



SENSORCOMM 2022

The Sixteenth International Conference on Sensor Technologies and Applications

ISBN: 978-1-68558-005-6

October 16 - 20, 2022

Lisbon, Portugal

SENSORCOMM 2022 Editors

Anders Fongen, Norwegian Defence University College, Norway

SENSORCOMM 2022

Forward

The Sixteenth International Conference on Sensor Technologies and Applications (SENSORCOMM 2022), held on October 16-20, 2022, was a multi-track event covering related topics on theory and practice on wired and wireless sensors and sensor networks.

Sensors and sensor networks have become a highly active research area because of their potential of providing diverse services to broad range of applications, not only on science and engineering, but equally importantly on issues related to critical infrastructure protection and security, health care, the environment, energy, food safety, and the potential impact on the quality of all areas of life.

Sensor networks and sensor-based systems support many applications today on the ground. Underwater operations and applications are quite limited by comparison. Most applications refer to remotely controlled submersibles and wide-area data collection systems at a coarse granularity.

In wireless sensor and micro-sensor networks energy consumption is a key factor for the sensor lifetime and accuracy of information. Protocols and mechanisms have been proposed for energy optimization considering various communication factors and types of applications. Conserving energy and optimizing energy consumption are challenges in wireless sensor networks, requiring energy-adaptive protocols, self-organization, and balanced forwarding mechanisms.

We take here the opportunity to warmly thank all the members of the SENSORCOMM 2022 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to SENSORCOMM 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the SENSORCOMM 2022 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that SENSORCOMM 2022 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of sensor technologies and applications. We also hope that Lisbon provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

SENSORCOMM 2022 Chairs

SENSORCOMM 2022 Steering Committee

Anne-Lena Kampen, Western Norway University of Applied Sciences, Norge
Masanari Nakamura, Hokkaido University, Japan

SENSORCOMM 2022 Publicity Chair

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

SENSORCOMM 2022

Committee

SENSORCOMM 2022 Steering Committee

Anne-Lena Kampen, Western Norway University of Applied Sciences, Norge
Masanari Nakamura, Hokkaido University, Japan

SENSORCOMM 2022 Publicity Chair

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

SENSORCOMM 2022 Technical Program Committee

Majid Bayani Abbasy, National University of Costa Rica, Costa Rica
Amin Al-Habaibeh, Nottingham Trent University, UK
Jesús B. Alonso Hernández, Institute for Technological Development and Innovation in Communications (IDeTIC) | University of Las Palmas de Gran Canaria (ULPGC), Spain
Youssef N. Altherwy, Imperial College London, UK
Mário Alves, Politécnico do Porto - School of Engineering (ISEP), Portugal
Thierry Antoine-Santoni, University of Corsica, France
Kazutami Arimoto, Okayama Prefectural University, Japan
Anish Arora, The Ohio State University / The Samraksh Company, USA
Andy Augousti, Kingston University London, UK
Michel Bakni, ESTIA Recherche, Bidart, France
Reda Benkhouya, Ibn Tofail University, Morocco
Abdelmadjid Bouabdallah, University of Technology of Compiègne, France
Souheila Bouam, University of Batna 2, Algeria
An Braeken, Vrije Universiteit België, Belgium
Erik Buchmann, Hochschule für Telekommunikation Leipzig, Germany
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Oscar Carrasco Quilis, Casa Systems Inc., Valencia, Spain
Vítor Carvalho, 2Ai Lab- School of Technology - IPCA / Algoritmi Research Center - Minho University, Portugal
Mario Cifrek, University of Zagreb, Croatia
Federico Corò, Sapienza University of Rome, Italy
Luis J. de la Cruz Llopis, Universitat Politècnica de Catalunya, Spain
Baban P. Dhonge, Indira Gandhi Center for Atomic Research, Kalpakkam, India
Alexis Duque, Rtone, Lyon, France
Loubna El Amrani, Ibn University Tofail Kénitra, Morocco
Mohamed El Bakkali, Sidi Mohamed Ben Abdellah University, Morocco
Youssef Elgholb, University Sidi Mohamed Ben Abdellah, Fez, Morocco / Technical University of Cartagena, Spain
Biyi Fang, Microsoft Corporation, USA
Stefan Fischer, University of Lübeck, Germany
Fernando Juan García-Diego, Universitat Politècnica de València, Spain

Dinesh R. Gawade, Tyndall National Institute | University College Cork, Ireland
Lyamine Guezouli, University of Batna 2, Algeria
Arda Gumusalan, IBM, USA
Malka N. Halgamuge, The University of Melbourne, Australia
Agostino Iadicicco, University of Naples Parthenope, Italy
Miao Jin, University of Louisiana at Lafayette, USA
Anand Y Joshi, G. H. Patel College of Engineering and Technology, India
Grigoris Kaltsas, University of West Attica, Greece
Anne-Lena Kampen, Western Norway University of Applied Sciences, Norge
Lutful Karim, Seneca College of Applied Arts and Technology, Toronto, Canada
Azeddine Khat, ENSET Mohammedia | University Hassan II of Casablanca, Morocco
André Kokkeler, University of Twente, Netherlands
Boris Kovalerchuk, Central Washington University, USA
Hanane Lamaazi, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
Chiu-Kuo Liang, Chung Hua University, Hsinchu, Taiwan
David Lizcano, Madrid Open University (UDIMA), Spain
Flaminia Luccio, University Ca' Foscari of Venice, Italy
Abdalah Makhoul, University of Franche-Comté, France
Stefano Mariani, Politecnico di Milano, Italy
Jose-Manuel Martinez-Caro, Technical University of Cartagena, Spain
Carlo Massaroni, Università Campus Bio-Medico di Roma, Italy
Lei Mei, KLA Corporation, USA
Carosena Meola, University of Naples Federico II, Italy
Ahmad Mohamad Mezher, University of New Brunswick, Canada
Fabien Mieyeville, Polytech Lyon - University Claude Bernard Lyon 1, France
Mohammad Moltafet, University of Oulu Finland - Centre for Wireless Communications, Finland
Raúl Monroy, Tecnológico de Monterrey, Mexico
Rafael Morales Herrera, Universidad de Castilla-La Mancha, Spain
Masanari Nakamura, Hokkaido University, Japan
Dmitry Namiot, Lomonosov Moscow State University, Russia
Jagriti Narang, Jamia Hamdard, New Delhi, India
Mir Mohammad Navidi, West Virginia University, USA
Michael Niedermayer, Berliner Hochschule für Technik, Germany
Rosdiadee Nordin, UniversitiKebangsaan Malaysia, Malaysia
Brendan O'Flynn, Tyndall National Institute | University College Cork, Ireland
Pouya Ostovari, San Jose State University | Charles W. Davidson College of Engineering, USA
Carlos Enrique Palau Salvador, Universitat Politècnica de València, Spain
Bruno Pereira dos Santos, University of Ouro Preto, Brazil
Ivan Miguel Pires, Instituto de Telecomunicações - Universidade da Beira Interior / Polytechnic Institute of Viseu, Portugal
Patrick Pons, LAAS-CNRS, France
Càndid Reig, University of Valencia, Spain
Girish Revadigar, Huawei International Pte Ltd, Singapore
Christos Riziotis, National Hellenic Research Foundation, Greece
Lorenzo Rubio Arjona, Universitat Politècnica de València, Spain
Ulrich Rückert, Bielefeld University, Germany
Sain Saginbekov, Nazarbayev University, Nur-Sultan, Kazakhstan
Prasan Kumar Sahoo, Chang Gung University, Taiwan

Addisson Salazar, Universitat Politècnica de València, Spain
Sevil Sen, Hacettepe University, Turkey
Sheng Shen, University of Illinois at Urbana-Champaign, USA
Francesco Betti Sorbelli, University of Perugia, Italy
Mu-Chun Su, National Central University, Taiwan
Alvaro Suárez Sarmiento, University of Las Palmas de Gran Canaria, Spain
Rui Teng, Advanced Telecommunications Research Institute International, Japan
Carlos Travieso González, University of Las Palmas de Gran Canaria, Spain
Yu Chee Tseng, NCTU, Taiwan
Eirini Eleni Tsiropoulou, University of New Mexico, USA
Stefan Valentin, Darmstadt University of Applied Sciences, Germany
Manuela Vieira, CTS/ISEL/IPL, Portugal
You-Chiun Wang, National Sun Yat-sen University, Taiwan
Julian Webber, Osaka University, Japan
Murat Kaya Yapici, Sabanci University, Turkey
Hong Yang, Nokia Bell Labs, Murray Hill, USA
Sergey Y. Yurish, International Frequency Sensor Association (IFSA), Spain
Zhenghao Zhang, Florida State University, USA
Zhe Zhou, Huzhou University, China
Chengzhi Zong, eBay Inc, USA

Copyright Information

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

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

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

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

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

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

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

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

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

Table of Contents

Design of Knitted Pressure Sensors for Tactile Perception Applications <i>Beyza Bozali, Sepideh Ghodrat, Joris van Dam, and Kaspar Jansen</i>	1
Centralized Routing for Lifetime Optimization Using Genetic Algorithm and Reinforcement Learning for WSNs <i>Elvis Obi, Zoubir Mammeri, and Okechukwu Emmanuel Ochia</i>	5
Integration of Digital Mobile Radio in a Sensor Network <i>Anders Fongen</i>	13

Design of Knitted Pressure Sensors for Tactile Perception Applications

Beyza Bozali

*Faculty of Industrial Design Engineering
Delft University of Technology
Delft, The Netherlands
e-mail: B.Bozali@tudelft.nl*

Sepideh Ghodrat

*Faculty of Industrial Design Engineering
Delft University of Technology
Delft, The Netherlands
e-mail: S.Ghodrat@tudelft.nl*

Joris van Dam

*Faculty of Industrial Design Engineering
Delft University of Technology
Delft, The Netherlands
e-mail: J.J.F.vanDam@tudelft.nl*

Kaspar Jansen

*Faculty of Industrial Design Engineering
Delft University of Technology
Delft, The Netherlands
e-mail: K.M.B.Jansen@tudelft.nl*

Abstract—As an emerging technology, smart textiles attract a lot of interest in various fields in healthcare, sports, entertainment, etc. The new trend in smart textiles is to design sensor elements that are entirely textile-based consisting of conductive and non-conductive yarns which make the traditional sensor systems more comfortable and available for everyday use. In this study, stretchable knitted pressure sensors with different configurations and designs were produced by using commercial conductive yarns. By controlling the knit structure, the overall resistance can also be fine-tuned for practical pressure sensor applications. The impact of different knit densities and knit designs on the pressure sensing performance is investigated for tactile sensing and force distribution applications for flat-knitted and dome-like knitted sensors. Initial studies showed that the flat structures knitted less densely and in more stretcher knit designs result in better sensor performance with a sensitivity of 0.71 kPa^{-1} and within the pressure ranges of 0 to 0.71 kPa and a sensitivity of 0.13 kPa^{-1} within a pressure range of 0.70 to 2.5 kPa. This pressure sensor is knitted with the Rib 3x3 design and with a stitch length of 3 mm, and the response time of the sensor is measured as 220 msec. For dome-like pressure sensors, Rib 3x3 showed a relatively better sensitivity value within a pressure range of 0 to 10 kPa. A practical application of these sensors can be for touch detection in human-computer interaction applications.

Keywords—pressure sensors; textile-based sensors; knitting

I. INTRODUCTION

As an emerging technology, textile-based sensors gained a lot of attention over the years due to their intrinsic softness, flexibility and comfort properties. They can distinguish between various environmental stimuli such as strain, pressure and temperature which enables them to be used in applications such as human-computer interaction [1], health monitoring applications [2], human motion recognition [3], and so on.

Textile-based pressure sensors show high-pressure sensing ability and improved flexibility when compared to traditional pressure sensors. Depending on the application area, textile pressure sensors can be tuned in to have the desired characteristics. Monitoring physiological characteristics such as speaking, aspiration and pulse require low pressure, ranging from

0 to 10 kPa. The medium pressure range (about 10-100 kPa) is suitable for the detection of finger movements like clicking a mouse button, whereas the high-pressure range (100 to 1000 kPa) can be used for pressure sensing in socks and wheelchairs [4]. In addition to the pressure range, other performance indicators are sensitivity, durability, detection range and response time [5].

