



# **MOBILITY 2021**

The Eleventh International Conference on Mobile Services, Resources, and Users

ISBN: 978-1-61208-866-2

May 30th – June 3<sup>rd</sup>, 2021

**MOBILITY 2021 Editors**

Omar Sami Oubbati, University of Laghouat, Algeria

# MOBILITY 2021

## Foreword

The Eleventh International Conference on Mobile Services, Resources, and Users (MOBILITY 2020), held between May 30 – June 3rd, 2021, continued a series of events dedicated to mobility-at-large, dealing with challenges raised by mobile services and applications considering user, device and service mobility.

Users increasingly rely on devices in different mobile scenarios and situations. "Everything is mobile", and mobility is now ubiquitous. Services are supported in mobile environments, through smart devices and enabling software. While there are well known mobile services, the extension to mobile communities and on-demand mobility requires appropriate mobile radios, middleware and interfacing. Mobility management becomes more complex, but is essential for every business. Mobile wireless communications, including vehicular technologies bring new requirements for ad hoc networking, topology control and interface standardization.

We take here the opportunity to warmly thank all the members of the MOBILITY 2021 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to MOBILITY 2021. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the MOBILITY 2021 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that MOBILITY 2021 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the areas of mobile services, resources and users.

### **MOBILITY 2021 Chairs:**

#### **MOBILITY 2021 Steering Committee**

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France

Omar Sami Oubbati, University of Laghouat, Algeria

#### **MOBILITY 2021 Publicity Chair**

Marta Botella-Campos, Universitat Politecnica de Valencia, Spain

Daniel Basterretxea, Universitat Politecnica de Valencia, Spain

# MOBILITY 2021

## COMMITTEE

### MOBILITY 2021 Steering Committee

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France  
Omar Sami Oubbati, University of Laghouat, Algeria

### MOBILITY 2021 Publicity Chairs

Marta Botella-Campos, Universitat Politècnica de Valencia, Spain  
Daniel Basterretxea, Universitat Politècnica de Valencia, Spain

### MOBILITY 2021 Technical Program Committee

Mohamed A.Aboulhassan, Pharos University, Egypt  
Osama M.F. Abu-Sharkh, Princess Sumaya University for Technology, Amman, Jordan  
Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France  
Mohamad Badra, Zayed University, Dubai, UAE  
Matthias Baldauf, Eastern Switzerland University of Applied Sciences, Switzerland  
Chaity Banerjee, University of Central Florida, USA  
Leandro Becker, Federal University of Santa Catarina (UFSC), Brazil  
Peter Brída, University of Žilina, Slovakia  
Ivana Bridova, University of Zilina, Slovakia  
Simeon Calvert, Delft University of Technology, Netherlands  
Carlos Carrascosa, Universitat Politècnica de València, Spain  
Chao Chen, Purdue University Fort Wayne, USA  
Daniel Delahaye, Ecole Nationale de l'Aviation Civile (ENAC), Toulouse France  
Anatoli Djanatliev, University of Erlangen-Nuremberg, Germany  
Mohand Djeziri, Aix Marseille University, France  
Mohamed El Kamili, Higher School of Technology - Hassan II University of Casablanca, Morocco  
Ayman El-Saleh, A'Sharqiyah University, Oman  
Brahim Elmaroud, Higher Institute of Maritime Studies - Casablanca, Morocco  
Javier Fabra, Universidad de Zaragoza, Spain  
Hassen Fourati, University Grenoble Alpes, France  
Jordi Garcia, CRAAX Lab - UPC BarcelonaTECH, Spain  
Gelayol Golcarenenji, University of the West of Scotland, UK  
Javier Ibanez-Guzman, Renault, France  
Sergio Ilarri, University of Zaragoza, Spain  
Jin-Hwan Jeong, SK telecom, South Korea  
Christian Jung, Fraunhofer IESE, Germany  
Georgios Kambourakis, University of the Aegean, Greece  
Hassan A. Karimi, University of Pittsburgh, USA  
Hamzeh Khalili, Fundació i2CAT, Barcelona, Spain  
Imran Khan, Insight Center for Data Analytics | University College Cork, Ireland  
Francesca Martelli, IIT-CNR, Pisa, Italy

Ignacio Martinez-Alpiste, University of the West of Scotland, UK  
Antonio Matencio-Escolar, University of the West of Scotland, UK  
Subhasish Mazumdar, New Mexico Tech, USA  
Weizhi Meng, Technical University of Denmark, Denmark  
Farshad Miramirkhani, Isik University, Turkey  
Javad Mohammadi, Carnegie Mellon University, USA  
Alireza Morsali, McGill University, Canada  
Tathagata Mukherjee, The University of Alabama in Huntsville, USA  
Tatsuo Nakajima, Waseda University, Japan  
Omar Sami Oubati, University of Laghouat, Algeria  
Hyoshin (John) Park, North Carolina A&T State University, USA  
Wuxu Peng, Texas State University, USA  
Alejandro Ramirez, Siemens AG, Germany  
Aurora Ramirez, University of Córdoba, Spain  
Ruben Ricart-Sanchez, University of the West of Scotland, UK  
Anna Lina Ruscelli, TeCIP Institute - Scuola Superiore Sant'Anna, Pisa, Italy  
Mahsa Sadeghi Ghahroudi, Glasgow Caledonian University, UK  
Viliam Sarian, Scientific Research Institute for Radio, Russia  
Régine Seidowsky, COSYS-GRETTIA | Univ. Gustave Eiffel | IFSTTAR, France  
Alireza Shahrabi, Glasgow Caledonian University, Scotland, UK  
Haichen Shen, Amazon Web Services, USA  
Danny Soroker, IBM T.J. Watson Research Center, USA  
Harald Sternberg, HafenCity Universität Hamburg, Germany  
Daxin Tian, Beihang University, Beijing, China  
Markku Turunen, Tampere University, Finland  
Sudip Vhaduri, Fordham University, USA  
Dario Vieira, Efrei Paris, France  
Ulrich Walder, Graz University of Technology, Austria  
Rainer Wasinger, University of Applied Sciences Zwickau, Germany  
Mudasser F. Wyne, National University, USA  
Cong-Cong Xing, Nicholls State University, USA  
Renjun Xu, Zhejiang University, China  
Hong Yang, Nokia Bell Labs, Murray Hill, USA  
Paul Yoo, Birkbeck, University of London, UK  
Lisu Yu, Nanchang University, China  
Mariusz Zal, Poznan University of Technology, Poland  
Jianhua Zhang, Clarkson University, USA  
Xiao Zhu, University of Michigan, USA  
Wolfgang Zirwas, Nokia Bell Labs, Munich, Germany  
Makia Zmitri, CNRS/GIPSA-Lab, France  
Kamil Zyla, Lublin University of Technology, Poland

## 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

Dual Decomposition Method for Reliable Allocations of Wireless Network Resources <i>Erkki Laitinen, Igor Konnov, and Alexey Kashuba</i>	1
A Conceptual Digital Twin for 5G Indoor Navigation <i>Vladeta Stojanovic, Hossein Shoushtari, Cigdem Askar, Annette Scheider, Caroline Schuldt, Nils Hellweg, and Harald Sternberg</i>	5
SDN-based MANET Using Existing OpenFlow Protocol <i>Rabia Saleh, Idris Skloul, and Lilia Georgieva</i>	15

# Dual Decomposition Method for Reliable Allocations of Wireless Network Resources

Igor Konnov  
*dept. of Sys. Anal. and Inf. Tech.*  
 Kazan Federal University  
 Kazan, Russia 420008  
 email: konn-igor@ya.ru

Alexey Kashuba  
*dept. of Sys. Anal. and Inf. Tech.*  
 Kazan Federal University  
 Kazan, Russia 420008  
 email: leksser@rambler.ru

Erkki Laitinen  
*unit of Math. Sci.*  
 University of Oulu  
 Oulu, Finland FI-90014  
 email: erkki.laitinen@oulu.fi

**Abstract**—In the paper, we consider the problem of allocation of network resources in telecommunication networks with respect to both utility and reliability goals. We suggest a solution method based on decomposition and gradient methods for this problem. We present numerical results for the suggested method on test examples.

**Index Terms**—Telecommunication networks; allocation of links; decomposition method; gradient method.

## I. INTRODUCTION

The current development of information technologies and telecommunications gives rise to new control problems related to efficient transmission of information and allocation of limited network resources. All these problems are determined on distributed systems where the spatial location of elements is taken into account. Due to strong variability and increasing demand of different wireless telecommunication services, fixed allocation rules usually lead to serious congestion effects and inefficient utilization of network resources despite the presence of very powerful processing and transmission devices. This situation forces one to replace the fixed allocation rules with more flexible mechanisms, which are based on proper mathematical models; see e.g., [1]– [3]. For example, solution methods for network resource allocation based on optimization formulations of network manager problems and decomposition techniques were presented in [4] [5]. In these problems, the goal function is the total network profit obtained from the total income of users payments and the implementation costs of the network. Otherwise, the total network users utility can serve as a goal function.

At the same time, wireless networks should be reliable with respect to various attacks. The most commonly seen in wireless networks are eavesdropping in which attackers aim at acquiring important/private information of users, jamming and Distributed Denial of Service (DDoS) attacks, which attempt to interfere and disrupt network operations by exhausting the resources available to legitimate systems and users. These attacks may lead to degrading the network performance and Quality of Service (QoS), as well as losing important data, reputations, and revenue; see e.g. [6]–[9]. Hence, the network resource allocation problem should take into account reliability estimates.

In [10], we considered a problem of telecommunication network link resources allocation among users under reliability control of network connections with the pre-defined non-reliability level. For this problem, it was suggested a penalty method. This method attained a solution, but its convergence does not allow one to attain high accuracy of solutions.

In this paper, we consider also the problem of allocation of link resources in telecommunication networks with respect to both utility and reliability goals. However, unlike [10], we do not indicate any pre-defined non-reliability level. Our cost function is a difference of the total network utility and non-reliability. By using the dual Lagrangian method with respect to the balance constraint, we replace the initial problem with an unconstrained optimization problem, where calculation of the cost function value leads to independent solution of single-dimensional problems. We present results of computational experiments which confirm the applicability of the new methods.

In Section 2 we describe the link resources allocation problem in telecommunication networks with respect to both utility and reliability constraints. In Section 3 we describe how to apply the dual Lagrange method for solving the original optimization problem. Finally, in Section 4 we give computational results which confirm rather stable performance of the method.

## II. PROBLEM DESCRIPTION

We first take the optimal link distribution problem in computer and telecommunication data transmission networks, which was suggested in [11]. This model describes a network that contains a set of transmission links (arcs)  $L$  and accomplishes some submitted data transmission requirements from a set of selected pairs of origin-destination vertices  $I$  within a fixed time period. Denote by  $x_i$  and  $\alpha_i$  the current and maximal value of data transmission for pair demand  $i$ , respectively, and by  $c_l$  the capacity of link  $l$ . Each pair demand is associated with a unique data transmission path, hence each link  $l$  is associated uniquely with the set  $I_l$  of pairs of origin-destination vertices, whose transmission paths contain this link. For each pair demand  $x_i$ , we denote by  $u_i(x_i)$  the

utility value at this data transmission volume. Then, we can write the network utility maximization problem as follows:

$$\max \rightarrow \sum_{i \in I} u_i(x_i)$$

subject to

$$\begin{aligned} \sum_{i \in I_l} x_i &\leq c_l, \quad l \in L; \\ 0 &\leq x_i \leq \alpha_i, \quad i \in I. \end{aligned}$$

If the functions  $u_i(x_i)$  are concave, this is a convex optimization problem.

Let us now consider the same telecommunication network where the reliability factor should be taken into account. Namely, we associate the reliability to each arc flow and determine  $\mu_l(f_l)$  as the non-reliability of the  $l$ -th arc having the flow  $f_l$  for  $l \in L$ . Then  $\sum_{l \in L} \mu_l(f_l)$  is the total network non-reliability and we formulate the network manager problem as follows:

$$\max \rightarrow \sum_{i \in I} u_i(x_i) - \sum_{l \in L} \mu_l(f_l), \quad (1)$$

subject to

$$\sum_{i \in I_l} x_i = f_l, \quad l \in L; \quad (2)$$

$$0 \leq f_l \leq c_l, \quad l \in L; \quad (3)$$

$$0 \leq x_i \leq \alpha_i, \quad i \in I. \quad (4)$$

If the functions  $u_i(x_i)$  and  $-\mu_l(f_l)$  are concave, this is a convex optimization problem with the polyhedral feasible set. However, solution of problem (1)–(4) is not so easy due to large dimensionality and inexact data. In this paper we consider the case where the functions  $u_i(x_i)$  and  $-\mu_l(f_l)$  are strictly concave. Then, we can apply the known dual decomposition technique.

