investigated the upper limitation of carrier sense multiple access with collision avoidance-based MAC protocols. Bouhamed et al. [10] found that MAC protocols have flight-path controls and trajectory optimization for UAV swarms, e.g., an adaptation of machine-learning techniques. As observed from the above studies, wireless connectivity has been investigated elementally, including antenna design and interference cancellation.

Investigation of UAV-IC-WSN's MAC protocols and physical protocols have remaining research problems [11]. To the best of my knowledge, the cooperative MAC protocols can be basically classified as being either receiver-side or transmitter-side cooperation schemes. In a previous study by the author [12], a type of cooperative MAC protocol design was investigated to remove interference among SNs, which is categorized as a cooperative sensing-data-collecting framework [11]. Receiver-side cooperation is suitable for wireless networks to maximize their network lifetime because the rich receiver-side station nodes undertake complicated cooperative procedures. In fact, the 5G and beyond (B5G) wireless network systems use UAVs as airborne base stations, and UAV swarms provide integrated receiver-side cooperative reception [13]. However, cooperation at receivers is not sufficient to provide large sensing-data transmissions. To tackle this issue, the proposed scheme combines sensing, forwarding, and storing, which includes transmitter-side cooperation.

To improve throughput and robustness, a data-encoding method has been investigated for ICNs. As an initial challenge, Montpetit et al. [14] studied an introduction of the network coding technique [15] for effective data transfer. In order to improve the network throughput, Malik et al. [16] achieve to combine the three techniques, such as ICN caching, multipath forwarding, and network coding. Unlike other schemes, the proposed scheme uses erasure code for data-coding method to fit into frame segmentation and cooperative communication. This method has been studied for distributed storage reliability [17] in order to build a reliable distributed data storage system. For example, Kishani et al. [18] investigated the redundant array of independent disks. Several studies have applied this method to network research, e.g., Sharma et al. [19] used it to develop multipath diversity-based packet loss-tolerant network systems.

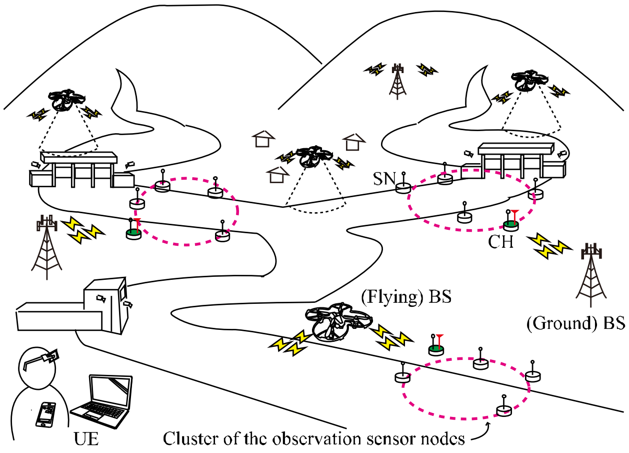


Figure 1. Target scenario of proposed scheme

III. Proposed Scheme

This section presents a network model of the proposed scheme, and the scheme's protocol stack, including the ICN, MAC, and physical layers.

A. *Network model of the proposed IC-WSNs*

In the proposed scheme, the wireless network consists of sensor nodes (SNs), cluster head nodes (CHs), base stations (BSs), and user (terminal) equipment (UEs), as shown in Figure 1. The SNs are scattered on the ground in the smart-city area and observe and store the sensing data. Several SNs are divided into groups, which correspond to the small monitoring areas, and a CH is deployed for individual clusters. CHs coordinate the SNs that are located in a CH's coverage area and act as relay (gateway) nodes between BSs and SNs. Typically, SNs and CHs are considered member nodes of WSNs, and it is assumed that SNs are low-cost and resource-limited hardware; whereas, CHs provide multi-functional capabilities with high-performance hardware. The proposed scheme uses a combination of these, i.e., not unifying all nodes as SNs, because certain transactions cannot be conducted in an SN, but the communication traffic will increase if such transaction transfer takes place on the cloud server. BSs act as brokers between local WSNs and outside networks. It is assumed that BSs are assigned not only as (traditional) fixed station nodes deployed on the ground but also as UAV-based BSs in the sky. With this layout, the scheme can reasonably provide wide-area coverage in smart cities.

For the collection and analysis of sensing data, the data are transmitted to the cloud server in traditional IoT-based schemes and analyzed and stored. The difference between the proposed scheme and these schemes is that the data are not collected uniformly but as needed. Namely, the SNs observe and store the data in their cache memories, and the UEs or CHs request the data if needed. Introducing IC-WSN methodology into the proposed scheme does not always send all data, which contributes to the long lifetime of the SNs. For this study, WSNs were designed on the basis of ICNs instead of IP networks. Since the current scheme does not always support IC-WSNs, the proposed scheme is designed with compatibility considerations, which is described in Section IV.

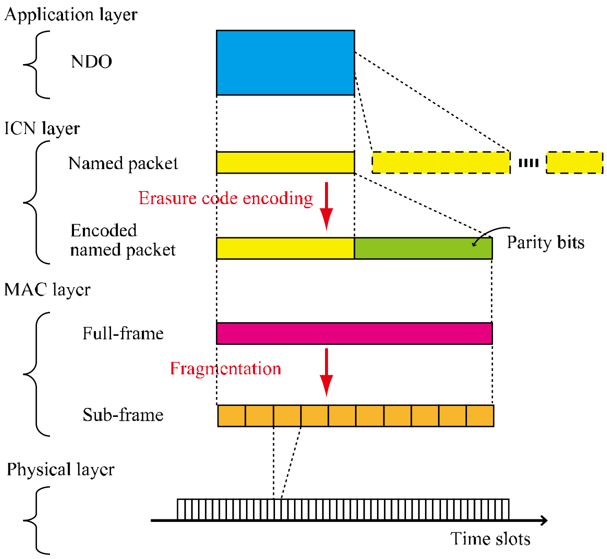


Figure 2. Structure of protocol-data unit in propsoed scheme

B. *Structure of the protocol stacks and protocol data unit*

The protocol stack of the proposed scheme consists of four layers: application layer, ICN layer, MAC layer, and physical layer. Note that if we implement the system on IP networks, we can optionally insert a TCP/IP or UDP/IP layer between the ICN and MAC layers. Figure 2 shows the structure of the protocol data unit in each layer. In the application layer, the data is logically organized as content data depending on individual application services. In IC-WSN systems, the content data are treated as a Named Data Object (NDO) with characteristics and attributes like tags. When an NDO is a large-size data, such as video and photograph data, the NDO is divided into several named packets at the ICN layer. For a named packet, the full-frame is structured in the MAC layer, and the full-frame is divided into several sub-frames. The length of the sub-frames is decided depending on the wireless communication method. Therefore, in the system, data fragmentation occurs in two patterns: the case that an NDO is divided into named packets and the case that a full-frame is divided into sub-frames. In particular, we focus on the latter case and concentrate on efficient wireless transfer technology since the former fragmentation is typically addressed in the ICN platforms.