Textile-based pressure sensors manufactured from commercially available materials have been studied previously by several researchers. Riley, Oliveria, Morris and Dias focused on knitted spacer pressure sensors for monitoring wheelchair users' seat posture [6]. They explored the effect of pressure, applied pressure area and hysteresis of the sensor and found out that knitted sensors could measure up to a pressure of 25 kPa. Baribina, Oks, and Eizentals studied the comparative analysis of the effect of pressure loads on knitted pressure sensors with different shapes and knitted with copper-coated acrylic yarns. They concluded that filled-shaped sensors in rectangular forms show the highest sensitivity at lower pressures [7]. Tian et al. produced a piezoresistive pressure sensor in the form of a composite pillow and noted that the sensor showed a high sensitivity of 3.5 kPa^{-1} and good durability. They also concluded that the change in diameter under pressure results in a big change in sensitivity which proves the importance of the structural design of pressure sensors [8]. Although metallic particle-coated yarns and conductive blended yarns have been used for making knitted pressure sensors [9], there is still a lack of attention on directly implementing conductive yarn into piezoresistive pressure sensors. In this study, we investigate the performance of knitted pressure sensors using commercial conductive yarns and materials. The proposed knitted pressure sensor designs possess the potential to be used in different application areas.

This paper is structured as follows: In Section I, there will be an introduction to textile-based pressure sensors, followed by an experimental section in Section II describing the material and measurement technique. Results and discussion will be summarized in Section III and the paper is finalized with the conclusion in Section IV.

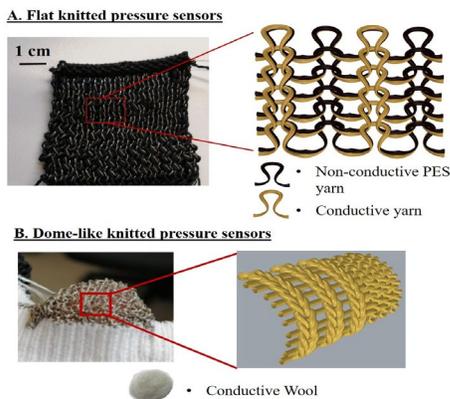


Figure 1. Knitted pressure sensors: A. Flat knitted, B, Dome-like knitted pressure sensors.

II. EXPERIMENTAL SECTION

In this section, the materials used in sensor design and the measurement technique that we used to evaluate the sensor performance will be explained respectively.

A. Materials

In this paper, we conducted a study on the electromechanical performance of knitted pressure sensors in which the material and process parameters were varied systematically. In terms of the materials, silver-plated nylon yarn from Shieldex with a resistivity of $\geq 600\Omega/m$ and non-conductive PES yarn with a linear density of dtex 110/32 is used. The stitch density and the knit designs were the two parameters that we changed as knitting parameters for the pressure sensors. As shown in Fig. 1, two types of designs were produced one of which is rectangular and the other in a dome-like form. These forms were selected to investigate the importance of the structural design on sensing performance. The advantages and disadvantages of two different sensors were compared in Table I.

TABLE I. COMPARISON OF FLAT AND DOME-LIKE KNITTED PRESSURE SENSORS

A&D	Comparison of flat and dome-like sensors	
	Flat knitted sensor	Dome-like sensors
Advantages	Easy to produce at one go. Can be directly attached to the skin.	Short response time Easy to use for tactile perception applications – bump form
Disadvantages	Low detection range	Complex sensor design

Rectangular samples were knitted via plating technique in such a way conductive and non-conductive yarn were knitted together. The dimension of the flat knitted sensors is $4 \times 4 \text{ cm}^2$. NP number and knit design were changed to examine the sensitivity, detection range and stability performance of the samples. The NP number which refers to the number of pitch directly change the loop length and changes the fabric tightness. In dome-like structures, the sensor part was produced only by using conductive yarn. The diameter (D) of the dome-like pressure sensor is 24 mm, and its area is calculated with the formula of $S=\pi(D/2)^2$. To gather the resistance change values under deformation, conductive wool which was purchased from Bekinox® W12/8 as a filler was inserted inside and optimized. The knitted pressure sensors used in this study have been weft-knitted on a Stoll CMS 530 flat knitting machine with an E8 machine gauge and 0.30 m/

m/sec carriage speed.

B. Measurement Technique

Knitted pressure sensors were evaluated in terms of sensitivity, in which P refers to applied pressure while ΔR refers to the resistance change and R_0 is the initial resistance of the sensor.

$$S = (\Delta R/R_0)/P \quad (1)$$

Sensitivity tests were conducted for flat knitted sensors by applying the same size blocks as pressure. And for dome-like knitted pressure sensors, sensitivity tests were conducted by using a rheometer in compression mode. During the test, a digital multimeter was used to record the sensor's resistance by connecting the cables on the sensor's edges, and 2 wire resistance measurement was conducted. The test speed was set as $20 \mu\text{m/sec}$. The axial force range is set to be between 0 and 10N.

III. RESULTS AND DISCUSSION

In this section, the sensor performances of flat-knitted and dome-like pressure sensors will be analyzed and the results will be discussed in terms of sensitivity and detection range.

A. Analysis of flat knitted pressure sensors

Flat knitted sensors in different knit designs were evaluated in terms of sensitivity and detection range. Resistance changes were collected by loading different weights on top of the sensors and tests were conducted three times per set. Pressure sensors in the forms of Rib 1x2, 2x2 and 3x3 were knitted in different NPs of 14 and 15. As can be seen in Table II, the pressure sensitivity of the Rib 2x2 and Rib 3x3 sensors increases as the loop length increased from NP 14 to NP 15. Fabrics with high stretchability allow the creation of more pressure-sensitive sensors. [3]. In addition to that, Rib 3x3 sensors have a higher elasticity than the Rib 2x2 which can explain the better sensitivity values. Under loading, yarn-to-yarn contacts are adjusted which leads to a change in overall fabric resistance [9]. Response time is another important parameter to minimize delays during application. The Rib 3x3 design knitted with NP15 has a response time of 220 msec with a detection range of 0-0.7 kPa.

TABLE II. SENSITIVITY RESULTS OF FLAT KNITTED PRESSURE SENSORS

Knit Design	Electromechanical Properties		
	NP	Sensitivity (kPa^{-1})	Detection Range (kPa)
Rib 1x2	15	0.10 ± 0.006	0-2.6
Rib 1x2	14	0.23 ± 0.01	0-0.7
Rib 2x2	15	0.28 ± 0.02	0-2.0
Rib 2x2	14	0.09 ± 0.004	0-1.3
Rib 3x3	15	0.71 ± 0.01	0-0.7
Rib 3x3	14	0.05 ± 0.03	0-2.0

^aLoop length.

As a preliminary study, Rib 3x3 sensors knitted with NP 15 were tested to explore whether the sensor distinguish the different pressures. Blocks of the same size corresponding to 1.2N are used to investigate the effect of pressure on resistance by applying pressure to the sensor area. As shown in Fig. 2, the Rib 3x3 flat knitted sensor has a good ability to distinguish the different loading forces.

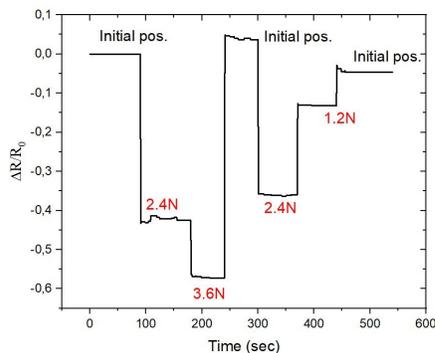


Figure 2. The response of the Rib 3x3 sensor knitted with NP15 when different pressure is loading and unloading.

According to the preliminary results in Table II, Rib 2x2 and Rib 3x3 were chosen for the following knit design.

B. Analysis of dome-like knitted pressure sensors

In this section, dome-like knitted pressure sensors were prepared and the performance was evaluated in terms of sensitivity and detection range. Because of the difficulties in the manufacturing process, samples were not produced with higher NPs like in the flat knitted sensors and were knitted with NP 9. Rib 2x2 and Rib 3x3 samples were produced. The conductive wool amount is set as 0.046g. The results were depicted in Table III. The Rib 3x3 sensor showed better sensitivity performance than Rib 2x2 at a pressure range of 0-10 kPa.

As mentioned in Table I, the detection range of flat-knitted sensors is lower than the dome-like sensors because of the contact areas and the interfaces related to the sensor design. When we think about the tactile perception application, a higher detection range is needed to be efficiently used which makes dome-like structures more promising. And higher sensitivities were observed for flat-knitted sensors at low detection ranges with looser samples. This can be related to the diameter variation in the conductive-filled assembly fibers, sensitivity changes can be obtained [9]. For future studies, dome-like sensors will be produced in a looser form and the sensitivity range will be compared with the flat knit sensors at lower pressures.

TABLE III. SENSITIVITY RESULTS OF DOME-LIKE KNITTED PRESSURE SENSORS

Sample Names	Electromechanical Properties		
	Filler (g)	Sensitivity (kPa ⁻¹)	Detection Range (kPa)
Rib 3x3	0.046	0.020 ± 0.004	0-10
Rib 2x2	0.046	0.015 ± 0.001	0-10

The Rib 3x3 dome-like sensor knitted with NP 9 was further investigated to explore its performance during loading and unloading. As shown in Fig. 3, 4 cycles were applied and it demonstrates that the sensor responds to the pressure changes. The dome-like knitted sensors have much lower sensitivity than the flat knitted sensors but do react properly when activated by a light finger press.

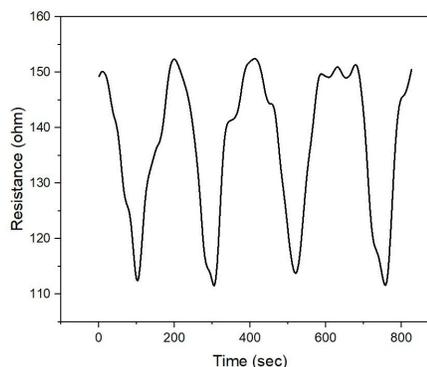


Figure 3. Cyclic test of Rib 3x3 dome-like knitted sensor

IV. CONCLUSION

This study demonstrates the importance of the structural design of knitted pressure sensors. Two different types of knit designs were produced and their sensor performance was evaluated. In the flat knitted form, Rib 3x3 with NP 15 possesses a high sensitivity of 0.71±0.01 kPa⁻¹. Apart from flat knitted structures, dome-like pressure sensors in the Rib design were produced and their sensor performance was evaluated. In the flat knitted form, Rib 3x3 with NP 15 possesses a high sensitivity of 0.71±0.01 kPa⁻¹. Apart from flat knitted structures, a dome-like pressure sensor in the Rib 3x3 form also possesses a potential design with a quick response time and a good sensitivity.

Flat knitted can be directly attached to the skin or large area textiles and be utilized for human-computer interface systems. Dome-like sensors were found to be promising for tactile applications such as the detection of finger movements by pinch or pressure sensing mats for shape or object recognition. Also, the user can perceive the sensors as protrusions by touching them, which provides convenience in the application.

For future studies, dome-like pressure sensors will be evaluated in detail by changing the filler content, type of filler and the shape of the dome. The production of multiple dome-like pressure sensors will be done at one-go and be tested for pressure sensing performance.

REFERENCES

- [1] L. Zhang, J. He, Y. Liao, X. Zeng, N. Qiu, Y. Liang, P. Xiao, T. Chen, "A self-protective, reproducible textile sensor with high performance towards human-machine interactions," Journal of Materials Chemistry A, vol. 46, pp. 26631-26640, 2019.
- [2] J. Zhang, T. G. Yun, M. A. MJ, S. A. Imam, and A. Zhao, "A review of passive RFID tag antenna-based sensors and systems for structural health monitoring applications," Sensors, vol. 17, pp. 265-298, 2017.
- [3] C. Jingyuan, M. Sundholm, B. Zhou, M. Hirsch, and P. Lukowicz, "Smart-surface: Large scale textile pressure sensors arrays for activity recognition," Pervasive and Mobile Computing, vol. 30, pp. 97-112, 2016.
- [4] L. Possanzini, M. Tassarolo, L. Mazzocchetti, E. Campari, and B. Fraboni, "Impact of fabric properties on textile pressure sensors performance," Sensors, vol. 19, pp. 4686-4701, 2019.
- [5] J. Zhang, Y. Zhang, L. Yuan-yuan, and P. Wang, "Textile-based flexible pressure sensors: A review," Polymer Reviews, vol. 62, pp. 65-94, 2022.