### III. DUAL DECOMPOSITION METHOD

Let us define the Lagrange function of problem (1)–(4) as follows:

$$L(x, f, y) = \sum_{i \in I} u_i(x_i) - \sum_{l \in L} \mu_l(f_l) + \sum_{l \in L} y_l \left( \sum_{i \in I_l} x_i - f_l \right)$$

for

$$x \in X = \prod_{i \in I} [0, \alpha_i] \quad \text{and} \quad f \in F = \prod_{l \in L} [0, c_l].$$

By duality, we can replace problem (1)–(4) with the dual unconstrained optimization problem:

$$\min \rightarrow \varphi(y), \quad (5)$$

where

$$\varphi(y) = \max_{x \in X, f \in F} L(x, f, y). \quad (6)$$

Clearly, the dual cost function  $\varphi$  is convex. Moreover, under the strict convexity of the functions  $\mu_l$  and the strict concavity of the functions  $u_i$  it is differentiable. Calculation of its

value and its gradient is rather simple and decomposed into independent solution of single-dimensional problems. Denote by  $L_i$  the set of links belonging to the path associated with the origin-destination pair  $i$ . By definition,

$$\begin{aligned} \varphi(y) &= \max_{x \in X, f \in F} L(x, f, y) \\ &= \sum_{i \in I} \max_{0 \leq x_i \leq \alpha_i} \left\{ u_i(x_i) + x_i \sum_{l \in L_i} y_l \right\} \\ &\quad - \sum_{l \in L} \min_{0 \leq f_l \leq c_l} \{ \mu_l(f_l) + y_l f_l \} \\ &= \sum_{i \in I} \left\{ u_i(x_i(y)) + x_i(y) \sum_{l \in L_i} y_l \right\} \\ &\quad - \sum_{l \in L} \{ \mu_l(f_l(y)) + y_l f_l(y) \}, \end{aligned}$$

where  $x_i(y)$  and  $f_l(y)$  are unique solutions of the single-dimensional optimization problems

$$\max_{0 \leq x_i \leq \alpha_i} \left\{ u_i(x_i) + x_i \sum_{l \in L_i} y_l \right\}$$

and

$$\min_{0 \leq f_l \leq c_l} \{ \mu_l(f_l) + y_l f_l \},$$

respectively. Next, we obtain

$$\frac{\partial \varphi(y)}{\partial y_l} = \sum_{i \in I_l} x_i(y) - f_l(y), \quad l \in L.$$

These properties enable us to apply the usual Uzawa gradient method to find a solution of the dual problem (5):

$$y^{k+1} = y^k - \lambda_k \varphi'(y^k), \quad \lambda_k > 0.$$

### IV. COMPUTATIONAL EXPERIMENTS

As part of the work, a numerical study of the suggested method was carried out. The method was implemented in C++ with a PC with the following facilities: Intel(R) Core(TM) i7-4500, CPU 1.80 GHz, RAM 6 Gb.

In the experiments, we used quadratic functions of utility of origin-destination pairs (QuadC)

$$u_i(x_i) = u_{1,i} x_i^2 + u_{0,i} x_i, \quad u_{1,i} < 0, u_{0,i} > 0, i \in I,$$

quadratic functions of non-reliability of arcs (QuadA)

$$\mu_l(f_l) = \mu_{1,l} f_l^2 + \mu_{0,l} f_l, \quad \mu_{1,l}, \mu_{0,l} > 0, l \in L,$$

logarithmic functions of utility of origin-destination pairs (LogC)

$$u_i(x_i) = u_{2,i} \ln(u_{0,i} + u_{1,i} x_i), \quad u_{j,i} > 0, j = 0, \dots, 2, i \in I,$$

and logarithmic functions of non-reliability of arcs (LogA)

$$\mu_l(x_i) = \mu_{0,l} f_l - \ln(1 + \mu_{1,l} f_l), \quad \mu_{0,l}, \mu_{1,l} > 0, l \in L.$$

All the arcs and origin-destination pairs were indexed as  $l = 0, \dots, |L| - 1$  ( $|L|$  is the cardinality of  $L$ ) and  $i = 0, \dots, |I| - 1$  ( $|I|$  is the cardinality  $I$ ), respectively.

The coefficients  $\mu_{1,l}$ ,  $\mu_{0,l}$ ,  $u_{0,i}$ ,  $u_{1,i}$ , and  $u_{2,i}$  were formed on the basis of trigonometric functions:

(i) for the functions QuadC

$$u_{0,i} = 2|\sin(2i+2)| + 2, \quad u_{1,i} = -|\cos(2i+1)| - 1,$$

(ii) for the functions QuadA

$$\mu_{0,l} = |\cos(l+1)| + 3, \quad \mu_{1,l} = 2|\sin(2l+2)| + 1,$$

(iii) for the functions LogC

$$u_{0,i} = 2|\sin(2i+2)| + 1, \quad u_{1,i} = |\sin(i+2)| + 1, \\ u_{2,i} = 3|\sin(2i+2)| + 1,$$

(iv) for the functions LogA

$$\mu_{0,l} = 10|\cos(l+1)| + 10, \quad \mu_{1,l} = 2|\cos(2l+2)| + 1.$$

The maximal arc flow capacity  $c_l$  was selected in [1] [10] as follows:

$$c_l = 10|\cos(l+3)| + 1.$$

The maximal path flow capacity  $\alpha_i$  associated with a origin-destination pair was selected in [1] [7] as follows:

$$\alpha_i = 7|\sin(i)| + 1.$$

The stepsize parameter  $\lambda_k$  in the dual gradient method was fixed and equal to 0.6.

In our tests, we used the following combinations of functions: QuadA-LogC, QuadA-QuadC, LogA-LogC. The distribution of the available arcs across the origin-destination pairs was carried out either uniformly or according to the normal distribution law. In the gradient method, we used two different initial points: the vector  $e$  of units and vector  $100e$ .

We now introduce additional notations:

- 1)  $\varepsilon$  is the accuracy of finding solution of the problem,
- 2)  $T_{\varepsilon,1}$  and  $T_{\varepsilon,100}$  are the time (in seconds) of the method with the starting point  $e$  and  $100e$ , respectively,
- 3)  $I_{\varepsilon,1}$  and  $I_{\varepsilon,100}$  are the numbers of iterations spent searching for a solution to the problem with the starting point  $e$  and  $100e$ , respectively.

The gradient method was stopped if the norm  $\|\varphi'(y^k)\|$  appeared less than  $\varepsilon$ . In Tables I–IV we give the results of finding a solution of the problem with QuadA-LogC combination of functions. In Table I, we give the results for the case where  $|I| = 620$ ,  $|L| = 310$  and for different values  $\varepsilon$ . In Table II, we give the results for the case where  $|I| = 310$ ,  $|L| = 620$  and for different values  $\varepsilon$ . In Table III, we give the results for the case where  $\varepsilon = 10^{-2}$ ,  $|L| = 310$  and for different values  $|I|$ . In Table IV, we give the results for the case where  $\varepsilon = 10^{-2}$ ,  $|I| = 310$  and for different values  $|L|$ . In this series of numerical experiments the solution time was less than one second. The experiments for the cases QuadA-QuadC and LogA-LogC gave similar results, which are given in Tables V–VIII and IX–XII, respectively.

## V. CONCLUSIONS

We presented a general problem of allocation of network resources in telecommunication networks using both utility and reliability factors. We suggest to apply the dual decomposition method to this problem. The results of computational experiments confirmed rather stable performance of the method.

## ACKNOWLEDGMENT

The first author was supported by Russian Foundation for Basic Research, project No. 19-01-00431 and by grant No. 331833 from Academy of Finland. The third author was supported by grant No. 331954 from Academy of Finland. The work of the second author was performed within the Russian Government Program of Competitive Growth of Kazan Federal University.

## REFERENCES

- [1] C. Courcoubetis and R. Weber, Pricing Communication Networks: Economics, Technology and Modelling. Chichester: John Wiley & Sons, 2003.
- [2] S. Stanczak, M. Wiczanowski and H. Boche, Resource Allocation in Wireless Networks. Theory and Algorithms. Berlin: Springer, 2006.
- [3] A. M. Wyglinski, M. Nekovee and Y. T. Hou (eds.), Cognitive Radio Communications and Networks: Principles and Practice. Amsterdam: Elsevier, 2010.
- [4] I. Konnov, A. Kashuba and E. Laitinen, "Dual methods for optimal allocation of telecommunication network resources with several classes of users," Math. Comput. Appl., vol. 23, no 31, pp. 1–14, 2018.
- [5] I. V. Konnov, A. Yu. Kashuba, "Application of the conditional gradient method to a network resource allocation problem with several classes of users," Journal of Physics: Conference Series. 1158, no 032015, pp. 1–8, 2019.
- [6] S. T. Zargar, J. Joshi and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," IEEE Communications Surveys & Tutorials, vol. 15, no 4, pp. 2046–2069, 2013.
- [7] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacsar and J.-P. Hubaux, "Game theory meets network security and privacy," ACM Computing Surveys (CSUR), vol. 45, no 3, pp. 25–64, 2013.
- [8] S. Vadlamani, B. Eksioğlu, H. Medal and A. Nandi, "Jamming attacks on wireless networks: A taxonomic survey," International Journal of Production Economics, vol 172, pp. 76–94, 2016.
- [9] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, Z. Han, "Applications of economic and pricing models for wireless network security: a survey," IEEE Communications Surveys & Tutorials, vol. 19, no 4, pp. 2735–2767, 2017.
- [10] I. Konnov, A. Kashuba, E. Laitinen, "Penalty method for reliable allocations of wireless network resources," CEUR Workshop Proceedings, no 2642, 2020.
- [11] F. P. Kelly, A. Maulloo and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," J. Oper. Res. Soc., vol. 49, no 3, 237–252, 1998.

TABLE I  
COMPUTATIONS FOR  $|I| = 620$ ,  $|L| = 310$  (QUADA-LOGC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.016	50	0.078	200
$10^{-2}$	0.028	57	0.094	220
$10^{-3}$	0.031	70	0.125	248
$10^{-4}$	0.047	91	0.172	407

TABLE II  
COMPUTATIONS FOR  $|I| = 310, |L| = 620$  (QUADA-LOGC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.016	35	0.235	466
$10^{-2}$	0.027	65	0.344	629
$10^{-3}$	0.032	67	0.375	703
$10^{-4}$	0.035	77	0.377	733

TABLE III  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |L| = 310$  (QUADA-LOGC)

$ I $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.028	57	0.094	220
930	0.047	84	0.078	157
1240	0.078	84	0.109	139

TABLE IV  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |I| = 310$  (QUADA-LOGC)

$ L $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.027	65	0.344	629
930	0.047	63	0.563	716
1240	0.062	70	0.750	783

TABLE V  
COMPUTATIONS FOR  $|I| = 620, |L| = 310$  (QUADA-QUADC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.016	24	0.094	198
$10^{-2}$	0.023	51	0.093	229
$10^{-3}$	0.031	77	0.109	268
$10^{-4}$	0.047	124	0.172	408

TABLE VI  
COMPUTATIONS FOR  $|I| = 310, |L| = 620$  (QUADA-QUADC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.015	32	0.234	467
$10^{-2}$	0.031	58	0.365	691
$10^{-3}$	0.032	61	0.370	770
$10^{-4}$	0.047	97	0.374	793

TABLE VII  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |L| = 310$  (QUADA-QUADC)

$ I $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.023	51	0.093	229
930	0.032	48	0.094	159
1240	0.031	51	0.094	150

TABLE VIII  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |I| = 310$  (QUADA-QUADC)

$ L $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.031	58	0.365	691
930	0.047	53	0.640	815
1240	0.078	66	0.797	856

TABLE IX  
COMPUTATIONS FOR  $|I| = 620, |L| = 310$  (LOGA-LOGC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.016	31	0.078	161
$10^{-2}$	0.032	57	0.084	186
$10^{-3}$	0.047	93	0.125	262
$10^{-4}$	0.062	129	0.156	334

TABLE X  
COMPUTATIONS FOR  $|I| = 310, |L| = 620$  (LOGA-LOGC)

$\varepsilon$	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
$10^{-1}$	0.016	28	0.187	265
$10^{-2}$	0.041	69	0.203	313
$10^{-3}$	0.047	93	0.265	491
$10^{-4}$	0.063	118	0.344	649

TABLE XI  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |L| = 310$  (LOGA-LOGC)

$ I $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.032	57	0.084	186
930	0.047	71	0.094	127
1240	0.063	72	0.125	163

TABLE XII  
COMPUTATIONS FOR  $\varepsilon = 10^{-2}, |I| = 310$  (LOGA-LOGC)

$ L $	$T_{\varepsilon,1}$	$I_{\varepsilon,1}$	$T_{\varepsilon,100}$	$I_{\varepsilon,100}$
620	0.041	69	0.203	313
930	0.047	64	0.312	434
1240	0.094	70	0.328	342

# A Conceptual Digital Twin for 5G Indoor Navigation

Vladeta Stojanovic<sup>1</sup>, Hossein Shoushtari<sup>2</sup>, Cigdem Askar<sup>3</sup>, Annette Scheider<sup>4</sup>,  
Caroline Schuldt<sup>5</sup>, Nils Hellweg<sup>6</sup>, and Harald Sternberg<sup>7</sup>

<sup>1, 2, 3, 4, 5, 6, 7</sup>Department of Hydrography and Geodesy, HafenCity University Hamburg, Germany  
Email: {name.surname}@hcu-hamburg.de

**Abstract**—With the introduction of practices from Industry 4.0 in various Architecture, Engineering, Construction, Owner and Occupant (AECOO) sectors, especially in Facilities Management (FM), new requirements concerning integration of digital data sources arise with prominence. Particularly with the increased use of the Digital Twin (DT) paradigm, the need to capture, store, process and present various data concerning the current and predicted states of the built environment becomes paramount. A conceptual and prototypical system design, focusing on integration, processing, analysis and visualization of data related to indoor navigation within modern buildings, is presented and discussed. Approaches for processing and presentation of key data sources, particularly indoor point clouds and as-is Building Information Modeling (BIM) data, combined with simulated 5G signal as localization approximation, are presented and discussed. A particular focus is placed on leveraging the Service-Oriented Architectures and Systems (SOA/SOS) paradigm for implementation of complex systems for meeting integration requirements of such diverse data sources. Finally, we discuss how such approaches can benefit from current and predicted use of 5G technologies, and provide experimental results from a case study conducted within a modern university campus building. The presented case study results demonstrate the feasibility of our approach, and provides a framework for future expansion, integration and evaluation with planned 5G infrastructure.