C. *Fragmentation of named data packets using LDPC-based erasure code*

In a simple frame-by-frame transfer, the original packet cannot be recovered unless all sub-frames correctly arrive. When frame loss occurs in typical networks due to poor communication-channel conditions, it can be recovered using error control coding or re-transmitted on the basis of an automatic repeat request process. The proposed scheme, however, encodes the named packet using erasure code; therefore, the packet can be recovered even if some sub-frames are lost. Since a specified sub-frame is not required to recover the packet with this mechanism, i.e., if any sub-frame can replace the packet to recover it, any sub-frame that helps to recover the packet is more likely to be stored in the neighbor nodes. As a result, the proposed scheme reduces retransmission procedures and ensures low energy consumption. We can select erasure codes with strong resistance against burst bit errors, such as the low-density parity-check (LDPC) and Reed-Solomon codes. This is because the packets with any lost sub-frames have continuous bit errors in the sector of the lost sub-frames.

The LDPC code and Reed-Solomon code are both types of linear block code; in particular, the LDPC code nears the Shannon limit if the code length is sufficiently long [20]. Therefore, we use the LDPC code as erasure code. The LDPC code generally has a feature in the parity-check matrix that is used for the decoding process at the receiver side. Let $\boldsymbol{H}$ denote the M-by-N parity check matrix, where M and N denote the parity length and codeword length (i.e., the size of encoded named packet), respectively. This matrix is a sparse matrix consisting of many zeroes and few ones. Let $\boldsymbol{G}$ denote the K-by-M generator matrix, where K ($\triangleq N-M$) denote the plain text length (i.e., the size of the named packet). In particular, $\boldsymbol{H}$ and $\boldsymbol{G}$ have the relationship $\boldsymbol{H}\,\boldsymbol{G}^{\mathrm{T}}=\boldsymbol{0}$, and the operator of $^{\mathrm{T}}$ means the transposed matrix.

On the transmitter side, let $\boldsymbol{c}=[c_1,c_2,\cdots,c_K]$ ($c_k\in\{0,1\},k=1,2,\cdots,K$) denote the named packet and $\boldsymbol{p}=[p_1,p_2,\cdots,p_M]$ ($p_m\in\{0,1\},m=1,2,\cdots,M$) denote the parity bit, respectively, the encoded packet of $\boldsymbol{x}$ consists of these, which is given by

$$\boldsymbol{x}=[\,\boldsymbol{c}\,\boldsymbol{p}]=\boldsymbol{c}\,\boldsymbol{G} \tag{1}$$

where the equations are satisfied on the Galois Fields (2). At the MAC layer, the full-frame encapsulates the encoded named packet and divides it into L sub-frames in accordance with the time slot of the physical layer. Assuming the protocol header is ignored for simplicity of explanation, the sequence of sub-frames is given by $\boldsymbol{x}=[\boldsymbol{x}_1\;\boldsymbol{x}_2\cdots\boldsymbol{x}_L]$ and the margin of the final sub-frame is filled with the zero-padding formula.

At the receiver side, we define the received sequence of $\boldsymbol{y}$ corresponding to the transmitted sequence of $\boldsymbol{x}$. For $y_\ell(\ell=1,2,\cdots L)$, if it is received correctly, the sequence of y_ℓ will be the same as x_ℓ; otherwise, some of the bits in y_ℓ differ from x_ℓ due to bit errors over wireless transmission. If a sub-frame is entirely lost, y_ℓ will be set as a random binary sequence that is not related to x_ℓ. Even if some sub-frames are organized in the above manner, they will be correctable due to the capability of the powerful error correction. In the decoding process, since $\mathbf{H}$ has the following fundamental feature: $\boldsymbol{x}\,\mathbf{H}^{\mathrm{T}}=\boldsymbol{0}$, for code error detection and correction, $\boldsymbol{s}=\boldsymbol{y}\,\mathbf{H}^{\mathrm{T}}$ gives the syndrome of $\boldsymbol{y}$. If $\boldsymbol{e}$ denotes the error pattern in $\boldsymbol{y}$, $\boldsymbol{e}$ and $\boldsymbol{s}$ satisfy $\boldsymbol{s}=\boldsymbol{e}\,\mathbf{H}^{\mathrm{T}}$; thus, we can recover the original $\boldsymbol{x}$ from $\boldsymbol{y}$ by using $\boldsymbol{e}$. In principle, the receiver-side procedure of the LDPC code has been described above, and several practical decoding algorithms have been developed, such as bit-flipping decoding, a posteriori probability decoding, and iterative decoding based on belief propagation, which is commonly known as the sum-product algorithm [20].

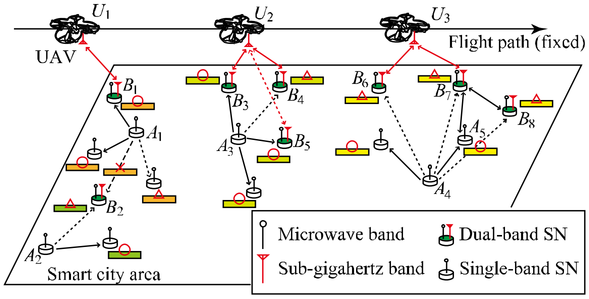


Figure 3. Cooperative communication scenario

D. *Wireless node devices with dual-band and cooperative communication for frame forwarding*

In the wireless air interface, the proposed scheme uses and switches to two radio-frequency bands: microwave and sub-gigahertz. Higher-frequency radio generally leads to larger data capacity and strong straightness (low diffraction). On the basis of the above feature, the proposed scheme assigns the radio frequency of the microwave band for sufficient bandwidth to forward relatively large amounts of data and the radio frequency of the sub-gigahertz band for long-distance data transmissions. It assigns the microwave and sub-gigahertz bands for the wireless transmission areas between an SN and CH and between a CH and BS, respectively. Therefore, CHs have two air interfaces (note that multi-band wireless communication modules have been utilized and adopted in several studies [21]), and we believe that it is realistic to equip CHs with the capability to hold such wireless modules.

