ITU Journal

Future and evolving technologies

FREE I FAST I FOR ALL

Volume 2 (2021), Issue 1

ITU Journal
Future and evolving technologies

Volume 2 (2021), Issue 1

ITU Journal
Future and evolving technologies

FREE I FAST I FOR ALL

The ITU Journal on Future and Evolving Technologies (ITU J-FET) is an international journal providing complete coverage of all communications and networking paradigms, free of charge for both readers and authors.

The ITU Journal considers yet-to-be-published papers addressing fundamental and applied research. It shares new techniques and concepts, analyses and tutorials, and learnings from experiments and physical and simulated testbeds. It also discusses the implications of the latest research results for policy and regulation, legal frameworks, and the economy and society. This publication builds bridges between disciplines, connects theory with application, and stimulates international dialogue. Its interdisciplinary approach reflects ITU's comprehensive field of interest and explores the convergence of ICT with other disciplines.

The ITU Journal welcomes submissions at any time, on any topic within its scope.

Publication rights

SUGGESTED CITATION: ITU Journal on Future and Evolving Technologies, Volume 2 (2021), Issue 1.

ADDITIONAL INFORMATION
Please visit the ITU Journal website at:
https://www.itu.int/en/journal/j-fet/Pages/default.aspx

Inquiries should be addressed to
Alessia Magliarditi at: journal@itu.int

Foreword

The ITU Journal embodies the inclusive character of ITU.

It seeks global representation in its published papers and teams of reviewers and editors, and it welcomes an interdisciplinary approach to the broad scope of topics addressed by ITU.

ITU is the United Nations specialized agency for information and communication technologies (ICTs). Our global membership includes 193 Member States and over 900 companies, universities, and international and regional organizations.

In contributing to the Journal and the work of ITU, leading minds in science and engineering are providing insight into the latest developments in ICT and prospects for future innovation.

ITU works in service of the public interest, aiming to ensure that all people share in the benefits of the ICT advances changing our world. This mission is very much aligned with the mission of the global academic community.

I express my gratitude to all contributors to this second issue of the ITU Journal and I would especially like to thank our Editor-in-Chief, Ian F. Akyildiz, for the great dedication and conviction that define his leadership.

Houlin Zhao
Secretary-General
International Telecommunication Union

Foreword

We are entering new frontiers in information and communication technology (ICT) as digital transformation reshapes our societies and economies. As future technologies are conceived and the state of the art evolves, the ITU Journal will lead an open conversation at the forefront of this innovation.

Our Journal, free of charge for both readers and authors, is the latest initiative to contribute to the growing strength of ITU's relationship with academia.

Researchers participate alongside policymakers and industry-leading engineers in ITU expert groups responsible for radiocommunication, standardization and development. Contributions from research communities bring greater strength to the work of ITU, and participation in ITU helps these communities to increase the impact of their research.

I thank all contributors to this second issue of the ITU Journal and I welcome you to the ITU community. I also express my gratitude to our Editor-in-Chief, Ian F. Akyildiz, for the vision and passion with which he is creating unique value to readers and authors worldwide.

ITU Academia membership, ITU Kaleidoscope conferences and the ITU Journal are key avenues for the academic profession to engage in ITU's work. I encourage you to join us in our work to help all innovators achieve their ambitions in digital transformation.

Chaesub Lee
Director
ITU Telecommunication Standardization Bureau

Editor-in-Chief's Message

I am pleased to welcome you on behalf of the International Telecommunication Union to the second issue of the ITU Journal on Future and Evolving Technologies (ITU J-FET). This journal is the peer-reviewed publication of the Union, available at no cost for both authors and readers. Releasing this complete issue soon after the announcement of our inaugural publication at the end of 2020 confirms that the ITU J-FET is indeed free, fast, for all.

With the commitment of an expert, international editorial board and of the researchers who contributed to this issue, eight papers have been published, providing analyses into emerging technologies and forward-looking solutions around networks and communication systems.

I trust that our readers will enjoy the research included in this publication and be encouraged to contribute to the academic discussions ongoing here.

To begin this issue, the paper "*Dynamic power control for time-critical networking with heterogeneous traffic*" focuses on future wireless networks and presents a stochastic network optimization problem, developing an algorithm to solve this in real-time. Considering further algorithms and developments towards protocol standards, "*Reconfiguration algorithms for high precision communications in time sensitive networks: Time-aware shaper configuration with IEEE 802.1Qcc*" investigates the impact of IEEE 802.1Qcc management protocol to configure Time Aware Shaper reconfigurations in response to dynamic network conditions.

With the increasing discussions related to cryptocurrencies today and the rise in the number of users on payment channel networks, privacy concerns are being brought to the fore. "*An evaluation of cryptocurrency payment channel networks and their privacy implications*" examines these issues and evaluates the various payment networks according to privacy metrics.

Software Defined Networks are likely to revolutionize network deployment and contribute to economic growth. The published paper, "*Controller placement optimization software defined wide area networks (SDWAN)*" provides a tested approach for network operators integrating SDN or developing new SDN deployment plans for optimal controller placement. With a similar lens towards the future and economic growth, "*Economic efficiency of spectrum allocation*" makes the case for a digital spectrum management system to encourage a more sustainable model of development in a time where the demand for spectrum is continuously increasing.

Aiming to balance the efficiency trade-offs between low implementation complexity and high error correction in OFDM communication systems, a performance analysis of new Hamming Coding design is provided in "*Performance of a parallel Hamming Coding in short-frame OFDM sensor's network*".

Deployment of wireless sensor networks is often limited by the Radio Access Technologies (RATs) used and now, authors present a new RODENT (Routing Over Different Existing Network Technologies protocol). This technology is explained as being more energy efficient, with increased coverage and multiple data requirements support, according to the research in "*RODENT: A flexible TOPSIS based routing protocol for multi-technology devices in wireless sensor networks*".

The eighth paper "*A multi-link communication connectivity game under hostile interference*" adopts the principles of game theory to show how decision making can be improved for determining communication stabilization including in MLCC systems.

Beyond this issue, a series of five special issues will be published later this year, covering areas of research pertinent to the work of ITU. I look forward to announcing these publications and continuing to encourage researchers to submit to the ITU J-FET, as we accept submissions throughout the year.

I would like to thank the authors and expert editors that have contributed to this issue of the ITU J-FET.

I wish to express my gratitude to the ITU Secretary-General, Houlin Zhao, for his continued support and for entrusting me with the responsibility to lead this Journal as Editor-in-Chief. I would also like to thank the Director of the ITU Telecommunication Standardization Bureau, Chaesub Lee, for his support, and the ITU Journal Team.

Dr. Ian F. Akyildiz
CEO
Truva Inc.,
Alpharetta, GA 30022, USA
Ian.akyildiz@itu.int

EDITORIAL BOARD

TABLE OF CONTENTS

Page

LIST OF ABSTRACTS

Dynamic power control for time-critical networking with heterogeneous traffic

Pages 1–12

Emmanouil Fountoulakis, Nikolaos Pappas, Anthony Ephremides

Future wireless networks will be characterized by heterogeneous traffic requirements. Examples can be low-latency or minimum-througput requirements. Therefore, the network has to adjust to different needs. Usually, users with low-latency requirements have to deliver their demand within a specific time frame, i.e., before a deadline, and they coexist with throughput oriented users. In addition, mobile devices have a limited-power budget and therefore, a power-efficient scheduling scheme is required by the network. In this work, we cast a stochastic network optimization problem for minimizing the packet drop rate while guaranteeing a minimum throughput and taking into account the limited-power capabilities of the users. We apply tools from Lyapunov optimization theory in order to provide an algorithm, named Dynamic Power Control (DPC) algorithm, that solves the formulated problem in real time. It is proved that the DPC algorithm gives a solution arbitrarily close to the optimal one. Simulation results show that our algorithm outperforms the baseline Largest-Debt-First (LDF) algorithm for short deadlines and multiple users.

View Article

Reconfiguration algorithms for high precision communications in time sensitive networks: Time-aware shaper configuration with IEEE 802.1Qcc

Pages 13–34

Ahmed Nasrallah, Venkatraman Balasubramanian, Akhilesh S. Thyagaturu, Martin Reisslein, Hesham ElBakoury

As new networking paradigms emerge for different networking applications, e.g., cyber-physical systems, and different services are handled under a converged data link technology, e.g., Ethernet, certain applications with mission critical traffic cannot coexist on the same physical networking infrastructure using traditional Ethernet packet-switched networking protocols. The IEEE 802.1Q Time Sensitive Networking (TSN) Task Group is developing protocol standards to provide deterministic properties, i.e., eliminates non-deterministic delays, on Ethernet based packet-switched networks. In particular, the IEEE 802.1Qcc, centralized management and control, and the IEEE 802.1Qbv, Time-Aware Shaper (TAS), can be used to manage and control Scheduled Traffic (ST) streams with periodic properties along with Best-Effort (BE) traffic on the same network infrastructure. We investigate the effects of using the IEEE 802.1Qcc management protocol to accurately and precisely configure TAS enabled switches (with transmission windows governed by Gate Control Lists (GCLs) with Gate Control Entries (GCEs)) ensuring ultra-low bounded latency, zero packet loss, and minimal jitter for ST TSN traffic. We examine both a centralized network/distributed user model (hybrid model) and a fully-distributed (decentralized) 802.1Qcc model on a typical industrial control network with the goal of maximizing the number of ST streams.