**Index Terms**—Point Clouds; BIM; Service Oriented Computing; Localization; 5G Simulation

## I. INTRODUCTION

Current advancements in adaptation of Industry 4.0 practices for Architecture, Engineering, Construction, Owner and Occupant (AECOO) stakeholders, particularly in the realm of Facility Management (FM), have created a paucity for the research and development of integration strategies for digital data sources [1]. Such data sources may include as-designed/built/as-is Computer-Aided Design (CAD) and Building Information Modeling (BIM) data, digital and digitized FM documents as well as historical or current sensor data - all pertaining to the past, present and future operational state of a building [2]. With the current standardization of 5G technology, additional data sources captured via 5G infrastructure should be available for assessing the current state of the built environment, for, e.g., indoor navigation [3] or FM within a Smart Building context [4]. The fusion of all these data sources into a dynamic representation is based on the currently promoted Digital Twin (DT) paradigm [5]. A DT is able to encapsulate and represent the physical state

of an object as its cyberphysical counterpart [6]. The use of DTs has potential benefit for engagement of stakeholders through interactive visualization [7] and enables an up-to-date virtual representation of their facilities and related states and processes [8].

### A. Problem Statement

The focus of this research is on indoor navigation based on the DT paradigm. The deployment and use of a DT requires access to various data sources related to the specific lifecycle stage of the building. For FM applications, historical and current data sources are needed in order to perform key analytics and generate results for furthering FM stakeholder engagement. Additionally, future states also need to be predicted. This poses three key problems:

- 1) The selection and analysis of required data used to represent the historical and current states of a building (e.g., existing BIM data).
- 2) The selection and analysis of required data used to generate predictions for future states of a building (e.g., signal and sensor data obtained from 5G infrastructure).
- 3) The combined analysis and representation of these data sources.

### B. Research Contributions

To address these problems, we present an approach for integration, processing, analysis and visualization of key data sources for indoor environments. Our focus is on indoor localization and navigation tasks with a FM context (Fig. 1). An SOA for integration of such data for DT representations is presented and discussed. A prototypical implementation of key system components is tested using data from a real-life university building with a simulation for 5G-based localization. We also use synthetically generated point clouds, based on as-is BIM data, in order to test our point cloud processing and reconstruction methods. The fusion of historical, current and predicted data sources is then used to generate indoor navigation results, which are interactively visualized alongside existing BIM and FM data (e.g., 2D floor plans, 3D models and point clouds). The final output is presented as a web-based, interactive visualization aimed at engaging FM stakeholders for indoor routing and navigation tasks.

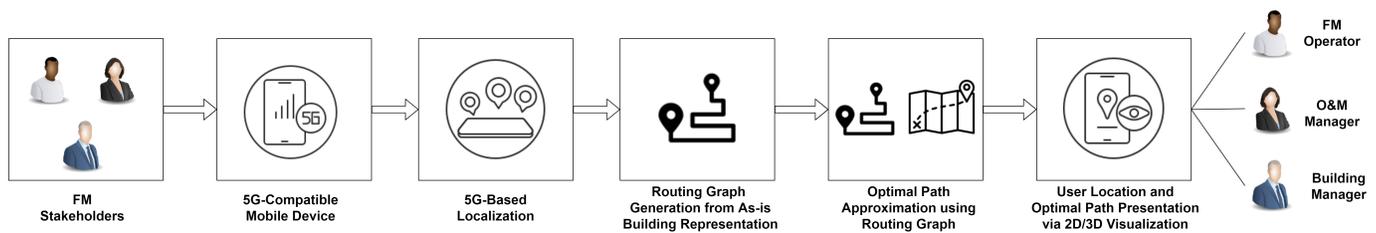


Fig. 1. The proposed approach of using 5G data and BIM data within a DT paradigm for indoor navigation and localization tasks related to FM.

## II. RELATED WORK

### A. Requirements for Indoor Navigation Modeling

A key requirement of indoor navigation is the determination of navigable areas within the chosen representation (e.g., 2D floor plan, BIM, etc.). A navigable area is defined as a region in 2D or 3D space where an agent representing a user or an autonomous entity is able to move around - in order to get from one location to another. A common name for data structures used to represent such navigable areas are "routing graphs" [9] or "navigation meshes" [10]. The use of routing graphs is therefore a requirement for pedestrian indoor navigation, especially for approximation of optimal paths (e.g., using Dijkstra's algorithm and its variants [11]). Existing BIM data and semantics (e.g., Industry Foundation Classes (IFC) schematics [12]) can be used in order to determine the geometric boundary constraints of areas that are navigable [13]. Methods for discretization of 3D space and approximation of topological derivatives using geometric computation principals also play an important role for generation of routing graphs [14] [15].

### B. Capture and Representation of Indoor Environments

In AECOO applications, the use of point clouds enables the area-wise capture of intricate physical details of indoor and outdoor built environments, real-world objects and locations, with varying levels of visual fidelity [16]. Point clouds can be generated by using photogrammetric techniques and by laser scanning [17] [18]. In laser scanning, laser signals emitted by 3D scanners are used to capture the reality based on time of flight or phase shift methods [19]. For AECOO applications, in particular Operations & Maintenance (O&M) procedures within FM, the current state of the physical environment can be captured using point clouds and analyzed for further decision making [20].

The simulation of the laser scanning process can also be performed, known as "synthetic data" generation, using as-designed/as-built or as-is 3D models of indoor environments as the main processing input [21]. The output of such a process is a point cloud with similar qualities as that of a physically captured point cloud, and can be used for various testing key processes (e.g., training machine and deep-learning models for semantic segmentation [22]).

Apart from manual generation of as-is and as-built BIMs, the application of the "Scan2BIM" process enables automated reconstruction of segmented point clouds to semantically-rich

and higher-level geometric representations (e.g., as-is BIM data in the form of IFC models [23]). This can also include both 2D and 3D floor plan generation [24]. Since point clouds themselves are ambiguous, they need to be processed and enriched semantically in order to make them useful for visualization and analysis applications [25].

### C. Localization Methods within the 5G Paradigm

The increased use of smartphones with integrated cost and energy efficient sensors for navigation and localization correspond to the ubiquitous positioning vision proposed by Park in [26]. Determining the optimal route to get between two points is often a bottleneck for FM tasks, and solving such a problem requires the approximation of the user's current location within a building as well as the approximate location of the destination point. One key approach to solving this problem is based using autonomous localization methods - independent from any other infrastructure solution, e.g., WLAN or UWB networks, which need an app registration or a predefined label dataset [27]. However, such autonomous localization performance would be limited both in physics-based solutions [28] [29], or machine learning-based approaches [30] [31], due to the well known sensor drift and the absence of a realistic dataset respectively [27].

The current introduction and adaptation of 5G networks is of great interest for application of indoor localization within a BIM-based FM usage context [32]. The use of 5G infrastructure and technology could increase the accuracy of indoor navigation applications, and would therefore be useful to consider for FM-related tasks by providing benefits within a Smart Building context (e.g., stable and low-latency communication, multi-device support and use for data-heavy streaming applications such as Augmented and Virtual Reality (AR/VR) [4]). Since the 5G technologies would be available to the vast majority of the population, it can be expected to be a staple technology to benefit indoor navigation, and capable of solving current open localization challenges in the built environment [27]. Variants of such 5G infrastructure (e.g., using a specialized 5G network that uses slightly different frequencies than a 'normal' network), can also provide similar benefits.

### D. Service-Oriented Architectures and Systems

The use of the DT paradigm further places demand on availability of historical and current digital information for

a given building [33]. The use of service-oriented computing and system architecture enables the design and deployment of complex software components and services, which are capable of meeting the requirements of accessing, storing and processing versatile data sources - along with the benefit of hardware decoupling between the server and the client systems [34].

A particular advantage of using SOA when implementing SOS-based FM software systems (e.g., CAFM, BAS and IWMSs [35]), is the processing and streaming of specific results related to FM tasks for a given building (i.e., visualizations of recent spatial changes of office furniture and/or equipment, occupancy levels, environmental sensor data, approximations for indoor localization, etc. [36]). Once received, these results can be presented on any connected client devices of stakeholders running, e.g., a compatible web-browser on a commodity mobile device [37].

### III. APPROACH

The main paradigm for presentation, interaction and decision making is envisioned with the use of a DT. The type of data utilized by the DT would depend on its intended use case, but in most cases the data needs to represent the physical characteristics of the entity that is presented digitally. For FM-specific applications this would include as-is point clouds and their generated semantics, as-is BIM data, as well as any existing data pertaining to the current representation and operational status of a building (e.g., Mechanical, Electrical and Plumbing (MEP) plans and status reports, floor plans, as-built/as-designed BIM, current or historical sensor data, etc.). The fusion of these data sources when generating the required result of the user's computation tasks is the essence of the DT [38].

The Data Processing and Data Analytics components of such a DT system would be implemented as services that are run when a specific task is requested, and would process and stream the result of the task back to the user. Such an approach fits well with the use of service-oriented computation, and is the recommended approach for implementation of DTs for AEC applications [39]. All of the processes can be automated and/or simulated, and included as software components and services within a SOS implementation (Fig. 2).

*a) Point Cloud Processing:* The capture of the as-is physical state of indoor environments can be accomplished with the use of point clouds. The use of indoor laser scanning can produce point clouds representing intricate details and can provide useful base-data for further processing and semantic enrichment (Fig. 3) However, if the capture of point clouds is not possible or prohibited due to various factors, their capture can be simulated using existing 3D models of indoor environments as the main inputs. Using methods such as Monte Carlo point sampling [40], or complete LiDAR device simulation [41], synthetically generated point clouds can be used as data sources for experimental assessment of key processing, semantic enrichment and visualization methods.

Further processing of captured or generated point clouds commonly involves registration, sub-sampling, noise filtering and per-point normal vector computation. Such tasks can be automated within a point cloud processing pipeline, and implemented with an SOS [42]. Most point cloud data management is currently done at the file level and there are different file formats specifically used for storing point cloud data (e.g., LAS [43]). As point clouds can be very large, they cannot always be loaded completely into system memory. To enhance performance for accessing, analyzing and viewing point clouds, different spatial data structures such as octrees or  $k$ -d trees can be used [44].

*b) Database Integration:* For FM use where point clouds need to be generated frequently, storing and distributing thousands of separate point cloud files is not sufficient for basic point cloud data manipulation. Here, a Database Management System (DBMS) represents an alternative that is easier to operate and scalable [45]. A relational or non-relational DBMS can be used, with additional special spatial indexing operations, in order to quickly access point cloud representations of, e.g., entire floors or specific rooms of a building, for processing and analysis. This also allows for the integration of point clouds with additional static (e.g., IFC models [46]), and spatio-temporal data sources (e.g., collected or real-time sensor data [36]), using non-relational DBMS that can accommodate the changing representation specifications for smart building related data.

*c) Semantic Enrichment of Point Clouds:* While point clouds can be used to represent intricate details of indoor environments that can be interpreted by viewers with domain expertise, point clouds themselves do not contain any semantics by default. Therefore, the use of supervised and unsupervised machine and deep-learning algorithms is needed to add useful semantics to point clouds (e.g., prior to reconstruction to as-is BIM representations). Unsupervised methods include segmentation and clustering methods such as RANSAC, Region Growing,  $k$ -means and DBSCAN clustering [47] [48]. These methods attempt to group together points that share similar features (e.g., spatial position, color, normal vector direction, etc.) or fit to a geometric primitive (e.g., a hyperplane using variants of the Least Squares method).

Supervised methods include the use of learning models where examples of what the model should output is based on observations from existing data (i.e., *a posteriori* knowledge). This commonly includes the use of Convolutional Neural Networks (CNNs) that can classify 2D and 3D data (e.g., point clouds, images of point clouds and higher-level geometric representations, e.g., voxels) into user defined categories using a classification model trained on previous examples [49].

*d) As-Is BIM and Routing Graph Data Generation:* The generation of as-is BIM data is based on the detection of semantically segmented point clusters, from which the geometry is used to infer the dimensions of the element to be reconstructed, and the label is used to infer the semantic associated with the element, usually with association to an IFC specification at a given Level of Detail (LOD). Semantically

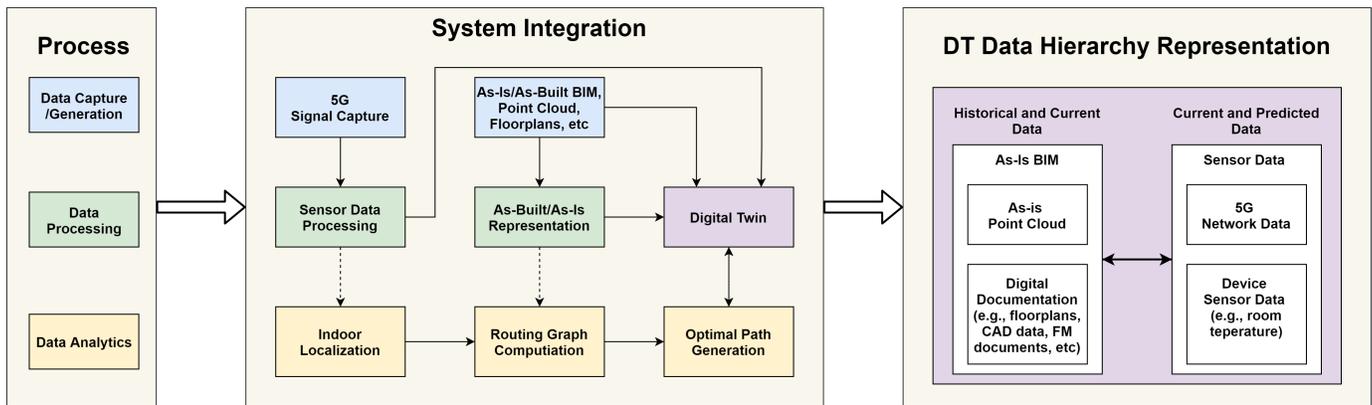


Fig. 2. High-level representation of key processes, system integration, data sources and hierarchy representation for the conceptual use of a DT for indoor navigation tasks.