- [6] T. Hughes-Riley, C. Oliveira, R. Morris, and T. Dias, "The characterization of a pressure sensor constructed from a knitted spacer structure," *Digital Medicine*, vol. 5, pp. 22-29, 2019.
- [7] N. Baribina, A. Oks, I. Baltina, and P. Eizentals, "Comparative analysis of knitted pressure sensors," 7th International Scientific Conference, May 2018, pp. 1599-1604, ISSN: 1691-5976.
- [8] M. Tian, L. Yunjing, Q. Lijun, S. Zhu, X. Zhang, and S. Chen, "A pillow-shaped 3D hierarchical piezoresistive pressure sensor based on conductive silver components-coated fabric and random fibers assembly," *Industrial Engineering Chemistry Research*, vol. 58, pp. 5737-5742, 2019.
- [9] J. Xie, J. Yongtang, and M. Miao, "High sensitivity knitted fabric bi-directional pressure sensor based on conductive blended yarn," *Smart Materials and Structures*, vol. 28, p. 035017, 2019.

Centralized Routing for Lifetime Optimization Using Genetic Algorithm and Reinforcement Learning for WSNs

Elvis Obi

Computer Science Research Institute
Paul Sabatier University
Toulouse, France
email: elvis.obi@irit.fr

Zoubir Mammeri

Computer Science Research Institute
Paul Sabatier University
Toulouse, France
email: zoubir.mammeri@irit.fr

Okechukwu E. Ochia

Department of Electrical Engineering
University of Calgary
Calgary, Canada
email: okechukwu.ochia1@ucalgary.ca

Abstract—The sensor nodes' energy-efficient utilization is a major challenge in the design of Wireless Sensor Networks (WSNs). This is because the network lifetime is determined by the sensor nodes' limited energy sources whose replacement or recharging is almost impossible due to the mostly deployment of the sensor nodes in harsh environments. An effective way to prolong the network lifetime is by designing an energy-efficient routing protocol for WSNs. This paper discusses an optimization method for routing in WSNs to extend the network lifetime. Although the conventional method can extend the network lifetime, the computation time increases exponentially with the number of sensor nodes. Therefore, this method cannot apply to large-scale WSNs. This paper proposes a method to reduce the computation time using a genetic algorithm and shows that the proposed method can provide a suboptimal routing path through evaluation experiments.

Index Terms—reinforcement learning, genetic algorithm, routing, wireless sensor network, network lifetime, path optimization

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of spatially deployed sensor nodes in a geographical area to monitor and/or track physical properties, such as motion, pressure, temperature, etc., to collect, process and communicate the data to a sink using the wireless medium [1]. This has made WSNs useful in different fields of application, such as battlefield monitoring, object tracking, environmental monitoring, disaster management, etc. The sensor nodes have limited resources, such as power, bandwidth, memory, storage, processing, and computing speed. The sensor node is made up of four units, namely: the sensing unit, power unit, processing unit, and communication unit. The power unit is limited in energy source and supplies energy to the other units. Sensor nodes are mostly deployed in harsh environments, which makes their battery replacement difficult [2]. This makes the energy-efficient utilization of the sensor nodes vital to increase the network lifetime [3].

Distributed control with self-organized management has been the way of operating WSNs. This approach consumes a lot of energy due to the control overhead, which emanates

from the periodically broadcast messages for topology discovery by each sensor node to discover neighbors within its transmission range. But, recently Software-Defined WSNs (SDWSNs) have been proposed to enable WSNs to utilize the benefits of Software-Defined Networking (SDN). SDN is an emerging technology that intends to overcome the limited resources of sensor nodes and static architectures of networks by decoupling the network control plane from the data plane, bringing about programmability, and grouping the network into the application plane, control plane, and data plane. The network policies for routing, load balancing, etc. are being defined and administered at the application plane. The control plane evaluates and transforms the network policies into routing rules. The generation and forwarding of traffic using the routing rules are carried out in the data plane under the management and supervision of the controller in the control plane. The controller has wholistic information of the network and communicates with the sensor nodes using multi-hop communication [4].

The energy constraint problem in WSN, which determines the network lifetime is therefore alleviated in the SDWSN paradigm by removing some of the energy-consuming functions, such as sending control packets from the sensor nodes to the controller. This makes the sensor nodes not have any intelligence and functions, such as load balancing, routing, etc. to be handled at the controller or/and application layer. Routing in SDWSN is the selection of paths by the controller for the sensor nodes in forwarding data packets to the sink using multihop communication. Since the controller has a wholistic knowledge of the network topology, it can generate possible Minimum Spanning Trees (MSTs) to be used as the routing paths [5].

Traditional SDWSN has a limitation of finding the best routing path(s) such that the residual energy of sensor nodes is balanced, this is because it uses predetermined routing paths for data transmission. This thereby degrades the network lifetime since the predetermined routing path in advance does not depict the real status of the network, for instance, the energy consume by the sensor nodes to send packets to the

sink. This problem of finding an energy-efficient way of using the best routing path(s) in SDWSN can be solved by deploying artificial intelligence, such as Reinforcement Learning (RL) and Genetic Algorithm (GA) at the controller. RL is a kind of machine learning that learns to solve a problem by trial and error [6]. GA is a search-based heuristic technique based on the concept of natural selection and genetics [7]. The GA can be used to remove the NP-hardness of the All MSTs algorithm for a large-scale WSNs. This enables the generation of a subset of the network MSTs which serve as the routing tables in polynomial time. The RL enables the controller to learn the MSTs that maximize the network lifetime.

This paper aims at improving the time and space complexity of the Lifetime-Aware Centralized Q-Routing Protocol (LACQRP) [5]. This is achieved by using the proposed genetic MSTs algorithm to generate the routing tables. Also, the network lifetime considered in LACQRP is extended from the time for the first sensor node to deplete its energy source to the time that the sink is not reachable by the alive sensor nodes. This is because sensing is still possible by the alive sensor nodes and their data can be forwarded to the sink if the network graph is still connected.

The rest of the paper is structured as follows: Section II is the review of similar works; Section III explains the design of the Centralized Routing Protocol for Lifetime Optimization using GA and RL (CRPLOGARL); Section IV provides the discussion of the simulation and results and the conclusion of the paper is given in section V.

II. REVIEW OF SIMILAR WORKS.

The first RL-based routing protocol in networks is called Q-routing by Boyan and Littman [8]. Q-routing is a hop-by-hop routing protocol that minimizes packet delivery delay in networks. The limitations of Q-routing are Q-value freshness, parameter setting sensitivity, and slow convergence. Different routing protocols such as [9] - [14] have also applied RL to optimize delivery delay in networks.

Network lifetime and energy optimization routing protocols based on RL for WSNs have been done by different authors. The sequel presents some of these works.

Zhang and Fromherz [15] proposed an energy-efficient routing protocol based on RL for WSNs called constrained flooding. The protocol enables energy saving by adapting Q-routing to optimize the cost of sending data packets to the sink in WSNs with flooding. This reduces the number of packet transmissions and a corresponding reduction in the energy consumption of the WSNs. Dong et al. [16] designed for ultra-wideband sensor networks a RL based Geographical Routing (RLGR) protocol. RLGR improved the network lifetime by uniformly distributing energy consumption among nodes and reducing packet delivery delay. RLGR considers residual energy of nodes and hop counts to the sink in its reward function. Simulation results showed that RLGR has better network lifetime by at least 75 percent with respect to Greedy Perimeter Stateless Routing (GPSR) [17].

Hu and Fei [18] proposed for Underwater WSNs (UWSNs) a Q-learning-based Energy-efficient and Lifetime-Aware Routing (QELAR) protocol for finding the optimal routing path in the network. The protocol makes the residual energy of the nodes to be distributed evenly and thereby increasing the network lifetime. The packets in the network are forwarded based on a reward function that takes into consideration the energy distribution of a group of nodes and the residual energy of each node.

Jafarzadeh and Moghaddam [19] proposed a routing protocol for WSNs called Energy-aware QoS Routing RL-based (EQR-RL) protocol. EQR-RL optimizes the energy in WSNs while guaranteeing the delivery delay of packets. EQR-RL employs the probability distribution-based exploration strategy to choose the next hop to forward data packets. The reward function of the protocol is based on weighted metrics of selected forwarder residual energy, link delay, and the ratio of packets between packet sender and the selected forwarder.

Geo et al. [20] proposed an intelligent routing protocol for WSNs built on RL named RL-based Lifetime Optimization (RLLO) routing protocol. RLLO uses residual energy of sensor node and hops count to the sink in its reward function to update agents' Q-values. The agents in RLLO are the sensor nodes. The routing protocol is implemented in NS2. Simulation results show improved performance when compared with energy-aware routing (EAR) and improved energy-aware routing (I-EAR) using network lifetime and packet delivery as performance metrics.

Debowski et al. [21] proposed a hybrid protocol called Q-Smart Gradient-based (QSGrd) routing protocol for WSNs. QSGrd optimizes the energy consumption in WSNs by combining transmission gradient and Q-learning. In QSGrd, each neighbor of a node is associated with transmission success probability which depends on the maximum transmission range and the distance between nodes. The transmission success probabilities of the neighbors of a node result in a transmission gradient. Subsequently, the transmission probabilities are used to update the Q-values. The best routing paths are learned and selected based on the residual energy of the next hop and the average least number of transmissions to the sink using RL.

Mutumbo et al. [22] proposed an RL-based Energy Balancing Routing (EBR-RL) protocol for WSNs. EBR-RL protocol maximizes the network lifetime by balancing the energy consumption between sensor nodes. EBR-RL protocol operates in two stages. The first stage set up the network and the second stage carries out the data transmission using RL. EBR-RL protocol has better performance in terms of network lifetime and energy saving when compared with existing energy-efficient routing protocols.

Obi et al. [5] proposed a Lifetime-Aware Centralized Q-Routing Protocol (LACQRP) for WSNs. The state space and action space of LACQRP are all MSTs generated by the controller which served as the routing tables. LACQRP maximizes the network lifetime by learning the optimal routing tables that minimize the maximum energy consumption of the sensor

nodes. Simulation results show that LACQRP converges to the optimal routing table(s) and has an increased network lifetime when compared with RL-Based Routing (RLBR) [23] and RL for Lifetime Optimization (R2LTO) [24]. However, LACQRP computational complexity varies exponentially with the number of deployed sensor nodes. This makes LACQRP not to be feasible in practice for large-scale WSNs.

III. METHODOLOGY

This paper considers a WSN consisting of a set of sensor nodes and a sink. The sink also acts as an SD controller. After the network initialization, each sensor node broadcasts its status information, which includes a unique identifier (Id), $x - y$ location, load (number of data octets to transmit per second to the sink), residual energy, and maximum transmission range. The sink collects all the sensor nodes' status information and builds the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of sensor nodes and \mathcal{E} is the set of network links between two connected distinct nodes in the network. Two sensor nodes are only connected if their Cartesian distance is less than or equal to the maximum transmission range of the sensor nodes.

The controller computes a list of routing tables using the proposed genetic MSTs algorithm based on distance edge weight. The distance edge weight is used to compute the MSTs because the transmission energy of a sensor node depends on distance. The controller chooses an MST in each round using RL and broadcasts it to all sensor nodes for data transmission. This enables the lifetime optimization of the WSN globally. The genetic MSTs algorithm and the centralized routing protocol for lifetime optimization using GA and RL are presented in the sequel.