View Article

An evaluation of cryptocurrency payment channel networks and their privacy implications

Pages 35–44

Enes Erdin, Suat Mercan, Kemal Akkaya

Cryptocurrencies redefined how money can be stored and transferred among users. However, public blockchain-based cryptocurrencies suffer from high transaction waiting times and fees. To address these challenges, the payment channel network concept is touted as the most viable solution to be used for micro-payments. The idea is exchanging the ownership of money by keeping the state of the accounts locally which provides transaction approvals in seconds. Such attention on payment channel networks has inspired many recent studies that focus on how to design them and allocate channels such that the transactions will be secure and efficient. However, as payment channel networks are emerging and reaching a large number of users, privacy issues are becoming more relevant, this raises concerns about exposing not only individual habits but also businesses' revenues. In this paper, we first propose a categorization of the existing payment networks formed on top of blockchain-backed cryptocurrencies. After discussing several emerging attacks on user/business privacy in these payment channel networks, we qualitatively evaluate them based on a number of privacy metrics that relate to our case. Based on the discussions on the strengths and weaknesses of the approaches, we offer possible directions for research for the future of privacy based payment channel networks.

View Article

Controller placement optimization for Software Defined Wide Area Networks (SDWAN)

Pages 45–66

Lusani Mamushiane, Joyce Mwangama, Albert Lysko

Software Defined Networking (SDN) has emerged as a promising solution to revolutionize network deployment, operations and economic growth. This paradigm aims to address management and configuration complexities in legacy networks so as to reduce the total cost associated with deploying and running telecommunication infrastructures. At the heart of SDN is a controller which oversees orchestration of resources. An important problem that must be addressed during the initial design of an SDN-based network deployment is to find the optimal number of controllers and their locations, to achieve desired operational efficiency. This problem constitutes competing objectives such as latency, load balancing, and reliability. We apply Silhouette Analysis, Gap Statistics and the Partition Around Medoids (PAM) algorithms and, unlike previous work, we add a new method for solving the controller placement problem using an emulation orchestration platform. Our approach aims to optimize controller-to-node latency, alleviate control-plane signalling overhead and ensure control-plane resiliency. Our results for South African national research network (SANReN) reveal that deploying two controllers yields the lowest latency, reduces control-plane signalling overhead and guarantees control-plane resiliency. Our approach can be used by network operators as a guideline to start integrating SDN or plan a new SDN deployment, by helping them make quick automatic decisions regarding optimal controller placement.

View Article

Economic efficiency of spectrum allocation

Pages 67–76

Vadim Nozdrin

Spectrum demand is rapidly expanding, driven by the developments of incumbent radio systems, as well as with the requirements of new technologies and market players to get spectrum access. In this regard, one of the fundamental functions of the state spectrum regulator is to create an enabling administrative and legal environment to ensure efficient spectrum use and to mitigate scarcity of this valuable natural resource. This paper analyses the economic value of spectrum use and suggests an extension of the spectrum's traditional technical boundaries, in order to take into consideration energy, environmental, sanitary and bio-geotechnical limits. The study reviews the phenomena of increasing transaction costs and negative externalities of spectrum access requiring economic criteria and assessment to achieve optimum resource allocation with a spectrum's demand growth. Results develop an input-output matrix as an allocation instrument to achieve spectrum efficiency and optimum equilibrium based on analysis of the economic effects from different band utilization scenarios, supplemented by currently used technical criteria. Practical application of proposed economic methods should improve the existing spectrum management system.

View Article

Performance of a parallel Hamming coding in short-frame OFDM sensor's network

Pages 77–88

Raouia Masmoudi Ghodhbane, Jorge Fernandez-Mayoralas

In this paper, we focus on the most relevant Error Correcting Codes (ECCs): the Hamming code and the Reed-Solomon code in order to meet the trade-off between the low implementation complexity and the high error correction capacity in a short-frame OFDM communication system. Moreover, we discuss and validate via simulations this trade-off between complexity (Hamming is the easiest to code) and error correction capability (Reed-Solomon being the most effective). Therefore, we have to either improve the correction capacity of the Hamming code, or decrease the complexity cost for the Reed-Solomon code. Based on this analysis, we propose a new design of parallel Hamming coding. On the one hand, we validate this new model of parallel Hamming coding with numerical results using MATLAB-Simulink tools and BERTool Application which makes easier the Bit Error Rate (BER) performance simulations. On the other hand, we implement the design of this new model on an FPGA mock-up and we show that this solution of a parallel Hamming encoder/decoder uses a few resources (LUTs) and has a higher capability of correcting when compared to the simple Hamming code.

View Article

RODENT: A flexible TOPSIS based routing protocol for multi-technology devices in wireless sensor networks

Pages 89–100

Brandon Foubert, Nathalie Mitton

Wireless Sensor Networks (WSN) are efficient tools for many use cases, such as environmental monitoring. However WSN deployment is sometimes limited by the characteristics of the Radio Access Technologies (RATs) they use. To overcome some of these limitations, we propose to leverage the use of a Multiple Technologies Network (MTN). What we refer to as MTN is a network composed of nodes which are able to use several RAT and communicating wirelessly through multi-hop paths. The management of the RAT and routes must be handled by the nodes themselves, in a local and distributed way, with a suitable communication protocol stack. Nodes may reach multiple neighbors over multiple RAT. Therefore, each stack's layer has to take the technologies' heterogeneity of the devices into account. In this article, we introduce our custom Routing Over Different Existing Network Technologies protocol (RODENT), designed for MTN. It enables dynamic (re)selection of the best route and RAT based on the data type and requirements that may evolve over time, potentially mixing each technology over a single path. RODENT relies on a multi-criteria route selection performed with a custom lightweight TOPSIS method. To assess RODENT's performances, we implemented a functional prototype on real WSN hardware, Pycom FiPy devices. Unlike related prototypes, ours has the advantage not to rely on specific infrastructure on the operator's side. Results show that RODENT enables energy savings, an increased coverage as well as multiple data requirements support.

View Article

A multi-link communication connectivity game under hostile interference

Pages 101–112

Andrey Garnaev, Wade Trappe, Narayan B. Mandayam, H. Vincent Poor

In this paper, we consider a communication connectivity problem involving a primary user (transmitter, for example, a Ground Control Station (GCS)) servicing a group of secondary users (receivers, for example, drones) under hostile interference. We formulate this multi-link communication connectivity problem, where the channels are affected by Rayleigh fading, as a zero-sum power resource allocation game between a transmitter and an adversary (jammer). The transmitter's objective is to maximize the probability of communication connectivity with all the receivers. It is proven that the problem has unique equilibrium in power allocation strategies, and the equilibrium is derived in closed form. Moreover, we reduce the problem of designing the equilibrium in power resource allocation strategies to the problem of finding a fixed point of a real-valued function. An algorithm based on the bisection method to find the fixed point (and so equilibrium strategies) is developed, and its convergence is proven.

View Article

DYNAMIC POWER CONTROL FOR TIME-CRITICAL NETWORKING WITH HETEROGENEOUS TRAFFIC

Emmanouil Fountoulakis[1], Nikolaos Pappas[2], Anthony Ephremides[1,2]
[1]Department of Science and Technology, Linköping University, Sweden, [2]Electrical and Computer Engineering Department, University of Maryland, College Park, USA

NOTE: Corresponding author: Emmanouil Fountoulakis, emmanouil.fountoulakis@liu.se

Abstract – *Future wireless networks will be characterized by heterogeneous traffic requirements. Examples can be low-latency or minimum-througput requirements. Therefore, the network has to adjust to different needs. Usually, users with low-latency requirements have to deliver their demand within a specific time frame, i.e., before a deadline, and they coexist with throughput oriented users. In addition, mobile devices have a limited-power budget and therefore, a power-efficient scheduling scheme is required by the network. In this work, we cast a stochastic network optimization problem for minimizing the packet drop rate while guaranteeing a minimum throughput and taking into account the limited-power capabilities of the users. We apply tools from Lyapunov optimization theory in order to provide an algorithm, named Dynamic Power Control (DPC) algorithm, that solves the formulated problem in real time. It is proved that the DPC algorithm gives a solution arbitrarily close to the optimal one. Simulation results show that our algorithm outperforms the baseline Largest-Debt-First (LDF) algorithm for short deadlines and multiple users.*

Keywords – Deadline-constrained traffic, dynamic algorithms, heterogeneous traffic, Lyapunov optimization, power-efficient algorithms, scheduling.

1. INTRODUCTION

5G and beyond networks are poised to support a mixed set of applications that require different types of services. There are two main categories of applications. The first category includes applications that require bandwidth-hungry services and the second includes delay-sensitive applications. The second category differentiates the current networks from future networks. These applications require low-latency services and increase the need for time-critical networking. In time-critical networking, applications are required to deliver their demands within a specific time duration [1]. In other words, each packet or a batch of packets has a deadline within which data must be transmitted, otherwise, it is dropped and removed from the system [2]. This is connected with the notion of timely throughput. Timely throughput measures the long-term time average number of successful deliveries before the deadline expiration [3, 4]. Each time-critical application belongs to a different category. For example, motion control, smart grid control, and process monitoring belong to the industrial control category. Furthermore, the growing popularity of real-time media applications increases the need for designing networks that can offer services with low latency. Such applications are media production, interactive Virtual Reality (VR), cloud computing, etc, that are under the umbrella of the Tactile Internet [5].

With the pervasiveness of mobile communications, such applications need to perform over wireless devices. In order to achieve reliable communication, the devices have to adapt their power transmission according to channel conditions. However, many devices may have a limited power budget. Therefore, energy-efficient communications have become a very important issue. In this work, we propose a scheduling algorithm that handles a heterogeneous set of users with heterogeneous traffic. In particular, we consider a network with deadline-constrained users and users with minimum-throughput requirements, with a limited-power budget. We provide an algorithm that solves the scheduling problem in real time. We prove that the obtained solution is arbitrarily close to the optimal.