A wireless communication system can generally overhear what neighbor nodes can receive whether they allow it or not. In the proposed scheme, to accelerate the effect of caching processing, the nodes should actively accumulate the overheard data, making it the so-called off-path caching mechanism. For example, in Figure 3, $\mathbb{A}_1$'s data should be cached in not only $\mathbb{B}_1$ but also in neighboring SNs. However, if $\mathbb{A}_2$'s data are sent at the same time as $\mathbb{A}_2$'s, the data will interfere with each other. Regardless of the circumstances, $\mathbb{B}_2$ should be caching some of $\mathbb{A}_2$'s data as imperfect frames.

To select the dual-band SN to which the UAV gives a transmission request, the UAV first broadcasts the interest packets to the area where the desired data might be located. The details of the data-retrieval procedure are described in the next section. If one node responds to the request, the UAV can accept or deny it, e.g., $\mathbb{U}_1$ selects $\mathbb{B}_1$. However, if there are several candidate SNs, the UAV can select the SN with the best wireless condition obtained using the signal strength of the responding packet among the dual-band SNs that have a perfect full-frame, e.g., $\mathbb{U}_2$ selects $\mathbb{B}_3$ among $\mathbb{B}_3$, $\mathbb{B}_4$, and $\mathbb{B}_5$. Moreover, if the candidate SNs have only imperfect data, the UAV attempts to combine and restore the data, e.g., $\mathbb{U}_3$ selects and recovers both $\mathbb{B}_6$ and $\mathbb{B}_7$. It is assumed that the wireless connection between the UAV and dual-band SNs is one hop because current sub-gigahertz wireless systems are typically single hop with the end devices connected to a central gateway through a direct link. However, we believe that further packet loss can be improved if multiple hops are acceptable, and this is for future work.

In accordance with the data wirelessly transmitted over IC-WSNs, there are control-plane data, such as interest packets, and user-plane data, such as sensing data. Sensing data can be classified into raw data and analysis data that SNs observe and obtain and CHs collect, analyze, and summarize, respectively. For example, in a river-monitoring system in smart cities [3], an individual SN captures the visual image data as raw data. The CH calculates the water-flow direction and velocity on the basis of those image data as analyzed data. Since SNs are resource-limited in large quantities at a low cost, we cannot alternate the analysis with CHs, even if we have to pay the cost of wireless transmission in terms of energy consumption. In addition, the data size of analysis data is smaller than that of raw data when the raw data are visual images, and the analysis data are text-based.

E. Procedure of data retrieval and data aggregation

In this section, the procedure each node follows when users request their required data or when the CH periodically conducts data analysis from the SNs is described. It is assumed that an SN periodically obtains sensing data (raw data) and stores them in its cache memory. Figure 4 shows the data-retrieval procedure of the proposed scheme. Figure 4 (a) represents the case in which all network nodes, such as UEs, BSs, CHs, and SNs, are constructed on the basis of an ICN platform. Figure 4 (b) represents the case in which the UE cannot support ICN technology.

As shown in Figure 4 (a), when a UE sends an interest packet to its neighbor BS, the BS forwards the interest packet to the CH that might have the required data. In the same manner, the CH further forwards the interest packet to an SN. The SN then sends back the required data, which are forwarded via the reverse route to the UE. If the BS and CH have the data that UE requests, such as copied raw data and analysis data, they reply with the data in their cache memories. As a result, the proposed scheme can shorten the data-retrieval time and reduce network traffic due to interest packet transfer. To implement the above steps, the BS and CH should have the appropriate forwarding information base (FIB) settings toward the next node related to the naming rule of the named packet. In addition, the BS, CH, and SN should automatically record the pending interest table (PIT) information in their cache memories to send data back to the requesting node. The evaluation on the feasibility of the proposed scheme using an ICN platform that enables these fundamental functionalities is discussed in Section IV.

When a UE cannot introduce an ICN-based protocol suite, as shown in Figure 4 (b), the proposed scheme solves this technical issue using broker enhancement to exchange between the conventional HTTP-based transaction and ICN-based transaction. In the scheme, the broker is introduced into a BS using the common gateway interface (CGI) process on the world wide web server. Specifically, the UE submits an HTTP-based request for the CGI process via the HTML-based website, then the BS triggers it to send an interest packet to the IC-WSNs. To prevent a huge amount of HTTP-based request, such as DoS attacks, the proposed scheme conducts the intermediary operation in two steps, i.e., acceptance of the HTTP-based request and retrieval of the data among IC-WSNs. After the data can be obtained in the same manner as in Figure 4 (a), a unique uniform resource identifier (URI) is assigned to the data to allow the UE to access the data via the traditional HTTP procedure. To achieve this, the proposed scheme uses two types of databases: one for the published URI and the other for requesting data. The database of the publish URI manages the URI that the BS published in its cache memory, and the database for requesting data records the request from UEs. This enables us to avoid duplicated interest and URI publishing work individually, even if several UEs request the same content.