A. A Genetic MSTs Algorithm

The set of nodes and links of the network graph are denoted by $\mathcal{V} = \{v_1, \dots, v_n\}$ and $\mathcal{E} = \{e_1, \dots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$, respectively. Each link $e \in \mathcal{E}$ is associated with an integer weight $w(e) > 0$ representing the link distance. The population of the GA is obtained from MSTs generated by a classical MST algorithm. The classical algorithms for finding MST of a connected undirected graph are Kruskal's algorithm [25], Prim's algorithm [26], and Boruka's algorithm [27]. Kruskal's algorithm and Boruka's algorithm can only find one MST of a network graph. This is because Kruskal's algorithm and Boruka's algorithm look at the network graph in its entirety and add the shortest edge to the existing tree until the MST is found. Subsequently, Prim's algorithm builds MST by initializing a random node as the root node. This makes Prim's algorithm find several MSTs for a network graph that does not have distinct edge weights when varying the root node [28]. Therefore, the number of MSTs generated by the Prim's algorithm by varying the root node is less than or equal to the number of nodes, $|\mathcal{V}|$ of the network graph.

The genetic MSTs algorithm uses unique MSTs of the network as the initial population extracted by the Prim's algorithm. Prim's algorithm is called using the network graph and varying root nodes as inputs. The algorithm for generating

the MSTs using Prim's algorithm runs in $O(|\mathcal{V}||\mathcal{E}| \log |\mathcal{V}|)$ time, where $|\mathcal{V}|$ and $|\mathcal{E}|$ denote the number of sensor nodes and network links, respectively. The algorithm for generating the initial population for the genetic MSTs algorithm is given in **Algorithm 1**.

Algorithm 1 Initial population using Prim's Algorithm

Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$

Output: $MSTs$

```

1:  $MSTs = \{\}$ 
2: for  $j$  in  $\mathcal{V}$  do
3:   Select vertex  $j$  as the root node
4:    $T = Prim(\mathcal{G}, j)$ 
5:   if  $T \notin MSTs$  then
6:      $MSTs \leftarrow T$ 
7:   end if
8: end for
9: Return  $MSTs$ 

```

Every chromosome in the population represents an MST, with its genes representing the graph edges [29]. The population evolves when passed through the crossover operator and the mutation operator, which generates new individuals (MSTs) by inheriting some of their parents' attributes (edges). The objective function used to measure the fitness of the newly formed individual is the cost of the MST of the graph. Individual fitness is given by the possibility of the new tree T^k formed by the crossover operator and the mutation operator having a total distance edge weight equal to the total distance edge weight of a minimum spanning tree T^* of the graph as given in (1).

$$fitness(k) = Poss \left[\sum_{i,j \in T^k} d_{i,j} = \sum_{i,j \in T^*} d_{i,j} \right] \quad (1)$$

where $d_{i,j}$ is a link distance weight.

The crossover operation is done by randomly selecting two individuals, T_1 and T_2 from the population as parents to form children for the next generation. T_1 and T_2 are united to form a sub-graph, \mathcal{G}_1 of \mathcal{G} by applying the union operation [30]. The new individual generated will be an MST of \mathcal{G}_1 and also of the network graph \mathcal{G} . The crossover rate, cr determines how many times of applying the crossover operator before taking the fitness individual.

Also, the mutation operation is done by randomly choosing an edge, $e_{i,j}$ from a selected individual, T_3 in the population. The $e_{i,j}$ is deleted from T_3 and the main network graph \mathcal{G} to form the sub-graphs, \mathcal{G}_2 and \mathcal{G}^* , respectively. A random edge, $e_{i,j}^*$ belonging to the cut set of \mathcal{G}^* is added to \mathcal{G}_2 to form a new sub-graph, \mathcal{G}_2^* [29]. The cut set of \mathcal{G}^* is the set of edges belonging to \mathcal{G}^* and does not belong to \mathcal{G}_2 . The new individual generated by the mutation operation must be a tree of \mathcal{G} before acceptance. The mutation rate, mr is used to specify the number of times of applying the mutation operator before selecting the fitness one. The genetic MSTs algorithm is as given in **Algorithm 2**.

Algorithm 2 A Genetic MSTs algorithm

Input: cr, mr , Number of generations (NG)

Output: MSTs

```

1:  $P = \{\}$ 
2: Generate  $k \leq |\mathcal{V}|$  unique MSTs using Algorithm 1
3:  $P \leftarrow k$  unique MSTs
4: for  $i = 1$  to  $NG$  do
5:    $P_i = \{\}$ 
6:    $n_c = \frac{100}{cr}$ 
7:   for  $j = 1$  to  $n_c$  do
8:     Randomly choose  $T_1, T_2$  from  $P$ 
9:      $\mathcal{G}_1 = T_1 \cup T_2$ 
10:     $T = Prim(\mathcal{G}_1, v)$ 
11:    if  $T \notin P_i$  then
12:       $P_i \leftarrow T$ 
13:    end if
14:  end for
15:   $n_m = \frac{100}{mr}$ 
16:  for  $j = 1$  to  $n_m$  do
17:    Randomly choose  $T_3$  from  $P$ 
18:    Randomly choose  $e_{i,j}$  from  $T_3$ 
19:     $\mathcal{G}_2 = T_3 - e_{i,j}$ 
20:     $\mathcal{G}^* = \mathcal{G} - e_{i,j}$ 
21:     $Cut\ Set = \{e_{i,j} \mid e_{i,j} \in \mathcal{G}^* \ \& \ e_{i,j} \notin \mathcal{G}_2\}$ 
22:    Randomly choose  $e_{i,j}$  from  $Cut\ Set$ 
23:     $\mathcal{G}_2^* = \mathcal{G}_2 + e_{i,j}^*$ 
24:    if  $\mathcal{G}_2^*$  is a tree of  $\mathcal{G}$  then
25:      if  $\mathcal{G}_2^* \notin P_i$  then
26:         $P_i \leftarrow \mathcal{G}_2^*$ 
27:      end if
28:    end if
29:  end for
30:  for  $T$  in  $P_i$  do
31:    Evaluate fitness using (1)
32:    if fitness is True then
33:      if  $T \notin P$  then
34:         $P \leftarrow T$ 
35:      end if
36:    end if
37:  end for
38: end for
39: Return  $P$ 

```

B. A Centralized Routing Protocol for Lifetime Optimization using Genetic Algorithm and Reinforcement Learning

A Centralized Routing Protocol for Lifetime Optimization using GA and RL (CRPLOGARL) is designed to remove the NP-hardness associated with the LACQRP [5]. This is achieved by replacing the All MSTs algorithm in LACQRP with the proposed genetic MSTs algorithm. The CRPLOGARL optimizes the time it takes for the sink not to be reachable by alive sensor node. This is achieved by finding the routing tables using Q-learning after each stage of the network graph building such that the minimum of the sensor nodes' energies

is maximized. This leads to the prolonging of the time taken for sensor node(s) to die and hence the maximization of the time taken for the sink not to be reachable by alive sensor nodes.

The learning agent of the CRPLOGARL resides in the controller of the SDWSN. The action space A and the state space S of the agent is the list of MSTs generated by the controller using **Algorithm 2**. The learning agent state is the current MST that is used by the sink in receiving data packets from the sensor nodes. The action of the learning agent is to choose an MST from the action space after a round of data transmission based on the performance of the previous MST used. The learning agent measures the performance of the chosen MST by using the maximum of the energies consumption of the sensor nodes to send data packets as the reward function. This is because there is a variation in the energy consumption of the sensor nodes when different MSTs are used in data transmission. The variation in the energy consumption is from the difference in the number of links crossing each sensor node in the different MSTs.

The quality of being in a state $s \in S$ and choosing an action $a \in A$ is measured by an action-value function called Q-value. The Q-value is a measure of the long-run reward that the agent gets from each pair of state-action. The estimate of this action-value function which is used to find the best action for a given state is realized by caching the Q-values $Q(s_t, a_t)$ of pairs of state-action using the iterative update rule given in (2) [6]. The learning of the agent is made meaningful by denoting the Q-value of the learning agent as the maximum of the sensor nodes' energies consumption when a particular MST is used in receiving data packets by the sink.

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha)Q^{old}(s_t, a_t) + \alpha \left[r_t + \gamma * \max_{a \in A} \{Q(s_{t+1}, a)\} \right] \quad (2)$$

The extent to which the new learned Q-value affects the old Q-value is dependent on the learning rate, $\alpha(0, 1]$. The closer the value of α is to one, the more the impact of the newly computed Q-value on the old one. If α is equal to one, then the recent learned Q-value replaces the old Q-value completely. The discount factor, $\gamma[0, 1]$ controls the agent's liking for the future rewards with respect to the current reward. If γ is equal to 1, both the immediate reward and the future reward are considered equally.

In the proposed protocol, $Q_0(s_0, a_0)$ is initialized as zero. This is because no energy is consumed by the sensor nodes when there is no sending of data packets to the sink. The achievable reward r_t in each learning episode is given in (3).

$$r_t = \max_{v \in \mathcal{V}} \{EC_v\} \quad (3)$$

The EC_v is calculated by the sink after each episode using the difference between the previously estimated sensor residual energy $ESRE_v^{Previous}$ and the currently estimated sensor

residual energy, $ESRE_v^{Current}$. Therefore EC_v is as given in (4).

$$EC_v = ESRE_v^{Previous} - ESRE_v^{Current} \quad (4)$$

The energy model adopted for CRPLOGARL is the same as LACQRP [5]. The reward function is minimized by selecting the MST that has a minimum Q-value using the epsilon-greedy technique [6]. Therefore, given a number, $r \in (0, 1)$ generated randomly in each episode and a likelihood epsilon value, $\epsilon \in [0, 1]$, the learning agent chooses its action in each round using the policy given in (5).

$$a_t = \begin{cases} \underset{a \in A}{\operatorname{argmin}}\{Q_t(s, a)\}, & \text{if } r < 1 - \epsilon \\ \text{Random action}, & \text{otherwise.} \end{cases} \quad (5)$$

The epsilon-greedy strategy employed by the learning agent ensures that the learning agent will converge to the optimal MST(s). The optimal MST at each stage of the network building is the MST with the highest utilization percentage. The utilization percentage of an MST is given in (6).

$$\zeta_{MST} = \frac{\tau_{MST}}{NE} \quad (6)$$

where ζ_{MST} is the utilization percentage of an MST, τ_{MST} is the number of episodes the MST is used, and NE is the number of episodes before the network is rebuilt.

The proposed CRPLOGARL for finding the optimal RT at each stage of network building with a view to maximizing the time taken for the network graph to be disconnected is given in **Algorithm 3**.

Algorithm 3 CRPLOGARL

Input: $G(V, E)$, α , γ , ϵ , Learning round (L)

Output: Optimal MST(s)

- 1: Controller executes **Algorithm 2**
 - 2: **for** $i = 1$ to L **do**
 - 3: Initialize $Q_0(s_0, a_0) = 0$.
 - 4: Initialize s_0 as a random MST
 - 5: Select an MST using (5).
 - 6: Broadcast the MST to all sensor nodes.
 - 7: Sink receive data from the sensors using the MST.
 - 8: Computes the reward using (3).
 - 9: Updates Q^{new} using (2).
 - 10: Updates s_t as the current MST.
 - 11: **if** Any sensor dies, **then**
 - 12: Delete the sensor(s) from $G(V, E)$
 - 13: Delete links connected to the dead sensor(s)
 - 14: Rebuild $G(V, E)$
 - 15: **if** $G(V, E)$ is connected, **then**
 - 16: Do step 1
 - 17: Do step 3 to 14
 - 18: **else**
 - 19: *break*
 - 20: **end if**
 - 21: **end if**
 - 22: **end for**
-

The initialized Q-matrix of **Algorithm 3** depends on the number of generated MSTs, N by **Algorithm 2** that are used as routing tables. This makes the time complexity of **Algorithm 3** to be the same as **Algorithm 2**. Likewise, **Algorithm 3** has a space complexity of $O(N)$.

IV. SIMULATION AND RESULTS DISCUSSIONS

The performance of the proposed genetic MSTs algorithm is first established for convergence using simulations and compared with the All MSTs algorithm [31] using the number of MSTs generated and computation time as the performance indices. Subsequently, the performance of the CRPLOGARL is achieved by simulations using the performance metrics of network lifetime, number of alive sensor nodes (NAN), and computation time. These metrics of the CRPLOGARL are compared with that of the recent LACQRP [5] as a means of validation. The network lifetime is computed as the time taken for the sink not to be reachable by alive sensor node(s). The NAN is the number of alive sensor nodes at the network lifetime. The computation time is the central processing unit (CPU) time taken to achieve the network lifetime.

