ICNS 2023

The Nineteenth International Conference on Networking and Services

ISBN: 978-1-68558-052-0

March 13th - 17th, 2023

Barcelona, Spain

ICNS 2023 Editors

Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain

Weizhi Meng, Technical University of Denmark (DTU), Denmark

ICNS 2023

Foreword

The Nineteenth International Conference on Networking and Services (ICNS 2023), held between March 13 – 17, 2023, continued a series of events targeting general networking and services aspects in multi-technologies environments. The conference covered fundamentals on networking and services, and highlighted new challenging industrial and research topics. Network control and management, multi-technology service deployment and assurance, next generation networks and ubiquitous services, emergency services and disaster recovery and emerging network communications and technologies were considered.

IPv6, the Next Generation of the Internet Protocol, has seen over the past three years tremendous activity related to its development, implementation and deployment. Its importance is unequivocally recognized by research organizations, businesses and governments worldwide. To maintain global competitiveness, governments are mandating, encouraging or actively supporting the adoption of IPv6 to prepare their respective economies for the future communication infrastructures. In the United States, government's plans to migrate to IPv6 has stimulated significant interest in the technology and accelerated the adoption process. Business organizations are also increasingly mindful of the IPv4 address space depletion and see within IPv6 a way to solve pressing technical problems. At the same time IPv6 technology continues to evolve beyond IPv4 capabilities. Communications equipment manufacturers and applications developers are actively integrating IPv6 in their products based on market demands.

IPv6 creates opportunities for new and more scalable IP based services while representing a fertile and growing area of research and technology innovation. The efforts of successful research projects, progressive service providers deploying IPv6 services and enterprises led to a significant body of knowledge and expertise. It is the goal of this workshop to facilitate the dissemination and exchange of technology and deployment related information, to provide a forum where academia and industry can share ideas and experiences in this field that could accelerate the adoption of IPv6. The workshop brings together IPv6 research and deployment experts that will share their work. The audience will hear the latest technological updates and will be provided with examples of successful IPv6 deployments; it will be offered an opportunity to learn what to expect from IPv6 and how to prepare for it.

Packet Dynamics refers broadly to measurements, theory and/or models that describe the time evolution and the associated attributes of packets, flows or streams of packets in a network. Factors impacting packet dynamics include cross traffic, architectures of intermediate nodes (e.g., routers, gateways, and firewalls), complex interaction of hardware resources and protocols at various levels, as well as implementations that often involve competing and conflicting requirements.

Parameters such as packet reordering, delay, jitter and loss that characterize the delivery of packet streams are at times highly correlated. Load-balancing at an intermediate node may, for example, result in out-of-order arrivals and excessive jitter, and network congestion may manifest as packet losses or large jitter. Out-of-order arrivals, losses, and jitter in turn may lead to unnecessary retransmissions in TCP or loss of voice quality in VoIP.

With the growth of the Internet in size, speed and traffic volume, understanding the impact of underlying network resources and protocols on packet delivery and application performance has assumed a critical importance. Measurements and models explaining the variation and interdependence of delivery characteristics are crucial not only for efficient operation of networks and network diagnosis, but also for developing solutions for future networks.

Local and global scheduling and heavy resource sharing are main features carried by Grid networks. Grids offer a uniform interface to a distributed collection of heterogeneous computational, storage and network resources. Most current operational Grids are dedicated to a limited set of computationally and/or data intensive scientific problems.

Optical burst switching enables these features while offering the necessary network flexibility demanded by future Grid applications. Currently ongoing research and achievements refers to high performance and computability in Grid networks. However, the communication and computation mechanisms for Grid applications require further development, deployment and validation.

We take here the opportunity to warmly thank all the members of the ICNS 2023 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICNS 2023.

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 ICNS 2023 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICNS 2023 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the fields of networking and services.

We are convinced that the participants found the event useful and communications very open. We also hope that Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ICNS 2023 Chairs:

ICNS 2023 General Chair
Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain

ICNS 2023 Steering Committee
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Mary Luz Mouronte López, Universidad Francisco de Vitoria - Madrid, Spain
Alex Sim, Lawrence Berkeley National Laboratory, USA
Poonam Dharam, Saginaw Valley State University, USA
Ivan Ganchev, University of Limerick, Ireland / Plovdiv University "Paisii Hilendarski", Bulgaria
Juraj Giertl, Deutsche Telekom IT Solutions, Slovakia

ICNS 2023 Publicity Chairs
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

ICNS 2023

ICNS 2023 General Chair

Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain

ICNS 2023 Steering Committee

Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Mary Luz Mouronte López, Universidad Francisco de Vitoria - Madrid, Spain
Alex Sim, Lawrence Berkeley National Laboratory, USA
Poonam Dharam, Saginaw Valley State University, USA
Ivan Ganchev, University of Limerick, Ireland / Plovdiv University "Paisii Hilendarski", Bulgaria
Juraj Giertl, Deutsche Telekom IT Solutions, Slovakia

ICNS 2023 Publicity Chairs

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

ICNS 2023 Technical Program Committee

Ryma Abassi, SUPCOM | University of Carthage, Tunisia
Abdelhafid Abouaissa, University of Haut-Alsace, France
Mohammad Reza Ghavidel Aghdam, University of Tabriz, Iran
Sami Marzook Alesawi, King Abdulaziz University, Rabigh, Saudi Arabia
Fatemah Alharbi, University of California, Riverside, USA / Taibah University, Yanbu, Saudi Arabia
Sadiq Ali, University of Engineering and Technology, Peshawar, Pakistan
Adel Alshamrani, University of Jeddah, Saudi Arabia
Mhd Tahssin Altabbaa, Istanbul Gelisim University, Turkey
Delaram Amiri, University of California Irvine, USA
Michael Atighetchi, Raytheon BBN Technologies, USA
F. Mzee Awuor, Kisii University, Kenya
Muhammed Ali Aydin, Istanbul University - Cerrahpasa, Turkey
Mohammad M. Banat, Jordan University of Science and Technology, Jordan
Ilija Basicevic, University of Novi Sad, Serbia
Imen Ben Lahmar, University of Sfax, Tunisia
Samaresh Bera, IISc Bangalore, India
Robert Bestak, Czech Technical University in Prague, Czech Republic
Bharat Bhasker, Indian Institute of Management Lucknow, India
Razvan Bocu, Transilvania University of Brasov, Romania
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Fernando Boronat Seguí, Universidad Politécnica De Valencia-Campus De Gandia, Spain
Abdelmadjid Bouabdallah, Université de Technologie de Compiègne (UTC), France
Christos Bouras, University of Patras / Computer Technology Institute and Press - Diophantus, Greece
An Braeken, Vrije Universiteit Brussel, Belgium
Claudia Canali, University of Modena and Reggio Emilia, Italy
Maria-Dolores Cano, Universidad Politécnica de Cartagena, Spain

Anne Kayem, Hasso-Plattner-Institute GmbH | University of Potsdam, Germany
Attila Kertesz, University of Szeged, Hungary
Samer Khamaiseh, Monmouth University, USA
Saeed R. Khosravirad, Nokia Bell Labs, USA
Pinar Kirci, Istanbul University-Cerrahpasa, Turkey
Masoomeh Javidi Kishi, Lehigh University, USA
Jerzy Konorski, Gdansk University of Technology, Poland
Josiane Kouam, INRIA & Ecole Polytechnique, France
Loïc Lagadec, Ensta Bretagne, France
Riccardo Lancellotti, University of Modena and Reggio Emilia, Italy
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Frédéric Le Mouël, INSA Lyon / University of Lyon, France
Diego Leonel Cadette Dutra, Federal University of Rio de Janeiro, Brazil
Beibei Li, Sichuan University, China
Chunguo Li, Southeast University, China
Xin Li, Google, USA
Tania Lorido Botrán, University of Deusto, Spain
Michael Mackay, Liverpool John Moores University, UK
Dharmendra Prasad Mahato, National Institute of Technology Hamirpur, India
Zoubir Mammeri, Toulouse University, France
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Daniel Marfil, Universidad Politécnica de Valencia, Spain
Weizhi Meng, Technical University of Denmark, Denmark
Ivan Mezei, University of Novi Sad, Serbia
Miroslav Michalko, Technical University of Košice, Slovakia
Mario Montagud, i2CAT Foundation & University of Valencia, Spain
Habib Mostafaei, Technical University Berlin, Germany
Mary Luz Mouronte López, Higher Polytechnic School | Universidad Francisco de Vitoria - Madrid, Spain
Avishek Mukherjee, Saginaw Valley State University, USA
Vasco N. G. J. Soares, Instituto de Telecomunicações / Instituto Politécnico de Castelo Branco, Portugal
Yu Nakayama, Tokyo University of Agriculture and Technology, Japan
Mort Naraghi-Pour, Louisiana State University, USA
Amar Nath, Sant Longowal Institute of Engineering and Technology Sangrur, India
Gianfranco Nencioni, University of Stavanger, Norway
Ranyelson Neres Carvalho, University of Brasília (UnB), Brazil
Diep N. Nguyen, University of Technology Sydney, Australia
Khoa Nguyen, Carleton University, Canada
Kim Khoa Nguyen, École de technologie supérieure | University of Quebec, Canada
Josef Noll, University of Oslo, Norway
Boubakr Nour, Beijing Institute of Technology, China
Serban Georgica Obreja, University Politehnica Bucharest, Romania
Roman Odarchenko, Bundleslab KFT, Budapest, Hungary
Vivek K. Pallipuram, University of the Pacific, USA
Luigi Pariota, University of Naples "Federico II", Italy
Jounsup Park, University of Texas at Tyler, USA
Lorena Parra, Universitat Politècnica de València, Spain
P. K. Paul, Raiganj University, India
Paulo Pinto, Universidade Nova de Lisboa, Portugal

Matin Pirouz, California State University, USA
Zsolt Alfred Polgar, Technical University of Cluj Napoca, Romania
Pavana Prakash, University of Houston, USA
Slawomir Przylucki, Lublin University of Technology, Poland
Cong Pu, Oklahoma State University, USA
Ziad Qais, Middle Technical University, Baghdad, Iraq
Tomasz Rak, Rzeszow University of Technology, Poland
Jiankang Ren, Dalian University of Technology, China
Éric Renault, Télécom SudParis | Institut Polytechnique de Paris, France
Sebastian Robitzsch, InterDigital Europe, UK
Will Rosenbaum, Max Planck Institute for Informatics, Saarbrücken, Germany
Vladimir Rykov, Piples' Friendship University of Russia (RUDN University), Russia
Anubhav Sachan, Samsung R&D Institute, Delhi, India
Ignacio Sanchez-Navarro, University of the West of Scotland, UK
Meghana N. Satpute, University of Texas at Dallas, USA
Arif Selcuk Ogrenci, Kadir Has University, Turkey
Mohamed Y. Selim, Iowa State University, USA
Purav Shah, Middlesex University, UK
Hamid Sharif, University of Nebraska-Lincoln, USA
Alex Sim, Lawrence Berkeley National Laboratory, USA
Jagdeep Singh, Sant Longowal Institute of Engineering & Technology (SLIET), India
Mukesh Singhal, University of California, Merced, USA
Rute Sofia, fortiss GmbH, Germany
Samy S. Soliman, University of Science and technology - Zewail City of Science and Technology, Egypt
Wen-Bin Sun, Northwestern Polytechnical University, Xi'an, China
Kevin (Kun) Suo, Kennesaw State University, USA
Fengxiao Tang, Tohoku University, Japan
Yoshiaki Taniguchi, Kindai University, Japan
Luis Tello-Oquendo, Universidad Nacional de Chimborazo, Ecuador
Yinglei Teng, Beijing University of Posts and Telecommunications, China
Giorgio Terracina, Università della Calabria, Italy
Mohamed Tounsi, Umm al Qura University Makkah, Saudi Arabia
Serpil Üstebay, Istanbul Medeniyet University, Turkey
K. Vasudevan, IIT Kanpur, India
Kefei Wang, Louisiana State University, USA
Huaming Wu, Center for Applied Mathematics - Tianjin University, China
Cong-Cong Xing, Nicholls State University, USA
Anjulata Yadav, Shri G.S. Institution of Technology and Science, Indore, India
Yiu-Wing Leung, Hong Kong Baptist University, Kowloon Tong, Hong Kong
Anna Zatwarnicka, Opole University of Technology, Poland
Justin Zhan, University of Arkansas | College of Medicine - University of Arkansas for Medical Sciences, USA
Tao Zheng, Orange Labs China, China
Fuhui Zhou, Nanjing University of Aeronautics Astronautics, China
Ye Zhu, Cleveland State University, USA
Galina A. Zverkina, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (ICS RAS), Moscow, Russia

Copyright Information

Table of Contents

A Topology Aggregation-based Approach for the Unsplittable Shortest Path Routing Problem

Hamza Ben-Ammar
Orange Innovation
Rennes, France
email: hamza.benammar@orange.com

Jean-Michel Sanner
Orange Innovation
Rennes, France
email: jeanmichel.sanner@orange.com

***Abstract*—The Unsplittable Shortest Path Routing (USPR) problem is one of the optimization problems that has been well studied in the field of traffic engineering for IP networks due to its importance for improving the network's Quality of Service (QoS). Given a directed graph representing the IP network and a set of commodities depicting the demands to be sent between its nodes, the USPR problem consists in identifying a set of routing paths and the associated administrative weights such that each commodity is routed along the unique shortest path between its origin and its destination following these weights. Due to the NP-hardness of this problem, we propose a topology aggregation-based approach to solve it, which consists in efficiently aggregating the network's graph to make the solving process more scalable. The experimental results show the efficiency of this proposal in terms of scalability and network's maximum load reduction compared to other methods from the state-of-the-art.**

Keywords—Traffic engineering; Mixed Integer Linear Programming; Topology Aggregation; Algorithms.

I. Introduction

With the continuous growth of traffic and the strong requirement for optimizing the use of networks' resources, Traffic Engineering (TE) becomes a major requirement of telecommunication networks. The goal of Traffic Engineering is to fit traffic flows with networks' constraints and operators' requirements. For instance, an objective can be to minimize the load of some network links to prevent the network from overload.

Unlike in Multiprotocol Label Switching (MPLS) networks, which embed some suitable centralized or distributed mechanisms for TE, intra-domain TE in Open Shortest Path First (OSPF) and IS-IS networks using Interior Gateway Protocol (IGP) based on Shortest Path First (SPF) algorithms is a very challenging task. This is particularly true if networks managers want to manage fine-grained internet traffic dynamically with short time slots.

The routing mechanism implemented in such IGP networks is based on the computation of the shortest path between nodes, using as lengths the weights assigned to each link in the network by the routing protocol. In each router, the next link on all shortest paths to all possible destinations is stored in a table. A flow arriving at the router is sent to its destination by splitting the flow between the links that are on the shortest path to the destination.

Solving the problem of finding the best and possibly unique set of weights assigned to each link to guarantee the correct flow of traffic and to fit some traffic constraints is NP-hard. It cannot be solved exactly in an acceptable delay when the network size exceeds some dozen of nodes. When it is required that each flow should be routed on a unique shortest path between its origin and its destination, it is called the Unsplittable Shortest Path Routing (USPR) problem.

The problem is even more difficult when the network is composed of thousands of nodes, without mentioning the fact that we can expect to adapt to short delay traffic variations. An empirical approach proposed by CISCO is to assign weights inversely proportional to the links capacities. This way provides a practical traffic engineering solution but it also provides results far from the optimal, without uniqueness guarantee for paths flows, and without adaptation to traffic variations in time.

The framework of telecommunications' networks is commonly characterized as Scale Free because the growing of the number of edges is sub-linear to the growing of the number of nodes. They are very often composed with some nodes, designed as hub, strongly connected with their neighboring nodes, and connected together with some edges. Telecommunications' networks can also be characterized as sparse networks because the number of edges is far lower than the maximum number of links within the network. In the end, sometimes some nodes or edges can be characterized as separators because when removed, they split the network in two parts.

These common features occurring frequently in telecommunication networks suggest that a heuristic to solve some NP-hard problems like the USPR problem applied to networks graphs could be designed based on their usage. The principle would be to simplify the topology by using the structure of the network and trying to keep some useful properties of the graph to guarantee that the algorithm will provide acceptable solutions, in acceptable delays.

In this work, we propose a topology aggregation-based (TA) approach to solve the USPR problem. The idea is to aggregate some parts of the network graph to tailor the new graph to the solver performances applied on a Mixed Integer Linear Programming (MILP) modelling of the problem. This process could possibly be done in an iterative way. We conducted some experimentation on a set of various networks, and we achieved some promising results. To the best of our knowledge, using topology aggregation to tackle the USPR problem has not been considered in previous studies.

The rest of the paper is organized as follows. In Section II, we present state-of-the-art of topology aggregations' techniques in telecommunications' networks and of MILP formulations of the USPR problem. Section III provides the detailed description of the problem and our proposal. In Section IV, we present a set of experimental results. Finally, Section V summarizes the achievements of this paper and describes future works.

II. Related Work

The shortest path routing problem (splittable and unsplittable cases) has been widely studied in the literature [1].

In [2], the authors studied the problem of computing a set of unique weights enforcing a given set of routing paths. They then proposed linear programs to solve this problem and boundaries for heuristics based resolution methods. The authors in [3] studied the Internet traffic optimization through OSPF weights setting. They proposed a local search heuristic which provided good solutions close to the optimal for their particular network. In [4], a MILP formulation of the problem's splittable version was developed using a family of valid inequalities. The MILP model is solved with a branch and cut algorithm capable of providing solutions in an acceptable delay for small networks with a controlled optimality gap. A local search heuristic is used to initialize the branch and cut algorithm.

Bley proposed in [5] an integer programming algorithm for the minimum congestion unsplittable shortest path routing problem. A general decomposition approach is adopted to provide a master and a client sub problems, which results in a significant improvement for small and middle size problems solving. The authors in [6] studied the delay constrained unsplittable shortest path routing problem. They extended the results presented in [5] by adding new valid inequalities, strengthening the linear relaxation of the problem. Then, they used this new formulation as a building block to perform a dynamic programming approach resolution.

Approximating dense graphs by sparse ones or preserving some useful properties like cuts or spectral representations of the graph are explored in some papers (for instance in [7]). The idea is that some algorithms can run faster on the sparse graph approximation to find good solutions in a reasonable time. However, there are no reasons that these techniques are suitable for each problem to be solved on graphs.

Topology Aggregation (TA) for routing purposes in networks is a well studied subject in the literature [8]. In [9] and [10], a TA tutorial for hierarchical routing in ATM networks containing the basics and conventional methods for topology aggregation along with performance evaluation of several aggregation schemes is provided. The authors in [11] analyzed the impact of TA on different QoS routing algorithms. Multiple TA methods were used with different networks settings. In [12], a novel QoS model for topology aggregation in delay-bandwidth sensitive networks was presented. Later on, several works [13]–[15] used TA dealing with problems like Virtual Network Embedding, Service Function Chain Orchestration and Virtual Network Function placement for scalability or security issues.

In this paper, we propose a TA-based approach adapted for solving the Unsplittable Shortest Path Routing Problem, extending the work presented in [6].

III. Problem statement and proposal

In this section, we present a formal description of our problem followed by an in-depth look of the proposed solution.

A. Problem description

Let $G = (V, E)$ be a directed graph representing an IP network where $V = \{v_1, \dots, v_M\}$ depicts the nodes of the network and $E \subset V \times V$ is the set of links connecting them. Each node $v \in V$ represents a router and each edge $e \in E$ corresponds to a logical link between adjacent nodes u and v with an associated capacity (bandwidth) denoted by c_{uv}. We denote by K a set of commodities representing the traffic demands to be routed between nodes of the graph G. Every commodity $k \in K$ is defined by a pair (s^k, t^k) where s^k and t^k being respectively the origin and destination of k. A traffic volume $D^k \geq 0$ to be routed from s^k to t^k is affected to each $k \in K$.

Let's recall that the USPR problem consists in finding a set of weights to assign to the graph's edges that will be used to generate a set of routing paths such that (i) there is a unique shortest path for each commodity and (ii) the network congestion is minimum. The load of an edge $uv \in E$ given a routing configuration $R \in \mathcal{R}(G, K)$, where $\mathcal{R}(G, K)$ depicts the set of all possible routing configurations of K in G, is the ratio between the total flow that goes through the edge and its capacity and it is defined as:

$$load(R, u, v) = \frac{1}{c_{uv}} \sum_{p^k \in R : uv \in p} D^k,$$

where a partial routing path for a commodity $k \in K$ in a graph G is a pair (p, k) denoted by p^k (p being a path).

The congestion $cong(R)$ of a routing configuration $R \in \mathcal{R}(G, K)$ is depicted as:

$$cong(R) = \max_{uv \in E} load(R, u, v).$$

It represents the maximum load over all arcs. The goal is then to minimize $cong(R)$, i.e., minimize the load of the most loaded link (denoted L).

B. MILP formulation

We define the binary variable x_e^k that takes the value 1 if commodity k is routed along a path containing edge e and 0 otherwise. Let u_e^t be a binary variable that takes 1 if e belongs to a shortest path towards destination t and takes 0 if not. We further depict w_{uv} as the weight assigned to the edge uv and r_v^u as the potential of node u, which is the distance between

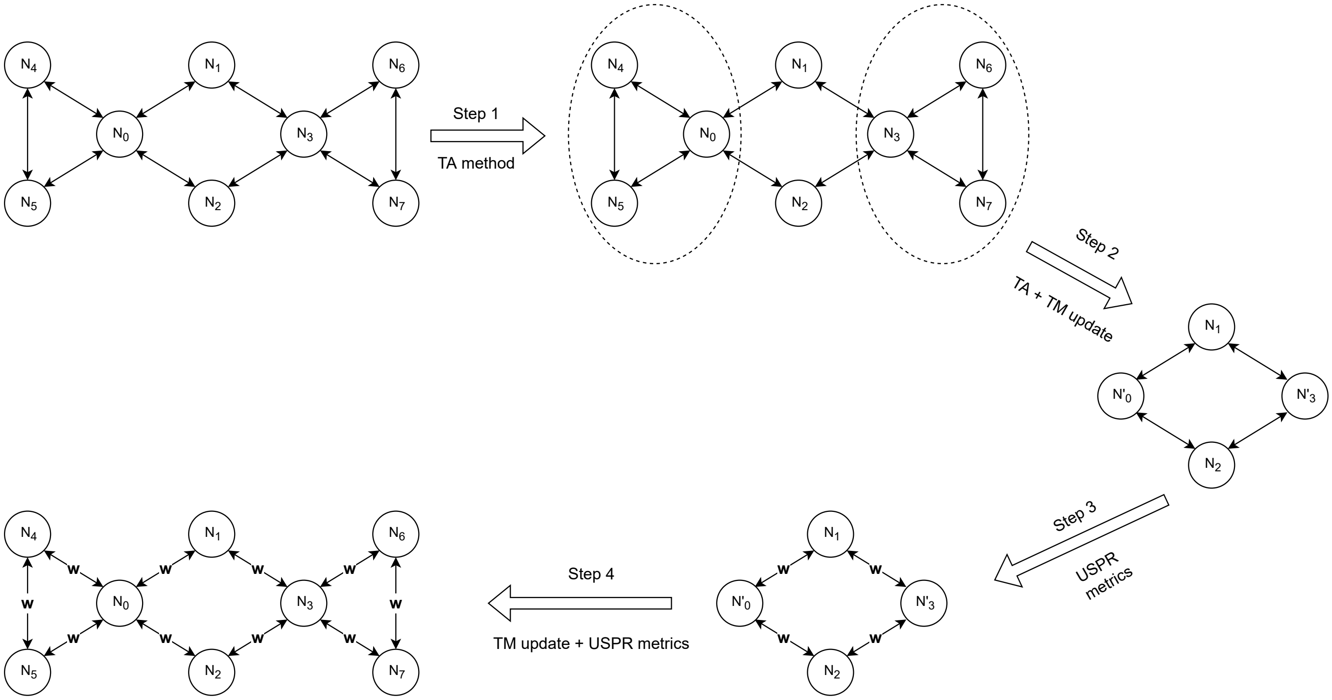


Fig. 1. Topology aggregation method for USPR.

nodes u and v. The USPR problem can then be described with the following MILP formulation [6]:

$$\text{minimize } L \tag{1}$$

$$\text{s.t.} \sum_{e\in\delta^+(v)} x_e^k - \sum_{e\in\delta^-(v)} x_e^k = \begin{cases} 1 & \text{if } v = s^k, \\ -1 & \text{if } v = t^k, \\ 0 & \text{otherwise}, \\ & {\scriptstyle \forall v\in V, \forall k\in K.} \end{cases} \tag{2}$$

$$\sum_{k\in K} D^k x_e^k \leq c_{uv} L, \forall e \in E, \tag{3}$$

$$\sum_{e\in\delta^+(v)} u_e^t \leq 1, \forall v \in V, \forall t \in T, \tag{4}$$

$$x_e^k \leq u_e^{t^k}, \forall e \in E, \forall k \in K, \tag{5}$$

$$u_e^t \leq \sum_{k\in K, t^k = t} x_e^k, \forall e \in E, \forall t \in T, \tag{6}$$

$$w_{uv} - r_u^t + r_v^t \geq 1 - u_{uv}^t, \forall uv \in E, \forall t \in T, \tag{7}$$

$$w_{uv} - r_u^t + r_v^t \leq M(1 - u_{uv}^t), \forall uv \in E, \forall t \in T. \tag{8}$$

The objective (1) is to minimize the load of the most loaded link, denoted L. Inequality (2) ensures that a unique path is associated to each commodity k and (3) expresses the load over an edge e. Inequalities (4) and (5)-(6) are anti-arborescence and linking constraints, respectively. In particular, inequality (4) ensures that there is at most one path traversing any node v towards a given destination $t \in T$, which is necessarily implied by Bellman property. Constraints (7) and (8) guarantee that the weight of any edge used by a shortest path towards a destination t corresponds to the difference of potentials between the end nodes of this edge and larger otherwise (a more detailed formulation with additional valid inequalities and constraints is presented in [6]).