1.1 Related works

Delay-constrained network optimization and performance analysis have been extensively investigated [6]. A variety of approaches have been applied to different scenarios. There is a line of work that considers the control of the maximum number of retransmissions before the deadline expiration. In [7], the authors consider a user transmitting packets over a wireless channel to a receiver. An optimal scheduling scheme is proposed that provides the optimal number of retransmissions for a packet. In [8], the authors consider users with packets with deadlines in a random-access network. They show how the number of maximum retransmissions affects the packet drop rate. In [9], the authors consider a single transmitter that transmits symbols to a receiver. Each symbol has a deadline and a corresponding distortion function. The authors consider the distortion-minimization problem while fulfilling deadline constraints. In [10], the au-

thors consider a system with an Age-of-Information (AoI) oriented user and a deadline-constrained user. The authors provide the distribution of the AoI and the packet drop rate and they examine the interplay between them.
Furthermore, energy and power efficient scheduling schemes for delay-constrained traffic have attracted a lot of attention over the last the few years [11–16]. In [11], the authors consider the minimization of drop rate for users with a limited power-budget. They propose an approximated algorithm that performs in real time. In [12], the authors propose an algorithm that minimizes the time average power consumption while guaranteeing minimum throughput and reducing the queueing delay. They also consider a hybrid multiple access system where the scheduler decides if the transmitter serves a user by orthogonal multiple access or non-orthogonal multiple access. In [13], [14], the authors utilize Markov decision theory to provide an optimal energy-efficient algorithm for delay-constrained users.
Lyapunov optimization theory has been widely applied for developing dynamic algorithms that schedule users with packets with deadlines. In [17–20], the authors consider the rate maximization under power and delay constraints. In [17], the authors consider the power allocation for users with hard-deadline constraints. In [18], the authors consider the rate maximization of non-real-time users while satisfying the packet drop rate for users with packets with deadlines. In [19, 20], consider packets with deadlines for scheduling real-time traffic in wireless environments. A novel approach for minimizing the packet drop rate while guaranteeing stability is provided in [21]. The authors combine tools from Lyapunov optimization theory and Markov decision processes in order to develop an optimal algorithm for minimizing the drop rate under stability constraints. However, the algorithm is able to solve small network scenarios because of the curse of dimensionality problem.
Besides delay-constrained traffic management, throughput-optimal algorithms have been developed over the years. Following the seminal work in [22], many researchers developed different solutions for the throughput-maximization problem by proposing a variety of approaches [23–25]. In [23], the authors consider the throughput-maximization while guaranteeing certain interservice times for all the links. They propose the time-since-last-service metric. They combine the last with the queue length of each user and they propose a max-weight policy based on Lyapunov optimization. In [24], [25], the authors consider the throughput-maximization in networks with dynamic flows. More specifically, in [24], the authors consider a hybrid system with both persistent and dynamic flows. They provide a queue-maximum-weight based algorithm that guarantees throughput-optimality while reducing the latency. In [25], the authors consider a network with dynamic flows of random size and they arrive in random size at the base station. The service times for each flow varies randomly because of the wireless channel. The authors provide a delay-MaxWeight scheduler that has proven its throughput is optimal. Research on scheduling heterogeneous traffic with Ultra-Reliable Low Latency (URLLC) users and enhanced Mobile Brodband (eMBB) has attracted a lot of attention by the community [26]- [33]. In [26], [27], the authors show the benefits of flexible Transmission Time Interval (TTI) for scheduling users with different types of requirements. In [28], the authors propose an algorithm that jointly schedules URLLC and eMBB traffic. They consider a slotted time system in which the slots are divided into mini slots. They consider the frequency and mini-slots allocation over one slot. In [29], the authors consider the resource allocation for URLLC users. They study resource allocation for different scenarios: i) OFDMA system, ii) system that includes retransmissions. In [30], [31], the authors propose a low-complexity algorithm for scheduling URLLC users. The authors in [32] consider the throughput maximization and HARQ optimization for URLLC users. Furthermore, reliable transmission is an important issue of URLLC communications. In [33], the authors consider a network in which multiple unreliable transmissions are combined to achieve reliable latency. The authors model the problem as a constrained Markov decision problem, and they provide the optimal policy that is based on dynamic programming.

1.2 Contributions

In this work, we consider two sets of users with heterogeneous traffic and a limited-power budget. The first set includes users with packets with deadlines and the second set includes users with minimum-throughput requirements. We provide a dynamic algorithm that schedules the users in real time and minimizes the drop rate while guaranteeing minimum throughput and limited-power consumption. The contributions of this work are the following.

- We formulate an optimization problem for minimizing the drop rate with minimum-throughput constraints and time average power consumption constraints.
- We provide a novel objective function for minimizing the drop rate. The objective function does not take into account only if a packet is going to expire or not, but also the remaining time of a packet before its expiration.
- We apply tools from the Lyapunov optimization theory to satisfy the time average constraints: throughput and power consumption.
- The proposed algorithm is proved to provide a solution arbitrarily close to the optimal.
- Simulation results show that our algorithm outperforms the baseline algorithm proposed in [3] for short deadlines and multiple users.

2. SYSTEM MODEL

We consider N users transmitting packets to a single receiver over wireless fading channels. Let $\mathcal{N} \triangleq \{1, \ldots, N\}$ be the set of the total users in the system. Time is assumed to be slotted, and $t \in \mathbb{Z}$ denotes the t^{th} slot.

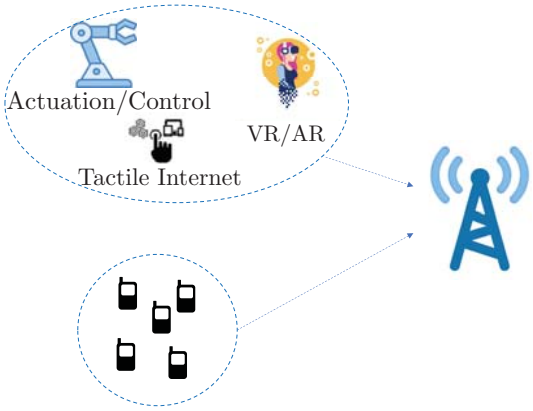


Fig. 1 – Example of the system model with time-critical users with packets with deadlines and users with minimum-throughput requirements.

We consider the users to be synchronized and controlled by a scheduler. In our system model, at most one user can transmit at each time slot. We consider two sets of users. The first set includes users that have arrivals of packets with deadlines. We denote the set of users with packets with deadlines by $\mathcal{R} \subseteq \mathcal{N}$. The second set of users, $\mathcal{U} \subseteq \mathcal{N}$, contains the users that have throughput requirements. We consider that each user $u \in \mathcal{U}$ is saturated and therefore, it has always packet to transmit. Note that $\mathcal{U} \cup \mathcal{R} = \mathcal{N}$ and $\mathcal{U} \cap \mathcal{R} = \emptyset$. An example of our system model is shown in Fig. 1.

For the deadline-constrained users, each packet that arrives in the queue of the users has a deadline by which the packet must be transmitted. Otherwise, it is dropped and removed from the system. We assume that the deadlines of the packets in the same queue are the same. However, deadlines of different queues may vary. We denote the packet deadline of the r^{th} queue with $m_r \in \mathbb{Z}_+, \forall r \in \mathcal{R}$. Let $d_r(t)$ be the number of slots left in the t^{th} slot before the packet that is at the head of queue r expires. Let $Q_r(t)$ be the number of packets in queue r in the t^{th} slot. A packet arrives with probability π_r at every time slot in the queue of user r. Let $\boldsymbol{\alpha}(t) \triangleq \{\alpha_r(t)\}_{r\in\mathcal{R}}$, where $\alpha_r(t) \in \{0, 1\}$, represents the packet arrival process for each user r in the t^{th} slot. The random variables of the packet arrival process are independent and identically distributed (i.i.d.). Let λ_r denote the arrival rate for user r and $\mathbb{E}\{\alpha_r(t)\} = \lambda_r$. Furthermore, at most one packet can be transmitted at each time slot and no collisions are allowed. In each queue of every user $r \in \mathcal{R}$, packets are served in the order that they arrive following the First In First Out (FIFO) discipline.

We assume that the channel state at the beginning of each time slot is known. The channel state remains constant within one slot but it changes from slot to slot. Let $\mathbf{S}(t) \triangleq \{S_i(t)\}_{i\in\mathcal{N}}$ represent the channel state for each user i during slot t. Also, the channel can be either in "Bad" state (deep fading) or in "Good" state (mild fading). The possible channel states of each user i are described by the set $\mathcal{S} \triangleq \{\text{B}, \text{G}\}$, and $S_i(t) \in \mathcal{S}, \forall i \in \mathcal{N}$. For simplicity, we assume that the random variables of the channel process $\mathbf{S}(t)$ are i.i.d. from one slot to the next.