The CRPLOGARL and LACQRP are coded with python 3.8 under the ‘‘PyCharm’’ development environment. The graphical structure of the WSN is implemented using the python networkx module [32]. The python code is executed on the SLURM (Simple Linux Utility for Resource Management) cluster on the IRIT’s OSIRIM platform. The Computer nodes of the OSIRIM platform adopted are the 4 AMD EPYC 7402 bi-processor computing nodes at 2.8 GHz, with 48 processors and 512 GB of RAM each. These nodes enable more than 24 threads and/or 192 GB of RAM for the same process. The simulation parameters used to implement the CRPLOGARL and the LACQRP are shown in Table 1.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Number of sink	1
Number of sensors	100
Area of deployment	1000 $m \times$ 1000 m
Deployment of Sensor node	Random
Sink coordinate	(500, 500)
Communication range	50 m
Bandwidth	1 $kbps$
Data packet size	1024 bits
Initial sensor energy	1 J to 10 J
Packet generation rate	1/ s to 10/ s
Discount factor	0.0
Learning rate	0.7
Epsilon	0.1
Number of generations	1000
Crossover rate	0.01
Mutation rate	1

As shown in Fig. 1, when the mutation rate is kept constant and the crossover rate is increased for the scenario when the number of network edges is 5000, the genetic algorithm for generating MSTs converges slower. This is because the higher the crossover rate, the lower the number of crossover operations that are performed by the genetic MSTs algorithm.

This leads to a reduced possibility of finding newer individuals in a given generation. There is a direct relationship between the speed of convergence and the crossover rate of the genetic MSTs algorithm. This is because the crossover operation always leads to the formation of an MST of the network graph.

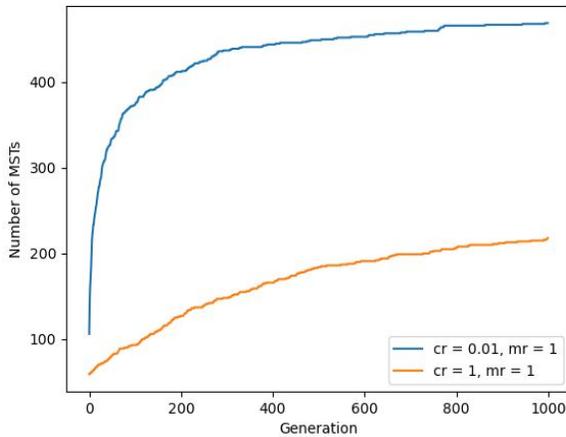


Fig. 1. Number of MSTs in each generation with varied crossover rate.

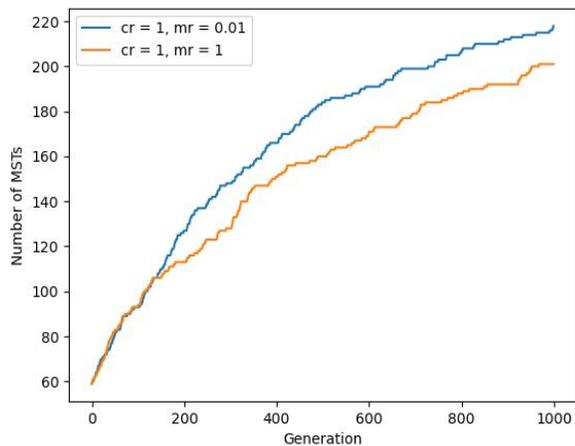


Fig. 2. Number of MSTs in each generation with varied mutation rate.

Consequently, as shown in Fig. 2, when the crossover rate is kept constant and the mutation rate is increased, the lower the number of mutation operations that are performed by the genetic MSTs algorithm. This leads to a reduced possibility of finding newer individuals in a given generation. Though there is no direct relationship between the speed of convergence and the mutation rate of the genetic MSTs algorithm. This is because the mutation operation does not always lead to the formation of an MST of the network graph. The choice of the crossover rate, and mutation rate for the CRPLOGARL is gotten from simulations as shown in Fig. 1 and Fig. 2.

The performance of the proposed genetic MSTs algorithm is ascertained by comparing it with the All MSTs algorithm using the number of generated MSTs and the computation time as the performance metrics. The number of the network

nodes is kept constant at 101, while the number of deployed edges is varied from 1000 to 5000 at an interval of 1000. The edges are formed randomly between nodes while ensuring there are no cycles formed in the network graph. The number of MSTs generated and the corresponding computation time of the proposed genetic MSTs algorithm and the All MSTs algorithm are given in Table II.

TABLE II
PERFORMANCE COMPARISON OF PROPOSED GENETIC MSTs ALGORITHM WITH ALL MSTs ALGORITHM

Edges	GA MSTs		All MSTs	
	MSTs	Time (s)	MSTs	Time (s)
1000	51	5.89	54	8.41
2000	103	12.74	168	23.92
3000	296	17.29	504	62.63
4000	345	26.01	1240	131.15
5000	496	41.65	4262	301.45

The number of MSTs generated by the genetic MSTs algorithm and the All MSTs algorithm increases with the number of randomly deployed edges as shown in Table II. The genetic MSTs algorithm can find an average of 20.73% when compared with all MSTs generated by the All MSTs algorithm with a reduced computation time of 80.48%.

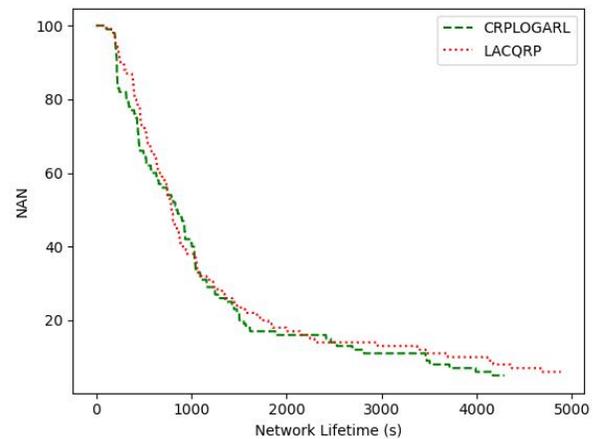


Fig. 3. Number of alive sensors with network lifetime.

Fig. 3 shows the comparison of the number of alive nodes in each round of data transmission of CRPLOGARL with that of the LACQRP when the initial energies and packet generation rate of the sensor nodes are kept arbitrarily at $1 J$ and $1 / s$, respectively. The number of alive sensor nodes of both routing protocols decreases with the network lifetime. The decrease in the number of alive sensor nodes is due to the depletion of the energy sources of the sensor nodes. The CRPLOGARL has 5 alive sensor nodes at the network lifetime. This is against the LACQRP with 6 alive sensor nodes before the sink is no longer reachable for data transmission. This resulted to CRPLOGARL having 16.67% degradation in NAN performance when compared with LACQRP. Therefore the CRPLOGARL makes the network graph to be disconnected faster when compared with LACQRP. This is because CRPLOGARL only uses a

subset of all MSTs that does not contain all the optimal ones. This led to a reduced network lifetime in CRPLOGARL when compared with LACQRP.

Subsequently, the network lifetime with increasing the initial sensor nodes energies for both routing protocols is as shown in Fig. 4. As the initial sensor nodes energies increases, the network lifetime of both routing protocols increases. This is because the network lifetime is proportional to the residual energies of the sensor nodes. The CRPLOGARL has a lower network lifetime performance of 9.88% when compared with LACQRP. This is because of LACQRP generates all MSTs which include the optimal ones and as the initial node energies of the sensor nodes increases, the LACQRP tends to use often the optimal MSTs which led to better network lifetime. However, due to the NP-hardness of generating all MSTs in LACQRP, CRPLOGARL has reduced computation time of 90.87% when compared with LACQRP as shown in Fig. 5.

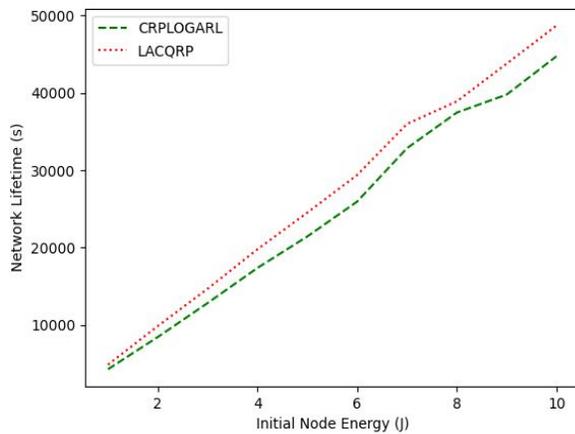


Fig. 4. Network lifetime with initial sensor energy.

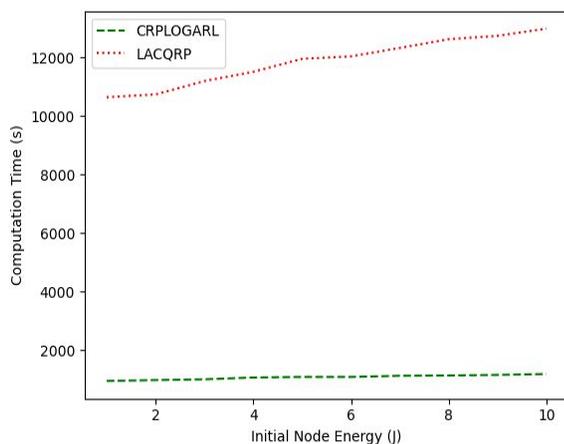


Fig. 5. Computation time with initial sensor energy.

Consequently, the network lifetime with increasing sensors packet generation rate for both routing protocols is as shown in Fig. 6. As the packet generation rate increases, the network

lifetime of both routing protocols decreases. This is because the network lifetime is inversely proportional to the packet generation rates of the sensor nodes. The CRPLOGARL has a lower network lifetime performance of 7.92% when compared with LACQRP.

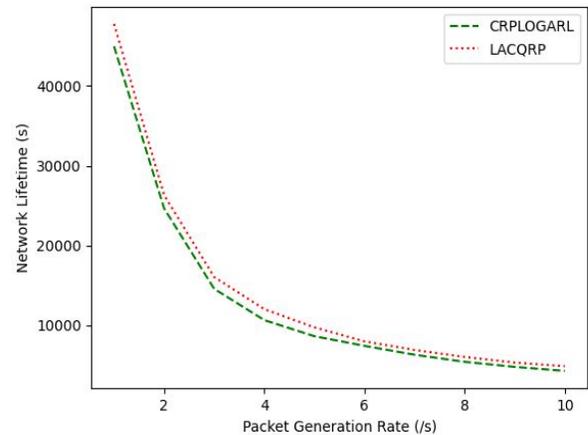


Fig. 6. Network lifetime with rate of packet generation.

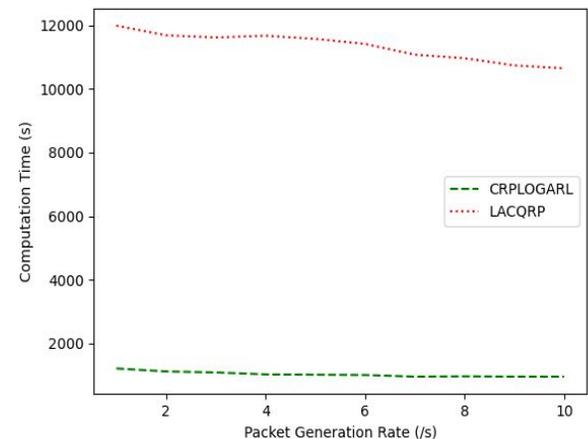


Fig. 7. Computation time with rate of packet generation.

This is because of LACQRP generates all MSTs which include the optimal ones and as the packet generation rates of the sensor nodes increases, the LACQRP tends to use less the optimal MSTs. However, due to the NP-hardness of generating all MSTs in LACQRP, CRPLOGARL has reduced computation time of 90.90% when compared with LACQRP as shown in Fig. 7.

V. CONCLUSION

This paper presented the design of a centralized routing protocol for lifetime optimization using a genetic algorithm and reinforcement learning for WSNs. The sink and the controller resided at the base station of the Software Defined WSN. The controller generated subsets of all minimum spanning trees of the network graph in polynomial time for routing purposes.