C. Solving USPR using TA

1) Topology aggregation-based approach: As mentioned earlier, the USPR problem is NP-hard and using the previously described formulation to obtain an optimal solution is not practical in terms of running time and scalability. In this paper, we propose a topology aggregation-based approach to be used combined with the MILP formulation to solve the USPR problem and generate near-optimal solutions efficiently. Topology aggregation is defined as a set of techniques that abstract or summarize the state information about the network topology to be exchanged, processed, and maintained by network nodes for routing purposes [8]. TA is generally applied for mainly two reasons: security concerns (hiding the network's internal topology) and scalability issues, which represents in this work our main concern. In our case, we want to use TA in order to compress the topology information and reduce its complexity to enable the application of the USPR solver and provide solutions in reasonable delays.

An overview of the approach proposed is depicted in Figure 1. The starting point is the network's topology to be studied, which is represented by the graph G and a traffic matrix K containing the set of demands to be routed between nodes. For sake of simplicity, G in Figure 1 is represented by a bi-directed graph, but in real networks, the upload and download links between nodes are separated and can have different weights. The first step consists in using an aggregation method (which will be detailed in the next section)

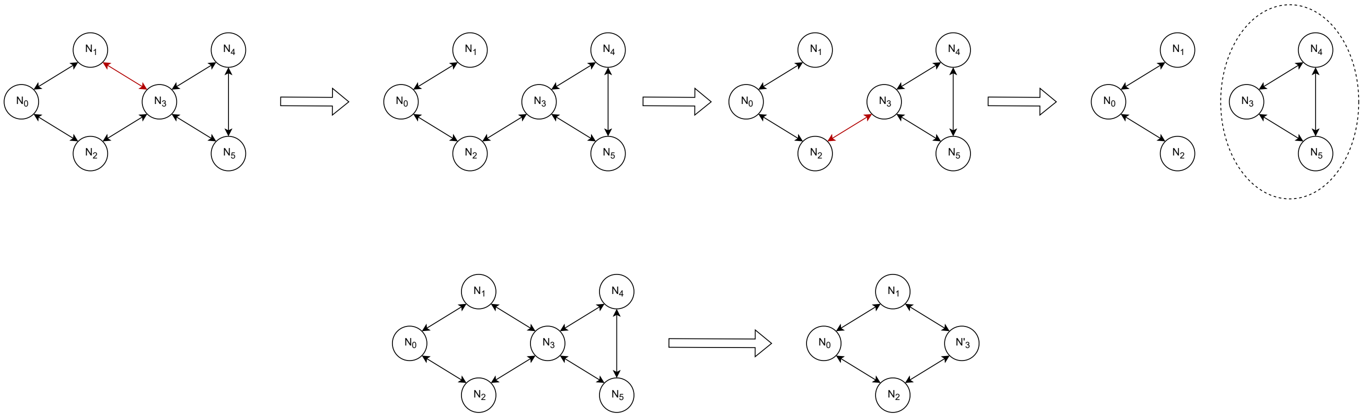


Fig. 2. GNC algorithm.

to select the nodes or the sub-graphs to be merged. Then, a new graph is generated according to the merged entities created along with a new traffic matrix where commodities will also be merged based on the aggregated graph. More specifically, the source (respectively the destination) of each commodity will be replaced by the aggregated node if it belongs to one. The next step now consists on applying the USPR solver on the newly generated graph to compute the set of different weights to be assigned to the links. In the next and final step, we adapt the initial traffic matrix (by keeping in each commodity only the sub-flows that have not been treated yet) in order to be able to apply the USPR solver on the aggregated sub-graphs. At the end, we have a full configuration of weights on all the links.

Algorithm 1 Aggregation algorithm

```
procedure GNC(Graph $G$)
    $List\_Aggregations \leftarrow \{\}$
    while $E \neq \emptyset$ do
        Calculate betweenness scores for all links in the graph
        Remove from the graph the link with the highest score
        if Disconnected sub-graph $G'$ forms a clique then
            $List\_Aggregations.insert(G')$
        end if
    end while
    return List_Aggregations
end procedure
```

2) Topology aggregation method: As we have seen previously, our goal is to select relevant sub-graphs from the initial topology to be aggregated. This choice will heavily impact the generated links' weights and thus, the network's maximum load that we want to minimize. In this paper, we propose a topology-aggregation method (see Algorithm 1) based on the Girvan-Newman (G-N) algorithm [16]. The Girvan–Newman algorithm detects communities by progressively removing edges from the original graph. The algorithm removes the "most valuable" edge, traditionally the edge with the highest betweenness centrality, at each step. The betweenness of a particular link is determined by computing the shortest paths for each couple of vertices in the graph representing the network and counting how many times each link appears on those shortest paths [17].

During each step of the G-N process and after computing all the links' betweenness values, the link having the largest value is removed. These steps will be repeated until a stopping condition is reached. Otherwise, the process will result in the removal of all the links, which reduces the network to its nodes. A stopping condition can be for instance the number of communities of a given size. In [16], the authors use the modularity metric as a quality indicator for the clustering process. This metric evaluates the partitioning of a graph by computing the ratio of intra-communities edges to the number of inter-communities edges (see [16] for details on the modularity metric's formula). In our process and at each link removal step, if an emerging community represented by a disconnected sub-graph forms a clique, it is aggregated (see Figure 2). Let's recall that a clique of a graph G is a sub-graph that is complete where each node is directly connected to all the others. In case where no cliques are found, we select sub-graphs having a maximum diameter of 2, *i.e*, the length of the shortest path between the most distanced nodes in the sub-graph.

In the proposed aggregation process, using the G-N algorithm to form communities will allow us to avoid aggregating the most valuable edges by removing them progressively from the network. These links are most likely to carry the maximum loads and as a result, their abstraction can lead the USPR solver to increase the overall maximum load. Conversely, the communities to be aggregated are set up with the less valuable edges.

IV. Results

We present in this section the experimental results related to our proposed TA-based approach for solving the USPR problem.

TABLE I. NETWORKS TOPOLOGIES CHARACTERISTICS.

Topology	$\|V\|$	$\|E\|$	$\|K\|$	Traffic pattern	Traffic volume	Links capacities	Avg node degree
Abilene	12	30	132	Uniform	Random	Uniform	2.50
Atlanta	15	44	210	Uniform	Random	Random	2.93
Newyork	16	98	240	Uniform	Random	Uniform	6.12
France	25	90	300	Random	Random	Uniform	3.60
Norway	27	102	702	Uniform	Random	Uniform	3.78
Nobel-us	14	42	91	Random	Random	Uniform	2.93
Nobel-ger	17	52	121	Random	Random	Uniform	3.06
Nobel-eu	28	82	378	Random	Random	Uniform	3.00

TABLE II. COMPARISON RESULTS.

Topology	InvCap	IGP-WO	NRPA	GNC	Exec time GNC	Exec time NRPA and IGP-WO	Exec time OPT
Abilene	48.12%	0%	0%	0%	0.3 min	10 min	1 min
Atlanta	54.58%	5.04%	5.04%	36.52% (0%)	2 min	10 min	4 min
Newyork	68.88%	37.77%	44.4%	28.8% (2.22%)	2 min	30 min	$>$ 4320 min
France	70.92%	19.50%	19.50%	60% (15.7%)	1 min	30 min	$>$ 4320 min
Norway	55.55%	7.40%	11.11%	59.31% (18.51%)	3 min	60 min	$>$ 4320 min
Nobel-us	53.51%	2.06%	2.06%	0%	6 min	10 min	58 min
Nobel-ger	43.15%	13.69%	13.69%	9% (6.75%)	1 min	10 min	122 min
Nobel-eu	24.74%	0.28%	0.28%	4.12% (1.31%)	8 min	30 min	1250 min

A. *Parameters*

The experiments were performed on networks with various sizes and characteristics taken from SNDlib [18]. The characteristics of the various network graphs tested are shown in Table I. A traffic pattern is considered uniform if it exists in K one commodity between each couple of nodes of the network (otherwise, it is a random one). A uniform traffic volume means that all the demand in the traffic matrix are equal (the same definition applies also for links capacities). Our proposal was implemented in Python using the NetworkX library for graph-related tasks, and Cplex (with the default settings) was used for the exact solving of USPR.

We compare our algorithm to the following common approaches from the literature:

- InvCap: a practical approach suggested by Cisco, this configuration sets the links weights inversely proportional to the capacities.
- IGP-WO: proposed in [19], this approach is based on a local search algorithm using dynamic graph algorithms to tackle the links' weights optimization.
- NRPA: in this approach [20], a Monte Carlo Search algorithm is used to solve the USPR problem through the application of Nested Rollout Policy Adaptation algorithm.

In order to evaluate the efficiency of our algorithm GNC and compare it to the other ones, we depict for each approach the execution time and the maximum load gap relative to the optimal value obtained by solving the MILP formulation of the problem using an exact approach OPT (or relative to a lower bound if no solution is found). The depicted results of IGP-WO and NRPA in Table II represent the average score of 5 executions as presented in [20].

B. *Performance evaluation*

The comparison results are shown in Table II. Regarding the execution time, we can see that GNC performs very well compared to NRPA, IGP-WO and especially OPT. The impact of topology aggregation on lightening the heavy computations to solve the USPR problem is very straightforward. In terms of maximum load gap, the performance depends on the tested topology network and its characteristics. For example, considering the Abilene results, all three approaches achieve a gap of 0%, which means that they achieve the optimal value. Looking at Atlanta topology, our algorithm does not perform well (36.52%) compared to IGP-WO and NRPA (5.04%). However, if a local search method is applied to the aggregations list generated by GNC (values reported between parentheses in GNC column), we can achieve excellent results (0%). Considering for example (N_0, N_1, N_2) as an initial aggregation, the local search method employed consists in replacing one node by another neighbor one (for instance to obtain (N_0, N_1, N_3)) or add a node to the aggregation list and check if it improves the obtained results. These results improvements show the potential of using our TA-based approach in general and the improvements that can be done by improving the aggregation level. We obtain similar results with France and Norway networks where a topology aggregation configuration can be found to achieve better results. For the remaining networks, our GNC algorithm outperforms IGP-WO and NRPA. For now, there is no clear correlation between the characteristics of the tested networks and the performance achieved by our algorithm (even in the case of a traffic matrix having a uniform pattern). Another point to be highlighted is that our approach (using the basic version of GNC) is deterministic compared to IGP-WO and NRPA, where results may differ with each experimental run.

V. Conclusion and future work

In this paper and in the context of traffic engineering, we tackle the Unsplittable Shortest Path Routing problem (USPR), which consists in finding the efficient weights of an IP network to handle traffics flows. The USPR problem is proven to be NP-hard and thus, many studies have used meta-heuristics to solve it. In this work, we propose a novel approach based on the Topology Aggregation (TA) paradigm in order to solve an USPR problem instance. The proposal consists of a TA-based methodology applicable to our use case in which an approach called GNC based on the Girvan-Newman community detection algorithm is implemented. The conducted experiments to evaluate GNC and compare it to other methods have shown the efficiency and the potential of using a TA-based method to solve the USPR problem. We intend in the future to improve our aggregation algorithm and conduct more extensive experiments to validate our work by testing other network configurations with challenging settings and analyze the correlation between the network's characteristics and the algorithm's performance. We also intend to study the delay-constrained version of USPR.

Acknowledgments

We want to gratefully thank our colleagues from Orange Innovation in Rennes Dr. Meryem Ouzzif for proofreading our paper and Xavier Grall for his help in producing the results shown in this paper.

References

[1] A. Altın *et al.*, "Intra-domain traffic engineering with shortest path routing protocols," *Annals of Operations Research*, vol. 204, no. 1, pp. 65–95, 2013.

[2] W. Ben-Ameur and E. Gourdin, "Internet routing and related topology issues," *SIAM Journal on Discrete Mathematics*, vol. 17, no. 1, pp. 18–49, 2003.

[3] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proceedings IEEE INFOCOM 2000. conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2. IEEE, 2000, pp. 519–528.

[4] A. Parmar, S. Ahmed, and J. Sokol, "An integer programming approach to the ospf weight setting problem," *Optimization Online*, 2006.

[5] A. Bley, "An integer programming algorithm for routing optimization in ip networks," *Algorithmica*, vol. 60, no. 1, pp. 21–45, 2011.

[6] A. Benhamiche and M. Chopin, "Toward scalable algorithms for the unsplittable shortest path routing problem," *arXiv preprint arXiv:2006.04324*, 2020.

[7] G. Goranci, "Dynamic graph algorithms and graph sparsification: New techniques and connections," *ArXiv*, vol. abs/1909.06413, 2019.

[8] S. Uludag *et al.*, "Analysis of topology aggregation techniques for qos routing," *ACM Computing Surveys (CSUR)*, vol. 39, no. 3, pp. 7–es, 2007.

[9] W. C. Lee, "Topology aggregation for hierarchical routing in atm networks," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 2, pp. 82–92, 1995.

[10] B. Awerbuch *et al.*, "Routing through networks with hierarchical topology aggregation," *Journal of High Speed Networks*, vol. 7, no. 1, pp. 57–73, 1998.

[11] F. Hao and E. W. Zegura, "On scalable qos routing: performance evaluation of topology aggregation," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 1. IEEE, 2000, pp. 147–156.

[12] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with topology aggregation in delay-bandwidth sensitive networks," *IEEE/ACM transactions on networking*, vol. 12, no. 1, pp. 17–29, 2004.

[13] M. Shen *et al.*, "Towards efficient virtual network embedding across multiple network domains," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 2014, pp. 61–70.

[14] G. Sun *et al.*, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1175–1191, 2018.

[15] C. Morin *et al.*, "Vnf placement algorithms to address the mono-and multi-tenant issues in edge and core networks," in *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*. IEEE, 2019, pp. 1–6.

[16] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[17] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[18] S. Orlowski *et al.*, "Sndlib 1.0—survivable network design library," *Networks: An International Journal*, vol. 55, no. 3, pp. 276–286, 2010.

[19] G. Leduc *et al.*, "An open source traffic engineering toolbox," *Computer Communications*, vol. 29, no. 5, pp. 593–610, 2006.

[20] C. Dang *et al.*, "Monte carlo search algorithms for network traffic engineering," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 486–501.

Host Migration Transparency Architecture by Cooperation between Multipath Transmission Control and VPN

1st Kohta Ohshima
Tokyo University of Marine Science and Technology
Tokyo, Japan
kxoh@kaiyodai.ac.jp

2nd Takehiko Kashiwagi
parallel networks LLC.
Tokyo, Japan

3th Yosuke Yano
iD corporation
Hokkaido, Japan

4rd Naoya Kitagawa
National Institute of Informatics
Tokyo, Japan
kitagawa@nii.ac.jp

***Abstract*—In this paper, we describe the basic design of network architecture with host mobility transparency for terminals equipped with multiple network interfaces. The features of the design are hardware acceleration using FPGA, functions to monitor and manage wireless connections, and the use of VPN to achieve both host migration transparency and seamless handover. We developed a prototype system and showed that it was possible to achieve host mobility transparency on the proposed architecture.**

***Index Terms*—host migration transparency, handover, Internet, VPN, mobility, multipath communication**

I. Introduction

Terminals equipped with multiple communication devices, e.g., smartphone and PCs, are increasing and there is a demand for continuity of communication using IP for mobile users. The problem of mobility in IP is that the address changes when the connected network changes due to movement. If the address changes during communication, a communication breakdown occurs and the terminal must be reconnected to continue communication. As a method of mobility support in IP environment, many methods of promptly notifying a new address to the source/destination terminal when an address changes, represented by Mobile IP [1], have been proposed. These methods are mainly intended for mobile terminals equipped with one network interface. Multipath TCP [2] is a protocol for improving the stability of communication in a terminal equipped with multiple network interfaces. Multipath TCP extends TCP to achieve increased bandwidth and fault tolerance by sending packets simultaneously using multiple paths. This method can be expected to have the effect of continuing communication even when moving, however using a Virtual Private Network (VPN) together may cause performance degradation due to the TCP-over-TCP problem.

This paper describes basic design of network architecture with host mobility transparency for terminals equipped with multiple network interfaces. The proposed architecture realizes a seamless handover that does not occur communication performance degradation and interrupt the communication session even when the address changes due to movement, while performing encrypted communication to ensure security.

II. Overview of Proposed Network Architecture

Figure II illustrates the proposed host migration transparency network architecture for multiple network interface environment. This architecture is constructed by terminals and routers. Terminals are information devices (PCs, smartphones, microcontrollers, etc.,) that can connect to the Internet via a gateway. Routers act as network gateways that enable site-to-site VPN connections, and also provide host migration transparency according to flow controller and connection controller. Flow controller has a function to manage and control how packets are distributed on multiple network interfaces. Distributing packets to multiple network interfaces can realize redundancy by copying packets, improvement of communication speed by round-robin transmission manner. However, when transmitting a large number of packets, the processing load becomes a bottleneck, and sufficient performance may not be obtained.

Our proposed architecture solves this problem using Field Programmable Gate Array (FPGA) hardware acceleration. Connection controller consists of the functions of monitoring wireless connection status and maintenance wireless connection. In this architecture, the router on the mobile side always establishes multiple wireless connections, and the wireless controller and flow controller work together to control at least one wireless connection to be in the connected state at all times. For example, when the signal strength of one wireless connection decreases, flow control is performed so that the other wireless connection is mainly used for communication, and the wireless connection whose signal strength has decreased is switched to a new connection. This operation allows the disconnect and reconnect delay times to be ignored. Reducing network switching time is important as it directly relates to QoS. A site-to-site VPN connection provides an encapsulation function that fixes the IP addresses between terminals even if the IP address on the Internet side of the router changes due to movement. Flow control and VPN can

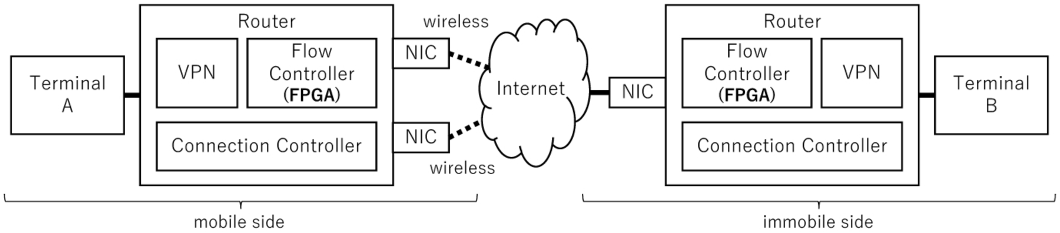


Fig. 1. Overview of the proposed network architecture

realize seamless handover function without adding functions to terminals.

III. Prototype Implementation and Evaluation

A. Prototype System

To verify that the host migration transparency feature of the proposed architecture works as intended, we developed a prototype system. This prototype system was developed using software and a wireless network interface, in order to confirm the effectiveness of the proposed architecture prior to implementation using FPGA,

Table I shows the specifications of the prototype system.

TABLE I
Specifications of the Router

Item	***Spec***
OS	Ubuntu 20.04 LTS
Wireless type	IEEE 802.11a
VPN	OpenVPN 2.54

B. Evaluation

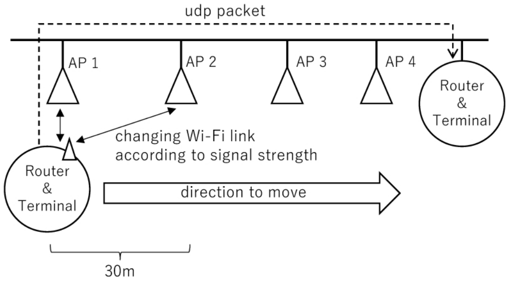


Fig. 2. Configuration of the evaluation environment

We conducted an evaluation experiment to confirm whether the proposed architecture can achieve host migration transparency. Figure 2 illustrates the evaluation environment. We evaluate host migration transparency and network reconnection latency by implementing a UDP packet transmitter in the mobile terminal that periodically sends UDP packets every 5 ms from the mobile terminal to the immobile terminal.

To evaluate the host migration transparency of this architecture, we used two different connection control methods: one that sets the SSIDs of all Wi-Fi Access Points (APs) the same and one that sets the SSIDs of all Wi-Fi APs to different names. Both connection control method automatically connects to nearby APs with stronger signal strength when the signal strength with the connected Wi-Fi AP become weak. The former aggressively switches connections, and the latter tries to maintain ongoing connections as much as possible. These connection controls are used standard Linux functions in this experiment.

As a result, the terminal was able to continue communication even after switching the connected network due to movement. In the same SSID case, about 600 packets were lost when network switching occurred. The time it takes to complete switching is about three seconds because one packet is sent every 0.5 ms. In the different SSID case, about 3,600 - 5,900 packets were lost when network switching occurred, and network switching delay is 19 - 30 seconds.

IV. Conclusions

In this paper, we described a basic design of network architecture with host mobility transparency for terminals equipped with multiple network interfaces. The proposed method is characterized that communication can be continued even if the terminal moves in the network, and the hardware acceleration of packet processing by FPGA. As a result of evaluating the mobility transparency of the proposed method, it was found that although the communication could be continued even if the network connection changed, the communication interruption time changed depending on the connection control method. In the future work, we will apply FPGA and implement and evaluate seamless handover method using multiple established wireless connections. And we will develop a communication stability improvement method that uses two or more wireless communications at the same time.

Acknowledgment

This work was supported by JKA and its promotion funds from KEIRIN RACE, and supported in part by JSPS KAKENHI Grant Number 22K12008.

References

[1] C. Perkins, “IP Mobility Support for IPv4,” RFC 3344, IETF, Aug. 2002.
[2] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, “TCP Extensions for Multipath Operation with Multiple Addresses,” RFC 8648, IETF, Mar. 2020.

VPN User Authentication Using Centralized Identity Providers

Duarte Mortágua
IEETA, University of Aveiro
Aveiro, Portugal
email: duarte.ntm@ua.pt

André Zúquete, Paulo Salvador
DETI / IEETA, University of Aveiro
Aveiro, Portugal
email: {andre.zuquete,salvador}@ua.pt
0000-0002-9745-4361, 0000-0001-6832-9417

***Abstract*—The online access to an always growing set of services requires users to manage credentials to identify themselves to all of them. The reduce this burden on users, centralized authentication systems, ordinarily known as Identity Providers (IdPs), and Single Sign-On (SSO) protocols where developed and are often deployed. IdPs and SSO were mainly developed for Web-based interactions, first in the scope of a set of federated services belonging to one organization, later on wider scopes, such as for virtually everyone (e.g., Google or Facebook users) or for all citizens of a given country. The Portuguese national IdP, Autenticação.gov, is an example of this later case. Today, many adhering services, from both the public and the private sectors, enable users to authenticate themselves using the functionalities provided by Autenticação.gov. However, the use of this IdP, as well as of similar ones, is mostly limited to Web applications. The goal of this paper was to study the integration of IdP services with Virtual Private Network (VPN) setup processes, namely for the authentication of VPN users. To this end, we used a recent VPN technology, WireGuard, which became popular amongst vendors due to its speed, simplicity and adoption by the kernels of the mainstream operating systems. We propose a method for a WireGuard-based VPN client to connect to a VPN server and negotiate cryptographic keys associated to a user authenticated by a centralized, OAuth 2.0-enabled IdP. We implemented a VPN server that enables users to use two different IdPs, namely Google Identity and Autenticação.gov; they both support the OAuth 2.0, but in different ways.**

***Index Terms*—Identity Providers, Authentication, OAuth 2.0, VPN, WireGuard**

I. Introduction

Services provided by companies or public sector departments often require people to register themselves, i.e., to create an account. Such registration usually involves the provisioning of users' authentication credentials (usually a passphrase) and a recovery mechanism (usually an e-mail address or, more recently, a phone number). Furthermore, and normally more complex to validate, users associate extra identity data to their account that may be useful in the future (e.g., a P.O. box address, a payment method, etc.).