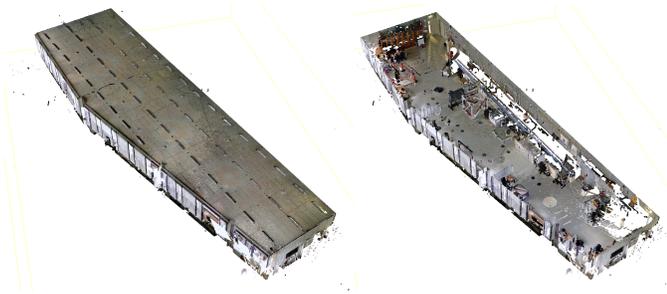


Fig. 3. An example of a point cloud of one of the areas in the university building presented in the case study, captured using indoor laser scanning and sub-sampled to 556 410 points.

segmented point clouds can also be used to generate 2D/3D floor plans. Such floor plans can either be 2D vector image data or 3D triangulated meshes.

For indoor navigation applications, besides geometric representations, an additional geometric data structure is needed to represent navigable areas. With a routing graph representation, the edges represent navigable paths, while the nodes are represented by vertices. Routing graphs and navigation meshes can be obtained both from 2D and 3D floor plan representations. For 2D floor plan representations, they are commonly generated using medial-axis approximation methods [50], while for 3D floor plans they can be derived either from the triangulated mesh topology representing the floor area, or from the semantics derived from the IFC elements representing navigable and room boundary elements [13].

*e) 5G Simulation, Processing and Localization Methods:*

For navigation applications, the use of localization methods is required for approximating the user's position in relation to their surroundings. In order to approximate the user's initial location, different absolute positioning algorithms (e.g., triangulation, trilateration and multilateration), can be used to determine the approximate location of the user [28]. Additionally, the user's current location can be estimated based on the fusion of the maps information, routing graphs, the absolute positions and the received sensor data readings by

using different state estimation algorithms (e.g., Monte Carlo Particle Filtering or Extended Kalman Filtering) [29].

With the availability of precise localization offered by the coming releases of 5G networks, user's mobile devices can capture and process localization data, as well as provide additional sensor data output such as barometer readings (i.e., used to measure height changes). The proposed approach relies on such input data in order to determine the user's approximate location, and can further make use of IoT and Smart Building paradigms by utilizing environmental sensor sources (e.g., RFID and NFC tags in building rooms [51]).

*f) Representation and Interactive Visualization:* Engagement with stakeholders through visualization plays an important role in furthering understanding amongst all levels of domain expertise. The use of interactive visualization, particularly for FM tasks, is crucial for enabling the visual representation and interpretation of key built environment elements and processes (e.g., planned renovations, item inventory, emergency route planning, etc.) [20]. The use of interactive visualization can greatly benefit indoor navigation tasks, where there is a need to combine as-is representations of buildings (either as 2D floor plans and/or with combined/exclusive 3D representations [52]). In such cases, stakeholders can use the as-is representation to interactively view their current location within the building, and use generated optimal paths for navigation between set markers.

Visualization of key data sources is enabled using real-time 3D rendering, mainly using existing game engines (e.g., Unity [53]) or Web3D frameworks (e.g., Three.js [54]). If dealing with complex data sources (e.g., visually complex 3D models or point clouds), out-of-core and SOS rendering methods can be utilized [44]. The representation of the as-is built environment, either as a raw or semantically enriched point cloud, or as a reconstructed higher-level geometric representation with associated semantics (e.g., 2D/3D floor plans, IFC models, etc.), enables stakeholders to visualize the environment as they may see it in real life, but with additional visualization metaphors and idioms [55].

## IV. CASE STUDY

### A. Overview

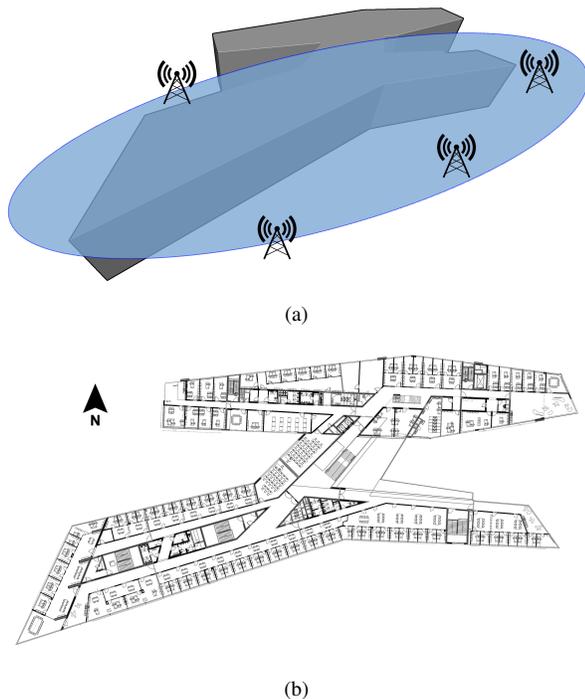


Fig. 4. (a) The planned 5G antenna locations around the perimeter of the building used in the case study. (b) An example of the architectural floor plan of one of the main floors of the university building.

A 5G Non-Standalone (5G NSA) campus network and an additional experimental system will be realized at the presented university building location between Q3 2021 and Q1 2022. Applications from the field of indoor navigation will be tested there and will serve research purposes. The planned campus network will consist of four outdoor antennas arranged around the building (Fig. 4(a)). Currently this process is in the planning stages, and more specific implementation details will be determined in the near future. For the planned outdoor and indoor network, the use of private frequencies in band B43/N78 (3.7–3.8 GHz) with a Long-Term Evolution (LTE) anchor band is planned. The aim is to set up, operate and use the networks as realistically as possible, as they can subsequently be set up and used as productive systems at other locations. The project will also set up a second experimental network. This network will work with frequencies in the range of 26 GHz–78 GHz to achieve even higher accuracy for the estimation of the position. In addition to this and for further research purposes, indoor units will also be provided on two floors of the building. In this way, the project will create different scenarios for testing indoor navigation.

Prior to the implementation of this infrastructure, it was decided to test key components in terms of their feasibility to generate and process data required for representation for indoor localization and navigation. As the 5G infrastructure is still absent, it was decided to simulate partially the key

missing aspects, in order to validate the feasibility claim of the DT paradigm, presented approach and prototypical SOA and SOS. We therefore simulate the indoor 5G network for the presented case study, using the basic cellular positioning measurements and algorithms. The reference points are used as the basis of such a simulation.

We particularly focus on the aspect of localization and navigation within a given floor space of the main university building (Fig. 4(b)). Using an as-is BIM data of the fourth floor area of the university building, we simulate the laser scanning, positions and signals of 5G antennas. We use this simulated 5G data to approximate the coordinates of the user as they are virtually navigating the indoor area. For the implemented case study, we present and discuss key SOS components for processing data sources for generating indoor navigation results that are presented to the user through interactive visualization. The virtual navigation and user interaction is enabled through a web-based application, while processing and analysis of data sources is implemented server-side as SOS components.

### B. Prototypical System Design and Implementation

We present a conceptual SOA, with key processes implemented as SOS software components and a Web3D-based client application (Fig. 5). The Web3D-based client is able to display either a point cloud or as-is BIM representation of an indoor environment, the user's simulated and approximated position. The generated optimal path between the user's current location and a defined point anywhere on the as-is BIM/point cloud can also be visualized. This optimal path, the as-is BIM/point cloud and the user's approximate location are displayed to the user interactively - enabling the user to inspect the each of the 3D scene elements from different views.

In terms of the back-end implementation, the prototype application is able to process point cloud data and create an initial version of the 2D floor plan. We also make use of a manually generated IFC model at LOD 300 of the same floor plan area, in order to provide additional data source for analysis and visualization of indoor navigation tasks. Additionally, the simulated 5G signal and localization estimates are implemented, along with a software component for optimal path generation using routing graphs generated from either the 2D floor plan or the 3D as-is BIM. The access to metadata related to the 2D floor plan, BIM, point cloud and FM data is provided via a DBMS also running on the server, while the actual files are stored on the same server within a specified directory structure. For the back-end systems, PostgreSQL DMBS is used as object-relational DBMS. With its spatial database extension PostGIS [56], it provides support for georeferenced objects that enable location queries (e.g., for processing of 2D floor plan representations). The simulated 5G capture works based on the reference points and can be combined with physics-based sensor fusion (i.e., pedestrian dead reckoning). The accuracy of such simulation can be defined by considering the current 3GPP release specification [57]. In other words, having the truth trajectory leads to a 5G simulation data by giving them some noise

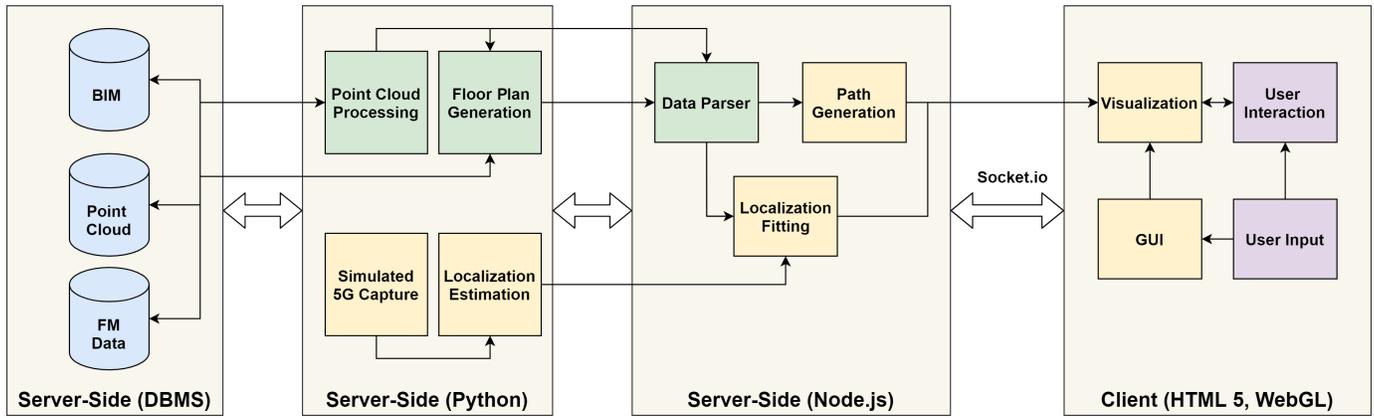


Fig. 5. A high-level system design for the presented prototype application, with key SOS components implemented server-side.

and timestamp. The time stamps are estimated using manual video analysis. The simulated points can be the basis for measurement estimation considering the pre-defined random noises. For instance, using the distance between one position and the antennas, measurement like time of arrival can be simulated.

The server-side software components are primarily implemented in JavaScript in Node.js [58], with additional components implemented in Python 3.6 and called as server-side scripts by a Node.js extension. Psycopg [59] can be used to connect to the database server and communicate with Python scripts. This DB API 2.0 compliant PostgreSQL [60] database adapter is designed for multi-threaded applications and maintains its own connection pool. It is mostly implemented in the C language, thus being efficient. Communication between the client and server is enabled via the Socket.IO [61] library. The client-side application is implemented in HTML5 [62], and uses the WebGL-based Three.js [54] rendering framework to facilitate real-time 3D viewing of the generated results.

## V. IMPLEMENTATION AND RESULTS

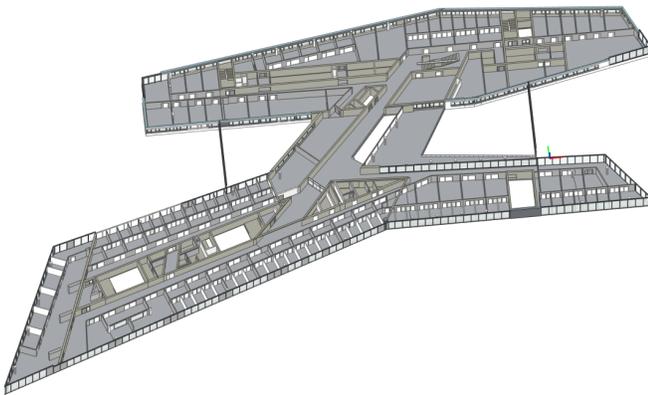


Fig. 6. The as-is BIM of the university building area used in the case study.

We present experimental results obtained from each of the key SOS component implementations, using mostly simulated

data related to a functional office floor space within a modern university building (Fig. 6). We present and discuss processing of point cloud representations of this environment (generated synthetically from the as-is BIM), and the resulting floor plan that is generated automatically from the point cloud representation using the initial version of the Floor Plan Generation software component. Furthermore, we discuss the use of simulated 5G signals for localization estimation, and provide examples of generating a routing graph from the approximated 2D floor plan as well as from the as-is IFC LOD 300 model of the entire floor area. We use these routing graphs for the computation of an optimal path for aiding in indoor navigation tasks. Finally, we present the initial visualization results running on a Web3D-enabled browser on a commodity client computer.

### A. Point Cloud Processing

For the point cloud processing results, the point cloud of the fourth floor area of the university building is generated using simulated laser scanning. We make use of the as-is LOD 300 IFC model, from which we extract a triangulated mesh, and sample points using uniformly distributed point sampling for each of the mesh triangles [63]. Once generated, further processing is performed via the Point Cloud Processing software component, using the Open3D Python framework [64] for key processing tasks (Fig. 7). These key tasks include: sub-sampling, outlier point removal and geometric segmentation. The result of these processes are point clusters representing key structural elements which can be used as floor plan layers. Registration was not performed for this case study, though the implementation is capable of accomplishing this using automated registration methods, e.g., Iterative-Closest Point (ICP) matching algorithm [65].