The reinforcement learning deployed at the controller learned the optimal MST(s) for lifetime optimization at each stage of the network building by the controller. This maximized the time taken for the sink not to be reachable by alive sensor nodes. The centralized routing protocol for lifetime optimization using genetic algorithm and reinforcement learning has improved computation time when compared with the lifetime-aware centralized routing protocol. This enabled the real-life implementation of the proposed protocol feasible. However, the proposed protocol suffered a reduction in the performance of network lifetime and the number of alive nodes when the network graph is disconnected. This is because the sub-set of the MSTs generated by the proposed protocol does not contain all the optimal MST(s). Future work will consider emulating the Software Defined WSN using Mininet considering real-world parameters of a typical WSN.

ACKNOWLEDGMENT

Elvis Obi was sponsored by the Petroleum Technology Trust Fund (PTDF) Overseas Scholarship Scheme of Nigeria government.

REFERENCES

- [1] D. S. Ibrahim, A. F. Mahdi, and Q. M. Yas, "Challenges and Issues for Wireless Sensor Networks: A Survey," *Journal of Global Scientific Research*, vol. 6, no. 1, pp. 1079-1097, January 2021.
- [2] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7333-7373, January 2020.
- [3] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916-55950, April 2019.
- [4] M. Ndiaye, G. P. Hancke, and A. M. Abu-Mahfouz, "Software Defined Networking for Improved Wireless sensor Network Management: A Survey," *Sensors*, vol. 17, no. 5, pp. 1-32, May 2017.
- [5] E. Obi, Z. Mammeri, and O.E. Ochia, "A Lifetime-Aware Centralized Routing Protocol for Wireless Sensor Networks using Reinforcement Learning," In 17th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2021, pp. 363-368.
- [6] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An introduction," 2nd ed., Cambridge, MA, USA: MIT press, 2018.
- [7] D. A. Whitley, "Genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65-85, June 1994.
- [8] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," In *Advances in neural information processing systems*, vol. 6 pp. 671-678, 1993.
- [9] S. P. Choi and D. Y. Yeung, "Predictive Q-routing: A memory-based reinforcement learning approach to adaptive control," in 9th Proceedings of Neural Information Processing Systems Conference, 1996, pp. 946-951.
- [10] D. Subramanian, P. Druschel, and J. Chen, "Ants and reinforcement learning: A case study in routing in dynamic networks," in 15th Proceedings of International Joint Artificial Intelligence Conference, 1997, pp. 832-839.
- [11] S. Kumar and R. Miikkulainen, "Confidence based Q-routing: An online network routing algorithm," in 16th Proceedings of Artificial Neural Network Engineering Conference, San Francisco, CA, USA, 1998, pp. 758-763.
- [12] L. Peshkin and V. Savova, "Reinforcement learning for adaptive routing," in Proceedings of International Joint Conference on Neural Networks, 2002, pp. 1825-1830.
- [13] D. Chetret, C. K. Tham, and L. Wong, "Reinforcement learning and CMAC-based adaptive routing for MANETs," in 12th Proceedings of IEEE International Conference on Networks, November 2004, pp. 540-544.
- [14] P. Fu, J. Li, and D. Zhang, "Heuristic and distributed QoS route discovery for mobile ad hoc networks," in 5th Proceedings of International Conference on Computer and Information Technology, Shanghai, China, 2005, pp. 512-516.
- [15] Y. Zhang, and M. Fromherz, "Constrained flooding: a robust and efficient routing framework for wireless sensor networks," In 20th IEEE International Conference on Advanced Information Networking and Applications, May 2006, pp. 1-6.
- [16] S. Dong, P. Agrawal, and K. Sivalingam, "Reinforcement learning based geographic routing protocol for UWB wireless sensor network," In IEEE Global Telecommunications Conference, November 2007, pp. 652-656.
- [17] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," In 6th annual international conference on Mobile computing and networking, August 2000, pp. 243-254.
- [18] T. Hu and Y. Fei, "QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796-809, February 2010.
- [19] S. Z. Jafarzadeh and M. H. Y. Moghaddam, "Design of energy-aware QoS routing algorithm in wireless sensor networks using reinforcement learning," In 4th IEEE International Conference on Computer and Knowledge Engineering, May 2014 pp. 722-727.
- [20] W.J. Guo, C.R. Yan, Y.L. Gan, and T. Lu, "An intelligent routing algorithm in wireless sensor networks based on reinforcement learning," *Applied Mechanics and Materials*, Vol. 678, pp. 487-493, October 2014.
- [21] B. Debowski, P. Spachos, and S. Areibi, "Q-learning enhanced gradient based routing for balancing energy consumption in WSNs," In 21st IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks, December 2016 pp. 18-23.
- [22] V. K. Mutombo, S. Y. Shin, and J. Hong, "EBR-RL: energy balancing routing protocol based on reinforcement learning for WSN," In 36th ACM Annual Symposium on Applied Computing Proceedings, April 2021, pp. 1915-1920.
- [23] W. Guo, C. Yan, and T. Lu, "Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, pp. 1-20, February 2019.
- [24] S. E. Bouzid, Y. Serrestou, K. Raouf, and M. N. Omri, "Efficient routing protocol for wireless sensor network based on reinforcement learning," In 5th IEEE International Conference on Advanced Technologies for Signal and Image Processing, October 2020, pp. 1-5.
- [25] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," In Proceedings of the American Mathematical society, vol. 7, no. 1, pp. 48-50, 1956.
- [26] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389-1401, 1957.
- [27] J. Eisner. "State-of-the-art algorithms for minimum spanning trees: A tutorial discussion", University of Pennsylvania, 1997.
- [28] Z. Halim, "Optimizing the minimum spanning tree-based extracted clusters using evolution strategy," *Cluster Computing*, vol.21, no.1, pp. 377-391, March 2018.
- [29] T. A. Almeida, V. N. Souza, F. M. S. Prado, A. Yamakami, and M. T. Takahashi, "Genetic algorithm to solve minimum spanning tree problem with fuzzy parameters using possibility measure," In IEEE NAFIPS Annual Meeting of the North American Fuzzy Information Processing Society, pp. 627-632, June 2005.
- [30] T. A. Almeida, A. Yamakami, and M. T. Takahashi, "An evolutionary approach to solve minimum spanning tree problem with fuzzy parameters," In 6th IEEE International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Vol. 2, November 2005, pp. 203-208.
- [31] T. Yamada, S. Kataoka, and K. Watanabe, "Listing all the minimum spanning trees in an undirected graph," *International Journal of Computer Mathematics*, vol. 87 no. 14, pp. 3175-3185, November 2010.
- [32] A. Hagberg, P. Swart, and S. C. Daniel, "Exploring network structure, dynamics, and function using NetworkX, In 8th SCIPY Conference, August 2008, pp. 11-15.

Integration of Digital Mobile Radio in a Sensor Network

Anders Fongen
 Norwegian Defence University College (FHS)
 Lillehammer, Norway
 email: anders@fongen.no

Abstract—The integration of a sensor network with services from voice communication, text messaging and Global Positioning System (GPS) creates opportunities for improved situational awareness, better safety for field operators, higher confidence in sensor readings and improved return on equipment investment. Digital Mobile Radio (DMR) has been the choice of communication technology for a series of experiments where these potentials have been investigated. Additional technology components used were mostly inexpensive and open source.

Keywords—Internet of things; Sensor Networks; Digital Mobile Radio; MQTT

I. INTRODUCTION

Examples of the services provided by a combination of digital handheld radios with GPS and text messaging services, and a sensor network are listed below:

- Unified user interface - One unit, one battery, one charger used for personal communication and interaction with the sensor network.
- Sensor selection - Based on the location of a field operator, events from nearby sensors are selected for reporting, while others are suppressed.
- Sensor activation - Based on location, sensors near the operator may be deactivated to avoid false alarms/events.
- Automatic geofencing - Based on sensor readings (radiation, toxins, gases), operators may be warned not to enter or exit certain areas.
- Based on the velocity of GPS readings, operators during transport (e.g., while driving a vehicle) can have notification sent as voice messages, while receiving text messages when stationary.

Digital Mobile Radio (DMR) [1] is a non-proprietary digital radio standard maintained by European Telecommunications Standards Institute (ETSI) and supported by a number of manufacturers of radio equipment as well as open source and radio amateur efforts. Contrary to other digital radio systems (e.g., TETRA), DMR operates with the same channel bandwidth as Frequency Modulation (FM) systems and owners of spectrum licences can migrate from FM to DMR using the same channel plan. The DMR physical layer uses Frequency Shift Keying (4FSK) modulation and offers 9600 bits per second (bps) transfer speed, so even FM radios equipped with a data port for 9600 bps can participate in a DMR network by connecting its data port to a DMR modem. Handheld DMR radios can be bought for as low as 120 Euro and are very cost-effective.

The DMR standards also specify the coding and transmission of Internet Protocol version 4 (IPv4) packets, which leads

to the assumption that DMR radios can be used as IP routers in a wide area wireless network. During the experiments, IP routers interconnected by DMR data links were constructed and will be presented and demonstrated in this paper, although the net bitrate offered by the DMR coding standard is far lower than 9600 bps and the applicability of IP services is limited.

DMR used for signalling of sensor readings and the operation of remote devices is a more promising application though, since such services are less hindered by bandwidth limitations and long transmission latency. For Internet-of-Things (IoT) and sensor network application, the DMR radio handset can be employed as a User Agent (UA) for the presentation of events to the operator (in text or voice form) and for the operator to invoke actuators in the IoT network, as well as to allow the operator to act as a sensor and enter information through text messaging.

Moreover, most DMR radio models are equipped with a GPS receiver and are able to report their location to a predefined receiver. The IoT network may use this information to select or modify information sent to the radios, or to provide location based services to their users.

During a field operation, these combined services of the handset greatly enhances the Situational Awareness (SA) of the operators without imposing additional equipment or increased cost. The integration of DMR radio handsets into a sensor network with Message Queuing Telemetry Transport (MQTT) [2] protocol will be demonstrated and presented in this paper as a proof-of-concept. The MQTT protocol is used to build a publish-subscribe based distributed systems, which offers a loose coupling between the system components and a high abstraction level on the information exchange.

The programmed interface between the DMR and MQTT protocol is relatively low level, and enables us to overcome some of the interoperability problems between radios from different manufacturers, e.g., during exchange of text messages.

Apart from the DMR radio handsets, all the technology and software components used in the experiments are open source and non-proprietary.

There are other wireless communication protocols that could be studied during our experiment, e.g., LORA and NB-IoT. These are offered as communication units (modems) without a user interface, and they do not offer GPS positioning or voice communication. They are therefore not taken into account for this experiment.

The contribution of the presented efforts is a middleware solution for others to build on using their preferred equipment of sensors and radios. Most industrial solutions in this area

are found to be vendor-specific stovepipe systems with little attention to interoperability.

The remainder of this paper is organized as follows: Section II will briefly present the technology aspects of DMR, followed by a description of the experimental design in Section III, and the choice of equipment in Section IV. Aspects of the Controller software design are discussed in Section V, while the efforts made on IP routing across DMR links are addressed separately in Section VI. Finally, Section VII presents a summary of the article and identifies future research activities.

II. A BRIEF PRESENTATION OF DMR

DMR is a digital communication standard developed and maintained by ETSI. The physical requirements are well suited for implementation in inexpensive hardware: The 4FSK coding for 9600 bps is available in, e.g., the ADF7021 IC from Analog Devices, which even includes small transceiver with transmission power up to 20 milliwatts.[3] The physical timing and signalling requirements need to be handled by a processor equipped with suitable firmware.

The transmission frames are divided in two independent time slots of 27.5 ms duration, each containing 264 bits. The signalling which takes place within a time slot is called a *burst*, while the structured information inside the burst is called a *block*. With necessary guard time, the two slots take up 60 ms transmission time. Two bit streams can flow independently using different slots, in the same or opposite directions. This division enables two simplex channels to be active at the same time, or a Single Frequency Repeater (SFR) operation using the two slots for in- and out-bound traffic. The two slots may also provide full-duplex communication, e.g., during connection to a telephone network.

One great advantage of DMR is the 12.5 kHz channel width, which facilitates a smooth migration from FM systems, since the licensed channels allocated for the FM service can be directly used by a DMR network. An older digital radio system also standardized by ETSI, TETRA, does not offer this advantage since it requires a large channel bandwidth.