Centralized Identity Providers (IdPs) appeared to reduce the users' burden regarding account management. They permitted to evolve from a so-called silo approach (where services do not share accounts) to accounts that can be shared by a set of federated services. Those services, often called Relying Parties (RPs), trust on the user authentication implemented by an IdP, and sometimes they can even enforce the use of specific approaches, by specifying Level of Assurance (LoA) indications. Furthermore, the RPs also receive, upon a user authentication, a set of user identification attributes which they assume that are accurate. Such accuracy is the responsibility of a back-office service used by an IdP, the Identity Manager (IdM).

Single Sign-On (SSO) is a concept that leverages centralized IdPs. Besides the centralization of the authentication in an IdP, SSO also enables users to remain authenticated during a time lapse, defined by the IdP (an authenticated session). Consequently, during that time they can access any federated RP without having to be authenticated for each of them.

IdPs and SSO were first explored in the context of Web interactions through the use of messages formatted with Secure Assertion Markup Language (SAML) [14] exchanged between an IdP and an RP through HTTP-based protocols, such as the Web Browser SSO Profile [13]. More recently, IdPs and RPs started to use OAuth 2.0, a protocol conceived to implement access control delegation, to allow RPs to access user identity resources maintained by an IdP. Nowadays, popular Internet services, such as Google and Facebook, which authenticate millions of people, an keep some relevant user identity attributes in their accounts, are often used as IdPs.

In order to facilitate the online identification of people in their interaction with services provided by the private or public sectors, several countries deployed an IdP for their citizens. This is the case of Autenticação.gov, created and maintained by the Portuguese state. This IdP enables citizens to authenticate themselves using two alternative methods, both implementing a two-factor authentication: a personal electronic identification device (Cartão de Cidadão, an eID crypto token) with a secret PIN or a combination of a secret PIN and a mobile phone number or e-mail address (Chave Móvel Digital). Other European countries followed a similar approach, for instance the ID Austria [3] or Cla@ve [9] in Spain.

This growing use of centralized IdPs for authenticating users and providing identification attributes about them to RPs happens mainly in the context of Web-based interactions, and considering that users use Web browsers to access the services provided by RPs. In this paper, we describe how we can explore an IdP for user authentication during a VPN setup, an action that became more frequent upon the recent Covid-19 pandemic. We chose to use a VPN technology, WireGuard, that uses as host (or user) authentication paradigm a set of public keys that must be pre-shared between VPN client and server. Then, we designed and implemented a protocol, involving a user browser and an IdP, which enables a VPN server to receive a trustworthy binding between a WireGuard public key and a user identity attribute. The provisioning of the identity attribute requires a user authentication by the IdP within the VPN setup protocol. For IdP services, we used Google Identity and Autenticação.gov. They both support OAuth 2.0, in different ways, to allow RPs (our VPN servers) to fetch a controlled set of identity attributes from the users they authenticate. A proof of concept implementation of the VPN client and server was successfully tested with those IdPs.

It is worth noting that we are not proposing a new authentication mechanism. We are proposing to benefit from the authentication mechanisms already explored by IdPs, and the subsequent provisioning of identity attributes associated to the authenticated person, during the setup of VPNs.

This paper is structured as follows. In Section II, we briefly describe the OAuth 2.0 details that are relevant to understand its use by IdPs and we detail how Google Identity and Autenticação.gov use it. In Section III, we explain the setup of a WireGuard VPN. In Section IV we describe the architecture of our solution. In Section V, we describe the implementation supporting two IdPs, Google Identity and Autenticação.gov. In Section VI, we discuss the security and usability of the final system. In Section VII, we present some related work. Finally, in Section VIII, we conclude the paper.

II. OAuth 2.0 in the context of IdPs

OAuth 2.0 is a protocol that can be used to authorize the access to protected resources in many different way. Thus, it can be explored differently by each IdP. In order to understand the integration of Autenticação.gov and Google Identity with our VPN setup process, it is therefore necessary to understand how these IdPs explore it in the context of users' identification. We start by an initial presentation of OAuth 2.0 concepts, and then we show how Autenticação.gov and Google Identity use it for user identification.

A. OAuth 2.0 concepts

OAuth 2.0 [10] enables an application (**client**) to access resources owned by a person (**resource owner**) kept by a **resource server**. In a nutshell, the client leads the resource owner (through their browser) to interact with the resource server in order to get an **authorization grant** to the client for accessing a set of resources. This interaction is conducted by the **authorization server** component of the resource server, which requires the authentication of the resource owner, shows the client identification and asks for permission to grant an access authorization to the set of resources listed by the client. This interaction ends successfully with the upload of an authorization grant to the client, which then uses it to get an **access token** from the authorization server. Finally, the client uses the access token as a bearer token to access the intended resources, kept by the resource server.

In the context of the centralized provisioning of personal identity attributes to federated services (RPs), the client is the RP, the resource owner is a person known by the IdP, and which the IdP knows how to authenticate, and the resource and authentication servers are parts of the IdP.

OAuth 2.0 was conceived for providing authorizations for clients wishing to access any kind of resource kept by a resource server. Thus, identity attributes are just a subset of those resources. There is an identity layer, called OpenID Connect (OIDC), that operates over OAuth 2.0, which uses ID tokens as resources. The knowledge of this layer is not fundamental to understand our system, and, in fact, we did not use it, because Autenticação.gov does not use it and Google Identity can be used without it. Consequently, we are not going to detail how it works.

B. OAuth 2.0 grant types

An OAuth 2.0 authorization grant can be obtained with 4 different approaches, which also define different interaction flows. Two of them, **resource owner password credentials** and **client credentials** grants, are not relevant, because they are meant to be used in special cases (when the client belongs to the resource owner and when the client is the resource owner, respectively) that are not suitable for our scenario.

The two grants that are of interest are **authorization code** and **implicit**. In the first case, the client receives an authorization code grant that it later uses to fetch an access token for accessing the resources of interest. The provisioning of the access token requires the client authentication by the resource server. In the second case, the client receives directly the access token. This approach is intended to be used by clients that are not meant to be authenticated by the resource server.

C. Registration of clients

The use of OAuth 2.0 implies a previous registration of the client in a resource server of interest. The registration requires the provisioning of the following items:

- Client type, either **confidential** or **public**. A confidential client is able to protect from disclosure a secret that can be used to get authenticated by the resource server (or its authorization server). A public client, on the other hand, cannot ensure the protection of such secret.
  In our case, the client will be an instance of a VPN server. It can be confidential, but in that case it requires a registration of each VPN server instance in the resource server. Or it can be public, if no secret is used or if the same secret is embedded in the code of all VPN server instances.
- Client identifier. This is a value that uniquely identifies the client in the scope of the resource server. It is provided by the later upon accepting the registration.
- Client authentication credentials. The resource server defines the alternatives (usually a password). Public clients may be required to make this registration, although not

enforcing a trustworthy identification (since they cannot protect the secrecy of the credentials).

- Redirection endpoint. This is a Universal Resource Identifier (URI) that is used by the resource server (or, more specifically, by its authorization server) to send the authorization grant to the client.

  In our case, we have several alternatives for defining this URI. We could have a URI per VPN server instance, but that would require a registration on each VPN server setup. Alternatively, since the URI is used in a communication initiated by a user browser (through HTTP redirection), the URI can contain an IP address that represents the browser host (e.g., 127.0.0.1). This solution is more appropriate for an exploitation scenario where there is a single registration for all VPN server instances (created by the VPN developers).
- Client identification items, such as application name, website, description, logo image, etc. Those items will allow users to recognize the client when it requires their identity attributes.

D. Autenticação.gov

Autenticação.gov uses the implicit grant flow with public clients without a shared secret. The registration of an OAuth 2.0 client in Autenticação.gov requires a manual agreement between the two parties, and the requester is naturally assumed to be an organization with an online portal. The registration includes 4 items: request issuer (portal URL), organization (the client identification to be presented to users), a contact e-mail (for technical issues) and a redirection endpoint domain (an IP address or a DNS domain). From this agreement results a client ID, which the client uses to identify itself in Autenticação.gov.

According to the public technical documentation of Autenticação.gov [1], the client needs to follow 3 steps regarding the user authentication and attribute provisioning:

1) Obtain an access token upon a successful user authentication by Autenticação.gov. This implies sending, to be presented to the user, the set of identity attributes the client wants to receive (scope). This step is initiated with a GET redirection of the user browser from the client to a specific Autenticação.gov authorization endpoint (see Table I). Upon a successful user authentication and authorization (to convey the indicated attributes), the browser is redirected to the client's redirection endpoint with an access token as a URL fragment.

   The client can specify how it wants the IdP to authenticate the user by means of an extra field in the initial request (`authentication_level`). If absent, the IdP will present to the user any available method.
2) Obtain an identifier of the authentication process (`authenticationContextId`) by presenting the access token and an optional subset of the attributes in the identification scope to Autenticação.gov with a JSON body.
3) Obtain the needed attributes by presenting the access token and the `authenticationContextId` to Autenticação.gov.

In our work we used a pre-production instance of Autenticação.gov. However, it is essentially a mirror of the production instance.

E. Google Identity

Unlike Autenticação.gov, the usage of the Google Identity IdP can be configured in an automated way through Google Cloud. This service allows the creation of Google Cloud projects, which may expose APIs and services to users. One of those services is Credentials, which allows the creation of OAuth 2.0 clients that may interact with Google APIs. The OAuth 2.0 consent screen presented to the users must also be configured, alongside the needed Google APIs that the OAuth 2.0 client may access, i.e., the OAuth 2.0 scopes.

The Credentials service allows the creation of multiple types of OAuth 2.0 clients [8], depending on the nature of the client application (Web App, JavaScript App, mobile App, etc.). In our case, the Desktop App was chosen, due to the fact that it allows the redirection of the Google's authorization server responses to a localhost-based redirection URI, with an arbitrary port, i.e., our VPN client running on the user's machine.

When a Credential is created, it generates a Client ID and a Client Secret, that can then be used by the client to implement the OAuth 2.0 flow. As opposed to Autenticação.gov, which uses the implicit grant flow, Google Identity uses the authorization code grant flow, which imposes client authentication towards the IdP.

In this case, there are also 3 steps for obtaining the attributes of a user:

1) Obtain an authorization code upon successful user authentication by Google Identity. Similar to the Autenticação.gov first step, this step implies making a GET request to Google's authorization endpoint (see Table I) that carries the OAuth 2.0 mandatory authorization parameters, such as the Client ID, the scopes and the redirection URI. After a successful user authentication, the user browser is redirected to the client's redirection endpoint with an authorization code in the URL query parameters.
2) Obtain an access token upon successful client authentication. In our case, this means the VPN server authentication with its Client ID and Secret, alongside with the authorization code. For this, one needs to use the Google Identity OAuth 2.0 token endpoint with URL encoded body).
3) Obtain the needed attributes upon presenting the access token to a Google API endpoint which the token is allowed to access, which is one of the token's scopes with the `Authorization` header as `Bearer` followed by the access token).

In the case of Google Identity, it was possible to publish the registered OAuth 2.0 client in production, which means

TABLE I
ENDPOINTS AND ACCESS METHODS FOR AUTENTICAÇÃO.GOV (TOP) AND GOOGLE IDENTITY (BOTTOM)

Endpoint	HTTP method and URL	
Authorization	GET	https://autenticacao.gov.pt/oauth/askauthorization?redirect_uri=...&client_id=...&response_type=token&scope=...
Client's redirection	GET	redirect_uri#token_type=bearer&expires_in=...&access_token=...
Authentication identifier request	POST	https://autenticacao.gov.pt/oauthresourceserver/api/AttributeManager
Attribute request	GET	https://autenticacao.gov.pt/oauthresourceserver/api/AttributeManager?token=...&authenticationContextId=...

Endpoint	HTTP method and URL	
Authorization	GET	https://accounts.google.com/o/oauth2/v2/auth?redirect_uri=...&client_id=...&response_type=code&scope=...
Client's redirection	GET	redirect_uri?code=...
Access token request	POST	https://oauth2.googleapis.com/token
Attribute request	GET	https://www.googleapis.com/oauth2/v3/userinfo

that our VPN server was able to authenticate anyone with a Google account.

III. WIREGUARD VPN SETUP

A WireGuard VPN is essentially a secure IP tunnel between two or more peers implementing WireGuard network interfaces [6]. These implement the concept of Cryptokey Routing, where each interface only needs to know its peer interfaces public keys and tunnel IPs [7].

In Linux systems, the WireGuard interfaces are configured using command line tools and configuration files. In order to connect a VPN client to a VPN server, we first need to create asymmetric key pairs for each one of them. To do so, we can run

```
wg genkey > privkey
wg pubkey < privkey > pubkey
```

on each machine, generating two pairs of keys. These are Curve25519 key pairs, which are used to run Elliptic Curve Diffie-Hellman key distribution protocols [2].

A configuration file for a WireGuard server (e.g., wg0.conf) would have a structure as follows:

```
[Interface]
PrivateKey = server private key
ListenPort = UDP port to listen for clients
Address = server VPN IP address / netmask bits

[Peer]
PublicKey = client public key
AllowedIPs = traffic to tunnel to/from the peer
```

This configuration essentially declares that the WireGuard's VPN server interface will have the indicated private key and tunnel IP address, and will use a given UDP port to interact with the peers.

Several peers can be indicated for each interface, and each is identified by its public key. The peers's tunnel UDP/IP addresses are not fixed, they can even vary over time (peers can roam). The tunneled traffic from peers, however, must come from the allowed IP addresses. Similarly, the interface should be used to tunnel all traffic to those IP addresses. The list of IP addresses in AllowedIPs is a routing table when choosing an interface for outbound traffic, and an access control list for filtering inbound traffic.

For the client, the configuration file would be:

```
[Interface]
PrivateKey = client private key
Address = client tunnel IP address / netmask

[Peer]
PublicKey = server public key
Endpoint = server initial t unnel UDP/IP port
AllowedIPs = traffic to tunnel to/from the peer
```

This configuration declares that the WireGuard VPN client interface will have the indicated private key and tunnel IP address and will communicate with a peer (server) with the provided public key and tunnel UDP/IP endpoint. It also declares, through the AllowedIPs parameter, which traffic should be routed to that peer (destination addresses matching the parameter) and, vice-versa, which traffic from the peer can be accepted (source addresses matching the parameter).

WireGuard can be used in scenarios where both hosts can act like client and server to each other, thus peers. In that case, they both should have a configuration similar to the server's one. However, that is not our case; we are considering a scenario where a user (client) establishes a VPN to a server. In this case, they are not peers *stricto sensu*.

In Linux, the WireGuard interfaces can be set up by storing the configuration files on the folder /etc/wireguard and running the command

```
wg-quick up wg0
```

This means that the set up of WireGuard VPN endpoints can be done in a simple way, just by running a few commands, upon knowing some of its peers attributes, namely their public keys and tunnel IPs and UDP ports.

In our approach, we start from a minimum of shared knowledge to initiate a VPN: the VPN server hostname and a certified public key, for the server, and a user identity attribute, for the client. Note that this information is not related with the identification elements that WireGuard uses. It is during our setup protocol that we exchange, in a trustworthy way, the public keys of both WireGuard endpoints. Once exchanged, the keys (and the IP addresses being used so far) are stored in configuration files and both sides initiate the VPN.

IV. Proposed approach

The architectural approach that we propose to instantiate WireGuard-based VPNs with an IdP-based user authentication and identification is briefly summarized in Figure 1, and described in more detail along the rest of this section.

A. Exploitation of different IdPs

Our main goal was to create a VPN upon an OAuth 2.0 user authentication with one external IdP. Therefore, we need to adapt our proposal to the way existing IdPs deal with OAuth 2.0.

As we saw in Section II-B, the two OAuth 2.0 flows that can be used by IdPs are authorization code flow and implicit flow. And they are both used, in fact. Therefore, we looked for a solution that could work with both flows, while keeping the system flexible to evolution.

Since some IdPs are very bureaucratic for the registration of new OAuth 2.0 clients (notably Autenticação.gov), we decided that, in the worst case, we could have to have a single client registration for all our VPN server installations. Therefore, we decided to take the OAuth 2.0 for Native Apps approach [4] and use an universal IP address, a localhost address (e.g., 127.0.0.1), for the OAuth 2.0 redirection endpoint for all IdPs. This endpoint is handled by the VPN client (see Figure 1), but it is not the real OAuth 2.0 client (it is the VPN server). Consequently, the VPN client forwards all the data received in that endpoint to the VPN server.

The use of a localhost address for an OAuth 2.0 redirection endpoint requires acceptance by the IdP on the client registration. However, both the IdPs that we have used (Autenticação.gov and Google Identity) accept it. Therefore, it may not be an architectural limitation.

The VPN server is able to work with several IdPs, and presents their list for the user upon an initial VPN setup request (through the Web browser). This list can vary for different VPN servers, and grow with time, and the server's code may have to be updated for dealing with new IdPs. The VPN client, on the contrary, is completely agnostic about the IdPs used. Thus, it can deal with different evolutions of VPN servers regarding the IdPs supported by them.

The user identification by the VPN server depends on the IdP selected by the user, from a list provided by the server. Different IdPs may provide different identity attributes, therefore the VPN server needs to maintain a list of attributes to be requested per IdP, and also a list of attributes known by each IdP for each enrolled user. For example, for Autenticação.gov we used the user Portuguese Civil Identifier, whilst for Google Identity we used the user e-mail address.

B. Exploitation of WireGuard

Since we chose to explore WireGuard VPNs upon a IdP-based user authentication and identification, our architecture necessarily involves a trustworthy exchange of the WireGuard's client public key within the protocol used to authenticate the VPN server host and the VPN user.

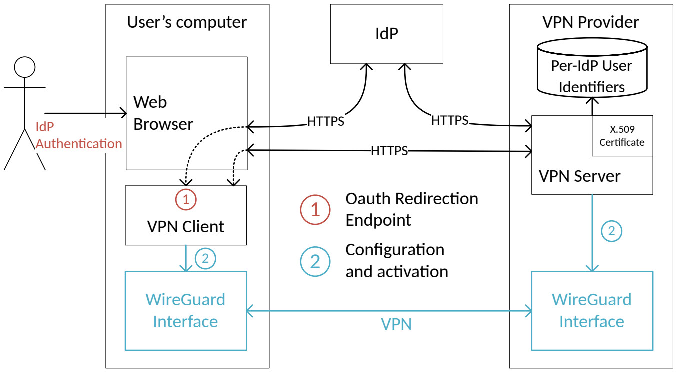


Fig. 1. High-level architecture and communication of our VPN solution

Furthermore, since our architecture requires the VPN server to handle HTTPS session, and these require a server-side X.509 public key certificate, we decided to separate the WireGuard's server side public key from the HTTPS certified public key. The first can be created on a needed basis (e.g., one per client), and are exchanged during the VPN setup, while the second, the certified public key, is expected to remain constant during the lifetime of its certificate.

Thus, our VPN client and server are applications that run a Web-based protocol that we partially designed, involving a user Web browser and an external IdP, which are able to configure WireGuard interfaces from scratch and initiate them to create a VPN. Figure 1 illustrates this approach.

C. Architecture overview

Our VPN client uses a (local) Web browser to initiate the VPN setup protocol for the current user (see Figure 2); this is required because of the way IdPs are normally explored. It also has a Web API (on localhost) to receive HTTP requests from external entities (IdPs and VPN server), redirected by the Web browser, in order to receive data required for the WireGuard setup. It does not participate on the user authentication; that is a responsibility of the IdP, and involves only the user and their Web browser.

The VPN server has a Web API accessible through HTTPS. The certificate used in the HTTPS server endpoint is the element that enables users to confirm that they are dealing with the right VPN server host. The certificate verification is performed by the users' Web browser.

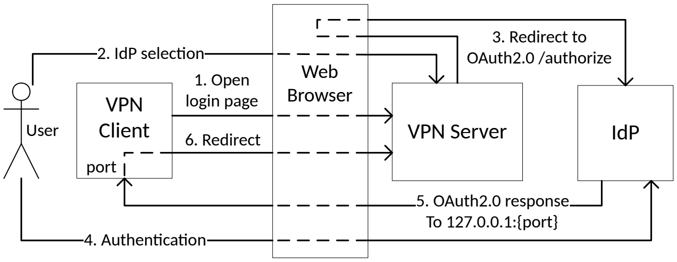


Fig. 2. Sequence of interactions for authenticating a user with an IdP

The user identification phase, which follows the user authentication by an IdP, requires the cooperation of the VPN client and server, and also uses the user's Web browser (for HTTP redirections). However, the VPN server is the sole entity responsible for retrieving the user's identity attributes from the IdP chosen by the user. Only upon the user identification, and the necessary authorization verification, the VPN server sends the server-side WireGuard interface parameters (public key and IP address) to the VPN client, allowing it to initiate the setup the intended WireGuard client interface.

All the communication between the VPN client and server is mediated through the user browser, and every communication between the user browser and the VPN server uses HTTPS. Therefore, no sensitive information (authorization codes, access tokens, keys, etc.) is sent in clear through the Internet.

D. VPN client Web API

The VPN client handles two HTTP endpoints in a single, variable TCP port, associated to a localhost IP address (see Figure 3).

The first endpoint is the OAuth 2.0 redirection endpoint (URL with path `/login_callback`). This endpoint is the one that receives the result of the user authentication on the selected IdP (through an HTTP GET redirection). The VPN client redirects the HTTP request to a VPN server endpoint (URL with path `/login_callback`), possibly with some parameters received in the request as part of a query string. The public key of the VPN client WireGuard interface is also provided as a parameter in the redirection to the VPN server.

When the IdP uses the access code grant flow is used, the URI contains the access code in one parameter of the URI query string (`code` [10, §4.1.2]). This parameter is added to the query string used for the VPN server redirection.

When the IdP uses the implicit flow, the access token is conveyed as a URL fragment [10, §4.2] (the last part of a URL, initiated by a `#` character). Since URL fragments are retained by browsers, and not conveyed to HTTP servers, the VPN server cannot immediately get the access token from the VPN client redirection. In this case, the VPN server needs to provide the user browser with a JavaScript-enabled resource that could read the fragment contents from the URL and upload them to the VPN server.

The second endpoint is used by VPN servers to add themselves, as WireGuard peers, to the a client WireGuard interface (URL with path `/vpn_parameters`). Upon a successful user identification, the VPN server makes a GET request to this local endpoint (through an HTTP redirection) carrying in the URL query string all the parameters required to set up a peer to an existing WireGuard interface: the server public key, the server UDP/IP port and the traffic to route to the server.

E. VPN server Web API

The VPN server handles three HTTP endpoints in a single, public HTTPS port (see Figure 3).

The first is the initial login endpoint, with a URL path `/login`. The VPN client launches a browser with this endpoint to allow the user to chose an IdP to authenticate with. As a query string parameter, the VPN client provides its localhost TCP port, so it can be included in the request to the IdP as part of the redirection URI. The response includes an HTTP cookie, which contains that port. This is done in order to retrieve the port later when the user actually chooses the IdP.

Client HTTP endpoints

`login_callback[?code=...]`
`vpn_parameters?pubkey=...&endpoint=...&ips=...`

Server HTTPS endpoints

`login?port=...`
IdP-specific login (e.g., `login/<IdP name>`)
`login_callback?pubkey=...[&code=...]`

Fig. 3. Web API of the VPN client and server. The parameters within square brackets are optional.

The second is the IdP-specific login endpoint. This is the endpoint which is requested when the user chooses a particular IdP. VPN servers can choose them freely, since they include them in the HTML resource presented to the user as result of the call to the generic login endpoint. The VPN server uses this endpoint, alongside with the port encapsulated in the cookie, to redirect the user browser to the chosen IdP Authorization endpoint, allowing the user to authenticate. The response also includes a new cookie with the name of the chosen IdP.

The third is login callback endpoint, with the URL path `/login_callback` (already referred in the previous section). This endpoint is where the VPN client redirects the response given by the IdP, possibly an authorization code as an URL query parameter. The cookie containing the IdP that was used will help the VPN server to select the appropriate approach to take in order get an access token from that authorization code or from the URL fragment that was kept in the user browser. Once having the access token, and knowing the IdP being used, the VPN server can request the necessary user identity attributes from the IdP, finalizing the user identification process.

V. IMPLEMENTATION

The focus regarding the implementation of the above approach was the IdP-based user authentication and identification, since the WireGuard VPN tunneling setup is trivial as long as the peers know each others public keys and IPs.

This section will start by explaining the implementation of the VPN client and server, followed by their interaction with the users, with their Web browser and with external IdPs.

A. VPN client

The VPN client is a Flask [15] application that runs in an arbitrary localhost TCP port (127.0.0.1:port). It starts by firing up a browser and opening the VPN server's login page, through HTTPS, with the query argument `?port=port`. It then listens to requests from the remaining entities in its two HTTP endpoints (`login_callback` and `vpn_parameters`).

B. VPN server

The VPN server is a Django [5] application. Its code is mostly generic for all IdPs. The handling of different IdPs happens when the user browser is redirected to the IdP selected by the user (each IdP has a specific URL for this purpose) and when the VPN server needs to get the user identity attributes from an IdP upon receiving a request on the `login_callback` endpoint.