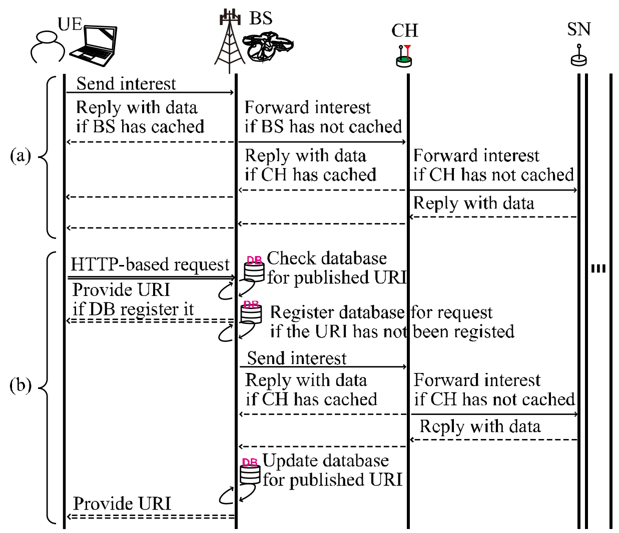


Figure 4. Data-retrival procedure, (a) all nodes are built on IC-WSNs, (b) UEs are built on HTTP-based networks, where BSs work as brokers between traditional networks and IC-WSNs

F. Physical protocol of proposed scheme

The signal processing of the proposed scheme is illustrated in Figure 5. As shown in Figure 5 (a), the full-frame is constructed at the erasure-code encoder by appending the parity bits calculated on the basis of the named packet, then the full-frame is divided into several sub-frames at the fragmentor. Each sub-frame is encoded using an error control coding, such as the convolutional code, for error detection and correction through wireless links. The codewords are then mapped into the analog signals using a modulator, such as the binary-phase-shift-keying method. As the multiple access mechanism, we use a slotted-Aloha method that is one of the

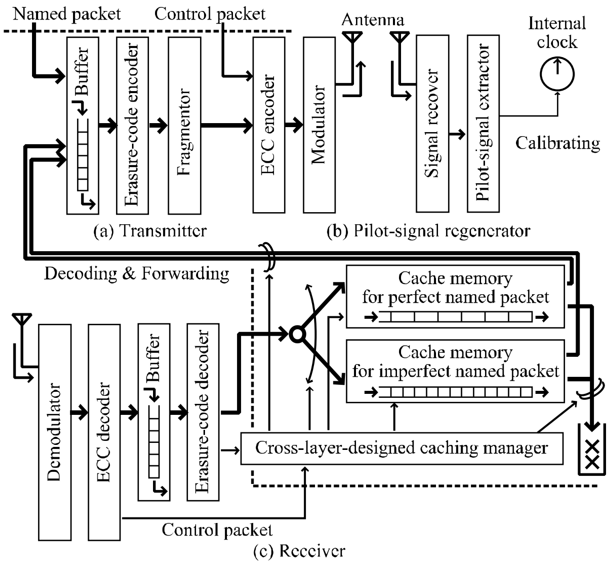


Figure 5. Procesure of proposed scheme

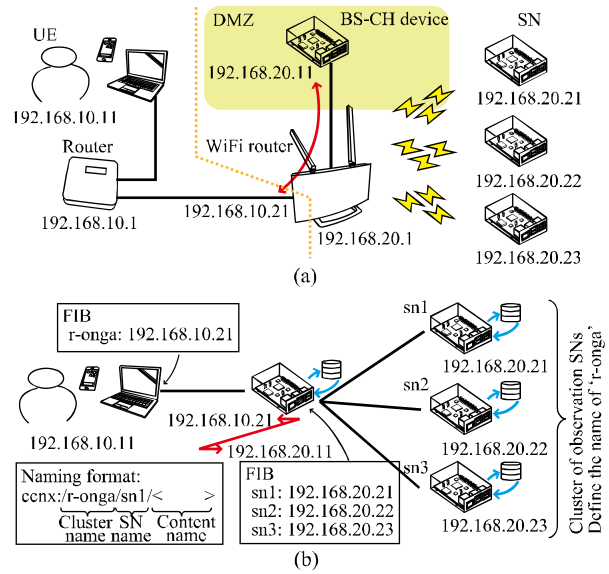


Figure 6. Network model of experimental environment, (a) network conections in TCP/IP layer, (b) relationship among nodes, FIB settings, and naming format in ICN layer

familiar random-access protocols using only collision-free slots. This method requires synchronization between the transmitter and receiver sides, thus the synchronization signals are obtained from the UAVs using the pilot-signal regenerator, as shown in Figure 5 (b).

At the receiver side, as shown in Figure 5 (c), the received signal is demodulated and interpreted using a method such as Viterbi decoding. The correctly received sub-frames are stacked into a temporary buffer, and the erasure-code decoder attempts to recover the original packet using sufficient sub-frames in the temporary buffer. If the restoring process is completed, the recovered packet is stacked in the cache memory for the perfectly named packet; otherwise, the failed packet is stacked in another cache memory for the imperfectly named packet. Therefore, the packets stored in those cache memories could be re-transmitted when the cooperative packet/frame transmissions are requested by other SNs and when the request is accepted. The proposed scheme requires collaboration beyond the boundaries among the lower three layers; thus, we believe that the caching manager must be created on the basis of cross-layer design [22].

IV. Hardware-based Experiments

The application and ICN layers, and the MAC and physical layers, were separately investigated to evaluate the effectiveness of the proposed scheme. This section focuses on the application and ICN layers, specifically, the feasibility and network performance of the proposed scheme through hardware-based experiments.

A. Layout and settings of experimental network

The network of the experiment contained a UE, BS-CH device, and SN, as shown in Figure 6 (a). Since the BS-CH device had fundamentally common functionalities in the experiment, we considered them to be the same device to simplify the testbed implementation. The network nodes belong to either of two networks, the UE-BS-CH network (192.169.10.0/24) or the BS-CH-SN network (192.169.20.0/24). The former is assumed to be the core network, and the latter is assumed to be a WSN. These two networks are separated by two routers. The network nodes are given a static IP address, as shown in Figure 6 (a). Since the BS-CH device should work as a gateway for both networks, it is placed in the demilitarized zone of the router and exchanges IP packets between the UE-BS-CH and BS-CH-SN networks. Two routers, that with the UE and that with the BS/CH device, are connected via the wired local area network (LAN). The SNs are connected to the Wi-Fi router via a wireless LAN on the basis of the IEEE 802.11 b/g standard due to Japan's regulations in outdoor environments.