Let $\mathbf{p}(t) \triangleq [p_1(t), \ldots, p_N(t)]$ denote the power allocation vector in the t^{th} slot. We consider a set of discrete power levels $\left\{0, P^{(\text{Low})}, P^{(\text{High})}\right\}$. The required power to have a successful transmission under "Bad" and "Good" channel conditions is denoted by P^{High} and P^{Low}, respectively. At each time slot, the set of selectable power levels $\mathcal{P}_i(t)$ for each user is conditioned on the channel state $S_i(t)$. For example, if the current channel state is "Bad", then $P^{(\text{Low})}$ cannot be selected. Thus, we have

$$p_i(t) \in \begin{cases} \left\{0, P^{(\text{High})}\right\}, \text{if } S_i(t) = \text{B} \\ \left\{0, P^{(\text{Low})}\right\}, \text{if } S_i(t) = \text{G} \end{cases}, \forall i \in \mathcal{N}. \quad (1)$$

Let $\mu_i(t)$ be the power allocation, or packet serving, indicator for the user i in the t^{th} slot, we have

$$\mu_i(t) \triangleq \begin{cases} 1, \text{if } p_i(t) > 0 \\ 0, \text{otherwise} \end{cases}, \forall i \in \mathcal{N}. \quad (2)$$

At most one packet can be transmitted in a timeslot t, i.e., the vector $\mathbf{p}(t)$ has at most one non-zero element. The set of power constraints for $\mathbf{p}(t)$ is then defined by

$$\mathcal{P}(t) \triangleq \left\{\mathbf{p}(t) : \sum_{i=1}^{N} \mathbf{1}_{\{\mu_i(t)=1\}} \leq 1\right\}, \quad (3)$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function. For each user $r \in \mathcal{R}$, a packet is dropped if its deadline has expired. Since the queue follows the FIFO discipline, a packet is dropped under the following conditions: 1) it is at the head of the queue; 2) the remaining number of the slots to serve the packet is 1; and 3) power is not assigned to user r at the current slot. Let $D_r(t)$ be the indicator of the packet drop for user r at time t. The queue evolution for each user $r \in \mathcal{R}$ is described as

$$Q_r(t+1) = \max\left[Q_r(t) - \mu_r(t), 0\right] + \alpha_r(t) - D_r(t), \forall i \in \mathcal{R}. \quad (4)$$

We define the packet drop rate for each user $r \in \mathcal{R}$, the average power consumption for each user $i \in \mathcal{N}$, and the throughput for each user $u \in \mathcal{U}$ as

$$\overline{D}_r \triangleq \lim_{t\to\infty} \overline{D}_r(t), \forall r \in \mathcal{R}, \quad (5)$$

$$\overline{p}_i \triangleq \lim_{t\to\infty} \overline{p}_i(t), \forall i \in \mathcal{N}, \quad (6)$$

$$\overline{\mu}_u = \lim_{t\to\infty} \bar{\mu}_u(t), \forall u \in \mathcal{U}, \quad (7)$$

respectively, where, $\overline{D}_r(\tau) = \frac{1}{t}\sum_{\tau=0}^{t-1} D_r(t)$, $\overline{p}_i(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} p_i(\tau)$, and $\bar{\mu}_u(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} \mu_u(\tau)$. The packet drop

Table 1 – Notation Table.

$\mathcal{N}$	Set of the total users in the system	$\mathcal{P}_i(t)$	Set of selectable power levels of user i
$\mathcal{R}$	Set of the deadline-constrained users	$Z_u(t)$	Length of throughput-dept queue of user u
$\mathcal{U}$	Set of the minimum-throughput requirements users	$f_r(t)$	Cost function for user r
t	t^{th} slot	$\mu_r(t)$	Power allocation indicator of user i
$Q_r(t)$	Number of packets in queue r	$\mathcal{P}(t)$	Set of power constraints for $\mathbf{p}(t)$
π_r	Packet arrival probability of user i	$D_r(t)$	Packet drop indicator of user r
$\alpha_r(t)$	Packet arrival indicator of user r	$\overline{D}_r$	Packet drop rate of user r
m_r	Deadline of packet of user r	$\overline{p}_r$	Average power consumption of user r
$d_r(t)$	Number of slots left before the deadline of user r	$X_r(t)$	Length of virtual queue of user i
$\mathbf{S}(t)$	Channel states	$L(\cdot)$	Quadratic Lyapunov function
$\mathbf{p}(t)$	Power allocation vector	$\Delta(L(\cdot))$	Lyapunov drift
γ_i	Allowed average power consumption for user i	$\boldsymbol{\alpha}(t)$	Packet arrival indicator vector

rate represents the average number of dropped packets per time slot. The average power consumption represents the average of transmit power over all time slots. The throughput represents the average served packets per time slot for each user $u \in \mathcal{U}$.
These metrics are connected and we will show in the following sections how the average power consumption affects the packet drop rate and the throughput.

3. PROBLEM FORMULATION

Our goal is to achieve the minimum drop rate for deadline-constrained users while providing a minimum throughput for each user $u \in \mathcal{U}$ and keeping the average power consumption for every user below a threshold. To this end, we provide the following stochastic optimization problem

$$\min_{\boldsymbol{p}(t)} \quad \sum_{r\in\mathcal{R}} \overline{D}_r \qquad \textit{(8)}\text{a}$$

$$\text{s. t.} \quad \overline{p}_i \leq \gamma_i, \forall i \in \mathcal{N}, \qquad \textit{(8)}\text{b}$$

$$\bar{\mu}_u \geq \delta_u, u \in \mathcal{U}, \qquad \textit{(8)}\text{c}$$

$$\boldsymbol{p}(t) \in \mathcal{P}(t), \qquad \textit{(8)}\text{d}$$

where $\gamma_i \in \left[0, P^{(\text{High})}\right]$ indicates the allowed average power consumption for each user i. Also, δ_u denotes the minimum throughput requirement for each user $u \in \mathcal{U}$. The constraint in *(8)*b ensures that average power consumption of each user i remains below γ_i power units.
The formulation above represents our intended goal which is the minimization of the packet drop rate. However, the objective function in *(8)*a makes the solution approach non-trivial. The decision variable, $\mathbf{p}(t)$ (power allocation), is optimized slot-by-slot for minimization of the objective function that is defined over an infinite time horizon. We have to cope with one critical point: we do not have prior knowledge about the future states of the channel and packet arrivals in the system. Therefore, we are not able to predict the values of the objective function in the future slots in order to decide on the power allocation that minimizes the cost. We aim to design a function whose future values are affected by the current decision and the remaining expiration time of the packets. To this end, we introduce a function incorporating the relative difference between the packet deadline m_r and the number of remaining future slots $(d_r(t)-1)$ before its expiration as described below

$$f_r(t) \triangleq \frac{m_r - (d_r(t)-1)}{m_r} \mathbf{1}_{\{\mu_r(t)=0\}}, \forall r \in \mathcal{R}. \qquad \textit{(9)}$$

The function in *(9)* takes its extreme value, $f_r(t) = 0$, when a packet of user $r \in \mathcal{R}$ is served, or $f_r(t) = 1$ when a packet of user $r \in \mathcal{R}$ is dropped. Therefore, that function takes the same values with those of *(5)* in the extreme cases. In addition, the function in *(9)* assigns the cost according to the remaining time of a packet to expire in the intermediate states, i.e., when a packet is waiting in the queue. The cost increases when there is less time left for serving the packet with respect to the defined deadline. The time average of $f_r(t)$ is

$$\overline{f_r} \triangleq \lim_{t\to\infty} \overline{f}_r(t), \forall r \in \mathcal{R}, \qquad \textit{(10)}$$

where $\overline{f}_r(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} f_r(\tau)$ and

$$f = \sum_{r\in\mathcal{R}} f_r(t). \qquad \textit{(11)}$$

Finally, we formulate a minimization problem by using (10) as shown below

$$\min_{\boldsymbol{p}(t)} \quad \sum_{r\in\mathcal{R}} \overline{f}_r \qquad (12)\text{a}$$

$$\text{s. t.} \quad \overline{p}_i \le \gamma_i, \forall i \in \mathcal{N}, \qquad (12)\text{b}$$

$$\bar{\mu}_u \ge \delta_u, u \in \mathcal{U}, \qquad (12)\text{c}$$

$$\boldsymbol{p}(t) \in \mathcal{P}(t). \qquad (12)\text{d}$$

4. PROPOSED APPROXIMATE SOLUTION

The problem in *(12)* includes time average constraints. In order to satisfy these constraints, we aim to develop a policy that uses techniques different from standard optimization methods based on static and deterministic models. Our approach is based on Lyapunov optimization theory [34].

In particular, we apply the technique developed in [35] and further discussed in [34] and [36] in order to develop a policy that ensures that the constraints in *(12)*b and *(12)*c are satisfied.

Each inequality constraint in *(12)*b and *(12)*c is mapped to a virtual queue. We show below that the power constraint and minimum throughput constraints problems are transformed into queue stability problems.

Before describing the motivation behind the mapping of average constraints in *(12)*b and *(12)*c to virtual queues, let us recall one basic theorem that comes from the general theory of stability of stochastic processes [37]. Consider a system with K queues. The number of unfinished jobs of queue i is denoted by $q_k(t)$, and $\mathbf{q}(t) = \{q_k(t)\}_{k=1}^{K}$. The Lyapunov function and the Lyapunov drift are denoted by $L(\mathbf{q}(t))$ and $\Delta(L(\mathbf{q}(t))) \triangleq E\{L(\mathbf{q}(t+1)) - L(\mathbf{q}(t))|\mathbf{q}(t)\}$ respectively [37]. Below we provide the definition of the Lyapunov function [37].

Definition 1 (Lyapunov function): A function $L : \mathbb{R}^K \to \mathbb{R}$ is said to be a Lyapunov function if it has the following properties

- $L(\mathbf{x}) \ge 0, \forall \mathbf{x} \in \mathbb{R}^K$,
- It is non-decreasing in any of its arguments,
- $L(\mathbf{x}) \to +\infty$, as $||\mathbf{x}|| \to +\infty$.