In the last case, the code needs also to handle different OAuth 2.0 flows. When handling OAuth 2.0 implicit grant flows, in which an access token is directly provided by the IdP, the only thing to do is to build the specific request for the IdP to retrieve the user's identity attributes. When handling OAuth 2.0 access code grant flows, in which an authorization code is first provided by the IdP, these require the VPN server (OAuth 2.0 client) authentication by the IdP (using the registered credentials) to get an access token, first, and then a request with the access token to get the user's identity attributes.

However, it is not possible to provide an abstraction for the retrieval of the user's identity attributes, not even per flow type, since each IdP implements that process in their own particular way, within the OAuth 2.0 framework boundaries. Therefore, the VPN server must know each IdP particular way to implement the respective OAuth 2.0 flow. Furthermore, the user identity attributes provided by each IdP are also different.

C. Detailed communication

When the user wants to login on a VPN server, they execute the VPN client with the hostname (and possibly a port) of the server's login endpoint. The VPN client fires up the user's browser with the VPN server login endpoint plus with the VPN client's localhost port as a URL parameter. This port is returned encapsulated in a cookie of the server's response, in order to be maintained along the following browser-server HTTP interactions (cookie1 in Figure 4).

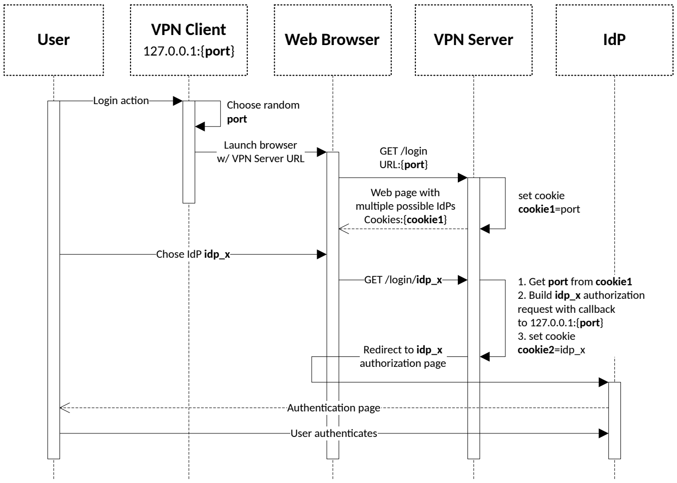


Fig. 4. VPN setup initial phase: IdP-based user authentication

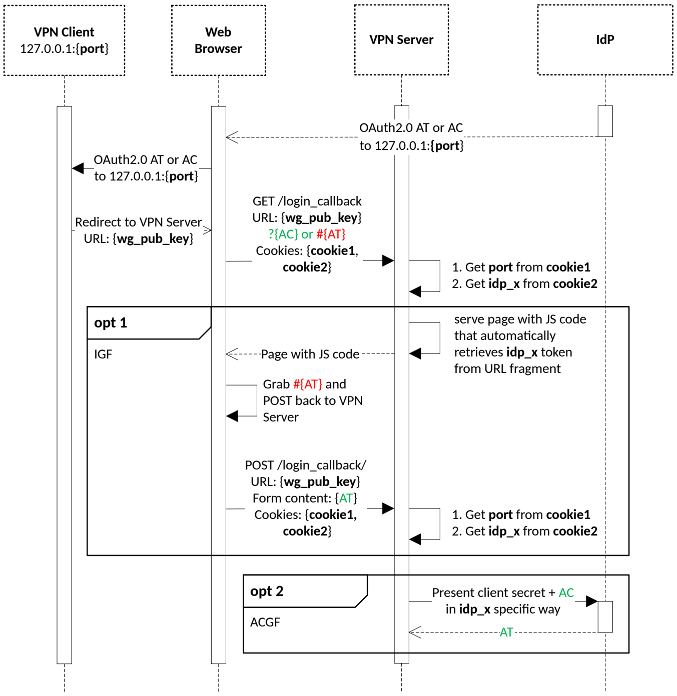


Fig. 5. OAuth 2.0 access token (AT) retrieval by the VPN server. The AC acronym stands for access token. The first optional interactions take place when the implicit grant flow (IGF) is used. The second optional interaction take place when the authorization code grant flow (ACGF) is used.

The VPN server response contains a Web page with all the possible IdPs that the user can use to authenticate with. According to the user's choice, a new request is made to the VPN server, which builds the appropriate OAuth 2.0 authorization redirection. Alongside with that redirection, another cookie is set in the user browser (**cookie2** in Figure 4), which contains the name of the chosen IdP. The user then authenticates with the chosen IdP. Figure 4 illustrates the protocol until this point.

The next phase is illustrated by Figure 5. After the user authentication, the IdP responds with an HTTP redirection to the redirection endpoint provided by the VPN server (which must conform with the registered one). This endpoint is a URL which is always uses the localhost IP address 127.0.0.1, combined with the port indicated by the VPN client. The parameters in the IdP response, together with the public key of the VPN client WireGuard interface, are then redirected to the VPN server.

If the VPN server receives a request to the `login_callback` endpoint, it expects that request to be a redirection from an IdP. The IdP is identified by **cookie2**, which makes it possible to the VPN Server to know which type of OAuth 2.0 flow that IdP implements (IGF or ACGF). If the VPN Server identifies an IdP that implements the IGF (optional block 1 in Figure 5), it returns a page containing JavaScript code that retrieves that specific access token, since URL fragments do not leave the browser. That JavaScript code will automatically run when the page is loaded, and will place the URL fragment token inside a

Form, which will be automatically submitted back to the VPN server.

If, on the contrary, the VPN server identifies an IdP that implements ACGF, and if the request contains an authorization code, it uses it, with the appropriate HTTP client authentication, to receive an access token (optional block 2 in Figure 5).

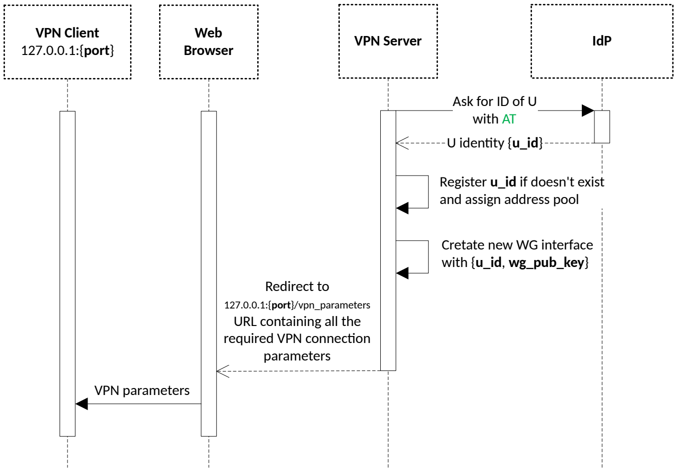

VPN Client
127.0.0.1:{port}
Web Browser
VPN Server
IdP
Ask for ID of U with AT
U identity {u_id}
Register u_id if doesn't exist and assign address pool
Cretate new WG interface with {u_id, wg_pub_key}
Redirect to 127.0.0.1:{port}/vpn_parameters URL containing all the required VPN connection parameters
VPN parameters

Fig. 6. User identity retrieval and configuration of the peer information in the user's WireGuard VPN endpoint

The final phase of the protocol deals with the user's identity attributes and with the setup of the WireGuard VPN, and is illustrated by Figure 6.

Once having the access token and knowing the IdP selected by the user (from **cookie2**), the VPN server retrieves the user identity attributes predefined for that IdP. When the identity attributes are received, the VPN server checks if they belong to an authorized user (registered with a set of attributes per IdP).

If the user is properly registered, the VPN server sets a peer to its WireGuard interface with the public key provided by the client and activates its WireGuard interface. Then, the VPN server sends a response with an HTTP redirection to the user's VPN Client containing the WireGuard setup information regarding its new peer (the server's WireGuard interface). This information includes the server WireGuard public key and IP. Once the configuration made, the VPN client can activate its WireGuard interface, allowing the peers to connect.

D. Testing

The solution underwent a timing analysis to evaluate its performance. The analysis focused on measuring the redirection delays between the VPN client and server, and the time to perform a login and create a WireGuard interface by the VPN server. The delays associated with the mandatory steps of the OAuth 2.0 framework and the user's authentication with the IdP were excluded as they are arbitrary and vary according to the IdP used. However, due to the lack of space in this document to present the results, and also due to the fact that the observed time figures are all less than a second, thus irrelevant in a login operation, we do not provide further details.

VI. Discussion

This section is dedicated to a discussion regarding the security and usability of the proposed solution. As the main goal is to have a VPN between a user host and a server (or network gateway), at the expense of having identity attributes involved, every confidentiality and trust aspect must be taken into account. Also, the solution should be easy, simple and secure for the user to benefit, and for the VPN server provider to implement and deploy.

A. Security

Our threat model excludes attacks against to and from the IdPs, because they are assumed to be trustworthy for the service they provide.

The threat model also excludes attacks against the VPN server from its host, because, when comparing with other solutions, we mainly remove the user authentication from it, therefore we reduced its attack surface. Furthermore, the OAuth 2.0 client credentials that it holds for all the registered IdPs are not critical for itself and for the VPN users, since they only allow it to fetch attributes from an IdP for a given user upon a proper interaction between that user and the IdP (therefore, with the consent of the former).

Finally, we also exclude attacks against the VPN client from its host, because otherwise, we would have to ultimately assume that user authentication credentials could be stolen and used by attackers to impersonate them.

Therefore, we assume that the threat model includes attacks against the exchanged messages (eavesdropping, tampering, or replaying) or against the communication endpoints. And we also assume that attackers may try to impersonate legit users or legit VPN servers. Finally, we assume that malicious VPN servers may attempt to steal user-related OAuth 2.0 data items provided by browsers in order to impersonate the associated users.

Confidentiality and integrity is assured between the VPN client and the VPN server, since every communication between the two entities is done over HTTPS (HTTP over TLS, mediated by the browser). The same happens between the VPN server and the external IdP. Thus, every communication regarding the VPN setup process between these three entities is properly ciphered, its integrity is assured and the server endpoints are authenticated with X.509 certificates.

Since HTTPS security is built upon using X.509 certificates, which provide a trustworthy binding between host names and public keys, the VPN client can be sure it is dealing with the intended VPN server. Thus, both VPN client and server can trust on the WireGuard configuration elements provided to each other, namely their interfaces' public keys.

Similarly, the VPN server can verify it is interacting with the correct IdPs, selected from a list that it provides and in which it trusts.

Regarding the correctness of the authorization code grant or access token received by the VPN server, that depends on the way the client registered on each IdP. In particular, they can only be fully trusted if the VPN server has a unique client ID and client secret shared with the IdP. Otherwise, if there is no client secret (implicit grant flow) or if the same client secret is used by all VPN server instances, then the credentials the VPN server receives from the IdP via the user browser may have been stolen from previous, similar interactions.

The steeling of IdP-provided credentials can occur in different ways.

When the IdP uses the implicit grant flow, the VPN server trusts on the correctness of the client user solely based in the presentation of a OAuth 2.0 access token, which is a bearer token. These tokens can be stolen inside the user's machine (e.g., by another application running locally [4]) or by an malicious agent running a server for which a similar access token was fetched by the user. For instance, with Autenticação.gov, from which we require the provisioning of a user's unique identifier (Portuguese Citizen Identifier), any malicious server requiring a similar access token from the user can use it later to impersonate the user in an access to an instance of our VPN server during a period of time defined by the IdP. However, this is not a problem of our solution, this is a problem of the way the IdP works.

This problem is similar when the authorization code grant flow is used with a well-known client secret (all VPN server instances use the same). In this case, stealing the authorization grant code from a similar interaction with the IdP (one involving the same client ID) allows its reuse in another instance of the client with the same client ID. Thus, a tampered version of our VPN server deployed by a malicious agent could reuse the access code grants received from users to impersonate them in the access to other similar VPN server deployments. Alternatively, the authorization code grant can be stolen in the user host by some malicious software and reuse to impersonate the user in the access to similar VPN server instances (they need to use the same client ID, otherwise the authorization code is useless).

The usage of the Proof Key for Code Exchange (PKCE) protocol [17] mitigates this last problem, since it implements a proof-of-possession extension to OAuth 2.0 that protects the authorization code from being used when stolen. It works by adding a code challenge to the first request to the IdP. This challenge results from the hashing of a random code verifier. The IdP stores the code challenge together with the provided authorization code. The client, then, sends the code verifier when requesting the access token, and the IdP verifies if hashing it produces the code challenge. Since hashing function are one-way functions (not invertible), only a legit client, with the original code verifier, can get the access token.

Google Identity supports PKCE [8], and we used it in our prototype, although we did not describe that step in Section V-C because the previous discussion was necessary to understand its relevance. The code verifier is generated by the VPN server prior to redirecting the control to the IdP, and stored in a ciphered cookie that it sends along with cookie2 (see Figure 5). The cookie encryption uses a secret key known only by the VPN server, created each time it is launched. This cookie will later be received along with the authorization code, and the embedded code verifier can then be used to request the access token.

Wrapping up, our VPN server can have more trust in the identity of the user when the authorization code grant is used, either with a client ID and secret per VPN server instance (possible with Google Identity, for instance) or with a client ID and secret shared by many VPN server instance, provided that PKCE could be used. On the other hand, when the implicit flow is used, there is more room for user identity stealing. In that case, the user identification by an IdP could be used mainly as a second factor authentication, in order to reduce the chances of impersonation.

B. Usability

In this field, we can discuss the usability of the proposed solution for both the users of the VPN client and for the VPN server provider.

Regarding the users, this solution is an advantage since it does not require learning a new authentication interface. In this case, the users will authenticate themselves using a Web interface which they already know and trust, while other VPN solutions rely on their own custom made interfaces, which are unknown to the users and always require some learning.

The proposed solution also provides a simple and scalable deployment strategy to the VPN server provider, since the VPN server requirements regarding user's identity are delegated in the possession of OAuth 2.0 credentials from the external IdPs. Redirect URLs do not need to be previously established with the IdPs, since they are always local to the VPN client. Regarding the VPN client instances, these just need to know a *priori* the VPN server domain.

VII. Related Work

In [11], the authors propose to authenticate VPN users with a X.509 certificate containing a SAML assertion as extension. This assertion contains the identity attributes required by the VPN server to authorize the user to create the VPN [12]. This solution requires a custom Certification Authority to issue those special certificates. Those certificates must also be obtained prior to instantiate the VPN.

A solution that supports VPN authentication using IdPs is Tailscale. With Tailscale, users can create VPNs that allow them to securely access resources on remote networks, as well as share files, printers, and other resources with other users on the network [16]. It is essentially a VPN software based on WireGuard that allows user authentication with some existing OIDC-based IdPs out of the box (Google Identity, Azure AD and GitHub) and with two SAML and OIDC IdPs (Okta and OneLogin). It also allows the integration of custom OAuth 2.0, SAML or OIDC providers [18]. This integration requires manual work and configuration and is only available through their paid Enterprise subscription [19].

VIII. Conclusions

This paper describes a VPN solution that resorts to external IdPs to authenticate client users. For the low-level

VPN implementation, we chose WireGuard, which supports a manual setup of the peers. The protocol designed for user authentication also distributes the critical elements (public keys and tunneling UDP/IP ports) that should be used for creating a WireGuard VPN. The majority of the user interface is handled by a Web browser, which is the usual tool the users use when they are authenticated by an IdP.

Our VPN solution was implemented with two IdPs, Autenticação.gov and Google Identity. They both support the use of OAuth 2.0 to authenticate people and fetch some of their identity attributes. However, they explore OAuth 2.0 in different ways, which were all considered in our architecture and tackled in the implementation.

Since each IdP can explore OAuth 2.0 in a different way, the code of our solution needs to be modified. Currently, we do not have a modular approach that could be used to add new IdPs while keeping the core system stable, but that can be done.

The security of IdP-based user identification depends on the way IdPs explore OAuth 2.0. We discussed some strategies and saw that some are weaker, namely the implicit grant flow. In that case, the user authentication by an IdP should be complemented with another authentication mechanism, to implement a two-factor authentication.

As a proof of concept, we implemented the system using Flask (for the client) and Django (for the server) and we deployed a VPN for tunneling all traffic from a user laptop to a VPN server deployed in the cloud, which would later route it to the Internet.

Acknowledgment

This research work was funded by Portuguese National Funds through the FCT - Foundation for Science and Technology, in the context of the research grant BI/DETI/9308/2022 within the project UIDB/00127/2020.

References

[1] Agência para a Modernização Administrativa. *Documentação técnica relativa ao serviço de autenticação do Autenticação.Gov e Chave Móvel Digital*. 2022. URL: https://github.com/amagovpt/doc-AUTENTICACAO (visited on 01/23/2023).

[2] Daniel J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". In: *Public Key Cryptography (PKC 2006, LNCS 3958)*. Ed. by Moti Yung et al. Springer, 2006, pp. 207–228. ISBN: 978-3-540-33852-9. DOI: 10.1007/11745853_14.

[3] Bundesministerium für Finanzen. *ID Austria: Mein Ich-organisiere-das-von-überall-Ausweis*. URL: https://www.oesterreich.gv.at/id-austria.html (visited on 01/23/2023).

[4] W. Denniss and J. Bradley. *OAuth 2.0 for Native Apps*. RFC 8252 (Proposed Standard). Oct. 2017. DOI: 10.17487/RFC8252.

[5] Django Software Foundation. *Django: The web framework for perfectionists with deadlines*. URL: https://www.djangoproject.com/ (visited on 01/23/2023).

[6] Jason A. Donenfeld. *WireGuard: fast, modern, secure VPN tunnel*. URL: https://www.wireguard.com (visited on 01/23/2023).

[7] Jason A. Donenfeld. "WireGuard: Next Generation Kernel Network Tunnel". In: *Network and Distributed System Security Symposium (NDSS'17)*. San Diego, CA, USA, Feb. 2017, pp. 1–12. DOI: 10.14722/ndss.2017.23160.

[8] Google. *Using OAuth 2.0 to Access Google APIs*. URL: https://developers.google.com/identity/protocols/oauth2 (visited on 01/23/2023).

[9] Government of Spain. *Get to know Cl@ve: Electronic Identity for the Administration*. URL: https://clave.gob.es/clave_Home/en/clave.html (visited on 01/23/2023).

[10] D. Hardt (Ed.) *The OAuth 2.0 Authorization Framework*. RFC 6749 (Proposed Standard). Oct. 2012. DOI: 10.17487/RFC6749.

[11] Eva Hladka et al. "Transparent security for collaborative environments". In: *Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*. 2007, pp. 79–84. DOI: 10.1109/COLCOM.2007.4553814.

[12] Petr Holub et al. "Secure and pervasive collaborative platform for medical applications". In: *Studies in Health Technology and Informatics* 126 (2007). PMID: 17476065, pp. 229–238.

[13] John Hughes et al. *Profiles for the OASIS Security Assertion Markup Language (SAML) 2.0*. Mar. 2005. URL: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[14] OASIS (Organization for the Advancement of Structured Information Standards). *Security Assertion Markup Language (SAML) V2.0 Technical Overview*. Committee Draft 02. Mar. 2008. URL: https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf.

[15] Pallets Projects. *Flask web development, one drop at the time*. URL: https://flask.palletsprojects.com/en/2.2.x/ (visited on 01/23/2023).

[16] Avery Pennarun. *How Tailscale works*. Mar. 2020. URL: https://tailscale.com/blog/how-tailscale-works/ (visited on 01/23/2023).

[17] N. Sakimura (Ed.), J. Bradley, and N. Agarwal. *Proof Key for Code Exchange by OAuth Public Clients*. RFC 7636 (Proposed Standard). Sept. 2015. DOI: 10.17487/RFC7636.

[18] Tailscale. *Custom SSO providers using SAML or OIDC*. July 2022. URL: https://tailscale.com/kb/1119/sso-saml-oidc/ (visited on 01/23/2023).

[19] Tailscale. *Pricing*. URL: https://tailscale.com/pricing/ (visited on 01/23/2023).

A Parallel Processing Method of Large-scale System Level Simulator for Advanced 5G System

Megumi Shibuya, Akira Yamaguchi, Takahide Murakami, and Hiroyuki Shinbo
KDDI Research, Inc.
Saitama, Japan
e-mail: {xmu-shibuya, ai-yamaguchi, tk-murakami, hi-shinbo}@kddi.com

Abstract—The advanced fifth generation mobile communication system (5G system) around 2025 is expected to introduce new technologies, such as virtualized Radio Access Network (vRAN) that can place base station functions on general servers, and base station function placement based on vRAN to fit the quality requirements of communication services. In order to evaluate the quality of end-to-end communication in mobile communication system, a System Level Simulator (SLS) is widely used. However, more simulation time for SLS with the advanced 5G system is required than with the 5G system. Because new technologies are added in SLS and it executed long-term and large-scale simulations are required. A reduction of simulation time for SLS is required for an effective evaluation. In this paper, we propose a software design of SLS for the advanced 5G system with RU-basis parallel processing by multiple computation nodes. Through the SLS executed on the supercomputer Fugaku, we confirmed that our proposed method can reduce SLS processing time.

Keywords - System level simulator; advanced 5G system; virtual RAN; parallel processing; MPI

I. Introduction

The fifth generation mobile communication system (5G system) has already become widespread in many countries. Around 2025, the 5G system will further advanced as the "*advanced 5G system*" accompanied by the introduction of new technologies. In order to introduce new technologies, it is necessary to evaluate the end-to-end communication quality of the entire advanced 5G system with the technologies due to clear how affect use-level packet. For example, the new technologies are wireless communication systems, such as massive Multiple Input Multiple Output (MIMO) [1], grant-free non-orthogonal multiple access for Internet of Things (IoT) [2], and virtualized Radio Access Network (vRAN). To evaluate the end-to-end communication quality with these technologies in a mobile communication system, a System Level Simulator (SLS) on computers is widely used.

The existing SLSs for the 5G system [3][4][5] can simulate the User Equipment (UE) layout and movement, Radio Unit (RU) layout, generation of application traffic, and the wireless communication technologies of the 5G system. For evaluations of the advanced 5G system, additional technologies will be simulated, such as a new wireless communication system and vRAN. In addition, the management methods for vRAN to maintain communication quality are also evaluated by SLS, such as base station function placement based on computation resources and transport resources [6], and radio resource assignment for each virtualized base station function [7]. The vRAN controls by the management methods are judged by the frequently changed status such as the status of the radio links between each of the RUs and UEs, and UE traffic generation. In addition, for example, base station function placement in vRAN is controlled by aggregating RAN-wide information, such as time-varying radio quality information and the generated traffic in each UE. Therefore, when base station function placement changes, the communication quality of user-level packets is affected by the placed base station function. Because of this interaction, the RAN portion and the radio portion must be simulated simultaneously. Although abstracted simulations are proposed [5], since it is not yet clear how the new technologies in the advanced 5G system will affect user-level packets, detailed simulations of individual technologies should be performed for the initial stage evaluation.

Figure 1 shows the simulation configuration of the advanced 5G system. There are approximately 1,000 RUs and 50,000 UEs inside approximately one square kilometer. The RUs create many areas of cells with various frequency bands. UEs (e.g., cars, IoT terminals, and smartphones) move in the areas. In addition, to judge the usefulness of the management methods, it is required an SLS simulation time of around 10 minutes to 1 hour period. The simulation will become large-scale about the number of RUs and UEs, and long-term about simulation time.

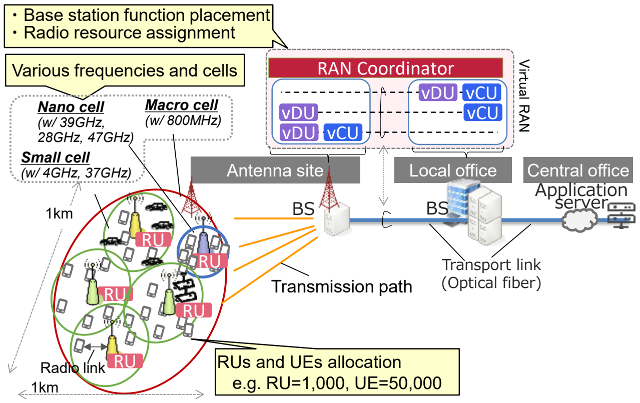


Figure 1. Simulation configuration of advanced 5G system.

Simulations with both new technologies and long-term/large-scale simulation takes a long time to execute the SLS for the advanced 5G system. The long simulation time affects effective evaluations by the SLS because many simulations will be executed to evaluate new wireless communication technologies and management methods. Therefore, we need to reduce the simulation time for the SLS. One method of reducing the simulation time of the SLS is a parallel processing method by one computation node with multi-cores [8]. However, since total computation power is limited, one computation node cannot effectively improve the simulation time. To obtain more computation power, there is a parallel processing method using multiple computation nodes with multi-cores [9]. To apply the SLS to this environment, the simulation result of each radio frame transmission (e.g., 1 msec) needs to be shared between the computation nodes because the next radio frame transmission is simulated based on previous radio frame transmissions. Since such data sharing and collecting between computation nodes is over a network, a "*memory access time*" is required. As shown in Figure 2, applying multiple computation nodes has pros and cons in that it can reduce SLS processing time in each computation node, but the memory access time between the computation nodes is increased. In order to reduce the total simulation time of the SLS, it is important to have balanced software design between the memory access time and the reduction of the SLS processing time.