As the gross bit rate of 9600 bps is divided in two time slots, and with bits for synchronization, error detection and signalling, the bit rate for voice communication is reduced to approx. 3600 bps, which requires a codec with heavy compression. The resulting sound quality is adequate for spoken language, but easily garbled if not spoken relatively clearly.[4] The choice of codec is not specified by ETSI, but the proprietary Advanced Multiband Excitation (AMBE+2) [5] algorithm seems to be the de facto standard.

During data communication, a thick layer of error correcting coding (Block Product Turbo Codes, BPTC) [6] reduces the number of information bits per time slot from 196 to 96, as seen on Figure 1. 96 bits every 60 ms minus transmission time for preambles, header blocks and checksums yields a net maximum transmission rate on 1300-1500 bps. This bandwidth is too low for general IP-based delay-sensitive applications like web access, and leaves the DMR link suitable mostly for

delay-tolerant tasks like e-mail, sensor readings and certain SA applications. Beside the transfer speed, one also needs to consider the acceptable duration for a shared channel to be occupied by data traffic. The radio turnaround time will also have to be taken into regard, since it will affect the Round Trip Time (RTT) of a simplex channel and the net throughput of a Transmission Control Protocol (TCP) connection.

DMR radio handsets include a short text messaging service (aka. SMS), which can be sent over a DMR data link in variety of ways, using a variety of character sets. The DMR specifications are vague on these details, and the resulting interoperability between manufacturer's equipment is weak.

A. Talkgroups and DMR-ids

Data and voice transmissions are given *addresses*. Each radio is configured with a unique DMR-id, and transmissions addressed to a DMR-id will only be received by this individual unit. Transmissions can also be sent to *Talk Groups* (TG), which is a way to separate radios into "Communities of Interest". Radios are normally only member of one TG and receives transmissions addressed to that TG in addition to its individual DMR-id. The radio will be allowed to change TG through the user interface. TG could be regarded as similar to the concept of *topics* in a Publish-Subscribe system.

B. Structure of data transmission

In order to send an IPv4 packet from one radio to another, it is first prepended with a 96 bits header block containing DMR-id of sender and receiver, transmission type, confirmation request etc. The IPv4 packet is split into 96 bits fragments which are placed in subsequent data blocks. The last block is padded to correct length and includes a 32 bit Cyclic Redundancy Check (CRC) checksum. Each block is coded using Turbo Coding which extends its length to 196 bits.

As seen on Figure 1, the 196-bit Turbo Coded block is split into two parts which are separated with a 68 bit synchronization and signalling field, bringing the total size of the burst to 264 bits. The signalling field is not Turbo coded so the transmission type of the burst can be observed without decoding the entire block.

When the IPv4 packet is addressed to a DMR unit, the IPv4 addresses used are derived from the DMR-ids of the sending and receiving units, by giving the first 8 bit the value $0 \times C0$ and the DMR-id, regarded as a decimal number, as the 24 next bits. For sending to a TG, a multicast IPv4 address is formed by using $0 \times E8$ as the first 8 bits.

C. Structure of text messages

As earlier stated, the structure and service aspects of text messages show variations between radio manufacturers and many interoperability problems are observed. Equipment from Motorola is observed not to use IPv4, but to embed the text directly into a data block. Other manufacturers are using UDP/IPv4 with a reserved UDP port (port 5016 is recommended by ETSI, port 4007 is also being observed). The syntax of the messages reveals frequent use of the UTF-16LE character set (although ETSI specifies UTF-16GE for

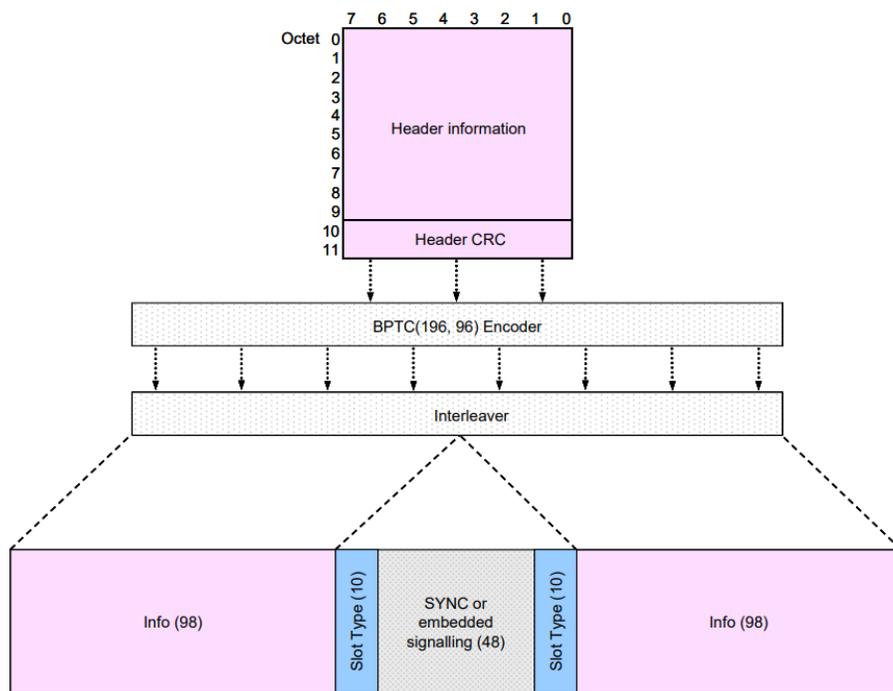


Figure 1. The burst structure of a data header block. Source: [1]

text messages) with leading characters of unknown function, so this part is totally left to the manufacturer to decide.

D. Structure of voice bursts

Voice transmissions are treated a little differently, where the start and end of a transmission are marked with data blocks indicating addresses etc, and the voice blocks in between is a mix of 4 voice blocks for each sync block. The actual sound information is coded with the proprietary AMBE+2 codec.

The transcoding of AMBE+2 sound to coding standards used in Voice over IP (VoIP) systems, e.g., the G.729 [7], would certainly add interesting opportunities to the experiments. Due to the proprietary status of AMBE+2, this effort has been left for future research.

III. EXPERIMENTAL SET-UP

Figure 2 illustrates how DMR may be employed in an IoT/sensor network configuration. The MQTT protocol for Publish-Subscribe (PubSub) distribution was chosen as a backbone for communication between sensors, actuators, decision makers and UAs. Decision makers are actors which contribute to the value chain by processing published data into a refined form for publishing to actuators or other decision makers. The DMR link on the right is shown used as connection between two IP routers, and the link on the left for communication with ordinary radio handsets for user information and actions. The two roles of the DMR links are quite different; While the IP router connection is purely a network layer service, agnostic of what is happening at the transport and application layer, the interface to the MQTT broker is fitted with programming

code specific to the information messages being published and subscribed to during system operation. For the traffic across the IP router (right side), the MQTT-SN protocol was chosen. The MQTT-SN is designed exactly for this type of environment, and employs a compact UDP payload for reasons of brevity and fewer round-trips.

The IP router connection has been tested for transmission rate, latency and RTT. It is not expected to successfully support general IP-based applications, but remain useful for delay-tolerant applications and simple application protocols.

IV. CHOICE OF DMR RADIO EQUIPMENT

This section describes the process of equipment selection. The radio which will serve as a part of the DMR-IoT interface will see requirements quite different from a portable communication handset.

A. Using DMR handsets

All DMR radios provide voice communication and text messaging, more expensive units also provide arrangements (cable) for connecting to a PC through its USB port. For data communication, the radio connection will appear as an MS Windows virtual network interface, which can be operated either by special applications for file transfer and text messaging, or sometimes through standard networking programs like ping, ftp, ssh, etc. This arrangement allows two Windows PCs to communicate through the DMR link, but does not offer a link between networks (like a router is supposed to) since no physical Ethernet interface is provided. The dependence on the Windows OS also rules out applications that require the use of other OSes.

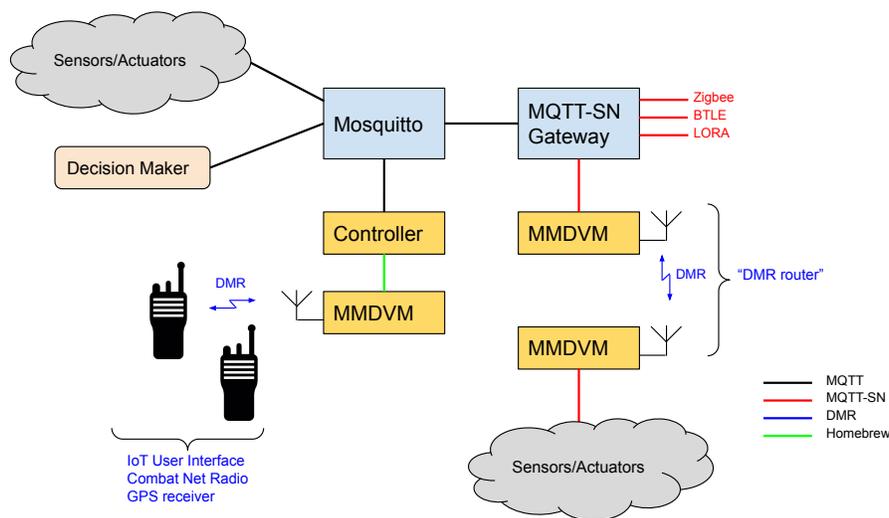


Figure 2. The design of the experimental network



Figure 3. Raspberry Pi with MMDVM-hat

B. Dedicated IoT modules

There are DMR radio modules available for IoT purposes [8], with connectors and an application protocol through which text messages can be sent and voice communication can be set up. These units are relatively inexpensive (e.g., DMR-828 from NiceRF) but provides limited functionality and rather low transmission power (2 W). They are probably exposed to the numerous interoperability problems related to how radios transmit text messages, and have not been studied for the use in this experiment.

C. MMDVM radio modem

In the amateur radio community, equipment called *hotspots* are used to connect DMR radios to world wide repeater networks in order to provide global connectivity. Hotspots are found as simplex link nodes or as repeaters, but with a low transmission power suited for private use within a household. A hotspot is equipped with a simple DMR radio module with up to 20 mW transmission power, and an IP interface for connecting to the repeater network through Internet. The Internet connection carries all voice and data traffic to a server

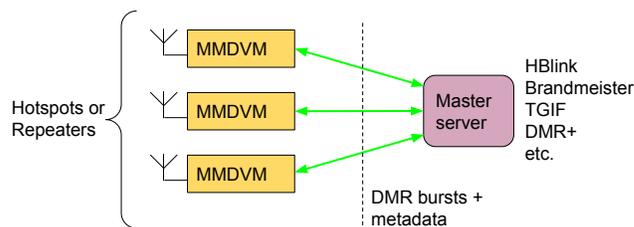


Figure 4. Structure of the DMR hotspot/repeater network as found in the Brandmeister system

which dispatches the traffic to other hotspots (and public DMR repeaters) in order to connect to other radios. The principle of this arrangement is shown on Figure 4.

The hardware of a hotspot will most often consist of a Raspberry Pi single-card computer and a “hat” of a MMDVM (Multi-Mode Digital Voice Modem) unit based on the ADF7021 IC and a micro controller with firmware for the operation of the DMR protocol, as shown on Figure 3. The MMDVM is also available without a radio module, for wired connection to radios with better receivers and higher transmission power.

D. The advantages of a low-level control protocol

The protocol between the hotspot and the Internet servers is called *MMDVM Homebrew* (shown as green lines on Figure 4) and is partially documented and easily reverse engineered. For voice, text and data traffic sent and received, the “raw” data bursts in the time slots are transmitted (layer 2 PDUs) together with some metadata. In the experiments described in this paper, the component called “Master server” has been replaced with a “Controller” component which emulates the Master server interface to the MMDVM and provides a service for conversion to/from the MQTT protocol. The Controller software was developed by the author as a part of this experiment, and its design will be described in the next section.

Although there is relatively more work in coding and decoding layer 2 Protocol Data Units (PDUs), it also allows the interpretation and construction of messages used by different radio manufacturers. For the sake of better interoperability between radios from different manufacturers, using this low-level interface is an appealing alternative, also due to the low cost of the MMDVM module and the availability of related source code. One serious drawback with the Homebrew protocol is that it is UDP-based, which means that there is no easy way to throttle the traffic from a connected computer, which may lead to buffer overrun problems.