After sub-sampling, using a voxelized sampling to ensure uniform removal of points, the original point cloud is reduced to a fraction of its original points (approx. 75% of points are removed). The overall visual fidelity of the point cloud is preserved, while increasing the coarsity and removing redundant overlapping and closely spaced points. For the outlier point

removal, a statistical outlier removal method was used [66], where the points forming clusters that are considered too far away from the main point cluster groups were removed. This enabled the removal of outlier points especially noticeable at the edges of the sub-sampled point cloud. Finally, the point clusters representing the planes of key structural components (namely walls, floors and ceilings) are detected and segmented using the iterative RANSAC method, which attempts to fit matching points to a set number of hyperplanes [67]. The horizontal point clusters forming the floor representation are then used for the 2D floor plan approximation by the Floor Plan Generation component.

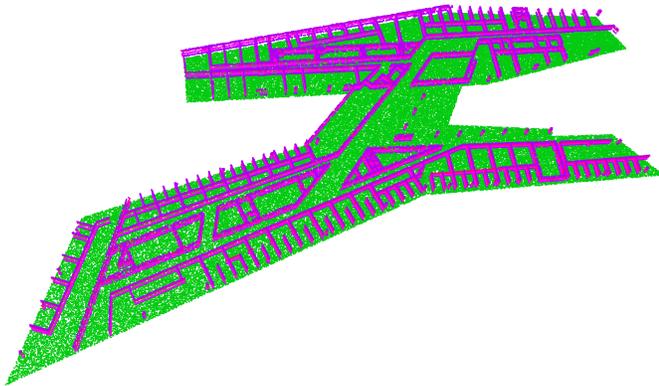


Fig. 7. Initially segmented point cloud, with the main floor plan boundary and secondary wall structure boundaries displayed.

**B. Floor Plan Generation**

A 2D floor plan can be generated either from the existing as-is BIM (i.e., IFC model), or from the as-is point cloud representation. Generating the 2D floor plan from the IFC requires selecting each building storey element and projecting it onto a 2D plane, where each of the IFC components are rendered as vectorized paths and associated element semantics, and exported together as a SVG file [68]. This SVG file is then further parsed and converted to a GeoJSON [69] file (Fig. 8), which can then be used as 2D floor plan data for required indoor navigation tasks.

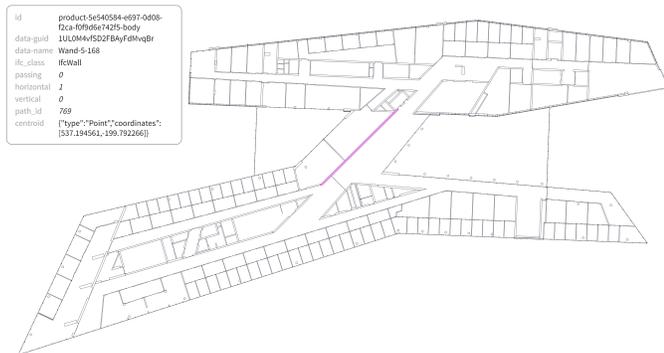


Fig. 8. An example of a semantically-rich GeoJSON file, containing vectorized paths along with IFC element shapes and semantics (e.g., a wall element).

A 2D floor plan can also be approximated from a horizontally sliced layer of the post-processed point cloud. The initial 2D concave or convex hulls of this point cloud layer (including both primary and secondary shape boundaries, i.e., shapes within shapes), are approximated based on the approach described in Stojanovic et al. (2019) [24]. This allows for the capture of boundaries of the floor plan, using adjustable parameter values (Fig. 9). Once the initial boundaries are approximated, they are simplified as their resulting boundary shape is usually noisy. The resulting 2D floor plan approximation is then exported as a GeoJSON file.

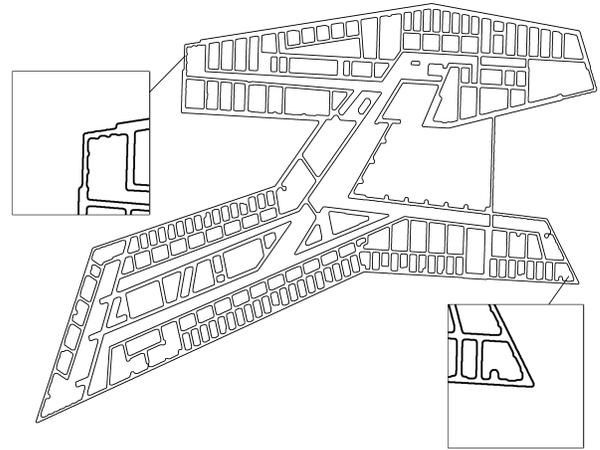


Fig. 9. A generated floor plan based on a point cloud, with boundary evaluation errors that need to be corrected in further post-processing steps.

**C. 5G Simulation and Localization**

Using the Simulated 5G Capture components, the antenna placement and three 5G positioning signals are simulated (Fig. 10). This is calculated based on some reference points and given noise. In this way, one may simulate 5G-based coordinates with different precision. This is needed for fusion algorithms when such a simulated infrastructure-based positioning has a high range of noise. One can enter the number of desired antennas and positions. Then, they can be entered to the simulated environment and by calling a calculation function, the predefined noises, frequencies and some measurement results can be computed.

**D. Optimal Path Approximation**

The optimal path generation is performed using the generated routing graphs either from the 2D floor plan or the 3D floor plan derived from the as-is IFC representation. For routing graph generation of 2D floor plans, we make use of a generalized Voronoi-based medial axis transform [50], in order to generate 2D line graphs representing navigable areas (Fig. 11(a)), while for routing graph generation from the 3D floor plan we use the triangulated mesh of the floor element boundary representation in order to generate a “navigation mesh” [10] (Fig. 11(b)).

For computation of the shortest paths using routing graphs, we make use of the A\* shortest path algorithm [70]. The A\*



Fig. 10. An example of localization of three points (red circles) in the simulated environment, with six different 5G antenna placements. This simulation component is tested via a simple user interface that allows the placement of 5G antennas anywhere on the vectorized GeoJSON version of the 2D floor plan (generated from the as-is BIM representation).

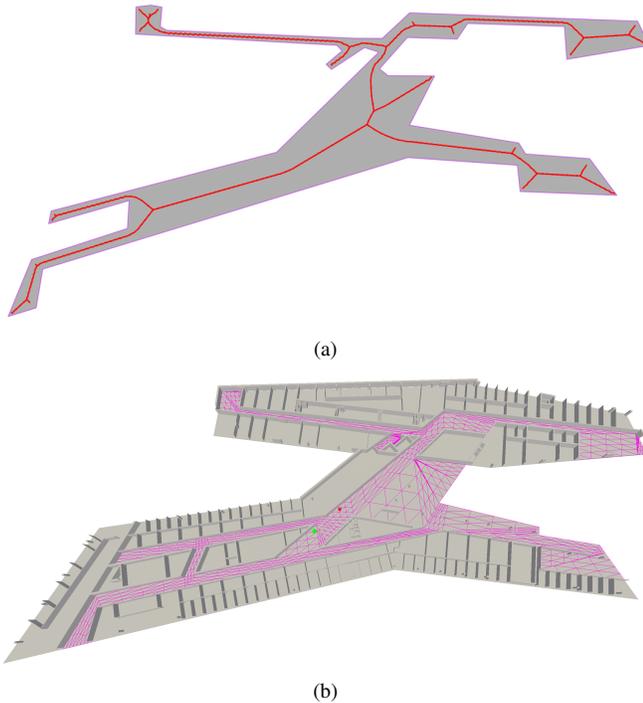


Fig. 11. (a) An example routing graph of a navigable floor plan area variant, based on the approximated primary boundary of the 2D floor plan. (b) An example routing graph/navigation mesh (purple) of the main navigable areas of the floor plan, based on the triangulated 3D model of the as-is BIM representation.

algorithm uses the vertex and edge connections that form the routing graph, in order to evaluate and construct the shortest navigation path. This approach was tested using the 3D model of an as-is BIM (Fig. 6). For testing the 3D floor plan and routing graph, the generated shortest path computation allows the user to select and set starting and ending points, between which the shortest path will be computed (taking into account any obstacles, e.g., walls that may be in between the starting and ending points (Fig. 12)).

## VI. DISCUSSION

We have presented a conceptual SOA and prototypical implementation of key SOS components for integrating 5G,

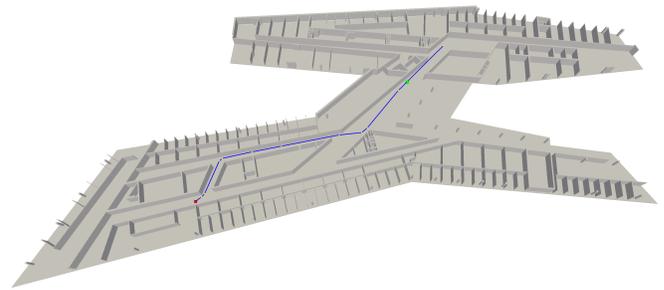


Fig. 12. An example of shortest path (blue) computation result between the starting and current point (green) and ending point (red) – based on the result of the A\* algorithm and making use of the underlying routing graph (i.e., navigation mesh).

BIM and point cloud data for DT representations of indoor built environments. For our case study, we have focused on the topic of indoor navigation, using an approach of combining point clouds, 2D floor plans, BIM and simulated 5G signal data. We can create 2D floor plan approximations from input point clouds, and derive and generate routing graphs from both 2D floor plans and 3D models (obtained from BIM data). Using the 5G-based simulated coordinates, we are able to simulate the expected localization approximation corresponding to the indoor areas within the university building - where the expected 5G infrastructure will be operational in the near future. In this way, one can approximate the user's indoor position. We further combine this with optimal path generation between the user's current location and target destination. The generated routing graphs are used with search algorithms to calculate the shortest path and visualize it within a 2D/3D floor plan context. The fusion of the outputs of all of these results from the SOS components is presented interactively via a Web3D interface, where the user can inspect the as-is point cloud, 2D floor plan, 3D BIM model, current approximated location and optimal path results. Our presented approach is based on the DT paradigm, which has evolved from previous research mainly focusing on BIM-based and IoT integration approaches [71] - though such approaches do not take into account, e.g., dynamically updated systems, frequently changing built environment states and indoor space configurations, and are not designed to predict future states. Apart from a Web3D-based approach for interactive visualization of the results obtained using service-oriented computation, another approach can be the use of existing game engines for computation and presentation for FM scenarios [72]. The use of the SOS paradigm is also not the only way to implement DT solutions, and approaches based on, e.g., Supervisory Control and Data Acquisition (SCADA) [73] and Enterprise Resource Planning (ERP) [74] have also been investigated as potential architecture solutions in previous research.

## VII. CONCLUSION

We have proposed our approach based on a concept of a DT, which enables the fusion of historical, current and predicted (or simulated) data sources for processing, analysis and

representation – within a service-oriented paradigm. We have implemented and tested all of the key software components responsible for processing, analysis and visualization of key data sources needed to generate valuable information about localization and indoor navigation within a university campus building. We have identified the use of point clouds as a key source for as-is representation of indoor environments, alongside existing as-is/as-built BIM data (e.g., IFC models, point clouds, etc.). We have presented and discussed methods for generation of approximated 2D floor plans from point clouds, and the generation of routing graphs from such floor plans as well as 3D mesh representations derived from existing IFC models. Finally, we have implemented and described state-of-the-art approaches for routing graph and optimal path computation. We have designed a simulation for 5G-based coordinate estimation to overcome with the localization task. The presented approach lays a solid foundation for future work focusing on developing a versatile indoor navigation software solution based on the DT and service-oriented computing paradigms.

The use of simulation processes for generation of point cloud and 5G sensor and signal data has enabled the rapid development and testing of the key software components – without needing to source such data from complex or non-existent sources. For future work, we plan to evaluate our approach using real-world point clouds captured using laser scanning as well as make use of the planned 5G infrastructure for capturing real signal and sensor data. The positioning approaches can achieve the fusion of sensor data, infrastructure input and consider the map matching features. While we make use of a DBMS for storage for data, we did not explicitly evaluate its performance for data retrieval and queries, and this is also planned for future work. We also plan to further optimize and improve the 2D floor plan generation methods, in order to have better approximations of secondary boundaries. Finally, our future work will focus the further development and integration of DT and 5G paradigms focusing FM related scenarios (e.g., real-time data can be fed into a DT with the help of 5G and serve to actively control the flow of people and the operational capacity of the building).

#### ACKNOWLEDGMENTS

This research and the L5IN project was funded by the Federal Ministry of Transport and Digital Infrastructure (BMVI), grant number VB5GFHAMB.