Theorem 1 (Lyapunov Drift)**.** *If there exist positive values B, ϵ such that for all time slots t we have $\Delta(L(\mathbf{q}(t)) \le B - \epsilon \sum_{k=1}^{K} q_k(t)$, then the system $\mathbf{q}(t)$ is strongly stable.*

The intuition behind Theorem 1 is that if we have a queueing system, and we provide a scheduling scheme such that the Lyapunov drift is bounded and the sum of the length of the queues are multiplied by a negative value, then the system is stable. Our goal is to find a scheduling scheme for which the inequality of Theorem 1 holds for our application.

Let $\{X_i(t)\}_{i\in\mathcal{N}}$ and $\{Z_u(t)\}_{u\in\mathcal{U}}$ be the virtual queues associated with constraints *(12)*b and *(12)*c, respectively. We update each virtual queue $X_i(t)$ at each time slot t as

$$X_i(t+1) = \max\left[X_i(t) - \gamma_i, 0\right] + p_i(t), \qquad (13)$$

and each virtual queue $Z_u(t)$ as

$$Z_u(t+1) = \max\left[Z_u(t) - \mu_u(t), 0\right] + \delta_u. \qquad (14)$$

Process $X_i(t)$ can be viewed as a queue with "arrivals" $p_i(t)$ and "service rate" γ_i. Process $Z_u(t)$ can be also viewed as a queue with "arrivals" δ_u and "service rate" $\mu_u(t)$.

We will show that the average constraints in *(12)*b and *(12)*c are transformed into queue stability problems. Then, we develop a dynamic algorithm and we prove that the algorithm satisfies Theorem 1 and achieves stability.

Lemma 1. *If $X_i(t)$ and $Z_u(t)$ are rate stable*[1]*, then the constraints in (12)*b *and (12)*c *are satisfied.*

Proof. See Appendix B. □

Note that strong stability implies all of the other forms of stability [34, Chapter 2] including the rate stability. Therefore, the problem is transformed into a queue stability problem. In order to stabilize the virtual queues $X_i(t), \forall i \in \mathcal{N}$ and $Z_u(t), \forall u \in \mathcal{U}$, we first define the Lyapunov function as

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{i\in\mathcal{N}} X_i^2(t) + \frac{1}{2}\sum_{u\in\mathcal{R}} Z_u^2(t), \qquad (15)$$

where $\mathbf{\Theta}(t) = [\{X_i(t)\}_{i\in\mathcal{N}}, \{Z_u(t)\}_{u\in\mathcal{U}}]$, and the Lyapunov drift as

$$\Delta(\mathbf{\Theta}(t)) \triangleq \mathbb{E}\left\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\right\}. \qquad (16)$$

The above conditional expectation is with respect to the random channel states and the arrivals.

To minimize the time average of the desired cost $f_r(t)$ while stabilizing the virtual queues $X_i(t)$, $\forall i \in \mathcal{N}$, $Z_u(t), \forall u \in \mathcal{U}$, we use the *drift-plus-penalty* minimization approach introduced in [36]. The approach seeks to minimize an upper bound on the following drift-plus-penalty expression at every slot t

$$\Delta(\mathbf{\Theta}(t)) + V\sum_{r\in\mathcal{R}} \mathbb{E}\left\{f_r(t)|\mathbf{\Theta}(t)\right\}, \qquad (17)$$

where $V > 0$ is an "importance" weight to scale the penalty. An upper bound for the expression in *(17)* is shown below

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) + V\sum_{r\in\mathcal{R}} \mathbb{E}\{f_r(t)|\mathbf{\Theta}(t)\} \le B \\ + \sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)(p_i(t) - \gamma_i)|\mathbf{\Theta}(t)\} \\ + \sum_{r\in\mathcal{R}} \mathbb{E}\{Z_u(t)(\delta_u - \mu_u(t))|\mathbf{\Theta}(t)\} \\ + V\sum_{r\in\mathcal{R}} \mathbb{E}\{f_r(t)|\mathbf{\Theta}(t)\}, \qquad (18)\end{aligned}$$

[1] A discrete time process $Q(t)$ is *rate stable* if $\lim_{t\to\infty} \frac{Q(t)}{t} = 0$ with probability 1 [34].

where $B < \infty$ and $B \geq \frac{1}{2}\sum_{i\in\mathcal{N}}\mathbb{E}\{p_i^2(t)+\gamma_i^2(t)|\mathbf{\Theta}(t)\} + \frac{1}{2}\sum_{r\in\mathcal{R}}\mathbb{E}\{\delta_r^2+\mu_r^2(t)|\mathbf{\Theta}(t)\} + \frac{1}{2}\sum_{r\in\mathcal{R}}\mathbb{E}\{\alpha_r^2(t)+\mu_r^2(t)|\mathbf{\Theta}(t)\}$. The complete derivation of the above bound can be found in Appendix A.

4.1 Min-Drift-Plus-Penalty Algorithm

We observe that the power allocation decision at each time slot does not affect the value of B. The minimum DPC algorithm observes the virtual queue backlogs of the virtual queues, the actual queue, and the channel states and makes a control action to solve the following optimization problem

$$\min_{\boldsymbol{p}(t)} \quad \sum_{i\in\mathcal{N}} X_i(t)(p_i(t)-\gamma_i) + \sum_{r\in\mathcal{R}} Z_u(t)(\delta_u-\mu_u(t)) + V\sum_{r\in\mathcal{R}} f_r(t) \qquad (19)\text{a}$$

$$\text{s. t.} \quad \boldsymbol{p}(t)\in\mathcal{P}(t). \qquad (19)\text{b}$$

Lemma 2. *The optimal solution to problem (19) minimizes the upper bound of the drift-plus-penalty expression given in the right-hand-side of (18).*

Proof. See Appendix C. □

Theorem 2 (Optimality of DPC algorithm and queue stability)**.** *The DPC algorithm guarantees that the virtual and the actual queues are strongly stable and therefore, according to Lemma 1, the time average constraints in (12)*b *and (12)*c *are satisfied. In particular, the time average expected value of the queues is bounded as*

$$\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\left(\sum_{i\in\mathcal{N}}\mathbb{E}\left\{X_i(t)\right\}+\sum_{u\in\mathcal{U}}\mathbb{E}\left\{Z_u(t)\right\}\right)\leq \frac{B+V\left(f^*(\epsilon)-f^{opt}\right)}{\epsilon}. \qquad (20)$$

In addition, the expected time average of function $f(t)$ is bounded as

$$\lim_{t\to\infty}\sup\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{f(\tau)\}\leq f^{opt}+\frac{B}{V}. \qquad (21)$$

Proof. See Appendix D. □

We summarize the steps of the DPC algorithm that solves the power control problem in *(19)* in Algorithm 1.

Algorithm 1: DPC

```
Input constant V, Initialization:
  X_i(0) = 0, γ_i, ∀i ∈ N, Z_u(0) = 0, ∀u ∈ U.
for t = 1 : ... do
    MinObj ← ∞
    for i = 1 : (|N| + 1) do
        p_i(t) ∈ P(t), Calculate f_r(t), ∀r ∈ R
        Obj ← V Σ_{r∈R} f_r(t) + Σ_{r∈R} X_r(t)(p_i(t) − γ_r)
              + Σ_{u∈U} Z_u(t)(δ_u − μ_i(t))
        if MinObj>Obj then
            p'(t) ← p(t)
            MinObj ← Obj
    p(t) ← p'(t)
    X_j(t+1) ← max[X_j(t) − γ_j, 0] + p_j(t), ∀j ∈ N
    Z_u(t+1) = max[Z_u(t) − μ_u(t), 0] + δ_u, ∀u ∈ U
```

In step 1, we initialize the importance factor V and the length of virtual queues $X_i(0), \forall i\in\mathcal{N}$, and $Z_u(0), \forall u\in\mathcal{U}$. We try all the possible power allocations from the set $\mathcal{P}$ in step 4, and we find the corresponding value of the objective function in step 6. In step 7, we check if the candidate power allocation gives a smaller value of the objective so far. In steps $11-12$, we updated the virtual queues. After the search, we obtain the solution, $\boldsymbol{p}'(t)$, for which the objective function takes its minimum value, $MinObj$.

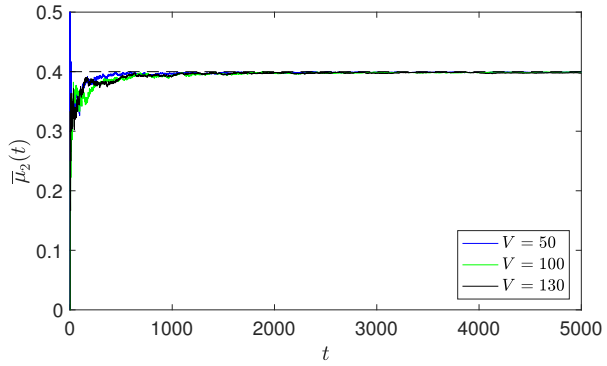
5. SIMULATION RESULTS

In this section, we provide results that show the performance of the DPC algorithm regarding the packet drop rate and the convergence time for the time average constraints, i.e, throughput, and average power consumption constraints. We use a MATLAB environment to perform our simulations. For the time average performance, we run each algorithm for 10^6 slots.
We first show the results of a system with two users. Our goal is to show the performance of DPC for different values of the importance factor V. Second, we compare our proposed algorithm with a baseline algorithm called LDF algorithm proposed in [3].