To implement the proposed scheme, the Cefore platform [23] is used to provide the fundamental ICN architecture. As shown in Figure 6 (b), an ICN-based network topology is established by stacking (overlaying) the ICN layer on the TCP/IP layer. In a testbed device, the FIB and PIT are managed by the cefnetd daemon process that runs in the background middleware and can enable ICN-based data transfer. The FIB settings are statically assigned before the configuration of the cefnetd daemon process and the naming rules are assumed organized on the basis of a hierarchical structure separated by the cluster name, SN name, and content name, as shown in Figure 6 (b). The caching data are managed in each physical memory via the csmgrd deamon process. Note that this process cooperates with the cefnetd daemon process and fulfills data storing, updating, and deleting on the basis of the first-in-first-out formula and data freshness.

```
01: #!/usr/bin/perl
02: $sn = "sn1";
03: @tm = localtime;
04: $fn = sprintf("%04d%02d%02d%02d%02d",
05:   $tm[5]+1900, $tm[4]+1, $tm[3], $tm[2], $tm[1]);
06: `"./get_raw_data ".$fn`;
07: `"cefputfile ccnx:/r-onga/".$sn."/".$fn`;
```

(a)

```
01: #!/usr/bin/perl
02: $flag=0;
03: open(LIST, "<./list.db");
04: while($list=<LIST>) {
05:   if($list =~ m/$data{'addr'}/){ $flag=1; break; }
06: }
07: close(LIST);
08: if($flag==0) {
09:   open(REQUEST, ">>./request.db");
10:   print REQUEST $data{'addr'}."\n";
11:   close(REQUEST);
12: }
13: $uri=$data{'addr'}; $uri =~ s/\//_/g;
14: print "Content-type: text/html\n\n";
15: print "<html>\n<head></head>\n<body>\n";
16: print "<a href=\"./".$uri."\">$data{'addr'}</a>\n";
17: print "</body>\n</html>\n";
```

(b)

```
01: #!/usr/bin/perl
02: open(REQUEST, "<./request.db"); @list=<REQUEST>
03: close(REQUEST); unlink("./request.db");
04: foreach(@list) {
05:   $uri = $_; $uri =~ s/\//_/g;
06:   `cefgetfile ccnx:/$_ -f ./$uri`;
07:   open(LIST, ">>./list.db");
08:   print LIST $_."\n";
09:   close(LIST);
10: }
```

(c)

Figure 7. The implemented program code for testbed devices, (a) SN obtains and publishes the sensing data for IC-WSNs via the Cefore platform, (b) BS-CH device accepts HTTP-based requests from UE, and publish URI for data, and (c) BS-CH device retrives data from IC-WSNs on the basis of accepted requests

B. Development of BS-CH device and SN

A testbed of an SN and BS-CH device was developed using Raspberry Pi 4 with the default Raspbian OS. The river-monitoring system used in the author's previous study [3] was considered as the experimental system. Thus, the system captures image files taken by the USB camera on an SN as sensing data. On the basis of the procedure mentioned in Section III E, we implement the control programs using the Perl language as shown in Figure 7. In Figure 7 (a), the SN obtains sensing data by executing the external software of 'get_raw_data' in line 6, the implementation of which was described in our previous study [3]. The name for the data is given on the basis of the naming rule, as shown in Figure 6, and the data is stored in the cache memory through the csmngrd daemon process in line 7.

Figure 7 (b) shows the acceptance process when the UE sends a request to the BS in the UE-BS section in Figure 4 (b). The software is placed as a CGI on the WWW server (Apache) in the BS-CH device and is carried out by calling using the POST method that is a type of HTTP-based request message sent from client to server through another HTML-based website. The ccnx-based address is assumed stored in the variable of $data('addr'), and the database is established on the basis of a standard filesystem to avoid complexity. In Figure 7 (b), the duplicate URIs are identified in lines 3–7, a new interest packet is requested to record the database in lines 8–12, and the URI is published for all content data to provide the UE in lines 13–17. Figure 7 (c) shows the BS-CH device requesting the interest packet from the SN on the basis of the database. The BS-CH device organizes the list of interest requests in lines 5–6, the ccnx-based interest packet is broadcasted among IC-WSNs via the Cefore daemon process in lines 6, and the information of the received data is recorded in the database in lines 7–9.

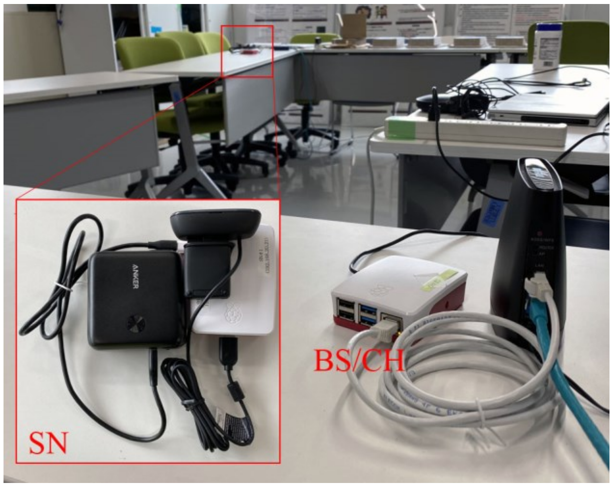


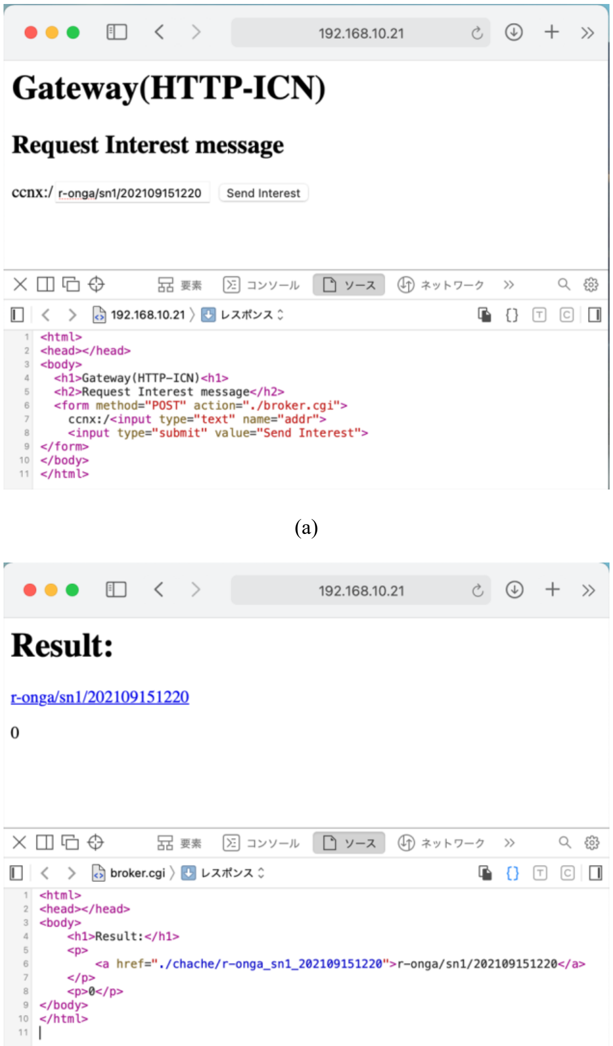
Figure 8. Indoor experimental testing environment

C. Demonstration

The testbed of the SN and BS-CH device is located on the desk, as shown in Figure 8. The experiment was conducted in indoor laboratory space, and the SN was 4 m from the Wi-Fi router. Although not shown in the figure, a PC (Macbook Air 13in) was used as the UE.