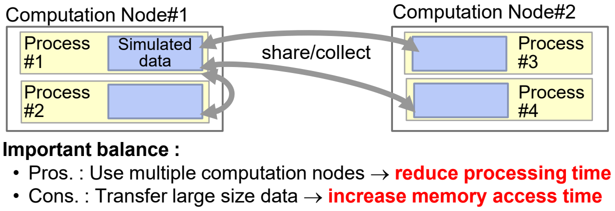

Computation Node#1
Computation Node#2
Process #1
Simulated data
Process #2
share/collect
Process #3
Process #4
Important balance :
• Pros. : Use multiple computation nodes → reduce processing time
• Cons. : Transfer large size data → increase memory access time

Figure 2. Pros and cons of data sharing in multiple computation nodes.

In this paper, in order to reduce the simulation time, we propose a software design of SLS for the advanced 5G system (A5G-SLS) with multiple computation nodes. The A5G-SLS introduces the RU-basis for parallel processing to reduce processing time. To evaluate the effectiveness of our proposed method, we use different types of parallel processing models, such as ALL MPIs and hybrid models, on the supercomputer Fugaku [10]. The rest of this paper is organized as follows. Section II presents the simulation targets and the problem of processing time reduction in the A5G-SLS. Section III proposes a software design for the A5G-SLS with parallel processing by multiple computation nodes. We present the evaluation of the proposed method in Section IV. Finally, we provide our conclusion in Section V.

II. SYSTEM LEVEL SIMULATOR FOR ADVANCED 5G SYSTEM AND PARALLEL PROCESSING PROBLEM

A. Overview of advanced 5G RAN

In the advanced 5G system, the functions of the base stations are divided into a Central Unit (CU), a Distributed Unit (DU) and an RU. In the vRAN environment, the CU and DU are virtualized as the vCU and vDU, and they work on commodity servers. The servers are placed at various locations in the RAN, i.e., an antenna site, a local office, and a central office. An optical fiber connects the local office and the antenna site as a physical link in the physical configuration as shown in Figure 3 (a). The application server is located in the central office (or local office), and various types of service traffic are generated, such as high-definition video streaming, connected vehicles, drone control, and the IoT.

The logical networks are constructed by the vCUs and vDUs, they are works on the commodity servers and the RUs in an area shown in Figure 3 (b). In order to maintain communication quality, a RAN slice is created for each service [6]. For example, one RAN slice is created for high-capacity traffic, and another RAN slice is created for low-latency, and massive connections in the area. The vRAN management method controls base station function placement based on computer resources and transport resources, and radio resource assignment based on the radio link quality between each of the RUs and UEs, and UEs traffic generation. For example, in Figure 3 (b), one RAN slice is reallocated from high-capacity traffic to massive connections, and the assignment of radio resource is controlled. To obtain user packet level results for throughput, delays, and errors in end-to-end communications, both the RAN and radio links are simulated at the same time. In addition, radio resource assignment is performed in relatively short-term control (100 msec to 1 sec), and base station function placement is performed in long-term control (10 minutes to 1 hour). On the other hand, the radio resource scheduling in the vDU for the radio link is performed in ultra-short control (approximately 1 msec).

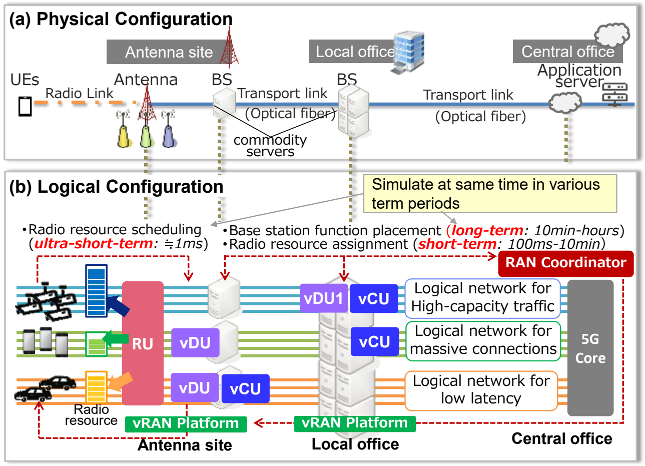

(a) Physical Configuration
Antenna site
Local office
Central office
UEs
Radio Link
Antenna
BS
Transport link
(Optical fiber)
BS
Application server
Transport link
(Optical fiber)
commodity servers
(b) Logical Configuration
Simulate at same time in various term periods
•Radio resource scheduling (ultra-short-term: ≒1ms)
•Base station function placement (long-term: 10min-hours)
•Radio resource assignment (short-term: 100ms-10min)
RAN Coordinator
RU
vDU
vDU
vCU
vDU1
vCU
vCU
Logical network for High-capacity traffic
Logical network for massive connections
Logical network for low latency
5G Core
Radio resource
vRAN Platform
vRAN Platform
Antenna site
Local office
Central office

Figure 3. Overview of 5G advanced RAN

B. A5G-SLS Implementation and Points for Simulation Targets

The A5G-SLS consists of three parts; initialization, preprocessing, and main simulation. As the initialization part, the physical and logical configurations, such as base stations (vCUs, vDUs, RUs), UEs, slices, and traffic servers, are set in the initial files as parameters. In addition, the services and simulation time are set in the same files. The traffic of each

ISBN: 978-1-68558-052-0

service for each UE and the propagation are created using the initial files in the preprocessing part. The main simulation conducts the simulation process, such as propagation at a transport link, the radio packet process includes radio resource scheduling at the vDU, and the radio link process includes error and retransmission in the radio links and the UE. The main simulation part conducts the processing for each time step until the simulation time is finished. We created the A5G-SLS for advanced 5G systems based on the SLS for 5G systems used in the evaluation of previous research [11].

From the above configuration, the A5G-SLS can simulate the following. A large number of RUs and UEs are placed, the RUs with various frequency bands, and UEs, which move in the area, are simulated. The propagations are simulated using a radio propagation model in 1 msec order, and the radio frames are sent/received using propagation data that include error and retransmission. In addition, the networks and transport links in a RAN are simulated, and the data packets are transferred through them. The data packets are transferred as radio signals or user data depending on changes in the base station functions. Furthermore, the UE can generate various services data, such as high-definition video streaming and connected vehicles. From these simulations, it is possible to clear how the generated traffic is affected by radio frames and transmission paths as communication, and the end-to-end communication quality can be simulated.

The simulation targets of the A5G-SLS are to evaluate the quality of end-to-end communication for each user in the large-scale environment of the advanced 5G system. There are two points for simulating the targets;

Point 1 is new technologies. One of the new technologies is the new wireless communication systems, such as MIMO. The other is vRAN and its management method, such as base station function placement and radio resource assignment. As described in section II A, the vRAN controls are judged by frequently changed statuses (e.g., radio links and UE traffic generation). In addition, the communication quality of user-level packets is affected by the changing placement of the base stations. Because of this interaction, the RAN portion and the radio portion must be simulated simultaneously.

Point 2 is large-scale simulation. One large-scale simulation is the environment. As described in Section I, there are approximately 1,000 RUs and 50,000 UEs in approximately one square kilometer. The RUs create many areas with various frequency bands, such as small-cell, nano-cell, and macro-cell. UEs are assumed to be the cars, IoT terminals, and smartphones, and they move in the areas. The other large-scale simulation is long-term simulation. To judge the usefulness of management methods, it is required a simulation time of SLS around 10 minutes to 1 hour period.

The above two points require that the simulations with both new technologies and long-term/large-scale simulation take a long time (e.g., a few days to a week) to execute on the SLS for the advanced 5G system. In addition, considering the new technologies, since the quantitative effects on user-level packets and end-to-end communication quality in the advanced 5G system are not yet clear, detailed simulations of how to work the radio frame transmission and user-level packets should be performed in 1 msec order. One reason is that a detailed simulation of the radio environment, such as each radio link between RUs and UE, is needed to evaluate new wireless communication systems. Since the effectiveness of new wireless communication system for user-level communication, such as communication quality, is unknown, the wireless communication system needs to evaluate the radio environment in detail on the order of milliseconds. The other reason is the evaluation of vRAN management methods and user-level communication quality based on the status of user traffic, usage of radio/transport/computation resources in RAN, and the status of radio links between the RUs and UEs. To evaluate the management methods and the communication quality, all items such as vRAN, user traffic generation, and radio links are required to simulate at the same time. In addition, long-term simulation is required because the management method controls vRAN per 1 or 10 minutes.

C. Related SLS (Existing System Level Simulators)

As the SLS for the 5G system, some simulation tools and open platforms are proposed in [3][4][12][13], and [14]. OMNet++ [3], NS-3 [4], and Veins [12] exist as popular event-based network simulators. In these simulators, most of the protocol stacks are modeled. However, in order to address computational complexity, it is difficult to simulate large-scale networks. 5G K-SimSys [13] has been developed to provide an open platform for evaluating SLS performance of the 5G standard. It is designed to be flexible, open, and has a modular form to make it easy to customize. To evaluate performance, a more complex testbed is required. OpenAirInterface [14] is implemented in part of the 3GPP LTE and provides an interface between the hardware platform and works as an emulator. When the complexity increases, it is difficult to conduct a large-scale simulation due to the number of nodes, which is limited.

The approaches of reducing the simulation time for the SLS have been conducted [15][16][8], and [5]. 5G-Lena [15], which is NS-3 based, introduces a method of reducing simulation time by abstracting the physical layer. D. S. Buse et al. [16] introduced an approach in which some part of the SLS process of the wireless signal attenuation model computation is asynchronously pushed to the background and offloaded. This approach is implemented in the Veins of VANET simulator. However, when the simulation scale becomes large and complex, the simulation takes a long time to compute due to the single-thread simulation run. Therefore, it has the limitation of reducing the simulation time for the SLS. As other approaches, Vienna5G [8] has been proposed. This simulator is based on MATLAB, and it can perform in a large-scale, multi-tier network with numerous types of network nodes. This approach uses multiple threads on a computation node. However, it is insufficient to reduce simulation time by only multi-threads processing in one computation node to increasing amount of calculation. Simu5G [5] is provided as an open-source simulator with an emulator function based on OMNeT++. In order to provide real-time emulation, it is introduced lightweight models of UEs and gNBs with abstracted limited functionalities for creating resource contention and interference.

Comparing the existing 5G SLS and the A5G-SLS, the existing 5G SLS in [16] presents the simulation result using a scenario, such as 30 RUs, 7,000 UEs, and 100 msec of simulation time. On the other hand, in the A5G-SLS, the simulation is performed using a scenario, such as approximately 1,000 RUs, 50,000 UEs, and one hour of simulation time. Hence, large-scale and long-term simulations are required. Moreover, in terms of the implementation of functions, the existing 5G SLS consists of the following functions: 1) services and the transmission paths process, 2) DU process, 3) RU process, and 5) other processes as shown in Table I. On the other hand, the A5G-SLS includes the function 4) the vRAN process in addition to the functions of the 5G SLS. These processes from 1) to 5) are performed repeatedly in the order of each time step (e.g., every 1 msec) until the end of the simulation. Namely, the A5G-SLS increases the simulation functions. Hence, to be adaptive of increasing calculation, it requires the parallel processing method using multiple computation nodes.

TABLE I. IMPLEMENTATION OF THE FUNCTIONS FOR EXISTING 5G SLS AND A5G-SLS

Function	Process	Existing 5G SLS	A5G-SLS
1) Services, transmission paths	• Traffic generation • Packet process (application server to vCU) , transmission paths	✓	✓
2) DU	• Wireless scheduler • Packet process (vDU to RU)	✓	✓
3) RU	• Packet process (RU to UE) • Radio communication quality measurement (calculate SINR) • Calculate receive power	✓	✓
4) vRAN	• Radio communication quality (calculate the RU changing indicator) • RAN coordinator (RUs and resources allocation)	-	✓
5) Others	• Radio communication (hand-over)	✓	✓

D. Problem with Parallel Processing

To realize parallel processing using multiple computation nodes, there are the following problems.

Problem-a) Transmission time of memory data: To use multiple computation nodes, in order to share the data between all processes, memory data are collected and shared between multiple computation nodes. In general, as shown in Figure 4, in existing scientific simulations, such as fluid dynamics [17] and weather forecasting [18], each process is an independent event, only the data used in each process is transferred, and the calculated result data is collected to approximately 320 Kbytes [18]. On the other hand, in the A5G-SLS, the radio allocation data of all UEs and RUs are required for calculating SINR. Therefore, the A5G-SLS transfers all simulated data in memory to all processes (shared data), and transfers all RUs and UEs information where changes, such as SINR, have occurred (collected data) (e.g., when the number of RUs is 598 and the number of UEs is 35 per RU, transmission data size, which includes shared data and collected data, is approximately 5 MBytes). Hence, the data transfer time becomes longer.

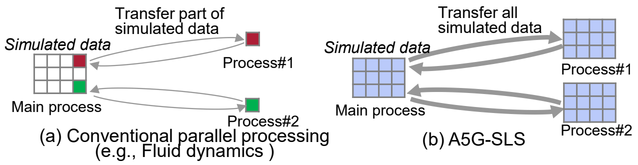

Transfer part of simulated data
Simulated data
Process#1
Main process
Process#2
(a) Conventional parallel processing (e.g., Fluid dynamics)
Transfer all simulated data
Simulated data
Process#1
Main process
Process#2
(b) A5G-SLS

Figure 4. Transmission data size for conventional parallel processing and A5G-SLS.

Problem-b) Process waiting time: Conventionally, the parallel processing method has the problem that it cannot proceed to the next process until all processes are finished because of the synchronization of all processes and the sharing of all simulation results between the computation nodes. In other words, the processes should wait until all processes reach this barrier of synchronization as shown in Figure 5. In the A5G-SLS, the simulation result of each radio frame transmission needs to be shared between computation nodes. However, increasing the waiting time increases the processing time for the simulation. In order to solve this problem, the A5G-SLS is required to reduce the waiting time. For this reason, it is necessary to design all parallel processes to have similar processing times as much as possible.

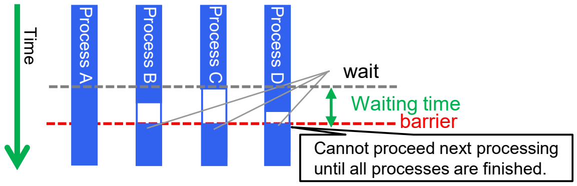
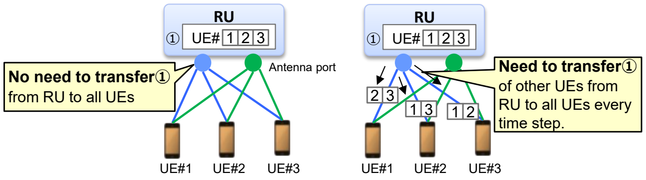

Time
Process A
Process B
Process C
Process D
wait
Waiting time
barrier
Cannot proceed next processing until all processes are finished.

Figure 5. Process waiting time.

Problem-c) Dividing basis for parallel processing: As the dividing basis, two types exist: RU unit (hereinafter "RU-basis") and the UE unit (hereinafter "UE-basis") as shown in Figure 6. The UE-basis processing needs to transfer the radio resource assignment with the channel status information (CSI) data of other UEs from the RU to all UEs for every time step. Therefore, the transference data size and times are increased. On the other hand, the RU-basis does not need to transfer the radio resource assignment with CSI data from the RU to all UEs because the RU already has them. Hence, if the dividing unit is inadequate, it takes a long processing time due to Problem-a) and Problem-b).


RU
① UE# 1 2 3
No need to transfer① from RU to all UEs
Antenna port
UE#1
UE#2
UE#3
RU
① UE# 1 2 3
2 3
1 3
1 2
Need to transfer① of other UEs from RU to all UEs every time step.
UE#1
UE#2
UE#3

・① : Radio resource assignment with Channel Status Information (CSI) data per UE

(a) RU-basis processing (b) UE-basis processing

Figure 6. Distribution of the radio resource assignment with CSI for each divided method.

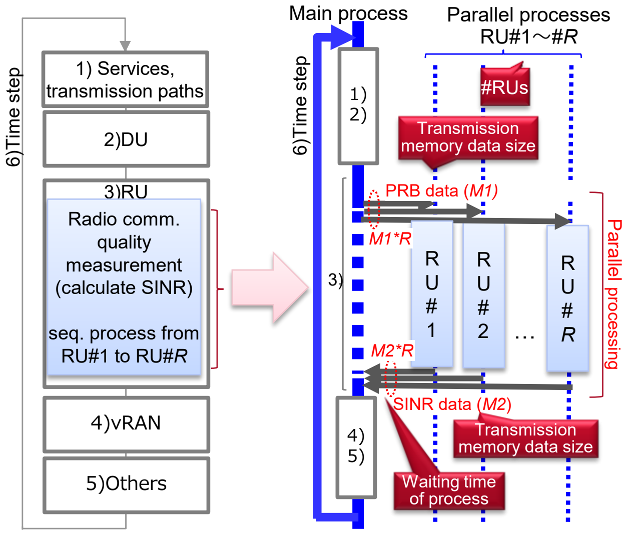
To address these problems, it requires reducing the simulation time by the parallel processing method using multiple computation nodes.

III. Proposal for A5G-SLS Design Method

In this section, we explain the parallel processing method for reducing simulation time by using multiple computation nodes to resolve Problem-a), Problem-b) and Problem-c) as described in the previous section.

For the design of the parallel processing target, the processing with a high computation load in the A5G-SLS is selected due to the effect of the processing time reduction. We analyzed the program execution time in the A5G-SLS from the following two viewpoints; the availability of parallel processing, and the process of the high computation load within A5G-SLS. As a result, the computation load imposed by the radio communication quality measurement (calculation of the Signal to Interference and Noise Ratio (SINR)) in RU processing is high (approximately 30% of the total) so this part is parallelized in multiple processing.

In addition, as a selection basis method for parallel processing, the UE-basis and RU-basis exist as shown in Figure 6. In order to calculate the SINR, the radio resource assignment data with the CSI of all UEs are required. In the case of the UE-basis, the RU transfers the radio resource assignment data with the CSI to all UEs at every time step in order to calculate the SINR at each UE. On the other hand, in the case of the RU-basis, it is not necessary to transfer the radio resource assignment data with the channel status information from each UE because the RU already has the data. By using these selection methods, the transmission volume is reduced, and the transmission time is reduced (Problem-a can be resolved). In addition, since there are no major differences in the process load of each RU, the processing waiting time may be reduced. Hence, we propose a method of reducing simulation time using the RU-basis (Problem-b and Problem-c can be resolved).


6)Time step
1) Services, transmission paths
2)DU
3)RU
Radio comm. quality measurement (calculate SINR)
seq. process from RU#1 to RU#R
4)vRAN
5)Others
Main process
Parallel processes RU#1～#R
6)Time step
1)
2)
#RUs
Transmission memory data size
PRB data (M1)
M1*R
3)
RU#1
RU#2
…
RU#R
Parallel processing
M2*R
SINR data (M2)
Transmission memory data size
4)
5)
Waiting time of process

(a) Non-parallel processing (b) Parallel processing of RU-basis

Figure 7. Sequence of parallel processing for A5G-SLS.

Figure 7 shows the specific processing sequence for the A5G-SLS. Non-parallel processing is shown in Figure 7 (a), and the parallel sequence of the RU-basis is shown in Figure 7 (b). We explain the sequence using the simulation scenario when the number of RUs is denoted as *R*. In the case of non-parallel processing, the SINR is calculated sequentially from RU#1 to RU#*R* in the measurement process of the radio communication quality in RU processing. On the other hand, in the proposed parallel processing method for the RU-basis, SINR calculation processing is divided into the *R* of each RU, then the *R* processes are conducted in parallel. Since introducing parallel processing, data transfer between the main process and each RU process is necessary due to sharing and collecting the data; when each process starts, the Physical Resource Blocks (PRBs) of data *M1* [Byte], which are allocated to all UEs by all RUs, are transferred from the main process to each RU process, and when each RU process ends, the SINR data of each UE *M2* [Byte] are transferred from each RU process to the main process. Hence, the total transmission memory data sizes from and to the main process of PRB and SINR are $M1 \times R$ [Byte], $M2 \times R$ [Byte], respectively.

IV. Evaluation

In this section, we verify our proposed method for the A5G-SLS to reduce the processing time on the RU-basis using parallel processing by multiple computation nodes.

A. Evaluation Viewpoints

We evaluate our proposed method from two viewpoints.

Viewpoint 1 is the reduction in processing time. The processing time is compared with some parallel models and a non-parallel model (the details are provided in Section IV.B.) varying the number of RUs and UEs.

Viewpoint 2 is the effect of Problem-a) and Problem-b). For Problem-a), it shows the balance between the memory access time and processing time. For Problem-b), it shows the relationship between the process waiting time and the increase in the number of RUs. From this relationship, we explain that the waiting time is reduced, and the calculation time is improved as a result.

B. Evaluation Method

As mentioned in subsection A, in order to evaluate the proposed method, four types of evaluation models, including three types of parallel model (Model-2, Model-3 and Model-4) and one non-parallel model (Model-1), which is used for comparing with the parallel processing models, are defined (shown in Figure 8). In the parallel models, considering the different configurations of execution types, such as process and/or thread, and interface types of transference memory data, such as Message Processing Interface (*MPI*) [19] and/or Open Multi-Processing (*Open*MP) [20], we set the following models:

- **Model-1) ALL threads :** Using multiple threads in one computation node, as "*non-parallel processing model*".

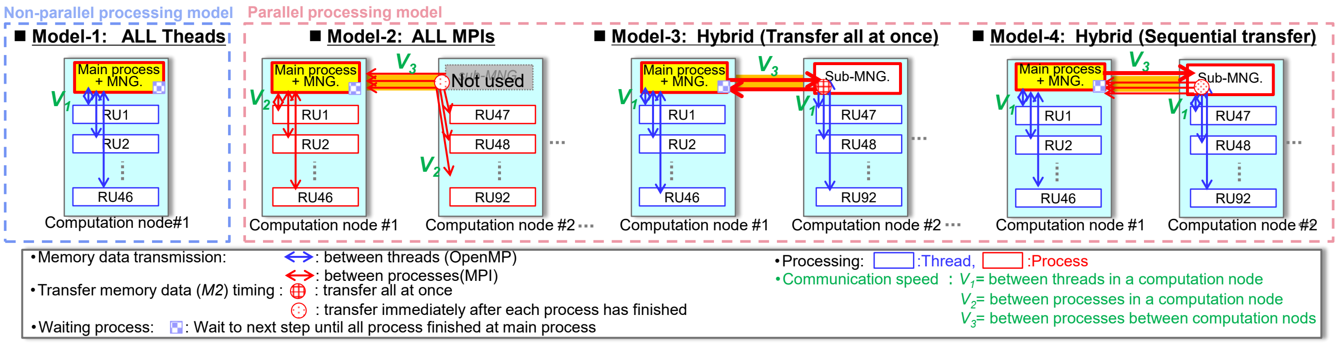

Non-parallel processing model
Parallel processing model
■ Model-1: ALL Theads
■ Model-2: ALL MPIs
■ Model-3: Hybrid (Transfer all at once)
■ Model-4: Hybrid (Sequential transfer)
Main process + MNG.
Sub-MNG.
Not used
V_1
V_2
V_3
RU1
RU2
RU46
RU47
RU48
RU92
Computation node#1
Computation node #1
Computation node #2
•Memory data transmission: ↔: between threads (OpenMP)
↔: between processes(MPI)
•Transfer memory data (M2) timing : ⊕ : transfer all at once
◌ : transfer immediately after each process has finished
•Waiting process: ▨: Wait to next step until all process finished at main process
•Processing: □:Thread, □:Process
•Communication speed : V_1= between threads in a computation node
V_2= between processes in a computation node
V_3= between processes between computation nods

Figure 8. Parallel Processing Models (Model-1: Single computation node, Model-2, Model-3, Model-4: Multiple computation nodes).

- **Model-2) ALL-MPI :** All processes in the multiple computation nodes are connected to the management process in node #1 by MPI. Data *M2* are transferred separately to the main process when the SINR calculation has finished at each thread.
- **Model-3) Hybrid (Transfer all at once) :** Multiple threads are connected to the management process within the computation node, and each management process is connected to the management process of computation node #1 by MPI. Data *M1* and *M2* are transferred through the sub-management process together with all data of each node by one MPI. Data *M2* are transferred to the main process through the sub-management process all at once when the SINR calculation of all threads in a computation node has finished.
- **Model-4) Hybrid (Transfer immediately) :** The configuration is the same as Model-2. Data *M1* are transferred through the sub-management process together with all data of each node by one MPI. Data *M2* are transferred separately to the main process through the sub-management process when the SINR calculation has finished at each thread.