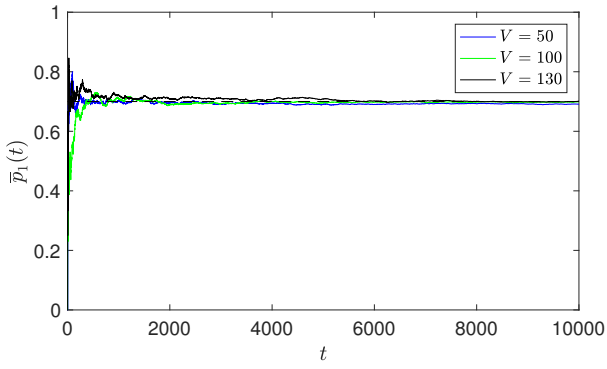
5.1 Performance and convergence of DPC algorithm for different values of V

In Fig. 2, we provide results for a system with two users; user 1: deadline-constrained user, user 2: user with minimum-throughput requirements. The average power thresholds are $\gamma_1 = 0.7$ and $\gamma_2 = 0.65$ for user 1 and user 2, respectively. The minimum-throughput requirements for user 2 is $\delta_2 = 0.4$ packets/slot, and the deadlines of the packet of user 1 is $m = 10$ slots.

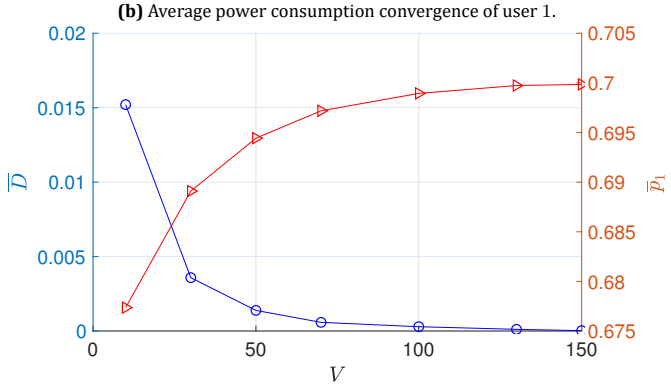
The probability of the channel being in "Good" state and in "Bad" state is 0.4 and 0.6, respectively. The high level power and the low level power is $P^{\text{High}} = 2$ power units



(a) Throughput-constraints convergence of user 2.



(b) Average power consumption convergence of user 1.



(c) Trade-off between packet drop rate and average power consumption of user 1 for different values of V.

Fig. 2 – DPC algorithm performance for different values of V. $\lambda_1 = 0.5$, $\gamma_1 = 0.7, \gamma_2 = 0.65, \delta_2 = 0.4, m = 10$.

and $P^{\text{Low}} = 1$ power units, respectively.

In Fig. 2a, we show the convergence of the algorithm regarding the minimum-throughput constraints for different values of the importance factor V. We observe that as the value of V increases, the time convergence increases as well. However, we observe that after approximately 2500 slots, the algorithm converges and the minimum-throughput requirements are satisfied. In Fig. 2b, we provide results for the converge of the DPC algorithm regarding the average power consumption constraints of user 1. The algorithm converges after approximately 8000 slots for each value of V. The probability of the channel of user 1 being in "Good" state is 0.4. Therefore, the user needs to transmit with a high power level for a large portion of the time and that affects the average power consumption. Thus, the average power consumption constraint with $\gamma_2 = 0.7$ is a tight constraint and the algorithm needs more time to converge.

In Fig. 2c, we show the trade-off between the packet drop rate and average power consumption of user 1. We observe that as the value of V increases the average power consumption increases and approaches threshold γ_1. For $V = 10$, we observe that the average power consumption is far from the threshold. In this case, the value of the virtual queue that corresponds to the average power consumption is larger than the cost function for a large period of time because the importance factor is relatively small. Therefore, the DPC seeks to minimize the larger term of the objective function that is the value of the virtual queue. On the other hand, as we increase V, the cost function is weighted more and therefore, the DPC algorithm seeks to minimize the cost function which is the most weighted term in the objective function. However, the average power consumption remains always below threshold γ_1.

5.2 DPC vs LDF

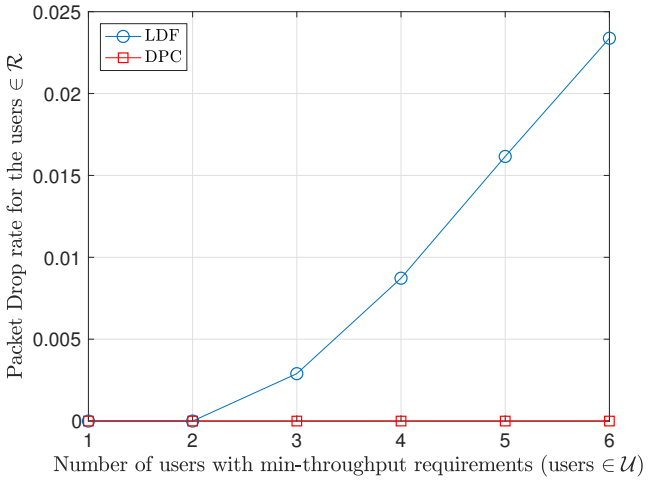
In this subsection, we compare the performance of our algorithm with that of LDF. The LDF algorithm allocates power to the user with the *largest-debt*. Algorithm LDF selects, at each time slot t, the node with the highest value of $y_i(t)$, where $y_i(t)$ is the throughput debt and is defined as

$$y_i(t+1) = tq_i - \sum_{\tau}^{t} \mu_i(t), \qquad (22)$$

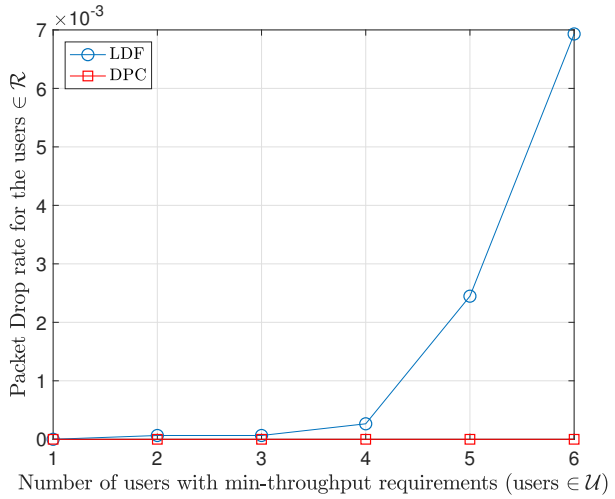
where q_i is the throughput requirements for user i. In our case, for the users with throughput requirements, $q_i = \delta_i$. For the deadline-constrained users, q_i will be equal to the percentage of the desired served packets for users with deadlines. For example, if our goal is to achieve a zero drop rate we set $q_i = \lambda_i$. However, this is not always feasible, i.e., zero drop rate and satisfaction of the throughput constraints. Therefore, in this case, we get a higher drop rate and lower throughput. Note that the LDF algorithm does not account for the average power constraints. It was shown in [3] that LDF is throughput optimal when the problem is feasible for systems with users with throughput requirements.

In this set-up, we consider one user with packets with deadlines and a set with multiple users with minimum-throughput requirements. The probability that the channel is in "good" state is equal to 0.9 for all the users. In order to observe a fair comparison between the algorithm, we consider that the average power threshold is 2 for all the users. Therefore, the average power constraint for every user is always satisfied. The arrival rate for user 1 is $\lambda_1 = 0.35$ packets/slot.

In Fig. 3, we compare the performance of the algorithms regarding the packet drop rate as the number of users with minimum-throughput requirements increases. In Fig. 3a, the deadline for the packets of user 1 is $m = 10$. We observe that the DPC algorithm outperforms the



(a) Packet deadline: $m = 10$.



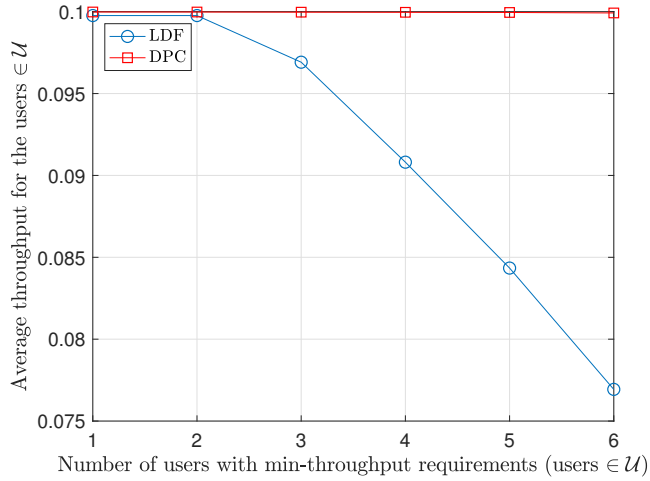
(b) Packet deadline: $m = 30$.

Fig. 3 – DPC vs LDF. Drop rate comparison for different values of packet deadline m.

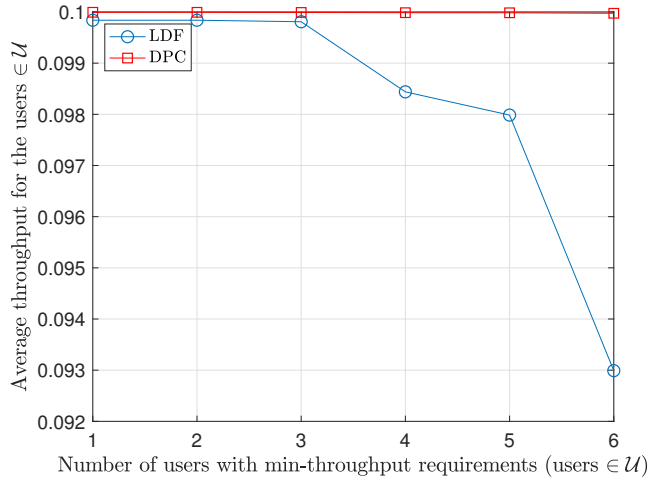
LDF algorithm in terms of drop rate as the number of minimum-throughput requirements increases. In Fig. 3b, the deadline for the packets of user 1 is $m = 30$. We observe that the performance of LDF has been improved. However, the DPC outperforms LDF in this case as well. We observe that LDF is more sensitive on the size of the deadline of the packets.