To evaluate the feasibility of the data retrieval for conventional HTTP-based requests, a website that can provide the name of the required data to the broker CGI was accessed, as shown in Figure 9 (a). For the above request, the BS-CH device replied with the URI for the data, as shown in Figure 9 (b). As a result, the proposed scheme provided the data in the IC-WSNs to conventional HTTP-based terminals. Therefore, the scheme is compatible with current HTTP/TCP/IP network systems.

Figures 10–12 show the results of the network-performance evaluation when all nodes support the ICN platform thanks to using the Cefore platform, including the UE. The results represent the average values for twenty trials of data retrieval, i.e., the UE sent twenty-interest packets for a different type of data. Figure 10 shows the data-retrieval time in which the UE could completely download after requesting the data. The first data acquisition required 6.90 s, whereas the average time after the second trial is decreased to 0.0402 s.



Gateway(HTTP-ICN)

Request Interest message

ccnx:/ r-onga/sn1/202109151220 Send Interest

192.168.10.21 〉 レスポンス

```
<html>
<head></head>
<body>
  <h1>Gateway(HTTP-ICN)<h1>
  <h2>Request Interest message</h2>
  <form method="POST" action="./broker.cgi">
    ccnx:/<input type="text" name="addr">
    <input type="submit" value="Send Interest">
</form>
</body>
</html>
```

(a)

Result:

r-onga/sn1/202109151220

0

broker.cgi 〉 レスポンス

```
<html>
<head></head>
<body>
    <h1>Result:</h1>
    <p>
        <a href="./chache/r-onga_sn1_202109151220">r-onga/sn1/202109151220</a>
    </p>
    <p>0</p>
</body>
</html>
```

(b)

Figure 9. Display screen of HTTP request sent from UE to BS and CH, (a) HTML-based website to intermediate content names to broker CGI, (b) broker CGI provides retrived data through HTML-based website

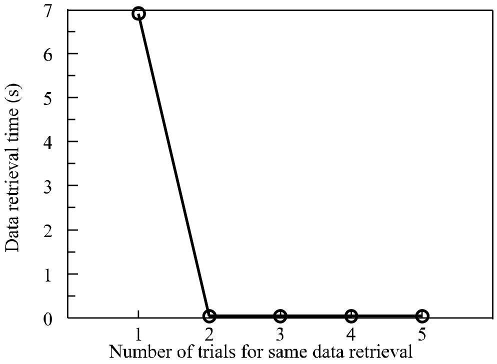


Figure 10. Data retrival time versus number of request to retrive for the same data

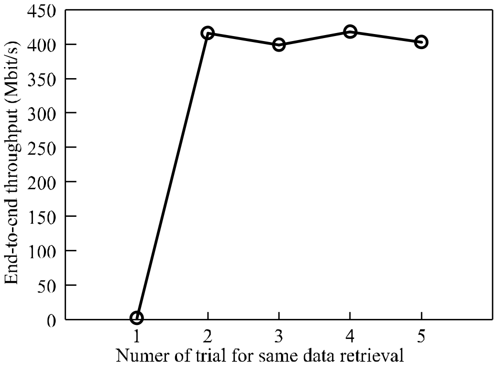


Figure 11. End-to-end network throughput versus number of requests to retrive for same data

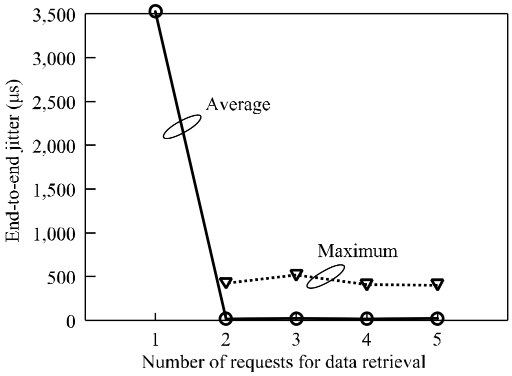


Figure 12. End-to-end jitter versus number of requests to retrive for same data

The reason behind this improvement was that the data were cached in the BS-CH device, and the second or later data retrievals were provided from the BS-CH device's cache memory. This fact can be evidenced by the results of the end-to-end throughput, as shown in Figure 11. In Figure 11, the first acquisition was 2.34 Mbit/s, and after the second trial, the end-to-end throughput improved to 409 Mbit/s. In this experimental network environment, the first trial needed to obtain the data from the SN, and the section between the BS-CH device and the SN included the wireless transmission, which led to a bottleneck. Since the wireless link was connected using IEEE 802.11 b/g, the ideal throughput was at a maximum of 54 Mbit/s at the physical layer. Figure 12 shows the end-to-end jitter versus the number of requests for data retrieval. The average jitter for the first data acquisition was 3.53 ms (up to 184 ms), whereas it improved to 19.7 µs (up to 438 µs) after the second acquisition. These results indicate that the IC-WSN can improve the stable delivery of data retrieval even if the data-transmission link involves a wireless channel.

V. Computer Simulation

In the previous section, feasibility and network performance were discussed in terms of the protocol stack at the upper layers. In this section, evaluation of the erasure-code decoder, frame reachability through wireless channels, and improvement in data caching among SNs at the lower layers are discussed.