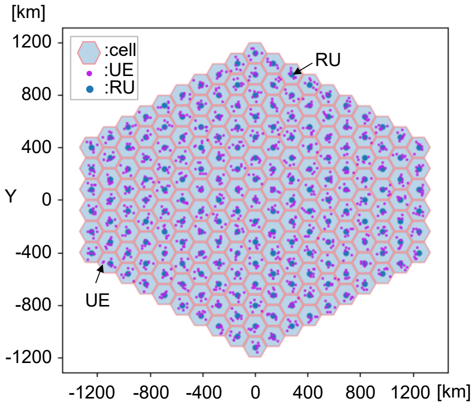


Figure 9. Example of RUs and UEs allocation in the scenario.

The evaluation scenario is that multiple RUs exist and a huge number of UEs are moving in many directions in a nano-area in the scenario (see Figure 9). We measure the processing time and waiting time when the number of RUs *R* varies from 46 to 1,012, and the number of UEs *U* varies from 690 to 55,660. We set the simulation time to 60 [min]. The other simulation parameters are shown in Table II. To obtain evaluation values, the A5G-SLS executes five times on Fugaku as shown in Table III.

TABLE II. SERVICE OPERATING AND MANAGEMENT DATA

Parameter	Value
# of RUs (*R*)	46 - 1,012 (46 RU intervals)
# of UEs (*U*)	15, 35, 55 (per RU) (Total UEs: 690 – 55,660)
# of cores	46 cores / node
#of PRBs	273
Simulation time	10 [min]
# of Time steps (*T*)	60,000
Communication speed (*Vi*)	V_1=8,192.0, V_2=159.6, V_3=50.1 [Gbps]

TABLE III. NODE SPECIFICATION ON FUGAKU@RIKEN [21]

Hardware	
Parameter	***Value***
CPU, # of Core	A64FX, 48 Cores/Node
Available # of nodes	Max 384
Node IF specification	Tofu Interconnected D (28 Gbps x 2 lane x 10 port)
Software	
Parameter	***Value***
OS	Red Hat Enterprise Linux 8
Compiler	C++ 17
MPI	FUJITSU MPI Library 4.0 (based on Open MPI)
OpenMP	OpenMP 4.5

The memory access time of the whole simulation *C* [sec] is calculated by equation (1):

$$C[sec] = MT \sum_{i=1}^{3} W_i/V_i \qquad (1)$$

where, $i = (1,2,3)$ denotes the transference points; $i = 1$ is between the management process and each thread in a computation node (Model-1, Model-3 and Model-4), $i = 2$ is between the management process and process in a computation node (Model-2), and i = 3 is between the management process and the sub-management process among two computation nodes (Model-2, Model-3 and Model-4), respectively. W_i is the number of transfer times per time step of i. V_i is the communication speed of point i [bps], each communication speed is V_1 and uses 8,192 [Gbps] referred from[17], V_2 and V_3 use the measurement values 159.6 [Gbps], and 50.1 [Gbps], respectively. Furthermore, T is the number of time steps, M is the transmission memory data size per RU [Byte] calculated by $M = M1 + M2$. From our simulation, M, $M1$, and $M2$ are the following values in Figure 10. $M1$ is the fixed size for every simulation step because it is calculated by the number of RUs and UEs. $M2$ is a different size for each simulation step, however, $M2$ size is in proportion to the number of RUs and EUs. The maximum transmission memory data size of M is approximately 13.9 [MBytes] per RU, when R is 1,012 and U is 55.

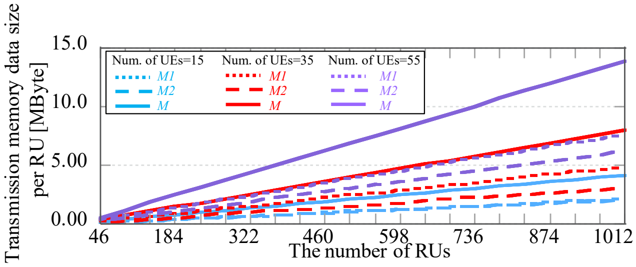


Figure 10. Transmission data size $M1$, $M2$ and, M of each RU.

C. Simulation Results

1) Improvement of Processing Time

First, we verify that our proposed method improves the processing time compared to ALL Threads (Model-1). Figure 11 shows the processing time varying R with a comparison between the parallel processing using multiple computation nodes and the non-parallel model using one computation node. The graph is normalized to the maximum processing time. From Figure 11, the parallel processing model can reduce the processing time more than the non-parallel processing model. Although the parallel processing model and the non-parallel processing model are almost same when R is small, the difference in the processing time is larger when R increases. Specifically, in the case of R=1021 and U=35 and 15, the processing time of Model-2 can be reduced by 1.8% and 12.5% compared to Model-1, respectively. The greatest reduction at U=35 is 11.3% when R=522. In addition, the processing time of Model-3 and Model-4 is reduced compared to Model-1. When U=15, it is obtained the greatest reduction, Model-3 and Model-4 reduce the processing time by 16.16% and 17.51%, respectively. Furthermore, Model-4 achieves more reduction compared to Model-3.

However, in cases where U=55 per RU (total 45,540 UEs), when R exceeded 828, Model-2 has a longer processing time than Model-1. This reason is estimated that the data size is larger (over 10MBytes per RU process), and transmission time is increased.

From this result, we confirm that the proposed method can reduce the processing time, unless in the case of Model-2 and the transmission data size becomes extremely large (e.g., over 10 MB per RU).

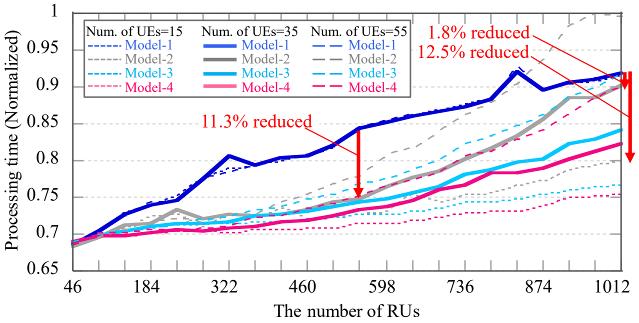


Figure 11. Improvement of SLS processing time.

2) Effectiveness of Parallel Processing

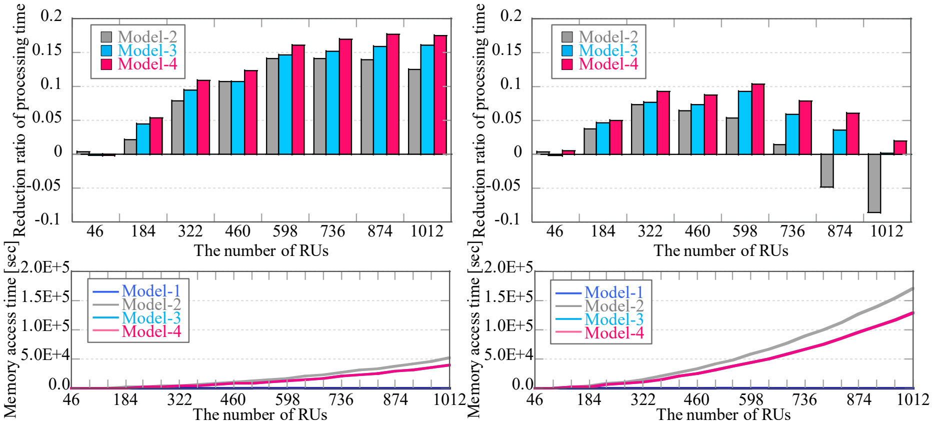
a) Memory Data Sharing

Next, the effect of processing time reduction and the balance between the memory access time and the processing time is confirmed. The upper graph in Figure 12 shows the reduction ratio of the processing time compared to Model-1, and Figure 12 below shows the memory access time. Both graphs vary R in which U=15 and 55, respectively. The reduction ratio is defined as the ratio of the difference between Model-1 and each model at each R. In the case of U=15, the reductions of processing time are obtained from Model-1. The maximum reduction ratio of the processing time is for Model-2, Model-3, and Model-4 when R=1,012, and they are 0.13, 0.16 and 0.18, respectively. On the other hand, in the case of U=55, the reduction ratio is improved compared to Model-1, and the ratio is less than when U=15. The maximum reduction ratio of the processing time is for Model-2, Model-3, and Model-4 when R=598, and they are 0.05, 0.09 and 0.10, respectively. However, these ratios are reduced after R=598, R exceeds 874 in Model-2, and the reduction ratio is less then Model-1.

The memory access time when U=15 and 55 is increased as R is increased, especially when U=55, and it is increased rapidly. When R=1,012 and U=55, the memory access time takes 3.3 times longer than when U=15. Hence, the reduction ratio is decreased due to the memory access time being increased. From this result, it can be seen that it is important to balance the processing time and memory access time.

b) Process Waiting Time

We evaluate the process waiting time of each model. Figure 13 shows the variance in waiting time varying R in the four models. The variance of waiting time for Model-2, Model-3, and Model-4 is smaller than that for Model-1. However, the variances of these models are almost constant despite the increase in the number of RUs when RU is less than or equal to 460. This result indicates that because the RU-basis divided method keeps the waiting time constant, the processing time can be improved. From another viewpoint,



(a) The number of UEs (U)=15 (b) The number of UEs (U)=55

Figure 12. Improvement of reduction ratio of processing time comparing the non-parallel processing model (Upper figure) and memory access times (Lower figure) of each UE.

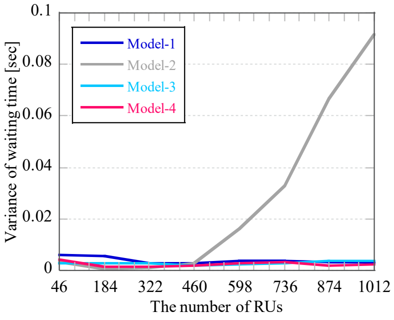


Figure 13. Process waiting time (The number of UEs (U) = 55).

Model-2 of R exceeds 460, and the variance of the waiting time is increased. We assume the reason for this is that the transmission data size when R is over 460 is large, such as 6MBytes, and the communication speed of Model-2 is slower than the other models.

From Figure 12 and Figure 13, it confirms that when the waiting time is long, the reduction of the processing time cannot be increased.

These results indicate that the hybrid Model-4 (transfer immediately) is the best method due to the balance among the reduction in processing time, memory access time and waiting time.

D. Discussion

In the above evaluation, we have verified the reduction of the processing time of A5G-SLS using three parallel processing models, Model-2, Model-3 and Model-4. Next, we evaluate quantitatively the best parallel model to execute A5G-SLS.

In order to conduct a quantitative evaluation, an evaluation score is introduced. The evaluation score E is calculated by (2) using three indications of processing time, memory access time and waiting time.

$$E = \alpha + \beta + \gamma \tag{2}$$

where, α is the processing time, β is the memory access time and γ is the waiting time, and all indications are normalized by the maximum value of each indication. Therefore, E is from 0.00 to 3.00 due to using the three indications. The smallest E indicates the best model. The evaluation score E is calculated for each model and each RU.

Figure 14 shows the evaluation score E of each model varying the number of RUs R. From this graph, when R is small, all models obtain almost the same score for E, which is less than 1. However, when R is large, the score for E is increased. Specifically, when R=1,012 and U=55, the evaluation scores E of Model-2, Model-3 and Model-4 are 3.00, 1.72 and 1.68, respectively. In comparing two hybrid models, E of Model-4 is 0.04 smaller than that of Model-3.

From the above results, in order to reduce the processing time, especially for large-scale simulation, it is desirable to conduct the simulation using hybrid Model-4 (transfer immediately).

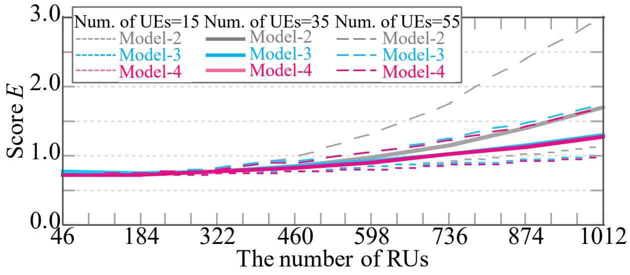


Figure 14. Score of each model.

V. Conclusion

Considering the advanced 5G system of approximately 2025, in this paper to reduce the simulation time, we propose a software design of SLS for the A5G-SLS with multiple computation nodes. Specifically, in the A5G-SLS, since the computation load of the radio communication quality measurement is high, in which the SINR calculation in the RU processing, the processing is divided into the RU-basis, and parallel processing is performed by multiple computation nodes. From our simulation results, we confirm that our proposed parallel processing method can improve the processing time compared to non-parallel processing models. However, when the number of RUs is large, the reduction ratio of the ALL_MPI model is less than that of the non-parallel processing model. In addition, as a result of the quantitative evaluation of four processing models, we verified that the hybrid Model-4 (Transfer immediately) is the best for large-scale simulations.

As a future work, in order to further reduction of the processing time, we study the reduction method of the transmission data size and the memory usage in the A5G-SLS. Furthermore, as another approach, a method is needed to reduce the processing time by abstracting the process while not affecting evaluations of the new technologies. The A5G-SLS is a closed software due to introducing our invention. Initially, it will be used for the evaluation of the advanced 5G technologies, and then we will publish our obtained research results.

ACKNOWLEDGMENT

This research is supported by the Ministry of Internal Affairs and Communications in Japan (JPJ000254). This work used computational resources of supercomputer Fugaku provided by the RIKEN Center for Computational Science through the HPCI System Research Project (Project ID: hp220145).

REFERENCES

[1] T. Murakami et al., “Research Project to Realize Various High-reliability Communications in Advanced 5G Network,” 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020, pp. 1-8.

[2] T. Hara, H. Iimori, and K. Ishibashi, “Grant-Free NOMA Using Time-Delay Domain for Low-Latency Massive Access over MIMO-OFDM,” IEEE International Conference on Communications (ICC 2022), May 2022.

[3] OMNeT++, “OMNeT++,” Simulation Models and Tools. [online] Available from: https://omnetpp.org/download/models-and-tools 2023.01.30.

[4] Network Simulator 3 (ns-3). [online] Available from: https://www.nsnam.org/ 2023.01.30.

[5] G. Nardini, G. Stea, and A. Virdis, “Scalable Real-time Emulation of 5G Networks with Simu5G”, IEEE Access, Vol. 9, pp. 148504-148520, 2021.

[6] Y. Tsukamoto, H. Hirayama, S. I. Moon, and H. Shinbo, “Adaptive Function Placement with Distributed Deep Reinforcement Learning in RAN Slicing,” 2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring), June 2022.

[7] H. Hirayama, Y. Tsukamoto, and H. Shinbo, “Feedback Control for QoS-Aware Radio Resource Allocation in Adaptive RAN,” IEEE Access, Vol. 10, pp. 21563-21573, Feb. 2022.

[8] M. K. Muller et al., “Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator,” EURASIP Journal on Wireless Communications and Networking, 227, pp.1-17, Sep. 2018.

[9] W. Tan, P. Lin, B. Liang, and H. Deng, “Influence of Network Bandwidth on Parallel Computing Performance with Intra-node and Inter-node Communication,” 2009 Second International Conference on Intelligent Networks and Intelligent Systems, pp. 534-537, Dec. 2009.

[10] Fugaku. [online] Available from: https://www.r-ccs.riken.jp/fugaku/system/ 2023.01.30.

[11] T. Ohseki, and Y. Suegara, “Fast outer-loop link adaptation scheme realizing low-latency transmission in LTE-Advanced and future wireless networks,” 2016 IEEE Radio and Wireless Symposium (RWS), pp. 1-3, Jan. 2016.

[12] M. Gutlein, R. German, and A. Djanatliev, “Performance Gains in V2X Experiments Using Distributed Simulation in the Veins Framework,” 2019 IEEE/ACM International Symposium on Distributed Simulation and Real Time Application (DS-RT), Oct. 2019.

[13] J. Lee, M. Han, M. Rim, and C. G. Kang, “5G K-SimSys for Open/Modular/Flexible System-Level Simulation: Overview and its Application to Evaluation of 5G Massive MIMO,” IEEE Access Vol. 9, pp. 94017-04032, Jun. 2021.

[14] N. Nikaein et al., “OpenAirInterface: A flexible platform for 5G research,” ACM SIGCOMM Comput. Commun. Rev. 44(5), pp.33-38, 2014.

[15] S. Lagen et al., “New Radio Physical Layer Abstraction for System-Level Simulations of 5G Networks,” 2020 IEEE International Conference on Communications (ICC), Jun. 2020.

[16] D. S. Buse, G. Echterling, and F. Dressler, “Accelerating the Simulation of Wireless Communication Protocols using Asynchronous,” Proc. MSWiM ’21, ACM, pp. 55-67, Nov. 2018.

[17] X. Guo et al., “Improving performance for simulating complex fluids on massively parallel computers by component loop-unrolling and communication hiding,” 2020 IEEE 22nd International Conference on High Performance Computing and Communications, pp.130-137, Dec. 2020.

[18] T. Saito et al., “Consideration of Data Transfer between Jobs,” IPSJ SIG Technical Report Vol. 2014-HPC-143 No.2, pp.1-6, Mar. 2014, (in Japanese).

[19] MPI Forum. [online] Available from: https://www.mpi-forum.org/ 2023.01.30.

[20] OpenMP, “Openmp application program interface version 4.6,” The OpenMP Forum Tech. Rep, 2008.

[21] Y. Nakamura, “Basic Performance of Fujitsu MPI on Fugaku,” The 7th meeting for application code tuning on A64FX computer systems, Jan. 2022.

ISBN: 978-1-68558-052-0

Security Analysis on Social Media Networks via STRIDE Model

Kamal Raj Sharma, Wei-Yang Chiu and Weizhi Meng
SPTAGE Lab, Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Denmark
Email: {weich, weme}@dtu.dk

***Abstract*—Security associated threats are often increased for online social media during a pandemic, such as COVID-19, along with changes in a work environment. For example, employees in many companies and organizations have started to work from home due to the COVID-19 pandemic. Such working style has increased many remote activities and further relied on email for communication, thus creating an ideal condition for email fraud schemes. Motivated by this observation, the main purpose of this work is to evaluate the privacy policy of online social media and identify potential security associated problems. First, we perform a risk analysis of online social media networks such as Facebook, Twitter and LinkedIn by using the STRIDE model. This aims to find threats and vulnerabilities in the online social media. Then in this analysis, the phishing attack was found to be a main threat in online social media, which is a social engineering attack, where users are convinced through some fake messages or emails to extract their personal credentials.**

***Index Terms*—Network Security, STRIDE Model, Social Media Network, Security Analysis, COVID-19 Pandemic**

I. Introduction

Social media is an Internet-based form of communication. Millions of people around the world are using social media to share information and communicate with each other [13]. By using social media, people get to have conversations, share information and create web content personally, professionally or at a company level. There are many forms of social media popularly used currently including blogs, micro-blogs, wikis, social networking sites, photo-sharing sites, instant messaging, video-sharing sites, Meta-virtual worlds, Facebook, Twitter, LinkedIn, Viber, WhatsApp and more [14]. Social networking media, especially in recent years, has been used in different application domains, such as Government, Business, Dating, Education, Finance, medical and health, and social and political application. According to the Statista (German online portal for statistic), 2958, 2000, 2000, 556 million users are active users in Facebook, Instagram, WhatsApp and Twitter, respectively, in January 2023 [1] and 734.7 million users are active in LinkedIn by the year of 2022 [2].

Social media can be divided into two categories: Web-based social network application and Mobile-based social network application.

- Web-based Social Network Applications:
  - Facebook: This is a popular social networking site, alowing people to connect with network of friends, business houses and organizations. Users can log in using both a browser or a mobile application.
  - LinkedIn: This is a business related social media platform mainly used for professional networking. It is an ideal site to post personal updates, job postings, academic programs, events and projects. Users can log in using web browser.
  - Twitter: This micro-blogging site allows users to post updates. Business houses and individuals expecting to engage with their followers at a high frequency rate should consider using Twitter. Users can log in using both a browser and a mobile application.
- Mobile-based Social Network Applications.
  - Viber: Free and secure calls and messages to anyone, anywhere. Used in mobile application.
  - WhatsApp: Free and secure call and messages. Available in smartphones, or a web browser.
  - Telegram: A famous cloud-based instant messaging application with completely free services.

Motivation. However, using too many social networking sites for conveying messages could dilute the entire social media strategy resulting in the ineffectiveness of entire planning and effort. So it becomes obvious that users have to be aware about which social media sites fit into their requirements and communication strategy. For example, it is better to choose social media sites that can be relevant to individual users. It is also easy to connect with others in social media by making new friends, creating new jobs or sharing new information whether it is for business or personal reasons [20]. However, there is a high risk of leaking private information and misusing the personal information because the bad actors can utilize those information for their own gain. Below are some motive examples about the risk:

- **Post information and update status:** Sensitive information may be revealed. It allows users to update status anyone, anywhere at any time.
- **Friends' Requests:** Carelessness in accepting friends' requests may result in adding 'enemies' instead of 'friends' who have more access to users' information.
- **Upload photos and videos:** It allows everyone to view photos and videos that are sensitive to either a user or an organization.
- **Third party applications and links to external sites:** While operating the applications or clicking on the links, malware may infect users' computing platforms.

Contributions. Though there are many studied investigating the risk of social media networks, to our knowledge, STRIDE model has not been used to analyze most social media applications. Also, due to the spread of COVID-19, there might be a change in the security landscape. Motivated by these, our work aims to bring in light how threats are increasing in the use of social media and most importantly how the attackers make most use of pandemic like the COVID-19 to spread the malicious contents through phishing attacks. In order to reach the aforementioned conclusion, a risk analysis on social media was performed, which results in phishing attack being the main threat in online social media that is in increment with the ever growing use of social media.

The remaining parts are structured as follows. Section II introduces related research on risk/security analysis on social media networks. In Section III, we explain our security analysis outcomes based on Facebook, Twitter and LinkedIn via STRIDE Model. Section IV discusses the form of phishing attack under COVID-19 situation and provides relevant countermeasures. Section V concludes our work.

II. Related Work

Risk in online social networks (OSNs) has received much attention from around 2010/2011. For instance, Tang *et al.* [23] introduced an early work that identified the privacy risks due to the lack of symmetric configurations in most of the OSNs, and designed a inference attack that can be used to infer users' private information, even users already made their friend list private. Creese *et al.* [10] figured out one key question about unchecked publishing and sharing of content and information in OSNs, and introduced a model to understand the potential risks faced should all of existing tools and methods be accessible to a malicious entity. The model enables easy and direct capture of the data extraction methods through the encoding of a data-reachability matrix.

Yang *et al.* [22] then figured out that users usually group their friends into social circles but the circles are not formed with privacy policies. They introduced a utility-based trade-off framework that models users' concerns and incentives of sharing, and made a trade-off between these two. Chan and Saqib [24] showed that online social circles such as 'Facebook friends' are akin to collectivistic communities by offering users a 'cushion' that mitigates financial loss, which increases users' financial risk-taking, consistent with the cushion hypothesis. Laleh *et al.* [9] introduced a risk measure, called *local risk factor*, with the key idea that the malicious users in OSNs may show some common features on the topology of their social graphs, which are different from those of legitimate users.

Aktypi *et al.* [5] examined the potential exposure of users' identity that is caused by information that they share online and personal data that are stored by their trackers. They then developed a tool to model online information shared by individuals and elaborated on how they might be exposed to the unwanted leakage. van Schaik *et al.* [21] focused on the security- and privacy settings of Facebook, and found there is a need for non-aggregated analysis and practical implications emphasise interventions to promote safe online social-network use. Han *et al.* [16] found that OSN users try to hide some information for privacy, but the hidden information is likely to be predicted by various powerful inference attacks. Then they proposed a general Framework for Private Attribute Disclosure estimation (F-PAD), which can estimate the disclosure risk for individuals in terms of disclosure probability and risk level.

TABLE I
Threats used in STRIDE Model.

Threat	Properties
Spoofing	Authenticity
Tampering	Integrity
Repudiation	Accountability/non-repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability

Chen *et al.* [7] focused on inference attack defence, and formulated the social network data sharing problem through an optimization-based approach. Then they proposed two privacy-preserving social network data-sharing methods to counter the inference attack. One is called the efficiency-based privacy-preserving disclosure algorithm (EPPD), and the other is to convert the original problem into a multi-dimensional knapsack problem (d-KP) using greedy heuristics. Fu and Yao [20] introduced an effective and reasonable privacy risk scoring method. It takes into account the granularity of the shared profile items, combines sensitivity and visibility, and generates a privacy risk score for each user.

There are many previous studies on this topic, but to the best of our knowledge, the STRIDE model [11] has not been widely used for risk analysis on OSNs. This indeed motivates our work, especially under COVID-19 situation, there could be some new attack vectors.