V. CONTROLLER SOFTWARE DESIGN

The chosen configuration, as shown in Figure 2, employs a Controller sitting between the MQTT broker service (Mosquitto) and the MMDVM unit as an adaption layer. Its responsibilities are:

- Emulate the Homebrew protocol to the MMDVM
- Construct and deconstruct the DMR bursts
- Route IPv4 packets between the MMDVM and the connected network
- Present an interface to the MQTT protocol and Mosquitto, the MQTT broker
- Construct text and voice messages from received MQTT publications
- Construct and publish MQTT publications from received text messages and GPS locations
- Record and playback voice messages

The chosen programming language is Python3, mostly due to the availability of library modules for operating on DMR PDU structures and interface to MQTT brokers, but also for its convenient programming style. The Controller is running a Linux OS, chosen for its easy access to raw sockets, threads and locks.

Figure 5 shows an interaction diagram with the flow of information between the different components. The color of the invocation lines indicates the protocol.

A. Coding and decoding

The process of DMR PDU coding and decoding is the most complicated part of the Controller program. There are Python libraries for parts of these processes (`dmr_utils3`), but they are mostly targeted towards voice communication and had to be modified for the purpose of constructing and deconstructing data traffic.

During data traffic, the IPv4 packet is split up in a series of blocks, each holding a 12 bytes fragment. A header block (cf. Figure 1) is prepended and the final data block is padded and given a 32 bit CRC value. The blocks also carry fields for synchronization and signalling, and the different parts are given different error correction and turbo coding. These tasks must be handled by the Controller, since the MMDVM simply passes the blocks on to the radio interface.

The construction of text messages uses much of the same code, since the messages are contained in UDP/IPv4 PDUs. As mentioned earlier, the UDP and IPv4 header details and

the syntax of the UDP payload depend on the radio manufacturer so programming code must be adapted to gloss over interoperability problems.

B. Synchronisation and flow control

The Homebrew protocol is simple and does not offer any flow control mechanisms, since all messages are included in UDP segments and no receiver feedback is provided. Possible buffer overrun problems could be mitigated through a choking mechanisms in the Controller, with the risk of creating premature timeouts in communication endpoints in the connected networks.

For short text messages and voice clips sent to the MMDVM, buffer overrun may not be a frequent problem. When the MMDVM engages in IP routing, however, additional steps may have to be taken, which will be discussed in Section VI.

C. The relation between DMR TGs and PubSub topics

On the matter of interfacing a DMR system to a PubSub distributed systems, the mapping between DMR Talk Groups (TG) and PubSub Topics must be addressed. PubSub topics in the case of the MQTT protocol are organized as a taxonomy. Clients can subscribe to a single topic or to a subtree of topics. Clients can thus choose to subscribe to all information in a general area of topics, or to specific topics with a narrow focus. Likewise, clients can publish information using any topic to indicate the nature and specificity of the information. A topic taxonomy which efficiently reflects the business logic is a core element of PubSub system design.

In a DMR system, the concept of Talk Groups can be used to indicate a topic of a voice or text transmission, but TGs are not hierarchically organized. Besides, the typical design of a DMR radio does not allow a radio to associate with different TGs for voice and text. Given that the radio is being used also for spoken communication in an operational environment, the TG does not appear as a useful tool to specify groups of receivers for publications. Individual transmissions to each radio, based on their DMR-id, will only be defensible on a small scale of operation due to the scarcity of communication capacity.

On the other hand, voice and text messages sent from a DMR radio can be annotated with any TG, which would be mapped into a MQTT topic before rewriting it to a common publication syntax and passed on to the PubSub broker. The radio is equipped with an address book where names can be given to TGs, which allows the operator to relate to meaningful topic names rather than TG numbers.

D. Voice recording and playback

In the absence of a proprietary software codec for AMBE+2, it is still possible to use the AMBE+2 codec embedded in the DMR radio handset for the recording and playback of voice messages.

All voice traffic will be sent from the MMDVM to the Controller over the Homebrew protocol as DMR bursts and can be stored there in the AMBE+2 coded form. Likewise,

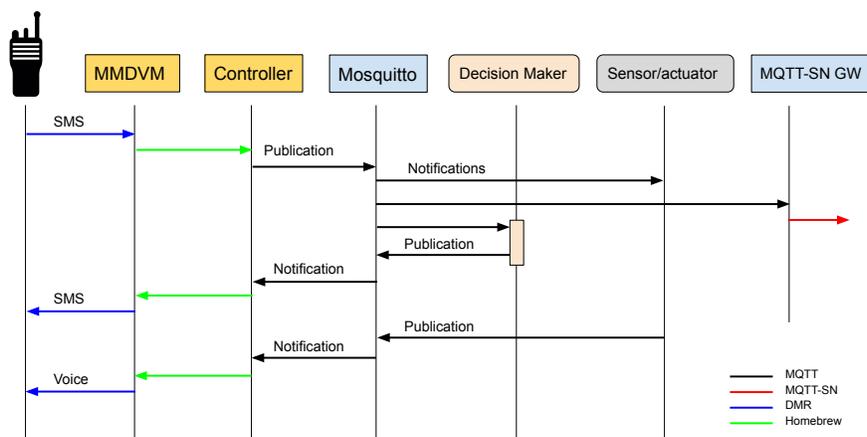


Figure 5. Protocol interaction diagram. The coloring of the invocation lines indicate the protocol used.

stored voice bursts can be sent from the Controller to the MMDVM which will be transmitted and received by the radios within range.

This arrangement does not allow for voice traffic across technology domains, but still is considered a useful addition to the otherwise text-centric information exchange. A set of voice messages can be recorded and assigned an index as a part of the system configuration. The voice clips can be sent to a radio unit or a TG as responses to MQTT notifications. The present implementation allows any radio to record and index voice clips, which allows for interesting voice information exchange: Voice clips can represent the most recent status or observation recorded by a field operator, which can be announced to other operators on a variety of conditions, including interactive queries.

VI. IP ROUTED ACROSS A DMR LINK

For the final part of this presentation, the use of a DMR link for general IP routing will be explained and demonstrated. The principle for such an arrangement is quite simply explained: The Controller uses a raw socket to take in full Ethernet frames, and decides if they should be routed across the DMR link. This decision is made on the basis of both MAC- and IP addresses. Frames addressed to the Controller IP interface are not routed, nor is any multicast or broadcast frames. The IP packet is split into a series of DMR data blocks as explained in Section II-B which are sent to the MMDVM using the Homebrew protocol. The receiving MMDVM restores the IP packet from the DMR data blocks and sends it through a raw socket interface.

Several problems were expected to appear when a simplex DMR channel is used to support two-way IP traffic, possible several two-way streams. No media access protocol is defined in DMR other than a “polite” mechanism which is a primitive form of Carrier Sense. No collision avoidance protocol is in place, nor any backoff method, etc.

Moreover, the equipment used for the experiment activated the transmitter even if the receiver was actively receiving a

TABLE I
RESULTING RTT AND BITRATE OVER SIMPLEX AND DUPLEX DMR LINKS

Link type	RTT (secs)	Bitrate (bps)
Simplex link	3.6	430
Duplex link	2.4	1100

signal (like how Push-to-talk (PTT) on a radio would override the receiver). Slow turnaround in the radios also caused problems for the reception.

Even though a simple ping command would successfully operate on the link, it became clear that a duplex link is necessary, which could operate in both directions simultaneously. Two way communication can be accomplished using the two time slots, or using two frequencies. The MMDVM units may have two independent radios, but their firmware does not support such configuration. The choice was made to use two pairs of MMDVMs to make two independent DMR links on different frequencies, and to modify the Controller software to use the two connected MMDVMs for traffic in opposite directions. This arrangement is illustrated in Figure 6.

This arrangement removed the channel access and collision problems, as well as problems related to radio turnaround time. Remaining problem were:

- Premature timeouts due to low transmission rate, both on the TCP protocol and certain application protocols.
- Buffer overrun in the sending MMDVM, due to lack of synchronized flow between the Controller and the MMDVM.

These problems affected the ability to use the iperf utility to measure communication rate. The familiar ping command combined with a stopwatch allowed for the estimation of Round Trip Time (RTT) and bit rate:

The average RTT for default ping message size (80 bytes), and the average bit rate when using packets of 240 bytes are shown in Table I. With a bit of patience, it was possible to fetch websites and make ssh connections over this duplex link, but in practice it would be better suited for short signalling

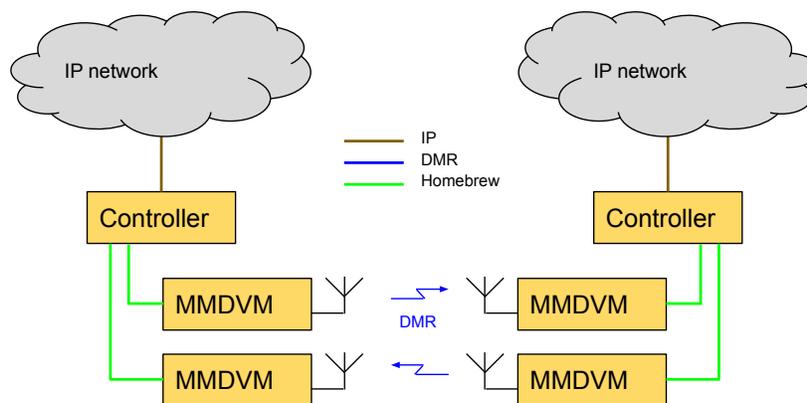


Figure 6. IP router arrangement using separate DMR links for opposite direction.

and short messages in a stop-and-wait type protocol. Any streaming application will fail due to the lack of MMDVM-Controller synchronization mentioned above.

VII. CONCLUSION

This paper has demonstrated a proof-of-concept effort on the integration of DMR technology with a sensor network using Publish-Subscribe technology and the MQTT protocol. The integration arrangement allows sensor readings to be reported to individual radios or to talk groups, using text or voice. It also allows the radio operators to send text messages and GPS position reports to the MQTT broker as publications. Additionally, the experiment demonstrates a voice recording service as well as IP routing across DMR links.

The combination of these services offers an improved situation awareness to field operations, better utilization of the handheld radios, and access to human observations into a sensor networks.

The use of MQTT protocol is used as an example for information dissemination in an IP network, other forms for distributed middleware will surely work in a similar manner.

For this laboratory experiment to evolve into a more realistic field study, there will be a need to use equipment better suited for the task than the MMDVM firmware. Modification to the firmware could be one option, or using different hardware, e.g., dedicated IoT/DMR modules. Voice traffic exchange

between DMR and VoIP systems requires the availability of a AMBE+2 codec and is an appealing extension to the present arrangement, which would also allow text-to-speech services to be implemented. Further studies will also look into the configuration and information presentation in a realistic field operation scenario, as well as usability aspects and interoperability issues.

REFERENCES

- [1] "The DMR Standards," <https://www.dmrassociation.org/dmr-standards.html>, European Telecommunications Standards Institute, [Online: Accessed 29.07.2022].
- [2] "MQTT Specification," <https://mqtt.org/mqtt-specification/>, OASIS, [Online: Accessed 29.07.2022].
- [3] "ADF7021," <https://www.analog.com/en/products/adf7021.html>, Analog Devices, [Online: Accessed 29.07.2022].
- [4] S. Campos Neto, F. Corcoran, J. Phipps, and S. Dimolitsas, "Performance assessment of 4.8 kbit/s ambe coding under aeronautical environmental conditions," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, 1996, pp. 499–502 vol. 1.
- [5] "AMBE+2," https://www.dvsinc.com/soft_products/ambe_p2.shtml, Digital Voice Systems, Inc., [Online: Accessed 29.07.2022].
- [6] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1003–1010, 1998.
- [7] L. Li, Q. Huang, and W. Wan, "Design of converting low-rate speech codec between ambe and g.729," in *IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013)*, 2013, pp. 185–188.
- [8] I. Ganchev, Z. Ji, and M. O'Droma, "Designing a low-cost dmr module for use in m2m/iot applications," in *2018 2nd URSI Atlantic Radio Science Meeting (AT-RASC)*, 2018, pp. 1–2.