#### REFERENCES

- [1] K. Kensek, “BIM guidelines inform facilities management databases: a case study over time,” *Buildings*, vol. 5, no. 3, pp. 899–916, 2015.
- [2] P. Teicholz et al., *BIM for facility managers*. John Wiley & Sons, 2013.
- [3] C. Schuldt, H. Shoushtari, N. Hellweg, and H. Sternberg, “L5in: Overview of an indoor navigation pilot project,” *Remote Sensing*, vol. 13, no. 4, p. 624, 2021.
- [4] M. Y. L. Chew, E. A. L. Teo, K. W. Shah, V. Kumar, and G. F. Hussein, “Evaluating the roadmap of 5G technology implementation for smart building and facilities management in singapore,” *Sustainability*, vol. 12, no. 24, p. 10259, 2020.
- [5] R. Alonso, M. Borrás, R. H. Koppelaar, A. Lodigiani, E. Loscos, and E. Yöntem, “SPHERE: BIM digital twin platform,” in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 20, no. 1, 2019, p. 9.
- [6] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in *Transdisciplinary perspectives on complex systems*. Springer, 2017, pp. 85–113.
- [7] J. Posada, M. Zorrilla, A. Dominguez, B. Simoes, P. Eisert, D. Stricker, J. Rambach, J. Döllner, and M. Guevara, “Graphics and media technologies for operators in industry 4.0,” *IEEE computer graphics and applications*, vol. 38, no. 5, pp. 119–132, 2018.
- [8] Q. Lu, A. K. Parlikad, P. Woodall, G. Don Ranasinghe, X. Xie, Z. Liang, E. Konstantinou, J. Heaton, and J. Schooling, “Developing a digital twin at building and city levels: case study of west cambridge campus,” *Journal of Management in Engineering*, vol. 36, no. 3, p. 05020004, 2020.
- [9] L. Yang and M. Worboys, “Generation of navigation graphs for indoor space,” *International Journal of Geographical Information Science*, vol. 29, no. 10, pp. 1737–1756, 2015.
- [10] S. Golodetz, “Automatic navigation mesh generation in configuration space,” *Overload Journal*, pp. 22–27, 2013.
- [11] D. E. Knuth, “A generalization of dijkstra’s algorithm,” *Information Processing Letters*, vol. 6, no. 1, pp. 1–5, 1977.
- [12] *IFC4.3 Specification*, buildingSMART, 2021, 4.3 RC2 - Release Candidate 2.
- [13] L. Liu, B. Li, S. Zlatanova, and P. van Oosterom, “Indoor navigation supported by the industry foundation classes (ifc): A survey,” *Automation in Construction*, vol. 121, p. 103436, 2021.
- [14] A. A. Diakité and S. Zlatanova, “Spatial subdivision of complex indoor environments for 3D indoor navigation,” *International Journal of Geographical Information Science*, vol. 32, no. 2, pp. 213–235, 2018.
- [15] M. Fu, R. Liu, B. Qi, and R. R. Issa, “Generating straight skeleton-based navigation networks with industry foundation classes for indoor way-finding,” *Automation in Construction*, vol. 112, p. 103057, 2020.
- [16] Q. Wang and M.-K. Kim, “Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018,” *Advanced Engineering Informatics*, vol. 39, pp. 306–319, 2019.
- [17] W. Moussa, “Integration of digital photogrammetry and terrestrial laser scanning for cultural heritage data recording,” Ph.D. dissertation, University of Stuttgart, 2014.
- [18] E. Angelats, M. Parés, and P. Kumar, “Feasibility of smartphone based photogrammetric point clouds for the generation of accessibility maps,” *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, no. 2, pp. 35–41, 2018.
- [19] R. Staiger, “Terrestrial laser scanning technology, systems and applications,” in *2nd FIG Regional Conference Marrakech, Morocco*, vol. 1, 2003, pp. 1–10.
- [20] V. Stojanovic, R. Richter, J. Döllner, and M. Trapp, “Comparative visualization of BIM geometry and corresponding point clouds,” *Building Information Systems in the Construction Industry*, p. 13, 2018.
- [21] J. W. Ma, T. Czerniawski, and F. Leite, “Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds,” *Automation in Construction*, vol. 113, p. 103144, 2020.
- [22] F. Engemann, T. Kontogianni, A. Hermans, and B. Leibe, “Exploring spatial context for 3D semantic segmentation of point clouds,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 716–724.
- [23] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques,” *Automation in construction*, vol. 19, no. 7, pp. 829–843, 2010.
- [24] V. Stojanovic, M. Trapp, R. Richter, and J. Döllner, “Generation of approximate 2D and 3D floor plans from 3D point clouds,” in *VISIGRAPP (I: GRAPP)*, 2019, pp. 177–184.
- [25] J. Döllner, “Geospatial artificial intelligence: Potentials of machine learning for 3D point clouds and geospatial digital twins,” *PFG—Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 1, pp. 15–24, 2020.
- [26] J.-g. Park, “Indoor localization using place and motion signatures,” Ph.D. dissertation, Massachusetts Institute of Technology, 2013.
- [27] H. Shoushtari, T. Willemsen, and H. Sternberg, “Many ways lead to the goal—possibilities of autonomous and infrastructure-based indoor positioning,” *Electronics*, vol. 10, no. 4, p. 397, 2021.

- [28] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 421–430.
- [29] T. Willemsen, "Fusionsalgorithmus zur autonomen positionsschätzung im gebäude, basierend auf mems-inertialsensoren im smartphone," Ph.D. dissertation, HafenCity Universität Hamburg, 2016.
- [30] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "Deep-learning-based pedestrian inertial navigation: Methods, data set, and on-device inference," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4431–4441, 2020.
- [31] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.
- [32] J. Liu, J. Luo, J. Hou, D. Wen, G. Feng, and X. Zhang, "A BIM based hybrid 3D indoor map model for indoor positioning and navigation," *ISPRS International Journal of Geo-Information*, vol. 9, no. 12, p. 747, 2020.
- [33] M. Deng, C. C. Menassa, and V. R. Kamat, "From BIM to digital twins: a systematic review of the evolution of intelligent building representations in the aec-fm industry," *Journal of Information Technology in Construction (ITcon)*, vol. 26, no. 5, pp. 58–83, 2021.
- [34] J. Döllner and B. Hagedorn, "Integrating urban GIS, CAD, and BIM data by servicebased virtual 3D city models," *Urban and regional data management-annual*, pp. 157–160, 2007.
- [35] K. Roper and R. Payant, *The facility management handbook*. Amacom, 2014.
- [36] V. Stojanovic, M. Trapp, B. Hagedorn, J. Klimke, R. Richter, and J. Döllner, "Sensor data visualization for indoor point clouds," *Advances in Cartography and GIScience of the ICA*, vol. 2, pp. 1–8, 2019.
- [37] J. Döllner, B. Hagedorn, and J. Klimke, "Server-based rendering of large 3D scenes for mobile devices using G-buffer cube maps," in *Proceedings of the 17th International Conference on 3D Web Technology*, 2012, pp. 97–100.
- [38] V. Stojanovic, M. Trapp, R. Richter, B. Hagedorn, and J. Döllner, "Towards the generation of digital twins for facility management based on 3D point clouds," in *Proceeding of the 34th Annual ARCOM Conference*, vol. 2018, 2018, pp. 270–279.
- [39] M. Borth, J. Verriet, and G. Muller, "Digital twin strategies for sos 4 challenges and 4 architecture setups for digital twins of sos," in *2019 14th Annual Conference System of Systems Engineering (SoSE)*. IEEE, 2019, pp. 164–169.
- [40] T. Birdal and S. Ilic, "A point sampling algorithm for 3D matching of irregular geometries," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6871–6878.
- [41] R. Pierdicca, M. Mameli, E. S. Malinverni, M. Paolanti, and E. Frontoni, "Automatic generation of point cloud synthetic dataset for historical building representation," in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, 2019, pp. 203–219.
- [42] V. Stojanovic, M. Trapp, R. Richter, and J. Döllner, "A service-oriented indoor point cloud processing pipeline," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 339–346, 2019.
- [43] *LAS Specification*, <https://www.ogc.org/standards/LAS>, OGC, 2018, version 1.0.
- [44] R. Richter and J. Döllner, "Out-of-core real-time visualization of massive 3D point clouds," in *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, 2010, pp. 121–128.
- [45] P. van Oosterom, O. Martinez-Rubi, M. Ivanova, M. Horhammer, D. Geringer, S. Ravada, T. Tijssen, M. Kodde, and R. Gonçalves, "Massive point cloud data management: Design, implementation and execution of a point cloud benchmark," *Computers & Graphics*, vol. 49, pp. 92–125, 2015.
- [46] S. Agarwal and K. Rajan, "Analyzing the performance of nosql vs. sql databases for spatial and aggregate queries," in *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*, vol. 17, no. 1, 2017, p. 4.
- [47] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, 2020.
- [48] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [49] E. Che, J. Jung, and M. J. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review," *Sensors*, vol. 19, no. 4, p. 810, 2019.
- [50] W. V. Toll, A. F. C. Iv, M. J. V. Kreveld, and R. Geraerts, "The medial axis of a multi-layered environment and its application as a navigation mesh," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 4, no. 1, pp. 1–34, 2018.
- [51] U. Isikdag, "BIM and IoT: A synopsis from GIS perspective," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, p. 33, 2015.
- [52] W.-L. Lee, M.-H. Tsai, C.-H. Yang, J.-R. Juang, and J.-Y. Su, "V3DM+: BIM interactive collaboration system for facility management," *Visualization in Engineering*, vol. 4, no. 1, pp. 1–15, 2016.
- [53] *Unity Game Engine*, <https://unity.com/>, Unity Technologies, 2021.
- [54] R. Cabello *et al.*, *Three.js*, <https://threejs.org/>, 2021.
- [55] R. B. Haber and D. A. McNabb, "Visualization idioms: A conceptual model for scientific visualization systems," *Visualization in scientific computing*, vol. 74, pp. 74–93, 1990.
- [56] *PostGIS*, <https://postgis.net/>, PostGIS Project Steering Committee, 2021, version 3.1.1.
- [57] *System Architecture for the 5G System*, 3GPP, 2021, document TS 23.501 V16.8.0.
- [58] *Node.js*, <https://nodejs.org/en/>, OpenJS Foundation, 2021.
- [59] D. Varrazzo *et al.*, *Psychopg*, <https://www.psychopg.org/>, 2021.
- [60] *PostgreSQL*, <https://www.postgresql.org/>, PostgreSQL Global Development Group, 2021.
- [61] G. Rauch *et al.*, *Socket.IO*, <https://socket.io/>, 2021.
- [62] *HTML 5.2 Specification*, <https://www.w3.org/TR/html52/>, W3C, 2017, version 5.2.
- [63] E. W. Weisstein, *Triangle point picking*, <https://mathworld.wolfram.com/>, Wolfram Research, Inc., 1999.
- [64] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [65] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [66] B. Skinner, T. Vidal-Calleja, J. V. Miro, F. De Bruijn, and R. Falque, "3D point cloud upsampling for accurate reconstruction of dense 2.5D thickness maps," in *ACRA*, 2014.
- [67] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [68] *SVG Specification*, <https://www.w3.org/TR/SVG11/>, W3C, 2011, version 1.1.
- [69] *GeoJSON Format Specification*, <https://tools.ietf.org/html/rfc7946>, Geographic JSON working group, 2016.
- [70] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a," *Journal of the ACM (JACM)*, vol. 32, no. 3, pp. 505–536, 1985.
- [71] A. Khan and K. Hornbæk, "Big data from the built environment," in *Proceedings of the 2nd international workshop on Research in the large*, 2011, pp. 29–32.
- [72] M. U. Khalid, M. K. Bashir, and D. Newport, "Development of a building information modelling BIM-based real-time data integration system using a building management system (BMS)," in *Building Information Modelling, Building Performance, Design and Smart Construction*. Springer, 2017, pp. 93–104.
- [73] A. Jain, D. Nong, T. X. Nghiem, and R. Mangharam, "Digital twins for efficient modeling and control of buildings: An integrated solution with scada systems," in *2018 Building Performance Analysis Conference and SimBuild*, 2018.
- [74] B. Tezel, Z. Aziz *et al.*, "From conventional to it based visual management: a conceptual discussion for lean construction," *Journal of information technology in construction*, vol. 22, pp. 220–246, 2017.

# SDN-based MANETs Using Existing OpenFlow Protocol

Saleh Rabia

*School of Mathematical and Computer sciences  
Heriot-Watt University  
Edinburgh, Scotland UK  
email res5@hw.ac.uk*

Skloul Idris I

*School of Mathematical and Computer sciences  
Heriot-Watt University  
Edinburgh, Scotland UK  
email i.s.ibrahim@hw.ac.uk*

Georgieva Lilia

*School of Mathematical and Computer sciences  
Heriot-Watt University  
Edinburgh, Scotland UK  
email L. Georgieva @hw.ac.uk*

**Abstract**—There is a continuous business need for network technologies to increase in functionality, performance and complexity. However, the present network paradigms show a lack of adaptability and are limited to single domain management. Thus, management of the network places a burden on the network's users. In addition, the high number and variety of stationary or dynamic devices make the network massive and intractable, with a complexity that leads to scalability challenges. Modern requirements cannot be supported by the current decentralized mobile ad hoc networks (MANETs) standard models. Additionally, MANETs suffer from packets/network overheads due to topology changes with the distributed and (decentralized) routing in each node. In a typical architecture, the mobile node is responsible for dynamically detecting other nodes in the network. The node can communicate directly or via an intermediate node, and to specify a route to other nodes. Thus, the node takes a decision with only a limited view of the network's topology. To this end, the deployment of the Software Defined Networking (SDN) paradigm has the potential to enable the redesign of these models. SDN provides a global view of network topology and a programmable network with centralized management. In this paper, we propose a new architecture for SDN-based MANETs, which is adding an Open Virtual Switch (OVS) per node to find the effect of OVS on the MANETs performance. We present a practical implementation for the new architecture using existing OpenFlow protocol. The tests have been carried out in an emulation environment based on Linux Containers (LXC V 2.0.11), Open Network Operating System (ONOS V 2.5.0) as a remote controller, NS3 and Open virtual Switch (OVS V 2.5).