In Fig. 4, we compare the performance of the algorithm regarding the average total throughput of users with minimum-throughput requirements. In Fig. 4a, we show results for packets deadline that is $m = 10$. Also in this case, we observe that the DPC algorithm outperforms the LDF. However, for larger deadlines, the performance of the LDF is improved, as shown in Fig. 4b.

In Fig. 5, we provide results for the convergence time of throughput requirements of one user. In this set-up, we consider a system with one user with packets with deadlines and six users with minimum-throughput requirements. In the previous results that are shown in Figures. 4 and 3, we observe that as we increase the value of m, we get a better performance of the LDF algorithm. In the system of which the results are shown in Fig. 5, we set a very large value to the deadlines of



(a) Packet deadline: $m = 10$.



(b) Packet deadline: $m = 30$.

Fig. 4 – DPC vs LDF. Throughput comparison for different values of packet deadline m.

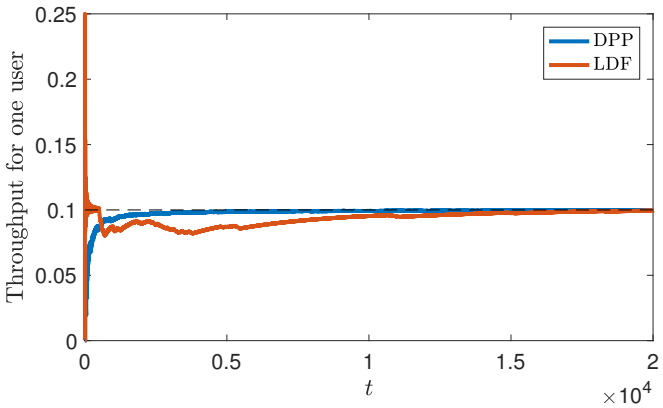


Fig. 5 – DPC vs LDF. Min-throughput requirements convergence. Packets deadline: $m = 100$ slots.

user 1. We observe that the DPC algorithm converges much earlier than the LDF algorithm. However, both algorithms converge after many slots. This explains the phenomenon of worst performance of LDF for small values of m. Since the LDF algorithm allocates power to the users with largest throughput-debt, it does not consider the remaining slots for each packet. Therefore, when the deadline, m, is small, the packets expire before the algorithm

converges in terms of throughput requirements and the drop rate increases.

6. CONCLUSIONS AND FUTURE WORK

In this work, we considered heterogeneous traffic with two sets of users. The first set contains users with packets with deadlines, and the second contains users with minimum-throughput requirements, all with a limited power budget. We considered the packet drop rate minimization with minimum-throughput guarantees. A dynamic algorithm was provided that solves the scheduling problem in real time. We proved that our scheduling scheme provides a solution arbitrarily close to the optimal. Simulation results show that the proposed algorithm outperforms the baseline algorithm, LDF, when the deadlines are short, and it is faster in terms of convergence. An interesting direction, for future work, would be the assumption that packets of the same user can have different deadlines, and there are multiple power levels.

7. ACKNOWLEDGMENTS

This work has been partially supported by the Swedish Research Council (VR) and by CENIIT.

Appendices

A. UPPER BOUND ON THE LYAPUNOV DRIFT OF DPC

Using the fact that $(\max[Q-b,0]+A)^2 \leq Q^2+A^2+b^2+2Q(A-b)$, we rewrite *(13)*, *(14)*, as

$$X_i^2(t+1) \leq X_i^2(t) + p_i^2 + \gamma_i^2 + 2X_i(t)(p_i(t)-\gamma_i), \quad (23)$$

$$Z_u^2(t+1) \leq Z_u^2(t) + \delta_u^2 + \mu_u^2(t) + 2Z_u(t)(\delta_u - \mu_u(t)), \quad (24)$$

respectively. Rearranging the terms in *(23)* and *(24)*, dividing them by two, and taking the summations, we obtain

$$\sum_{i\in\mathcal{N}} \frac{X_i^2(t+1)-X_i^2(t)}{2} \leq \sum_{i\in\mathcal{R}} \frac{p_i^2(t)+\gamma_i^2}{2} + \sum_{i\in\mathcal{N}} X_i(t)(p_i(t)-\gamma_i), \quad (25)$$

$$\sum_{u\in\mathcal{U}} \frac{Z_u^2(t+1)-Z_u^2(t)}{2} \leq \sum_{u\in\mathcal{U}} \frac{\delta_u^2+\mu_u^2(t)}{2} + \sum_{u\in\mathcal{U}} Z_u(t)(\delta_u-\mu_u(t)). \quad (26)$$

Taking the conditional expectations in *(25)* and *(26)*, and adding them together, we obtain the bound for the Lyapunov drift in *(18)*. To prove that B is bounded, we have to find an example and a scheduling scheme for which B takes its maximum value that is bounded. We consider the following set-up:

- The scheduler allocates power at every time slot with the maximum power level,
- $\gamma_i = P^{\text{High}}, \forall i \in \mathcal{N}$,
- $\delta_u = 1,\ \forall u \in \mathcal{U}$,
- $\lambda_i = 1,\ \forall i \in \mathcal{N}$.

Then, for the above scheduling scheme, we set $B = \frac{1}{2}\sum_{i\in\mathcal{N}} \mathbb{E}\{p_i^2(t)+\gamma_i^2(t)|\mathbf{\Theta}(t)\} + \frac{1}{2}\sum_{r\in\mathcal{R}} \mathbb{E}\{\delta_r^2+\mu_r^2(t)|\mathbf{\Theta}(t)\} + \frac{1}{2}\sum_{r\in\mathcal{R}} \mathbb{E}\{\alpha_r^2(t)+\mu_r^2(t)|\mathbf{\Theta}(t)\} = \frac{1}{2}|\mathcal{N}+1|(P^{\text{High}})^2 + \frac{1}{2}|\mathcal{R}+1| + \frac{1}{2}|\mathcal{R}+1| < \infty$. We observe that even in the scenario in which we take the maximum values of γ_i, δ_u, and λ_i, B is bounded.

B. PROOF OF LEMMA 1

Proof. Using the basic sample property [34, Lemma 2.1, Chapter 2], we have

$$\frac{X_i(t)}{t} - \frac{X_i(0)}{t} \geq \frac{1}{t}\sum_{\tau=0}^{t-1} p_i(\tau) - \frac{1}{t}\sum_{\tau=0}^{t-1}\gamma_i, \quad (27)$$

$$\frac{Z_u(t)}{t} - \frac{Z_u(0)}{t} \geq \frac{1}{t}\sum_{\tau=0}^{t-1} \delta_u - \frac{1}{t}\sum_{\tau=0}^{t-1}\mu_u(t). \quad (28)$$

Therefore, if $X_i(t)$ and $Z_u(t)$ are rate stable [2], so that $\frac{X_i(t)}{t} \to 0, \forall i \in \mathcal{N}$, and $\frac{Z_u(t)}{t} \to 0, \forall u \in \mathcal{U}$, with probability 1, then constraints *(12)*b and *(12)*c are satisfied with probability 1 [38]. □

C. PROOF OF LEMMA 2

Proof. Let $\mathbf{p}(t)$ represent any, possibly randomized, power allocation decision made at slot t. Suppose that $\mathbf{p}^*(t)$ is the optimal solution to problem *(19)*, and under action $\mathbf{p}^*(t)$ the value of $f_i(t)$ yields $f_i^*(t)$ and that of $\mu_u(t)$, $\mu_u^*(t)$. Then, we have

$$\begin{aligned} &Vf^*(t) + \sum_{i\in\mathcal{N}} X_i(t)(p^*(t)-\gamma_i) + \sum_{u\in\mathcal{U}} Z_u(t)(\delta_u - \mu_u^*(t)) \\ &\leq Vf(t) + \sum_{i\in\mathcal{N}} X_i(t)(p(t)-\gamma_i) + \sum_{u\in\mathcal{U}} Z_u(t)(\delta_u-\mu_u(t)). \end{aligned} \quad (29)$$

[2] A discrete time process $Q(t)$ is strongly stable if $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{|Q(\tau)|\} < \infty$, [34].