A. Simulation environment

The simulation parameters are listed in Table I. The LDPC code was used as the erasure code, and its parity-check matrix was determined on the basis of the DVB-S2 specifications, which are widely used in digital video broadcasting via telecommunications [24]. The full-frame length was determined on the basis of the codeword length of the LDPC code, and the sub-frame length was determined on the basis of typical LPWA systems. To avoid system complexity, the buffer size was assumed an ideal condition, i.e., the upper limit of the cache memory of hardware was ignored, and the selection of buffered sensing data was not considered. The radio-propagation models used Erceg's model [25], Amorim's model [26], and the theoretical free-space model. Note that the first two models were used on the basis of the practical measurement results, and fading and shadowing were taken into account, unlike with the theoretical free-space model.

B. Simulation results

Regarding the robustness of the LDPC-based erasure code, Figure 13 shows the probability of successful recovery of the original packet if several sub-frames were lost. When the code rate was *R* = 1/4, 1/3, 1/2, 2/3, and 3/4, the original packet could be reconstructed even if 4, 11, 7, 3, and 2 sub-frames were lost, respectively. Note that the code rate denotes the percentage of information-data length in the total codeword length, including parity bits. The LDPC code has strong resilience against burst errors but requires a long codeword to guarantee sufficient error correction; therefore, we need to overcome this barrier for short sensing-data messages. When the percentage of lost sub-frames is small, the curve in Figure 13 remained flat because enough sub-frames to recover a full-frame arrived. The recovery rate suddenly decreased because the received data were digitally decoded; thus, there was no resistance to noise, the same as in an analog system.

The LDPC-code decoder fulfills an iterative operation on the basis of the belief propagation, which is called the sum-product algorithm. Figure 14 shows the average number of iterations until successful recovery, i.e., the computational burden increases depending on the increased number of iterations. The number of iterations was ten times or less when the packet was successfully restored, and even if the number of iterative operations exceeded 50, no improvement occurred. In other words, the curve in Figure 14 remained flat when the number of iterations exceeded 50 because the iterative-decoding process reached the pre-defined upper limitation. As shown in Figures 13 and 14, the radio-propagation models were not taken into account because those simulations were conducted on the basis of lost sub-frames as parameters; thus, there was no difference among radio-propagation models.

TABLE I. Simulation parameters

Parameter	Values
Erase code	LDPC with sum-product decoding
Trans. Interval	600 s (= 10 min.)
Multiple access	Slotted-ALOHA
Number of channels	15
Full-frame length	64,800 bits
Number of fragmentations	60
Modulation method	BPSK
Error control coding	Convolutional coding
Radio Frequency	2.4 GHz (in microwave), 920 MHz (in sub-GHz)
Channel model	Rayleigh fading
Radio propagation model	Erceg's model (SN-BS), Amorim's model (SN-UAV)
Radio transmission power	0 dBm
Antenna gain	0 dBi
Circuit loss	0 dB
Thermal noise	−172 dBm

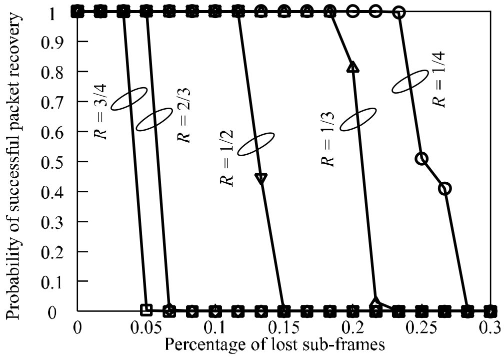


Figure 13. Probability of successful packet recovery versus percentage of lost subframes

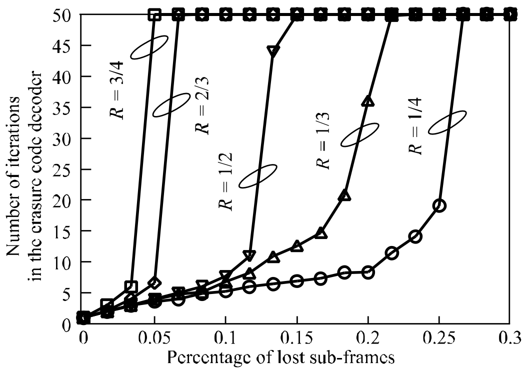


Figure 14. Number of iteratinos in the erasure-code decoder versus percentage of lost sub-frames

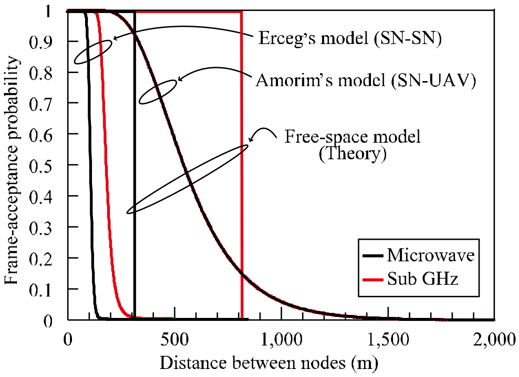


Figure 15. Frame-acceptance probability versus distance between nodes

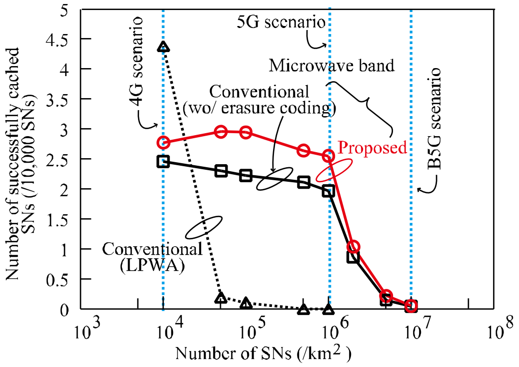


Figure 16. Number of successfully cached SNs versus density of distributed SNs

Figure 15 shows the frame-reception probability versus the distance between nodes. Erceg's and Amorim's models described smooth curves, and there did not appear to be a difference between radio frequency bands with Amorim's model.

Figures 13–15 show the effectiveness of the proposed scheme for packet caching, and Figure 16 shows the computer simulation results. In general, 10,000/km^2 (in the 4G scenario), 1,000,000/km^2 (in the 5G scenario), and 10,000,000/km^2 (in the Beyond 5G (B5G) scenario) were assumed as the number of SN deployments. In Figure 16, the LPWA systems achieved high reachability in the 4G scenario due to sufficient capacity for generated traffic. Therefore, the first computer simulation indicated that the proposed scheme could work in the 5G scenario by using the MAC and physical protocols, while a traditional IoT scheme cannot work. The proposed scheme improved data caching by 29.3% compared with the scheme without using an erasure-code decoder. This preliminary evaluation led to the conclusion that the proposed scheme has significant limitations for B5G scenarios and requires further analysis.