**Keywords**—MANET; Software Defined Networking (SDN); Linux Containers (LXC); NS-3; OpenFlow protocol.

## I. INTRODUCTION

A collection of mobile nodes able to communicate without any need for fixed infrastructure or administration is called a MANETs. The MANET's nodes have a limited wireless transmission range and energy. In traditional MANETs,

MANETs cooperative mobile nodes take decisions (e.g., routing) independently based on their limited view and without global network topology knowledge. Routing protocols run in each node in a distributed way. In addition, the routing decisions which are taken by the mobile nodes are very difficult and sometimes need all mobile nodes to participate in this process, which may lead to high energy consumption and overheads [1]. Furthermore, these networks are prone to vendor-locking, are not flexible enough to allow them to be updated considering the modern requirements of a system, and are complex to manage [2]. According to recent estimates, SDN is expected to grow from \$13.7 billion in 2020 to \$32.7 billion by 2025, which represents an annual growth rate of 19% [3]. Thus, the SDN model is considered a new generation of networking. The SDN paradigm has shown its effectiveness in wired networks, such as data centres. However, deployment of SDN in MANETs has attracted very little attention and remains an open research problem. The architectures of the completely decentralized MANETs are considered as one of the main reasons they are not used for large topologies. With the SDN paradigm, this philosophy can be revisited. SDN provides centralized control, and a programmable and wide view network [4]. However, due to the dynamic environment of the MANETs, it is challenging to deploy SDN within it.

The Open Networking Foundation (ONF) [5] which is the organization responsible for the OpenFlow protocol defines SDN as “decoupling the data plane from control plane where the networking devices are controlled or updated using the open flow protocol and centralized SDN controller”. SDN controller is an application (software) that works as a strategic control point to manage the whole network. Recently, SDN has shed new light on how to control and manage mobile ad-hoc networks. The control and management of a MANETs

with an SDN paradigm is offered more flexibility and enable new features and services in the network. Moreover, the SDN paradigm provides a centralized network where all control plane and management plane functions are pushed to a centralized unit known as the SDN controller. In other words, SDN allows network operators/managers to define policies and/or behaviours of the network in the central controller by making control decisions for each new flow. The SDN controller is effectively the brain of the network, and defines the collaborative behaviour of the network and directs the participating nodes on how to behave. The aim of this work is to take advantage of the efficiency of SDN and implement this paradigm in MANETs using an existing open flow protocol. Additionally, each node is configured to support open flow, which will logically decouple the data plane and control plane. In other words, each node needs to have an OpenFlow switch and acts as router and end host at the same time. Figure 1 considers MANETs where the ONOS controller [6] is directly reachable for each mobile node (out-of-band mode).

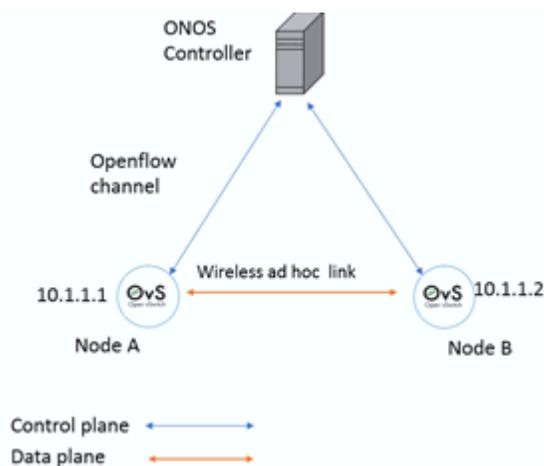


Figure. 1: Network topology.

The most challenging issue facing the SDN-based MANETs architecture is the dynamicity of the network topology, where MANETs nodes (e.g., vehicles and drones) move with a different direction and speed, in addition to security and reliability in the control plane to data plane communication between the SDN controller and OpenFlow-enabled nodes. There are two main approaches that have been proposed to address this problem, classified as out-of-band and in-band modes. Depending on the network architecture, the connection between the controller and forwarding devices can be either out-of-band, in which a different network is used for control communication, or in-band, in which the same network is used for both control communications and data [7]. The out-of-band mode is widely used because of its reliability, as it completely isolates control traffic from data traffic. However, it suffers from scalability issues, and thus, out-of-band mode is used in this paper.

This paper is organized as follow. Section II gives a brief

background of OpenFlow based SDN. Section III discusses related work. The proposed SDN-based MANETs architecture is described in Section IV, and the emulation scenario is presented in Section V. Additionally, in the same section, some emulation results are shown and discussed. Finally, Section VI concludes the paper.

## II. OPENFLOW BASED SDN BACKGROUND

The most important feature of OpenFlow based SDN is the separation of the control plane and data plane. In addition to this, the SDN paradigm provides programmable, simplified network configuration and management by pushing all control tasks to a centralized controller. An OpenFlow based SDN network is created by OpenFlow switches that communicate with one or more controllers and forward data packets using OpenFlow protocol.

A controller creates and configures the forwarding behaviour by setting rules in flow tables and sending them to each switch in the network. The rules consist of match criteria and actions. There is one or more flow tables in each OpenFlow switch. Thus, it is important to note that matching starts at the first flow table and may continue to additional flow tables in the pipeline [8]. When a packet arrives at the OpenFlow switch, it will first start in flow table (0) and check those entries based on priority. The packet with the highest priority will match first. If the flow needs to continue to another table, a “goto” statement tells the packet to go to the table specified in the instructions. If there is no match found in a flow table, the outcome depends on configuration of the table-miss flow entry. The actions define the packet processing to be applied by the switch, such as packet modification, output forwarding port or forwarding to the controller or flooding.

The OpenFlow protocol is used to enable the controller to install, update and delete the flow entries in the flow table of a switch in a reactive or proactive way. In the reactive form, this is in response to incoming packets, and the OpenFlow switch cannot make forwarding decisions, functioning as a dumb switch, and if the incoming packet does not match any rules in the flow table, the switch sends the packet to the controller as packet-in messages. The controller may perform different actions depending on its configuration. For example, the controller may calculate the path for this packet and send update flow entries to the requesting switch as packet-out messages.

An OpenFlow switch can operate at layer 2, layer 3 and up to the transport layer, according to the OpenFlow specification [8]. Thus, it can implement match criteria and actions at different protocol layers. The flexibility of the multi-protocol makes it possible to customize an OpenFlow switch to realize traditional network devices, such as an IP router, Ethernet, firewall, etc. In the OpenFlow switch/SDN, the control plane and data plane are separated. The control plane is processed by software called a “controller”, in which decisions are made, but the data plane is still implemented inside the switch. This is very different from the traditional router/switch, where routing is distributed in each device. In the legacy router/switch, IP

processing occurs when a packet arrives at the router, and the routing table associates the IP address of the destination with the next hop address and an outgoing interface. For instance, with an Ethernet as an interface, the router resolves the IP address of the next hop into a MAC address and forwards the packet by rewriting the MAC addresses. Furthermore, the MAC address of the destination becomes the resolved next hop MAC address and the MAC address of the source becomes the router interface MAC address. This is not the case in the Open virtual Switch (OVS)/SDN, where IP routing is replaced by the flow table. In an OpenFlow switch, a match field is used to match packets, consisting of the IP address source/destination, input port, MAC address source/destination etc.

#### A. SDN provides flowing functionality

SDN simplifies network operations, where it reduces complexity by decoupling the control and data planes, while making automation highly secure and scalable.

- It builds programmable networks.
- It has Easy Management (managed remotely), wherein SDN enables the administrator to manage their entire network as a single unit.
- It eliminates manual configuration. In legacy networks, the network is configured manually. Manual processes are usually resource-inefficient, cumbersome, complex and may be error-prone. With the SDN model, a network administrator/ operator can configure all the forwarding devices from a single unit (controller) using applications/API, and can test the configuration before pushing it to all network devices.
- Applications and services are deployed faster by leveraging open APIs.
- SDN provides the possibility of configuring and managing the network using software rather than hardware. It enables network administrators to configure their networks using a software application, rather than changing the configuration of physical hardware.
- There is a centralized MANETs and global view of the network topology. In open SDN, the brain is removed from the networking devices/nodes and placed into a centralized controller. Thus, SDN provides a holistic view of the network.
- Web Graphical User Interface (GUI) centralized controller: the Open Network Operating System (ONOS) controller web GUI is a single page web application which provides a visual interface for the controller [9].

### III. RELATED WORK

With the rapid development in wireless devices and the increased amount of data traffic transferred through channels, future expectations are that increase in wireless traffic will be larger than for wired traffic.

Few recent works have attempted to apply an SDN model in wireless networks and MANETs to improve performance. Kadhim, Seno and Shihab in [1] have proposed a new routing

protocol called the SDN-Cluster Based Routing Protocol (S-CBRP). In the proposed architecture, each cluster head works as a local SDN controller which relies on implementing an SDN agent in each node to manage one or more clusters. These local controllers connect to the central controller, which manages the whole network. Their proposed SDN architecture and routing protocol can increase the lifetime of the network and reduce delay in the route building/rebuilding process.

In [2], the authors provide an innovative method that can automatically configure the open flow in MANETs. The proposed method can be implemented in hybrid OpenFlow switches which support both OpenFlow and traditional routing protocols. The method was tested on MANETs that consisted of 20 nodes in less than 40 seconds, mobility was considered in each node, including the POX controller [10]. The results of the data traffic experiments show that the performance of the network using OpenFlow is lower than for a network that does not deploy OpenFlow. This is due to the use of tunnelling, such as in Generic Routing Encapsulation (GRE) and Virtual Extensible LAN VXLAN between the network nodes.

SDN integrated with wireless mesh networks (WMN) is proposed in [11], in which Detti et al. propose a hybrid approach in the OLSR-to-OpenFlow (O2O) architecture. The OLSR protocol is used in emergency conditions, such as the controller failing or being unreachable, in which the OLSR pushes rules in the OpenFlow switch. Additionally, OLSR is used to send updated network information to the POX controller if the network topology changes. The main disadvantage in this work is related to network updates, because updates to the network require extensive collection of information from the network when there are topology changes, which makes the updates extremely slow. The MANETs SDN-based quality management architecture proposed in [12] provides high flexibility by deployment of new flow management rules at provisioning time and ability to properly handle nodes join/leave event to reduce the network overhead. Authors in [13] have attempted to improve the route finding process. Each node has two different channels, the first is for the communication between the MANET's nodes and the second channel is used for connectivity between the MANET-Controller (MC) and mobile nodes. MC has a global view of the entire topology and receives updates of topology changes from the nodes. The results showed that including the bandwidth in the route finding process increases the reliability of transmissions and the network performance. However, delays slightly increase due to collisions in case where two nodes start transmitting at the same time.

An architecture for an SDN based MANETs is presented in [14]: however, the authors consider one hop links for control communication between mobile nodes and a controller. Additionally, an SDN bridge is used to transfer the OpenFlow frame over wireless links by making modifications on the Reactive Forwarding app. Our approach is almost the same, but we have not made any modifications to ONOS applications. In our proposed approach, each mobile node has a pure OVS (as one device) and works as forwarding and end device as

well. Furthermore, each node connects directly to a remote controller (out-of-band mode). The purpose of our contribution is to implement a new centralized MANETs architecture by using SDN paradigm and study their effect on the entire network performance. Thus, to evaluate the impacts of using existing OpenFlow protocol which was originally designed for wired networks on MANETs performance and management.

#### IV. THE PROPOSED SDN-BASED MANETS ARCHITECTURE

This section provides implementation details through which to understand the proposed SDN-based MANETS architecture.

The connection between nodes is in ad-hoc mode, which is implemented on the node. If OVSs have wireless ports and connect wirelessly, then a 4addr mode/WDS (wireless distribution system) or tunnels, such as VXLAN or GRE as proposed in [2] must be used. An alternative approach described in [14] uses an SDN bridge. In [15], the authors extensively study the use of 4addr mode in OpenFlow protocol/SDN. These options are used to resolve the problem of transfer of OpenFlow packets over a wireless link. OVS is based on the standard specification for Ethernet IEEE 802.3. However, the MANETs do not support the IEEE 802.3, and thus, this frame cannot be successfully transmitted or received over a wireless link unless it is retransformed to an IEEE 802.11 frame.

This work uses an emulated environment built with NS-3, Linux containers (LXC) [16] and an external ONOS controller [6]. An LXC is a set of processes that are detached from the rest of the Operating System (OS). These containers must be compatible with the underlying OS because LXCs share the kernel: only Linux distros (packages) can run with LXC (e.g., Fedora, Ubuntu, Gentoo, Debian. etc.), while Windows or any other OS cannot be run with LXC. Traditional virtualization, such as VMWARE and Virtual Box needs a full OS image for each instance. Thus, any OS can be run using traditional virtualization.

Ubuntu is selected as underlying OS and LXC. OpenFlow virtual Switches (OVS) [17] are attached to the LXCs. Each NS-3 node has a switch and each OVS has two ports: one connects to the controller, while the other connects to the NS-3 node using TapBridge in NS3 and Linux TapDevice. The TapBridge Model [18] allows the replacement of a particular NS-3 simulation node with real hosts. This module overwrites the MAC address of the NS-3 device (node) with the overlying real host MAC address. The real host considers the NS-3 net-device as a local device and all the NS-3 node’s incoming traffic will be sent by TapBridge through a virtual TAP interface which is connected to the LXC that host OVS through Linux Bridge, as Figure 2 shows. Moreover, the TapBridge model sends all outgoing traffic through the emulated wireless Ad-Hoc network. Thus, using the underlying network created by NS3, real devices can communicate with each other. The bridge and tap are configured completely outside NS3. Thus, TapBridge in NS3 uses an existing TAP interface previously configured and created by the user [18] [16] [19].

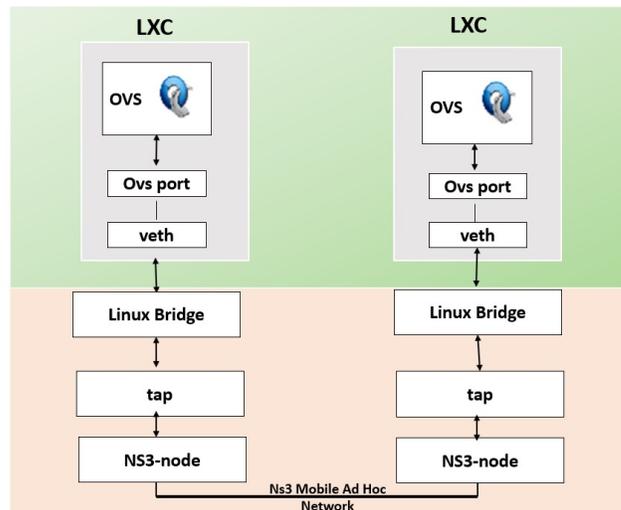


Figure. 2: Internal architecture of mobile Nodes.