III. Security Analysis with STRIDE Model

The rapid increase of online social media may also bring new types of threats that spill over from the Internet world to everyday life [7]. For example, it has become very easy for an intruder to exploit social media for malicious purposes, but organizations and governments find it difficult to accurately detect, identify, predict and prevent the malicious exploitation of social media. In this section, we aim to perform a STRIDE model-based risk analysis on popular social media networks such as Facebook, Twitter, and LinkedIn.

A. STRIDE Model

The STRIDE model was designed by Praerit and Loren at Microsoft, which can be used to threat modelling of software, hardware and network systems [11]. It provides a mnemonic for security threats as shown in Table I.

Spoofing is the process of manipulating data look like it has come from different sources. The main goal of spoofing is to cover the attacker tracks by misleading the server using a fake address. Examples include E-mail spoofing, MAC spoofing and IP address spoofing.

Denial of service is an attack in which the attacker attempts to make the victim unavailable to its legitimate users, through a temporary or indefinite interruption of provided services.

Tampering in STRIDE models means any improper modification of information. **Repudiation** is the ability to deny the participation in the communication or part of it. For example, the attacker can log into the system that does not have a log or tracing program running, so there is no evidence to decide who does what. **Non-repudiation** is to ensure that this repudiation does not occur.

Information Disclosure means to spying the information by attackers rather than his/her direct intention. For example, when a web server has a crash, there will be an error message utilized by administrators to discover the problem, but it may also give the attacker a chance to compromise the server.

For the properties of Information Security, STRIDE model mainly considers the followings:

- **Confidentiality:** Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- **Integrity:** Guarding data against improper modification or destruction of information.
- **Availability:** Ensuring the timely and reliable use of and access to information.
- **Authenticity:** The property of being genuine and being able to be verified and trusted.
- **Accountability/Non-repudiation:** The goal that generates the requirement for actions of an entity to be traced uniquely to that entity.

Below are the assets and objects that are critical for online social media.

- **Hardware:** Personal computer, mobile phone, data store server, etc.
- **Software:** Web browsing, mobile application, etc.
- **Data:** User information at the server.

During the security analysis, a number of threats have been identified, which we need to protect against to ensure that the security goals of the system are achieved.

To assess and determine the risk levels of different threats, risks are modeled with probabilities and impacts. In this work, both probability and impact are defined in three levels (low, medium, high).

- **Low:** A successful attack does not affect the functionality of a system.
- **Medium:** Requires active action, but does not render the system unable to function indefinitely.
- **High:** Irreversible or fatal damage to the system.

We show how to categorize an attack in different probability levels as follows:

- **Low:** The resources required for the attack outweigh the gain even if the attack is successful.
- **Medium:** The resources required for the attack are comparable to the gain of a successful attack.
- **High:** The gain from a successful attack should outweigh the resources needed to perform the attack.

B. Security Analysis of Facebook

Facebook introduced a privacy policy in 2009 for the first time, where users could select a personal privacy setting for their personal data. However, the default option was selected to be "Everyone", so many users accepted the default setting without being aware of the risks. This allowed much of the data to become publicly available.

After receiving feedback and criticisms about privacy concern, Facebook proposed a new privacy setting in 2010. There were different levels of privacy setting options on the page including Everyone, Friends of Friends, Friends Only for each data category. However, it was not sufficient to prevent privacy for users. Facebook did not possess strong privacy till 2011, where people could not reach some users' personal data and profiles without being friends.

Table II provides the threat model of Facebook. In most of the cases, attackers make use of the Facebook infrastructure to gather and expose the personal information of users and their friends. In doing so the attackers are able to make them go to malicious links, advertisements by generating fake profiles. Some of such common attacks in Facebook are shown below:

- **Compromised Account Attacks:** A compromised account is the condition of an account in which legitimate users lose complete or partial control over the login credentials [12]. Accounts can be compromised in different ways, e.g., by using a phishing scam to gather user login credentials, by utilizing cross-site scripting, and adopting bots to harvest login credentials. Compromised accounts can be very powerful means to spread out the malicious contents that can deteriorate the relationship established by the legitimate user in the past, and to communicate the malicious contents rapidly and effectively.
- **Sybil Attacks:** Malicious users create several fake identities, called Sybil, for influencing their identity within a target network [15]. After that, such malicious users send a friend request to rest of the users of that network. When one accepts the friend request, the malicious users will forward the malware and spam. Normally, Sybil attacks are found to be of two ways on Facebook. Attackers generate several fake identities to create legitimate accounts to spread malware and spam to friends in their friend list or form more social links to distribute malware and spam.
- **Socware Attacks:** For such attack type, contenders create malware, also known as *socware*, in the form of events, applications or pages capable of having links to malicious contents. False gift vouchers, coupons, and gifts are used as stimulants to attract victims, and then cheat them into installing or accepting the malware [8]. Once the malware has been installed in the system, attackers can easily gather personal information stored by the users. On top of that, the malware is posted on the user's wall, which will also spread on their friend's profile.
- **Identity Clone Attacks:** Malicious users, sometimes, create similar profiles pretending to be the victims and outspread malicious content into their network. To make

TABLE II
THREAT MODEL OF FACEBOOK.

Threat	Violated Property	Definition	Example
Spoofing	Authentication	Pretending to be someone else	Make fake Facebook account
Tampering	Integrity	Modify post on user's timeline	Delete/change post and message of others
Repudiation	Non-repudiation	Claim the real user	Multiple accounts and profile
Information Disclosure	Confidentiality	Unauthorized party gain access to Info	Malicious links, e.g., phishing URL
Denial of Service	Availability	Service unavailable to user	Overflow system, shutting down system

TABLE III
THREAT MODEL OF TWITTER.

Threat	Violated Property	Definition	Example
Spoofing	Authentication	Pretending to be someone else	Make fake Twitter account
Tampering	Integrity	Retweet false news	Promote false news
Repudiation	Non-repudiation	Claim the real user	Multiple accounts and profile
Information Disclosure	Confidentiality	Unauthorized party gain access to Info	An unauthorized person composes and sends tweets via text messages from a phone number associated with account
Denial of Service	Availability	Service unavailable to user	Overflow system, shutting down system

TABLE IV
THREAT MODEL OF LINKEDIN.

Threat	Violated Property	Definition	Example
Spoofing	Authentication	Pretending to be someone else	Fake job offer by using fake profile
Tampering	Integrity	Target potential victim	Convince users to open an email
Repudiation	Non-repudiation	Claim the real user	Multiple accounts and profile
Information Disclosure	Confidentiality	Unauthorized party gain access to Info	Malicious links, e.g., phishing
Denial of Service	Availability	Service unavailable to user	Overflow system, shutting down system

it possible, the malicious users normally first attempt to get a victim's personal information, such as occupation, name, and friends list. After collecting the information, attackers can copy the victim's profile and sends friend request to the victim's friends

- **Creepers Attacks:** Creepers are actual users who use online social media network functionalities in a wrong way [8]. For example, they would send a friend request to many unknown users and post spammy letters on their walls. Such attacks are mainly used for advertisements.
- **Cyberbullying attacks:** Cyberbullying is one of the most known and popular attacks in social network. This type of attacks can harass victims by posting sexual remarks, threat, repeated hurtful messages and irrelevant and disgusting contents. Besides, they can plant rumours about victims by posting awkward and embarrassing videos and / or photos online. According to a research study [6], 12% of the parents complained that their children have been cyber bullied.
- **Clickjacking Attacks:** For such attack type, it is also known as user interface (UI) redressing, where an adversary will trick users to click on some actionable contents, which are actually different from what they intend to click on. Afterwards, they can collect the personal information from these users and send spam messages and malicious links on their wall [17].

C. Security Analysis of Twitter

The popularity of Twitter has changed the way that users interact with technology. Generally, the users share their data with social network sites in a transparent way. Twitter is one of the famous public platforms, which provides developers with Application Programming Interface but with limited use for multiple reasons such as data volume, user privacy expectation, and Twitter business interests, since the platform will share some private information with advertisers such as how users interact with ads and which ones attract their attention. Sharing data is very crucial for the Twitter company because it has been proved that Twitter users interact with ads that advertisement companies post. As a result, these companies will pay Twitter and help it operate a free service.

Table III shows the threat model of Twitter. It is found that Twitter may suffer from various attacks but can also be used to intrude many users.

- **Short-URLs:** Due to the strict limitation on tweet length, users will use short-Universal Resource Locators (URLs) in tweets instead of standard URLs. The short URLs are indeed ordinary URLs that are encoded into URLs with the least characters, which thus best suit in tweets [18], [19]. A normal user has very limited knowledge of the target of the short URLs, and such users can easily be exploited and manipulated to download and / or spread malicious software without their knowledge. We have been experiencing plenty of such attacks in the recent times, as short URLs have increased in number along with the explosion of short messaging for mobile users. Attackers are able to exploit human shortcomings in various ways with the use of shortened URLs. Making it more difficult to understand and analyse, busy users do not take time to look into the link of the short-versioned URLs, therefore, the underlying URLs are more probable to be clicked. As many phishing emails are targeted to elicit quick emotional response from the recipients warning on negative consequences, an exhausted employee may hastily click on such links. Shortened URLs also benefit from the fact that several employees may not normally be aware of how a shortened URL looks like.

- **Compromise and control a user account:** In addition to sending direct messages to the users, attackers can use a compromised account to tweet to the followers. The probability of followers and the other users linked to the followers clicking on such ill-motivated links, in this case, is greater than the case of tweeting to the direct user, due to the fact that there is already a significant degree of trust between the users and their followers.
- **Clickjacking:** Clickjacking method is a very common and widely implemented attack among the advanced self-propagating attacks. The chance of clicking on a link is more likely by a follower of a user than by any other non-followers. In such attack, tweet retweets itself whenever a user of Twitter clicks on the link.
- **Indirect attack:** The clickjacking attack is remodelled to travel beyond Twitter. When the Twitter users are surfing in other public websites that allow users to enter links to other websites such as news sites or blogs. These sites provide a malicious short-URLs, and clicking on the links would result in a clickjacking attack when such a victim also has a Twitter handle.

D. Security Analysis of LinkedIn

Privacy is a great concern for LinkedIn, and that is why they have stressed upon it many times. Their main aim is to make transparency about the data they are collecting from the users. The privacy policy is applied on the users, who are using their service or product. LinkedIn provides their users an option to make a choice about the data collection, use and sharing as described in the privacy policy. The data collected by LinkedIn starts from creating a profile or an account, which includes user name, email address or mobile phone number and a password. If users would like to have a premium service, then they have to provide payment and billing information. After the registration phase, the user moves into profile setting. A user can fill in the information regarding his / her education, experience, skills and profile summary.

Table IV shows the threat model of LinkedIn. It is the same that many attacks are threatening the security and privacy of LinkedIn users.

- **Illegitimate Contact Requests:** Similar to other online social media platforms, the act of connecting with another LinkedIn user also leaves enough space for malicious activities. As a matter of fact, one of the most common tactics on LinkedIn is when a user gets a fake connection request from another member. Such requests may take on one of several different forms. In many cases, scammers may mostly claim that they are romantically interested at the recipients.
- **Fake Job Offers:** Users, sometimes, receive a LinkedIn message from claiming to be a job recruiter. The spammer then details a high-paying job and convinces the users that the duties can be performed from anywhere with Internet access. Such an offer lures a number of users as it sounds too good to be true.
- **Phishing:** The most customary type of phishing scam involves convincing people into opening emails or clicking on a link / url that appears to have been sent from a legitimate business or creditable source. LinkedIn hook has been found to have been used by more than half of social media phishing emails. LinkedIn has become the most trusted medium to target potential victims with more than half of all social media phishing emails using the Microsoftowned platform as a hook. KnowBe4's tests [3] revealed that LinkedIn had been used in 56% of the top phishing emails more than all other combined social media networks. The way such scam works is that one receives an email from someone that they might not know in person but is a business associate that they are connected to through LinkedIn. This kind of email would, at the first glance, look rather innocuous. Such emails use professional language, and one would be asked to click on a link that would direct them to a website, and the URL being used here would seem more or less legitimate, thereby, making one even less suspicious in the whole phishing process.

IV. Discussion on Phishing

According to the Infosecurity Magazine [4], email phishing attacks have spiked by over 600% since the end of February 2020, as many organisations and companies started working from home because of the COVID-19 restrictions. This working environment increased remote activities and the reliance on email for communication, thus, creating perfect conditions for email fraud schemes.

Criminals are taking advantage of the COVID-19 pandemic to launch phishing attacks. Below are some typical ways:

- **Zoom Users Become Targets:** This is an emerging type of phishing attack, in which intruder will send fake Zoom video-conferencing meeting notifications. This attack is designed to steal usernames and passwords from victims' Zoom accounts. Phishers can use these credentials to log into corporate video conferencing accounts, and try to collect passwords afterwards.
- **To use Covid-19 in business Email for Compromised Account Attacks:** In such attack, the phisher will send an

email to the person who has access to finance information of the organization. The phisher pretended to be a real supplier and requested for past due invoices including the information due to the Covid-19 in new account.

- **Fake registration process:** Victim may get an instant message with a link to claim for an official registration for the immediate withdraw of money from the Government. Thus, the phisher collects a victim's information via fake registration process.

Countermeasures. Phishing prevention can be reached by providing an extra layer of security in the login process. The extra layer could be a two-factor authentication scheme, which is a process to confirm the user's identity before he or she is granted to access an account.

For example, two-factor authentication can be performed via Short Message Service (SMS). When the user enters username and password, a verification code will be delivered to the other device. User can be granted access for login when he or she enters the verification code successfully. This method has been widely adopted in current market. However, this solution is not very usable in some cases, i.e., it requires an extra device from the user and causes extra cost to implement.

User awareness can help in educating the users by which they are able to identify phishing attempts. The big success of phishing attacks is mostly due to the negligence of users. To help reduce the phishing attacks, the awareness campaign program is very useful and important. Currently, there is no sufficient education and awareness campaign against phishing attacks. There are some anti-phishing methodologies such as games and security awareness tools in the server to familiarize threats like phishing attacks and identify malicious URLs and other phishing scams. The embedded server training program technique is used to teach users by sending a mock phishing email and asking them to open the attached emails or URL. Once the user opens the phishing contained email, they will find that the email contains fake information. In this way, the mock phishing awareness campaign increases the end user's knowledge to protect against phishing attacks.

V. Conclusion

Cyber criminals are now taking advantage of social media networks to collect personal information because users make a lot of private information public such as location, job, email, contact number and more. In this work, we provided a security analysis using STRIDE model on three major social media networks, such as Facebook, Twitter and LinkedIn. We found that phishing attack has a high probability of risk in online social media or using an online social media for launching attacks as compared with other potential attacks. Especially, criminals have taken advantage of the COVID-19 pandemic to design particular phishing attacks, i.e., the information about COVID-19 is included for convincing the user to click on designed URL. In the end, we also provided and discussed potential countermeasures to identify phishing content. This work aims to highlight the risk and issues in current social media networks and stimulate more defensive studies.

Acknowledgment

This research work was funded by the European Union H2020 DataVaults project with GA Number 871755.

References

[1] Most popular social networks worldwide as of January 2023, ranked by number of monthly active users. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[2] Forecast of the number of LinkedIn users in the World from 2019 to 2028. (accessed on 1 Jan 2023) https://www.statista.com/forecasts/1147197/linkedin-users-in-the-world

[3] Knowbe4 findings: LinkedIn Accounts for More than Half of Social Media Phishing Emails in Q2 2019. https://www.knowbe4.com/press/linkedin-accounts-for-more-than-half-of-social-media-phishing-emails-in-q2-2019-according-to-knowbe4-findings

[4] Covid-19 drives a phishing emails in a higher numbers under a month. (accessed on 1 Jan 2023) https://www.infosecurity-magazine.com/news/covid19-drive-phishing-emails-667/

[5] A. Aktypi, J.R.C. Nurse, and M. Goldsmith, "Unwinding Ariadne's Identity Thread: Privacy Risks with Fitness Trackers and Online Social Networks," in *Proc. MPS@CCS*, pp. 1-11, 2017.

[6] D. Bunga and O.S. Hiariej, "Cyberbullying on Children in Victimology Perspective," *Sociological Jurisprudence Journal* 2(2), pp. 1-6, 2019.

[7] J. Chen, J. He, L. Cai, and J. Pan, "Disclose More and Risk Less: Privacy Preserving Online Social Network Data Sharing," *IEEE Trans. Dependable Secur. Comput.* 17(6), pp. 1173-1187, 2020.

[8] O. Coban, A. Inan, and S.A. Ozel, "Privacy Risk Analysis for Facebook Users," in *Proc. SIU*, pp. 1-4, 2020.

[9] N. Laleh, B. Carminati, and E. Ferrari, "Graph Based Local Risk Estimation in Large Scale Online Social Networks," in *Proc. SmartCity*, pp. 528-535, 2015.

[10] S. Creese, M. Goldsmith, J.R.C. Nurse, and E. Phillips, "A Data-Reachability Model for Elucidating Privacy and Security Risks Related to the Use of Online Social Networks," in *Proc. TrustCom*, pp. 1124-1131, 2012.

[11] A.E.W. Eldewahi, A. Hassan, K. Elbadawi, and B.I.A. Barry, "The Analysis of MATE Attack in SDN Based on STRIDE Model," in *Proc. EIDWT*, pp. 901-910, 2018.

[12] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Towards Detecting Compromised Accounts on Social Networks," *IEEE Trans. Dependable Secur. Comput.* 14(4), pp. 447-460, 2017.

[13] N.A. Ghani, S. Hamid, I.A.T. Hashem, and E. Ahmed, "Social media big data analytics: A survey," *Comput. Hum. Behav.* 101: 417-428, 2019.

[14] W. Meng, W. Li, L. Jiang, and J. Zhou, "SocialAuth: Designing Touch Behavioral Smartphone User Authentication based on Social Networking Applications," in *Proc. IFIP SEC*, pp. 180-193, 2019.

[15] Y. Sun and L. Yin, "A Security Routing Mechanism against Sybil Attacks in Mobile Social Networks," in *Proc. APWeb Workshophs*, pp. 325-332, 2014.

[16] X. Han, H. Huang, and L. Wang, "F-PAD: Private Attribute Disclosure Risk Estimation in Online Social Networks," *IEEE Trans. Dependable Secur. Comput.* 16(6), pp. 1054-1069, 2019.

[17] U.U. Rehman, W.A. Khan, N.A. Saqib, and M. Kaleem, "On Detection and Prevention of Clickjacking Attack for OSNs," in *Proc. FIT*, pp. 160-165, 2013.

[18] N. Gupta, A. Aggarwal, and P. Kumaraguru, "bit.ly/malicious: Deep dive into short URL based e-crime detection," in *Proc. eCrime*, 14-24, 2014.

[19] D. Wang, S.B. Navathe, L. Liu, D. Irani, A. Tamersoy, and C. Pu, "Click traffic analysis of short URL spam on Twitter," in *Proc. CollaborateCom*, pp. 250-259, 2013.

[20] S. Fu and Z. Yao, "Privacy risk estimation of online social networks," in *Proc. NaNA*, pp. 1-8, 2022.

[21] P. van Schaik, J. Jansen, J.A. Onibokun, L.J. Camp, and P. Kusev, "Security and privacy in online social networking: Risk perceptions and precautionary behaviour," *Comput. Hum. Behav.* 78, pp. 283-297, 2018.

[22] M. Yang, Y. Yu, A.K. Bandara, and B. Nuseibeh, "Adaptive Sharing for Online Social Networks: A Trade-off Between Privacy Risk and Social Benefit," in *Proc. TrustCom*, pp. 45-52, 2014.

[23] C. Tang *et al.*, "Need for Symmetry: Addressing Privacy Risks in Online Social Networks," in *Proc. AINA*, pp. 534-541, 2011.

[24] E.Y. Chan and N.U. Saqib, "Online social networking increases financial risk-taking," *Comput. Hum. Behav.* 51, pp. 224-231, 2015.

Development of a Low-Cost Sensor to Optimise the Use of Fertilisers in Irrigation Systems

Francisco Javier Diaz[1], Ali Ahmad[1], Sandra Viciano-Tudela[1], Lorena Parra[1], Sandra Sendra[1], Jaime Lloret[1*]
[1]Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politècnica de València, 46730 Grau de Gandia, Spain

Email: fjdiabla@doctor.upv.es, aahmad1@upv.es, svictud@upv.es, loparbo@doctor.upv.es, sansenco@upv.es, jlloret@dcom.upv.es

***Abstract*— Fertilizers are widely used in agriculture to ensure the availability of nutrients for crops. Sensors are being used to determine fertilizers concentration in precision agriculture systems. In this paper, we present a low-cost sensor for determining fertilizer concentration based on optical and electromagnetic sensors. The combination of sensors prevents the overestimation of fertilizers. Four Two-Coil Systems (TCSs) were compared to determine which offered the most suitable data for determining the fertilizer in the water. The number of spires of powered and induced coils of TCSs ranged from 20 to 80 spires. A single configuration using a light source and light receptor is proposed for the optical sensor. Six calibration samples were prepared to calibrate both the TCSs and the optical-based sensor. The calibration samples vary from 0 to 10 mL/L of fertilizer. Results indicate that TCS 2 b is the one that offers the most accurate results among the tested TCSs. The single regression model obtained with data from TCS 2 b was characterized by a correlation coefficient of 0.988. Finally, the data for both sensors are used in an ANN model to predict the fertilizer concentration of samples. The correctly classified cases were 100 %.**

Keywords- Two-Coil System, copper coil, optical sensor, electromagnetic sensor, agriculture, artificial neural network.

I. INTRODUCTION

Soil fertility refers to the ability of soil to support and sustain plant growth. This ability is influenced by many factors, such as nutrient ad minerals presence, soil texture, soil organic matter, pH amount of water and biomass [1] [2]. Soil minerals usually cover up half of the soil volume, and their composition varies among different types of soils, changing their chemical composition and physical properties. At the same time, the mineral composition is affected by life forms present in the environment, modifying them. Moreover, the organic matter, complementary to soil minerals and fused together, makes up the solid phase of soil. Linked to these phases, another one occupies the rest of the soil space. The combination of water and air constitutes it. Together, all these phases allow plant growth, limiting or favouring organisms' growth and the living matter on the soil [3]. There are several types of fertiliser. They are classified according to the number of nutrients they are made with, being so, fertilisers with a single nutrient or fertilisers with several types of nutrients. Fertilisers with several types of nutrients are classified into double nutrient fertilisers, combining two necessary elements between Nitrogen (N), Phosphorus (P), or Potassium (K) [15].

Therefore, to perform the analysis of their phases, methods are traditionally used mainly based on chemical procedures, with extraction, digestion and processing samples, like colourimetry, for example [4]. Not so long ago, the use of sensors was an implemented idea with successful acceptance. They have the ability to transform physical or chemical readings from the environment into data signals, capable of being easily read by a system. Being said that, it is possible to place a large number of sensors around the study area so that they are capable of collecting data and sending it to a database. These sensors have two main purposes, monitoring the environment and tracking objects, animals, humans, etc. [5][6].

Typically, chemical methods are necessary for soil analysis. Nonetheless, they entail many associated disadvantages such as (i) high cost, (ii) long delay in order to obtain the results, (iii) use of reagents, and (iv) sample destruction [7]. Nowadays, sensors are able to determine soil fertility by taking measures of different parameters, such as soil pH, moisture, temperature, electrical conductivity, and nutrient levels. By giving these measures, it is possible to indicate whether applying any modification in the soil environment is necessary. Usually, sensors are able to provide relevant values regarding fertility, even in the presence of fertiliser [8]. The use of fertilisers allows an improvement in the physical soil properties and the numerous processes that the soil undergoes [9][10][11]. It is described, that a prolonged application and exposition to fertilisers, influence the quantity of solid matter [12], soil density, structure and ability to retain water [11][13][14]. When presicion agriculture is applied to fertilizer application in drip irrigation yield increases by 22% [15].

The aim of the paper is to verify two sensors, one based on EM fields and the other based on optical effects, are better to determine the minimum fertiliser concentration necessary to optimise its use. For the EM-based sensor, we utilised a Two-Coil System (TCS) that used mutual inductance. A magnetic field was generated by the first coil that was powered with an alternate current (AC) source, thereby inducing a magnetic flux in the second non-powered coil or induced coil (IC). In order to determine which coil was best suited for this test, 4 different coils were tested, changing the spires numbers. A light source and a light detector were used for the optical-based sensor. The fertiliser used was a liquid type with double nutrient composition, being that N and K. The main novelty of the proposed system is the combination of electromagnetic (EM) and optic sensors in a classification system so that it is possible to avoid overestimation of fertiliser. The system will save fertiliser, leading to economic savings for farmers.

The rest of the paper is structured as follows; Section 2 outlines the related work. The proposed system is fully described in Section 3. Following, Section 4 details the test bench. The results are discussed in Section 5. Finally, Section 6 summarises the conclusion and future work.