VI. Conclusion

The focus of this paper was on UAV-IC-WSNs, and an effective scheme for sensing-data collection and management in future smart city applications was proposed. The erasure code-encoded data-encoding method for large-scale data fragmentation and transmitter-side cooperative transmissions with dual-band node devices was addressed. A hardware-based experiment demonstrated that the feasibility of the scheme and its fundamental performance. The numerical results indicate that the proposed scheme can improve data caching by 29.3% in the deployment of future WSNs. For future work, the B5G scenarios will be expanded upon and analyzed in practical environments. It will also be necessary to discuss the feasibility and effectiveness of the proposed scheme through on-site field testing in terms of energy consumption and implementation cost.

Acknowledgments

A part of this work was supported by JSPS KAKENHI Grant Number JP19K20261 and JP21H03436.

References

[1] S. Mori, "A fundamental analysis of an erase code-enabled data caching scheme for future UAV-IC-WSNs," *Proc. IARIA the 20th Int. Conf. Networks (ICN 2021)*, Apr. 2021, pp. 8–12.

[2] A Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, "Future trends and current state of smart city concepts: A survey," *IEEE Access*, vol. 8, pp. 86448–86467, 2020.

[3] S. Mori, "Prototype development of river velocimetry using visual particle image velocimetry for smart cities and disaster area networks," *Proc. 20th Int. Sympo. Commun. and Info. Tech. (ISCIT 2021)*, Oct. 2021, pp. 169–171, doi: 10.1109/ ISCIT52804.2021.9590602.

[4] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent advances in information-centric networking-based Internet of things (ICN-IoT)," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2128–2158, Apr. 2019.

[5] F. Qi, X. Zhu, G. Mang, M. Kadoch, and W. Li, "UAV network and IoT in the sky for future smart cities," *IEEE Network*, vol. 33, no. 2, pp. 96–101, Mar. 2019.

[6] R. A. Nazib and S. Moh, "Energy-efficient and fast data collection in UAV-aided wireless sensor networks for hilly terrains," *IEEE Access*, vol. 9, pp. 23168–23190, 2021.

[7] S Mori, "A fundamental analysis of caching data protection scheme using light-weight blockchain and hashchain for information-centric WSNs," *Proc. 2nd Conf. Blockchain Research & Applications for Innovative Networks and Services (BRAINS 2020)*, Sept. 2020, pp. 200–201, doi: 10.1109/ BRAINS49436.2020.9223279.

[8] P. S. Bithas, V. Nikolaidis, A. G. Kanatas, and G. K. Karagiannidis, "UAV-to-ground communications: Channel modeling and UAV selection," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5135–5144, Aug. 2020.

[9] B. Li, X. Guo, R. Zhang, X. Du, and M. Guizani, "Performance analysis and optimization for the MAC protocol in UAV-based IoT network," *IEEE Trans. Vehicular Tech.*, vol. 69, no. 8, pp. 8925–8937, Aug. 2020.

[10] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, "A UAV-assisted data collection for wireless sensor networks: autonomous navigation and scheduling," *IEEE Access*, vol. 8, pp. 110446–110460, 2020.

[11] S. Poudel and S. Moh, "Medium access control protocols for unmanned aerial vehicle-aided wireless sensor networks: A survey," *IEEE Access*, vol. 7, pp. 65728–65744, 2019.

[12] S. Mori, "Cooperative sensing data collecting framework by using unmanned aircraft vehicle in wireless sensor network," *Proc. IEEE Int. Conf. Commun. (ICC2016)*, May 2016, pp. 1–6, doi: 10.1109/ICC.2016.7511187.

[13] S. Zhang, H. Zhang, and L. Song, "Beyond D2D: full dimension UAV-to-everything communications in 6G," *IEEE Trans. Vehicular Tech.*, vol. 69, no. 6, pp. 6592–6602, June 2020.

[14] M. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: An architectural case for information dispersion through native network coding," *Proc. 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications (NOM2012)*, June 2012, pp. 31–36, doi: 10.1145/2248361.2248370.

[15] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Info. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[16] H. Malik, C. Adjih, C. Weidmann, and M. Kieffer, "MICN: A network coding protocol for ICN with multiple distinct interests per generation," *Computer Networks*, vol. 187, pp. 1–14, Mar. 2021.

[17] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Information-theoretically secure erasure codes for distributed storage," *IEEE Trans. Info. Theory*, vol. 64, no. 3, pp. 1621–1646, Mar. 2018.

[18] M. Kishani, S. Ahmadian, and H. Asadi, "A modeling framework for reliability of erasure codes in SSD arrays," *IEEE Trans. Computers*, vol. 69, no. 5, pp. 649–665, May 2020.

[19] V. Sharma, S. Kalyanaraman, K. Kar, K. K. Ramakrishnan, and V. Subramanian, "MPLOT: A transport protocol exploiting multipath diversity using erasure codes," *Proc. Int. Conf. Computer Commun. (INFOCOM2008)*, Apr. 2008, pp. 121–125, doi: 10.1109/INFOCOM.2008.33.

[20] S. Lin and D. Costello, *Error control coding*, Prentice Hall, May 2004.

[21] Z. M. Fadlullah et al., "Multi-hop wireless transmission in multi-band WLAN systems: Proposal and future perspective," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 108–113, Feb. 2019.

[22] V. Srivastava and M. Motani, "Cross-layer design: A survey and the road ahead," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 112–119, Dec. 2005.

[23] Cefore: https://cefore.net (retrieved: Nov. 2021).

[24] DVB project: https://dvb.org (retrieved: Nov. 2021).

[25] V. Erceg et al., "An empirically based path loss model for wireless channels in suburban environment," *IEEE J. Sel. Areas in Commun.*, vol. 17, no. 7, pp. 1205–1211, July 1999.

[26] R. Amorim et al., "Radio channel modeling for UAV communication over cellular networks," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 514–517, Aug. 2017.