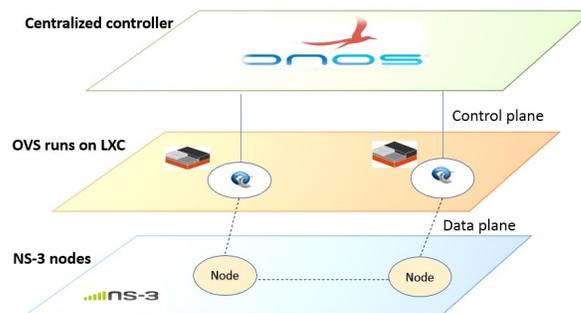


Figure. 3: Global view of the network.

The network is connected as out-of-band mode, as Figure 3 shown, the NS3 nodes are connected wirelessly using ad-hoc network mode. The first layer is remote ONOS controller, the second layer is OVSs where control plane messages are exchanged between the controller and OVSS. while the third layer consists of NS-3 nodes, which includes data plane traffic.

One of the main strengths of this work is that the system works with real applications, e.g., real hardware, Linux Containers, Virtual Machines, that can be directly used in real infrastructure afterwards. The framework is suitable for emulating both wireless (infrastructure and ad hoc) and wired networks. Because this implementation is designed to deploy MANETs scenarios, all the network scenarios provided are based on 208.11 Wi-Fi technology.

#### V. EMULATION SCENARIO

Our first approach was an attempt to adapt the OpenFlow switch13 module in NS3, as shown in Figure 4 to work with the MANETs by making some modifications in the NS3 mod-

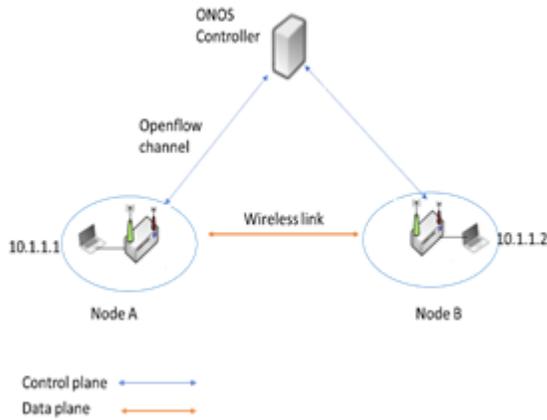


Figure. 4: First model (approach).

ule (source code), such as WifiNetDevice, Ad-hocWifiMac etc. We added Wi-Fi ports to OpenFlow switch, and each switch has two interfaces on the Wi-Fi port: one to connect the switches in ad-hoc mode; and the other one as an Ethernet port to connect the NS-3 node to the Wi-Fi switch. WDS technology [20] has been used in Wi-Fi ports to successfully transmit or receive OpenFlow packets over wireless. However, the communication was unsuccessful, due to NS-3's limited ability to enable OpenFlow.

The second approach was an adaption of an emulation environment using NS-3, LXC and OVS, as shown in Figure 1, where successful connection and data transfer was achieved. The system contains Virtual Box as a virtualization environment, on which runs Ubuntu 16.04 LTS image as an operating system (virtual machine VM), with 29 GB, 4096 MB RAM and 2 CPUs. NS-3, an LXC container attached to OVS, OpenFlow protocol and an ONOS controller running locally in a single machine with NS-3 and LXC. The experimental scenario consists of two mobile nodes: each node has OVS that are connected directly to an ONOS remote controller in out-of-band mode. Each node acts as forwarding and end device. The OVS consists of pure OpenFlow switches (fail mode set as secure). The difference between the approaches is that in the first approach, Wi-Fi interfaces are added to the switches and WDS enabled in the interfaces, while in the second approach, Wi-Fi is already implemented in the NS-3 nodes. In addition, the OVS and NS-3 nodes work as one device. Table 1 shows the software used in the model's implementation and experiments. Table 2 illustrates the network parameters.

To evaluate performance based on the new approach, we characterize three relevant metrics. These metrics are throughput, packet loss, and whether the controller is up or down. The performance measurements were carried out with traffic generated using the well-known tool iperf [21] in the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) for determining the performance of TCP and UDP flows over an SDN-based MANETs. In addition, the Wireshark tool was used to capture packets on interfaces of the network. Each test was executed more than ten times, while the simulation

TABLE I: SOFTWARE USED IN IMPLEMENTATION

Software	Function	Version
Ubuntu	VM OS	16.04.6
NS3	Nodes, mobility and 802.11 Link Emulation	NS-3.29
OVS	Forwarding devices (data plane)	2.5.0
Iperf	Throughput Measurement	V2
LXC	Ubuntu / guest (virtualized) operating system	2.0.11
ONOS	controller Control plane	2.5.0
OpenFlow	Protocol	1.3

TABLE II: NETWORK PARAMETERS

Parameter	Value
Propagation loss	Friis model
Network size	2 nodes, 2 OVS switches
Mobility	Random waypoint
Node speed	5m/s
Simulation area	100 x 200 m2
Simulation runtime	100s
Packet type	TCP / UDP
Traffic Size	1470 bps
TCP widow size	128 KByte
UDP buffer size	208 Kbyte
Wi-Fi standard	802.11a
Link	Ad hoc mode

run time was 100 seconds. These metrics may help the mission planner to gain an insight into the SDN-based MANETs using OpenFlow protocol. Additionally, these metrics show the effect of using a centralized unit (controller) and OpenFlow protocol in MANETs. TCP and UDP data traffic were generated and transmitted, from node B to node A, with node B as server and node A as client as shown in Figure 1, sending 1470 bits per second. Figure 5, and 6 show the TCP and UDP data traffic between two mobile nodes with the parameters stated in Table 2. When node A goes out of range of node B the connection is lost from about 23 seconds to 75 seconds. When after that, node A comes into the communication range of node B, it takes some time (delay) to start the data transmission again because the OVS sends packet-in to the controller requesting new flow table entries, and the controller sends packet-out to the OVS to install new entries. Thus, this process takes some seconds. The number of packets (TCP and UDP) that are lost in the network is shown in Figure 7.

These losses are due to the connection being lost when node leaves and joins the network. As UDP is a connectionless protocol, it is more likely that packets may be lost or that packets arrive out of order at their destination through transmission, whereas TCP is a connection-oriented and reliable

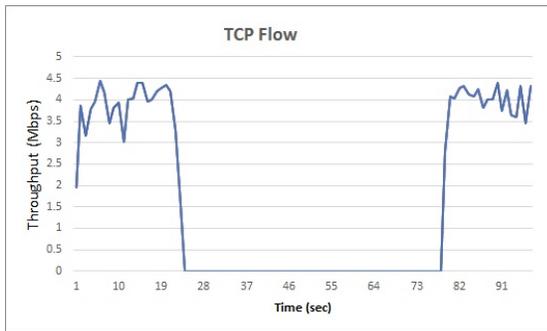


Figure. 5: Throughput in TCP flow.

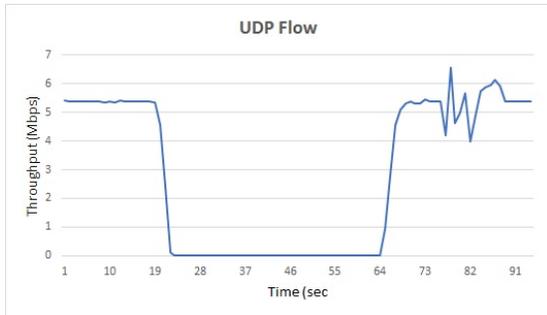


Figure. 6: Throughput in UDP flow.

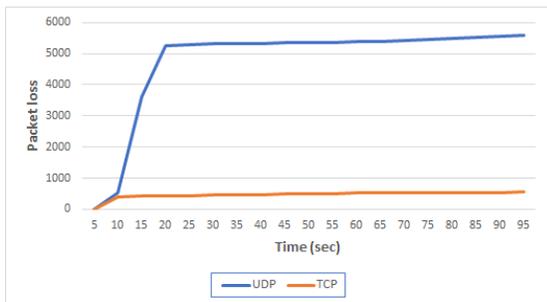


Figure. 7: Packet loss.

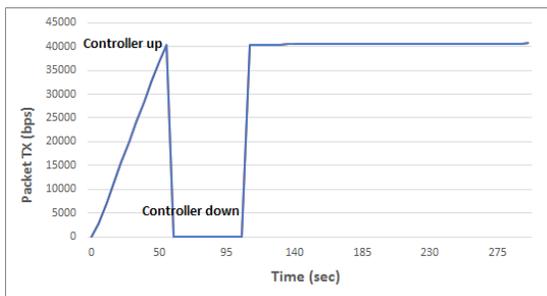


Figure. 8: Controller up/down.

data transfer protocol. Thus, as Figure 7 shows, the packet loss in UDP traffic is much higher than in TCP.

Pure OVS was used for this study, in which the switch does not support traditional route/switch protocols. Figure 8 illustrates when the controller is up or down. Therefore, when

the controller is down, there is no connection between the network nodes. On the other hand, the switches do not have a flow table installed on them (being dumb switches), and thus, they cannot forward packets by themselves without the controller.

## VI. CONCLUSION AND FUTURE WORK

The SDN paradigm has been increasingly extending its presence outside wired networks (e.g., datacentres, intranets) to wireless dynamic environments, such as MANETs.

In this work, a new proposed network architecture was described and discussed which showed how SDN may be applied in MANETs. Additionally, the paper presented a practical implementation of a centralized SDN based MANETs using existing OpenFlow protocol. The centralised control, and network applications which are provided by the controller make the network programmable and easy to manage. Pure OpenFlow switches were used in the implementation, which do not support traditional routing. Mobility is considered in terms of nodes only, but not including the controller. The results of our proposed approach are showing an average of 4 Mbps throughput for the TCP traffic and more than 4.5 Mbps in case of UDP, which is reasonable throughput. The reason for increased packet loss is due to nodes joining /leaving the network. The benefits of the SDN paradigm can be leveraged using a central remote controller and OVSs/OpenFlow enabled switches in the challenging environment of MANETs. The limitations of our proposed approach are due to the one hop link between each mobile node and the controller, and the single point controller. In both cases, there will be no communication at all in case of a controller failure, or no communication with a specific node in case of its single link failure. These two main limitations will be addressed in the future work as well.

In our future work, we plan to create larger complex scenarios and use multiple controllers in the network to avoid a single point of failure in a centralized system, which will also overcome scalability and congestion issues. Additionally, we will use the latest version of OVS (e.g., 2.15), and will compare our approach to the existing solutions which are provided in the related work.

## REFERENCES

- [1] A. Kadhim, S. A. Hosseini Seno, and R. A. Shihab, "Routing-protocol for sdn-cluster based manet," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 5398–5412, 2018.
- [2] S. Sharma and M. Nekovee, "Demo abstract: A demonstration of automatic configuration of openflow in wireless adhoc networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 953–954, IEEE, 2019.
- [3] "Software defined networking market size, share and global market forecast to 2025," 2020. accessed Apr. 09, 2021.

- [4] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Sdn-enabled tactical ad hoc networks: Extending programmable control to the edge," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 132–138, 2018.
- [5] ONF, "Software-defined networking (sdn) definition." accessed 1. 05, 2021.
- [6] "Open network operating system (onos)." accessed Mar. 30, 2021.
- [7] A. Dusia, V. K. Mishra, and A. S. Sethi, "Control communication in sdn-based dynamic multi-hop wireless infrastructure-less networks," in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, IEEE, 2018.
- [8] "Openflow switch specification 1.3.0," 2013.
- [9] A. Koshibe, "The onos web gui," 2020. accessed Mar. 10, 2021.
- [10] "noxrepo/pox: The pox network software platform.." accessed Apr. 03, 2021.
- [11] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless mesh software defined networks (wmsdn)," in *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 89–95, IEEE, 2013.
- [12] P. Bellavista, A. Dolci, and C. Giannelli, "Manet-oriented sdn: Motivations, challenges, and a solution prototype," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–22, 2018.
- [13] K. Streit, N. Rodday, F. Steuber, C. Schmitt, and G. D. Rodosek, "Wireless sdn for highly utilized manets," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 226–234, IEEE, 2019.
- [14] C. Y. Hans, G. Quer, and R. R. Rao, "Wireless sdn mobile ad hoc network: From theory to practice," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2017.
- [15] M. Rademacher, F. Siebertz, M. Schlebusch, and K. Jonas, "Experiments with openflow and ieee802. 11 point-to-point links in a wmn," *ICWMC 2016*, pp. 100–105, 2016.
- [16] "How to use linux containers to set up virtual networks nsnam." accessed May. 01, 2021.
- [17] "Open vswitch." accessed Mar. 20, 2021.
- [18] "Ns-3: Tap bridge model." accessed May. 02, 2021.
- [19] V. Sanchez-Aguero, F. Valera, B. Nogales, L. F. Gonzalez, and I. Vidal, "Venue: Virtualized environment for multi-uav network emulation," *IEEE Access*, vol. 7, pp. 154659–154671, 2019.
- [20] "ns-3-patch-wifi-wds.diff ." accessed Apr. 28, 2021.
- [21] "iperf - the tcp, udp and sctp network bandwidth measurement tool." accessed Mar. 23, 2021.