II. Related Work

Currently, different types of sensors are being employed for the detection of fertilisers. For instance, visible/near-infrared (Vis/NIR) spectroscopy was reported to be effective in determining fertiliser content and reducing fertiliser waste by enabling more precise application of fertilisers by Lin et al. [16]. Similarly, a nitrogen-phosphorus-potassium (NPK) based sensor was developed by Lavanya et al. [17] for soil fertility monitoring. The proposed sensor worked on the colourimetric principle consisting of Light Dependent Resistor (LDR) and Light Emitting Diodes (LEDs).

In 2019 [18], the evaluation of the effects of irrigation and the application of mineral and organic fertiliser was proposed by mapping the variability of the apparent electrical conductivity using multiconfiguration electromagnetic induction. Deductions obtained by using this procedure showed that long-term fertiliser application influenced the electromagnetic induction measures and that multi-coil can be used to determine the homogeneity of agricultural treatments. According to Basterrechea, et al. [19] proposed measuring the quantity of organic fertiliser by using inductive coils. The results were rather conclusive, showing that one of their prototypes was valid.

Silva et al. [20] developed a novel procedure for detecting contaminants in water and organic fertilisers using portable and disposable commercial electrodes. They reported the use of electrochemical sensors efficient in detecting low concentrations of the substances in both water and organic fertilisers and postulated it an appropriate tool for environmental monitoring and quality control in agriculture.

Based on different approaches, Qiu and Qu [21] reported a novel non-enzymatic electrochemical sensor for the detection of nitrite derived from nitrogen fertilisers. This sensor was of polyaniline and manganese dioxide, a binary nanocomposite material. Further, it was demonstrated that electrochemical detection of nitrite is non-enzymatic and does not require the use of costly equipment. Recently, Meenakshi and Naresh [22] used crop image identification to analyse soil health and fertiliser requirement. The study utilised deep learning algorithms and random forest regression, and the results suggested that this image identification approach could be useful for precision agriculture and reducing fertiliser waste. Similar outcomes were found by Wang et al. [23] when they employed machine vision technology to monitor the fertiliser use for corn fertiliser planters.

Nevertheless, all these currently employed fertiliser monitoring and estimation methods involve complex procedures in their employment, or data processing, or rely on a single parameter. Finally, some of the proposed solutions may require significant upfront investment or may not be financially viable for small-scale farmers, highlighting the need for cost-effective solutions. The system proposed in this paper is based on the combination of two parameters which can be measured using low-cost sensors. The use of two parameters reduces the possibility of overestimation due to irrigation water with elevated values of turbidity or salinity. The reasons to employ these sensors include their low cost, high accuracy, and easy deployment. In addition, promising results have been reported when similar sensors models were utilized in previous studies [18][24][25].

III. Proposal

In this section, we detail the proposed system for fertiliser monitoring in irrigation systems. First, the EM and optical sensors are described. Then, the used node is characterised. Finally, the ANN model proposed for data classification is presented.

A. *Sensor based on EM effects*

Two pairs of inductive coils are tested to evaluate which TCS that offers the best results. Each pair of inductive coils can be used in two configurations by changing the powered and induced coil; more details are provided in the subsequent section. Since an alternating current is required to power the coil, a specific electronic circuit is included to power the coil using the microprocessor. The used circuit is fully described in [24].

The operational principle of EM-based sensors is that the presence of salts in the fertiliser modifies the induction of the induced coil, as demonstrated in [18]. Thus, it is expected that the Vout of the induced coil changes when the concentration of fertiliser is modified.

B. *Sensor based on optical effects*

The second included sensor in our proposed system is based on light absorption. Thus, a LED and a Light Dependent Resistor (LDR) are included. A microprocessor module, including an LDR, KY-018 [26], is used. As a light source, a white LED has been selected, which is powered by the microprocessor.

The optical-based sensors' operational principle is that the fertiliser's presence modifies the water's colour and transparency. It has a direct effect on light absortion. Therefore, it is expected a reduction in the light transmitted to the LDR, which modifies the received Vout in the microprocessor.

C. *Node*

A microprocessor is used to power the LED, the powered coil and the LDR and to receive the signal from the LDR and the induced coil. The selected microprocessor must accomplish the required analogue input for the coil and the

LDR module. In addition, it must be able to run an ANN model. Thus, a Raspberry Pi 4 Model B [27] is selected.

D. *ANN model*

The fact of using two sensors is to avoid false detection of fertiliser. If the induced coil is the single sensor, when fertiliser is added to water with moderate salinity values, as happened in some coastal areas, salinity might be considered as fertiliser. It will provoke an overestimation of fertiliser concentration. Thus, the optical sensor is used. The Artificial Neural Network (ANN) is selected to combine the data of both sensors. We have selected it since it can be easly implemented in the microprocessor. The proposed ANN can be seen in Figure 1.

IV. TEST BENCH

In this section, the complete test bench is detailed. First of all, sample preparation is described. Then, used sensors and their assemblage are detailed. The equipment used to feed the TCS is explained. Finally, the measurement procedure for data gathering and data analysis is presented.

A. *Sample preparation*

Six fertiliser calibration samples were prepared to create a calibration curve to identify the appropriate sensor model. A commercial organic fertiliser [28], containing nitrogen, phosphorous and other nutrients was used. The recommended dose is 10 mL of fertiliser mixed with 1 L of water. Therefore, in order to identify the lowest possible amount of fertiliser, six dilutions were prepared, see Table I. Each calibration sample has a volume of 500 mL. The volumes used for the calibration include 100 mL.

B. *TCS description*

Two TCSs, TCS 1 and TCS 2, which can be connected in four configurations, were employed in these tests. TCSs can be seen in Figure 2a and Figure 2b.

The TCSs can be connected in two different configurations (a and b) since they have different spires. The number of spires in the IC and the PC of each configuration of the used TCSs are summarised in Table II. The wire used to craft the coils was enamelled copper coil of 0.4 mm diameter. These coils were wrapped on a 25 mm diameter Polyvinyl chloride (PVC) tube with an empty core.

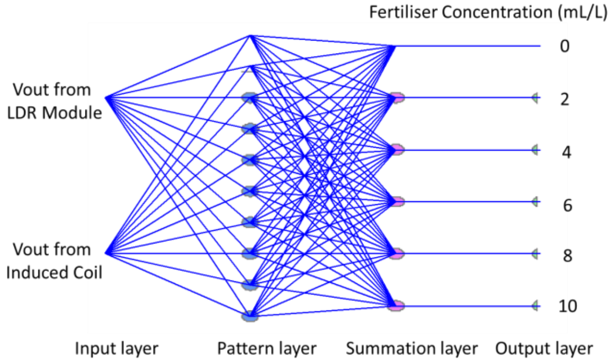


Figure 1. Proposed ANN model.

TABLE I. FERTILISER SAMPLES

Sampmle No	Sample content		
	Added Fertiliser	Added Water	Dose (mL/L)
1	0	500	0
2	1	499	2
3	2	498	4
4	3	497	6
5	4	496	8
6	5	495	10

TABLE II. SUMMARY OF TCSS' CHARACTERISTICS

Sampmle No	Features			
	Number of spires		Diameter (mm)	
	IC	PC	Coil	PVC Tube
TCS 1 a	80	40	0.4	25
TCS 1 b	40	80	0.4	25
TCS 2 a	40	20	0.4	25
TCS 2 b	20	40	0.4	25

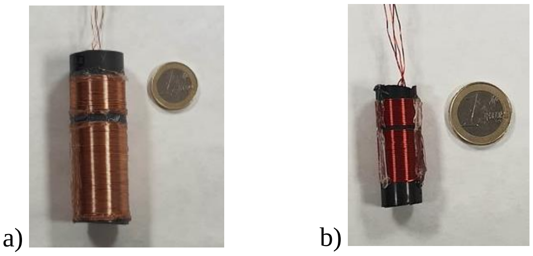


Figure 2. Used TCSs a) TCS 1, b) TCS 2.

C. *Optical sensor assemblage*

The measures were taken by connecting the LDR module to the microprocessor. The LED and the LDR were separated 3 cm by a methacrylate transparent tube. The tube was filled with the calibration samples. The Vout from the LDR module was obtained from the microprocessor using the analogue input.

D. *Equipment*

A wave generator, model AFG1022 [29], has been used to power the PC. Resistance of 330 Ω was connected in series to PC. A sine wave having an amplitude of 3.3 V peak-to-peak and 0.045 A have been used to power the coils. The used signal generator allows a range of frequencies from 25 MHz to 1 uHz.

The IC is connected to an oscilloscope, model TBS1104 [30]. A capacitor of 10 nF was added in parallel to the IC. The complete scheme of the used devices and the electronic circuit can be seen in Figure 3.

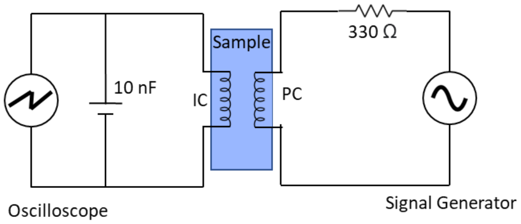


Figure 3. TCS circuit.

E. *Measurement protocol and data analysis*

The measurement protocol followed for the calibration process consists of measuring the Vout values of the TCS or the LDR module. Measurements started from the most diluted samples, sample 1, to the most concentrated one, sample 6. Each measurement was repeated three times.

In the TCS measurement procedure, the oscilloscope is used to gather the data as a previous step before adapting the required electronic circuit to measure it with the microprocessor. The first step is to identify the working frequency for each used PC. Once the working frequency is determined, calibration starts. For the calibration, the TCS was entirely submerged in the calibration samples. Induced voltage, Vout, was gathered individually for all used sensors.

For the optical-based sensors, the measurement procedure consists of adding fertiliser inside the methacrylate tube. Then, the white LED is powered by the microprocessor. At the same time, the microprocessor reads the value from the analogue input of the LDR module.

Obtained data is statistically analysed. The performed analyses include descriptive statistics, simple regression models and Discriminant analyses (DA). All these analyses, in addition to the ANN, have been performed with the Statgraphics Centurion XVIII.

V. Results

In this section, we present the results from the obtained Vout from both sensors. First, the results of the descriptive statistics are shown for the Vout of the induced coil to decide which coil and configuration are selected. Then, the results of the mathematical models are compared. Finally, the results of the ANN are analysed.

A. *Descriptive statistics of Vout from induced coils*

First of all, the working frequencies are presented. The working frequency for the used TCSs was: 147 kHz for TCS 1 a, 267 kHz for the TCS 1 b, 433 kHz for TCS 2 a, and 665 kHz for the TCS 2b.

The summation of descriptive analyses of data from IC can be seen in Table III. The Table shows that the coil that maximises the average Vout is TCS 1 a. TCS 1 b is characterised by a mean Vout of 2.52 V, which is similar to the mean Vout of TCS 2 b, 2.21 V. TCS 1 b is the one with lowers mean Vout, 1.88 V. The standard deviation (σ) is the minimum for TCS 2 a and TCS 2 b. Finally, values of Kurtosis and Skewness are between ± 2 for TCS 1 b, 2a, and 2b. Nonetheless, for TCS 1 a, the values are beyond the established thresholds to be considered a variable with normal distribution.

Finally, Figure 4 portrays the distribution of each variable as violin plots. For these plots, cosine has been used as a smoothing method. The interval width has been set at 35. The means and outliers are represented in the graphic.

B. *Simple Regression Models*

In this section, the single correlation models for each one of the EM-based sensors are presented. The regression models were selected among the available options, maximising the correlation coefficient's value. The summary of regression models can be seen in Table IV. In the table, the selected model type, the correlation coefficient and the R-squared (R^2). According to the R2 and correlation coefficients, we selected TCS 1 a and TCS 2 b as two alternatives for EM-based sensors for the proposed system.

The correlation models for TCS 1 a and TCS 2 b can be seen in Figure 5 and Figure 6, respectively. The mathematical model represents confidence and prediction intervals in both figures. We selected the second model for the obtained models due to their more accurate intervals. Moreover, the first option, inducer 1 a, has maximum values which are too high for the analogue inputs for the microprocessor.

C. *Classification Methodologies*

Finally, we combine the data for the Vout from the TCS 2 b and the LDR module for the classification methods. On the one hand, the classification diagram for the DA based on generated functions can be seen in Figure 7. On the other hand, the classification diagram for the ANN can be seen in Figure 8. For the ANN, the dataset was split into a training dataset (first and second replicas of tests) and a validation dataset (third replica). Regarding the classified cases, both tested classification methods, DA and ANN, allow the classification of 100% of cases.

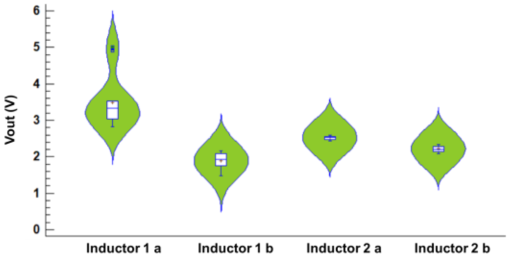


Figure 4. Violin plot of Vout for the different sensors.

TABLE III. SUMMARY OF CHARACTERISTICS OF THE SELECTED JELLYFISH

	Sensor 1		**Sensor 2**	
	TCS 1 a	***TCS 1 b***	***TCS 2 a***	***TCS 2 b***
Mean Vout (V)	3.49556	1.88444	2.52778	2.21333
Minimum Vout (V)	2.82	1.48	2.44	2.08
Maximum Vout (V)	5	2.16	2.6	2.34
σ	0.713914	0.226747	0.0470988	0.0840168
Kurtosis	2.58232	-0.963285	-0.807876	-0.165955
Skewness	0.94211	-0.763609	-0.781795	-1.12133

TABLE IV. SUMMARY OF TCSS' CHARACTERISTICS

TCS	Selected model	Correlation coefficient	R^2
TCS 1 a	Y = 1/(a + b*sqrt(X))	0.987	97.45
TCS 1 b	Y = sqrt(a + b*X)	-0.822	67.638
TCS 2 a	Y = sqrt(a + b*sqrt(X))	-0.312	9.725
TCS 2 b	Y = sqrt(a + b*X)	0.988	97.709

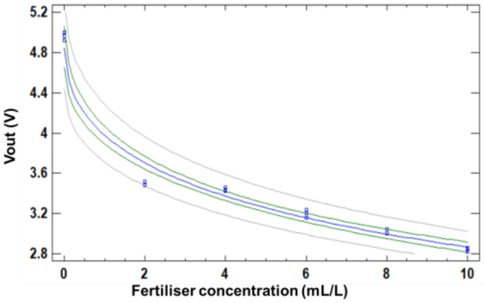


Figure 5. X-Y Plot for Vout of the TCS 1 a.

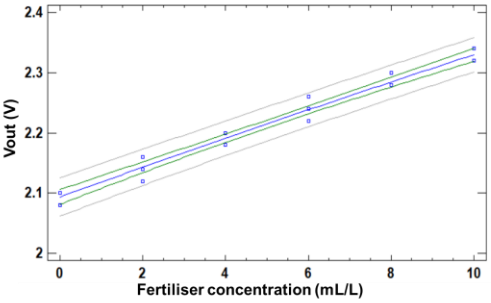


Figure 6. X-Y Plot for Vout of the TCS 2 b.

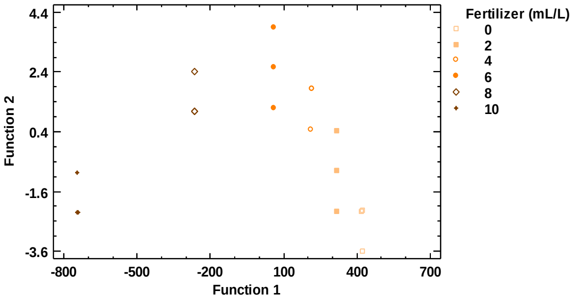


Figure 7. Classification diagram with DA.

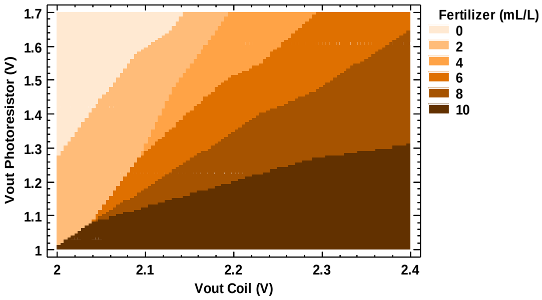


Figure 8. Classification diagram with ANN.

In both training and validation, 100 % of correctly classified cases were attained. A second classification was done by selecting the data for each dataset randomly, with the same percentage of cases correctly classified in both the training and validation dataset.

VI. CONCLUSIONS AND FUTURE WORK

The use of fertiliser has increased considerably in recent years. The need to produce food in the shortest possible time and protect the fruit against biological agents has become a current area of research.

In this paper, we presented a system for optimising liquid fertilisers for irrigation water. The system is based on TCSs that generate electromagnetic fields that allow the establishment of the concentrations of fertilisers. Thus, Vout measurements are established for each solution prepared. A function generator has been used and visualised in an oscilloscope to generate the electromagnetic field. In addition, an LDR and a photoresistor have been implemented to obtain data to avoid overestimating the fertiliser concentration. After applying the statistical analysis, the results show that the system can classify 100% fertiliser solutions with different concentrations.

In future work, we want to maximise the sensitivity of the coil at smaller concentrations and check in a real environment how the decrease in fertiliser does not affect plant growth. This fact would allow economic savings for the farmer, as well as a decrease in the use of products in agriculture.

ACKNOWLEDGMENT

This work has been funded by the “Ministerio de Ciencia e Innovación” through the Project PID2020-114467RR-C33/AEI/10.13039/501100011033, by “Ministerio de Agricultura, Pesca y Alimentación” through the “proyectos de innovación de interés general por grupos operativos de la Asociación Europea para la Innovación en materia de productividad y sostenibilidad agrícolas (AEI-Agri)”, project GO TECNOGAR, and by the “Ministerio de Economía y Competitividad” through the Project TED2021-131040B-C31. This study also forms part of the ThinkInAzul programme and was supported by MCIN with funding from European Union NextGenerationEU (PRTR-C17.I1) and by Generalitat Valenciana (THINKINAZUL/2021/002).

REFERENCES

[1] A. Desbiez, R. Matthews, B. Tripathi, and J. EllisJones, “Perceptions and assessment of soil fertility by farmers in the mid-hills of Nepal,” Agric. Ecosyst. Environ., vol. 103, no. 1, pp. 191–206, 2004.

[2] I. M. Cardoso and T. W. Kuyper, “Mycorrhizas and tropical soil fertility,” Agric. Ecosyst. Environ., vol. 116, no. 1–2, pp. 72–84, 2006.

[3] N. Uphoff et al., “Biological Approaches to Sustainable Soil Systems,” Books in ScienceBiol. Approaches to Sustain. Soil Syst., no. Jan., 2006.

[4] N.T. Fainthfull NT, “Methods in agricultural chemical analysis,” CABI Publishing, Oxfordshire, pp 57–104, 2002.

[5] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Comput. Networks, vol. 52, no. 12, pp. 2292–2330, 2008.

[6] J. Lloret, M. Garcia, F. Boronat, and J. Tomás, “A group-based protocol for large wireless AD-HOC and sensor networks,” 2008 IEEE Netw. Oper. Manag. Symp. Work. - NOMS 08, vol. 23, no. May, pp. 7–14, 2008.

[7] E. Lopez and F. Miñano, “Metodos rapidos de analisis de suelos,” Minist. Agric. Pesca y Aliment., p. 32pp, 2008.

[8] S. Postolache, P. Sebastião, V. Viegas, O. Postolache, and F. Cercas, “IoT-Based Systems for Soil Nutrients Assessment in Horticulture,” Sensors, vol. 23, no. 1, 2023.

[9] D. W. Reeves, “The role of soil organic matter in maintaining soil quality in continuous cropping systems,” Soil Tillage Res., vol. 43, no. 1–2, pp. 131–167, 1997.

[10] B. J. Zebarth, G. H. Neilsen, E. Hogue, and D. Neilsen, “Influence of organic waste amendments on selected soil physical and chemical properties,” Can. J. Soil Sci., vol. 79, no. 3, pp. 501–504, 1999.

[11] I. Celik, I. Ortas, and S. Kilic, “Effects of compost, mycorrhiza, manure and fertilizer on some physical properties of a Chromoxerert soil,” Soil Tillage Res., vol. 78, no. 1, pp. 59–67, 2004.

[12] D. K. Benbi, C. R. Biswas, S. S. Bawa, and K. Kumar, “Influence of farmyard manure, inorganic fertilizers and weed control practices on some soil physical properties in a long-term experiment,” Soil Use Manag., vol. 14, no. 1, pp. 52–54, 1998.

[13] J. J. Miller, N. J. Sweetland, and C. Chang, “Hydrological Properties of a Clay Loam Soil after Long-Term Cattle Manure Application,” J. Environ. Qual., vol. 31, no. 3, pp. 989–996, 2002.

[14] P. Schjønning, L. J. Munkholm, P. Moldrup, and O. H. Jacobsen, “Modelling soil pore characteristics from measurements of air exchange: The long-term effects of fertilization and crop rotation,” Eur. J. Soil Sci., vol. 53, no. 2, pp. 331–339, 2002.

[15] Shareef, T. M. E., Ma, Z., & Zhao, B.,” Essentials of drip irrigation system for saving water and nutrients to plant roots: As a guide for growers.” Journal of Water Resource and Protection., Vol 11, no 9, pp. 1129-1145, 2019.

[16] Z. Lin et al., "Accurate and rapid detection of soil and fertilizer properties based on visible/near-infrared spectroscopy," Applied Optics, vol. 57, no. 18, pp. D69-D73, 2018.

[17] G. Lavanya, C. Rani, and P. GaneshKumar, "An automated low cost IoT based Fertilizer Intimation System for smart agriculture," Sustainable Computing: Informatics and Systems, vol. 28, p. 100300, 2020.

[18] J. Nie et al., "Effect of drip irrigation with magnetized water and fertilizer on cotton nutrient absorption." IOP Conference Series: Earth and Environmental Science. vol. 697. no. 1, IOP Publishing, 2021.

[19] D. A. Basterrechea, L. Parra, M. Botella-Campos, J. Lloret, and P. V. Mauri, “New Sensor Based on Magnetic Fields for Monitoring the Concentration of Organic Fertilizers in Fertigation Systems,” Applied Sciences, vol. 10, no. 20, p. 7222, Oct. 2020.

[20] L. R. Silva, J. G. Rodrigues, M. de LS Vasconcellos, E. S. Ribeiro, E. D’Elia, and R. de Q. Ferreira, “Portable and simple electroanalytical procedure for simultaneous detection of dipyrone and norfloxacin with disposable commercial electrodes in water and organic fertilisers,” Ionics, vol. 28, no. 10, pp. 4833-4841, 2022.

[21] Y. Qiu and K. Qu, "Binary organic-inorganic nanocomposite of polyaniline-MnO2 for non-enzymatic electrochemical detection of environmental pollutant nitrite," Environmental Research, vol. 214, p. 114066, 2022.

[22] M. Meenakshi and R. Naresh, "Soil health analysis and fertiliser prediction for crop image identification by Inception-V3 and random forest," Remote Sensing Applications: Society and Environment, vol. 28, p. 100846, 2022.

[23] B. Wang, Y. Wang, H. Wang, H. Mao, and L. Zhou, "Research on accurate perception and control system of fertilisation amount for corn fertilisation planter," Frontiers in Plant Science, vol. 13, no. 1074945, 2022.

[24] L. Parra, S. Viciano-Tudela, D. Carrasco, S. Sendra, and J. Lloret, “Low-Cost Microcontroller-Based Multiparametric Probe for Coastal Area Monitoring,” Sensors, vol. 23, no. 4, 2023.

[25] Y. Wang, S. S. M. Rajib, C. Collins, and B. Grieve, “Low-cost turbidity sensor for low-power wireless monitoring of fresh-water courses,” IEEE Sensors Journal, vol. 18, no. 11, pp. 4689-4696, 2018.

[26] LDR module . KY-018. Available online: https://datasheet.lcsc.com/szlcsc/Shenzhen-Jing-Chuang-He-Li-Tech-GL5528-10-20_C10081.pdf. (accessed on 23 February 2023).

[27] Raspberry Pi 4 Model B. Available online: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf (accessed on 23 February 2023).

[28] Fertilizer Datasheet. Available at: https://www.compo.de/dam/jcr:e4246087-ca92-4ea5-83da-52aa203e1d42/2224501004_r0427195.PDF [Last access on 23/02/2023]

[29] Information of the current generator used. Available at: https://www.mouser.com/datasheet/2/403/AFG1022-Arbitrary-Function-Generator-Datasheet-1-540840.pdf [Last access on 23/02/2022].

[30] Information of the Osciloscpoe used. Available at: https://www.mouser.com/datasheet/2/403/AFG1022-Arbitrary-Function-Generator-Datasheet-1-540840.pdf [Last access on 23/02/2022].

ISBN: 978-1-68558-052-0