Taking the conditional expectations of *(29)*, we have the result as

$$V\mathbb{E}\{f^*(t)|\mathbf{\Theta}(t)\} + \sum_{i\in\mathcal{N}} \mathbb{E}\{X_i|\mathbf{\Theta}(t)\}(t)(p_i^*(t) - \gamma_i) + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)|\mathbf{\Theta}(t)\}(\delta_u - \mu_u^*(t)) \leq V\mathbb{E}\{f(t)|\mathbf{\Theta}(t)\} + \sum_{i\in\mathcal{N}} \mathbb{E}\{X_i|\mathbf{\Theta}(t)\}(t)(p_i(t) - \gamma_i) + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)|\mathbf{\Theta}(t)\}(\delta_u - \mu_u(t)).$$

□

D. PROOF OF THEOREM 2

Proof. Suppose that a feasible policy ω exists, i.e., constraints *(12)*b and *(12)*c are satisfied. Furthermore, assume that, for the ω policy, the following holds

$$\mathbb{E}\{p_i(t) - \gamma_i\} \leq -\epsilon, \quad (30)$$
$$\mathbb{E}\{\delta_u - \mu_u(t)\} \leq -\epsilon, \quad (31)$$
$$\mathbb{E}\{f^*(\epsilon)\} = f^*(\epsilon), \quad (32)$$

where $f^*(\epsilon)$ is a sub-optimal solution. Applying *(30)* and *(31)* into *(18)*, we obtain

$$\mathbb{E}\{L(\mathbf{\Theta}(t+1))\} - \mathbb{E}\{L(\mathbf{\Theta}(t))\} + V\mathbb{E}\{f(t)\} \leq B - \epsilon\left(\sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)\} + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)\}\right) + Vf^*(\epsilon),$$

taking $\epsilon \to 0$ and the sum over $\tau = 0, \ldots, t-1$ we obtain

$$\frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{f(\tau)\} \leq \frac{-\mathbb{E}\{L(\mathbf{\Theta}(t))\} + \mathbb{E}\{L(\mathbf{\Theta}(0))\} + Bt}{Vt} + f^{\text{opt}}, \quad (33)$$

taking $t \to \infty$, we obtain

$$\lim_{t\to\infty} \sup \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{f(\tau)\} \leq f^{\text{opt}} + \frac{B}{V}. \quad (34)$$

That concludes the second part of Theorem 2. In order to prove the stability of the queues, we manipulate *(33)*

$$\left(\sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)\} + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)\}\right) \leq \frac{B}{\epsilon} - \frac{\mathbb{E}\{L(\mathbf{\Theta}(t+1))\} - \mathbb{E}\{L(\mathbf{\Theta}(t))\}}{\epsilon} - \frac{V(f^*(\epsilon) - f(t))}{\epsilon}. \quad (35)$$

By taking the sum over $\tau = 0, \ldots, t-1$ and divide by t, we obtain

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\left(\sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)\} + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)\}\right) \leq \frac{B}{\epsilon} - \frac{\mathbb{E}\{L(\mathbf{\Theta}(t))\} - \mathbb{E}\{L(\mathbf{\Theta}(0))\}}{t\epsilon} + \frac{V(f^*(\epsilon) - f(t))}{\epsilon}, \quad (36)$$

neglecting the negative term and taking $t \to \infty$, we have

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left(\sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)\} + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)\}\right) \leq \frac{B + V(f^*(\epsilon) - f(t))}{\epsilon}. \quad (37)$$

Consider that $\mathbb{E}\{f(t)\} \geq f^{\text{opt}}$, we obtain the final result as

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left(\sum_{i\in\mathcal{N}} \mathbb{E}\{X_i(t)\} + \sum_{u\in\mathcal{U}} \mathbb{E}\{Z_u(t)\}\right) \leq \frac{B + V(f^*(\epsilon) - f^{\text{opt}})}{\epsilon}. \quad (38)$$

This shows that the queues are strongly stable for $\epsilon > 0$.

□

REFERENCES

[1] F. Alriksson, L. Boström, J. Sachs, E. Wang, and A. Zaidi, "Critical IoT connectivity," Ericsson review, vol. 102, pp. 52–64, 2020.

[2] I. Hou and P. R. Kumar, "Packets with deadlines: A framework for real-time wireless networks," Synthesis Lectures on Communication Networks, vol. 6, no. 1, pp. 1–116, 2013.

[3] I. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in Proc. IEEE INFOCOM, pp. 486–494, 2009.

[4] S. Lashgari and A. S. Avestimehr, "Timely throughput of heterogeneous wireless networks: Fundamental limits and algorithms," IEEE Trans. Inf. Theory, vol. 59, no. 12, pp. 8414–8433, 2013.

[5] O. Holland, E. Steinbach, R. V. Prasad, Q. Liu, Z. Dawy, A. Aijaz, N. Pappas, K. Chandra, V. S. Rao, S. Oteafy et al., "The IEEE 1918.1 "tactile internet" standards working group and its standards," Proc. of the IEEE, vol. 107, no. 2, pp. 256–279, 2019.

[6] Y. Cui, V. K. N. Lau, R. Wang, H. Huang, and M. Shunqing Zhang, "A survey on delay-aware resource control for wireless systems–large deviation theory, stochastic lyapunov drift, and distributed stochastic learning," IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1677–1701, Mar. 2012.

[7] N. Master and N. Bambos, "Power control for packet streaming with head-of-line deadlines," Performance Evaluation, vol. 106, pp. 1–18, 2016.

[8] N. Nomikos, N. Pappas, T. Charalambous, and Y.-A. Pignolet, "Deadline-constrained bursty traffic in random access wireless networks," in IEEE Proc. SPAWC, pp. 1–5, 2018.

[9] A. Faridi and A. Ephremides, "Distortion control for delay-sensitive sources," IEEE Trans. Inf. Theory, vol. 54, no. 8, pp. 3399–3411, 2008.

[10] E. Fountoulakis, T. Charalambous, N. Nomikos, A. Ephremides, N. Pappas, "Information Freshness and Packet Drop Rate Interplay in a Two-User Multi-Access Channel", to appear, in Proc. IEEE ITW, 2021.

[11] E. Fountoulakis, N. Pappas, Q. Liao, A. Ephremides, and V. Angelakis, "Dynamic power control for packets with deadlines," in IEEE Proc. GLOBECOM, pp. 1–6, 2018.

[12] M. Choi, J. Kim, and J. Moon, "Dynamic power allocation and user scheduling for power-efficient and delay-constrained multiple access networks," IEEE Trans. Wireless Commun, vol. 18, no. 10, pp. 4846–4858, 2019.

[13] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar, "An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel," IEEE J. Sel. Areas Commun., vol. 26, no. 4, pp. 732– 742, May 2008.

[14] M. Goyal, A. Kumar, and V. Sharma, "Power constrained and delay optimal policies for scheduling transmission over a fading channel," in Proc. IEEE INFOCOM, vol. 1, May 2003, pp. 311–320.

[15] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," IEEE Trans. Mobile Comput., vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[16] A. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling over a fading channel with energy and deadline constraints," IEEE Trans. Wireless Commun., vol. 5, no. 3, pp. 630–641, Mar. 2006.

[17] A. Ewaisha and C. Tepedelenlioglu, "Power control and scheduling under hard deadline constraints for on-off fading channels," in Proc. IEEE WCNC, March 2017, pp. 1–6.

[18] A. E. Ewaisha and C. Tepedelenlioglu, "Optimal power control and scheduling for real-time and non-real-time data," IEEE Trans. Vehic. Tech, vol. 67, no. 3, pp. 2727–2740, 2017.

[19] S. ElAzzouni, E. Ekici, and N. Shroff, "Is deadline oblivious scheduling efficient for controlling real-time traffic in cellular downlink systems?" in Proc. IEEE INFOCOM, pp. 49–58, 2020.

[20] K. S. Kim, C.-P. Li, I. Kadota, and E. Modiano, "Optimal scheduling of real-time traffic in wireless networks with delayed feedback," Allerton, pp. 1143–1149, 2015.

[21] M. J. Neely and S. Supittayapornpong, "Dynamic Markov decision policies for delay constrained wireless scheduling," IEEE Trans. Automatic Control, vol. 58, no. 8, pp. 1948–1961, 2013.

[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in Proc. IEEE CDC, pp. 2130–2132, 1990.

[23] B. Li, R. Li, and A. Eryilmaz, "Throughput-optimal scheduling design with regular service guarantees in wireless networks," IEEE/ACM Trans. Net., vol. 23, no. 5, pp. 1542–1552, 2014.

[24] X. Lan, Y. Chen, and L. Cai, "Throughput-optimal H-QMW scheduling for hybrid wireless networks with persistent and dynamic flows," IEEE Trans. Wireless Commun., vol. 19, no. 2, pp. 1182– 1195, 2019.

[25] B. Sadiq and G. De Veciana, "Throughput optimality of delay-driven maxweight scheduler for a wireless system with flow dynamics," Allerton, pp. 1097–1102, 2009.

[26] L. You, Q. Liao, N. Pappas, and D. Yuan, "Resource optimization with flexible numerology and frame structure for heterogeneous services," IEEE Commun. Letters, vol. 22, no. 12, pp. 2579–2582, 2018.

[27] E. Fountoulakis, N. Pappas, Q. Liao, V. Suryaprakash, and D. Yuan, "An examination of the benefits of scalable TTI for heterogeneous traffic management in 5G networks," in Proc. WiOpt, May 2017, pp. 1–6.

[28] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of urllc and embb traffic in 5G wireless networks," IEEE/ACM Trans. Net., vol. 28, no. 2, pp. 477–490, 2020.

[29] A. Anand and G. de Veciana, "Resource allocation and harq optimization for urllc traffic in 5G wireless networks," IEEE J. Sel. Areas Commun., vol. 36, no. 11, pp. 2411–2421, 2018.

[30] A. Karimi, K. I. Pedersen, and P. Mogensen, "Low-complexity centralized multi-cell radio resource allocation for 5g urllc," in Proc. IEEE WCNC, 2020, pp. 1–6.

[31] N. B. Khalifa, V. Angilella, M. Assaad, and M. Debbah, "Lowcomplexity channel allocation scheme for urllc traffic," IEEE Trans. Commun., 2020.

[32] A. Avranas, M. Kountouris, and P. Ciblat, "Throughput maximization and ir-harq optimization for urllc traffic in 5G systems," in Proc. IEEE ICC , 2019, pp. 1–6.

[33] A. Destounis, G. S. Paschos, J. Arnau, and M. Kountouris, "Scheduling urllc users with reliable latency guarantees," in Proc. WiOpt, 2018, pp. 1–8.

[34] M. J. Neely, Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan & Claypool, 2010.

[35] M. J. Neely, "Energy optimal control for time-varying wireless networks," IEEE Trans. Inf. Theory, vol. 52, no. 7, pp. 2915–2934, July 2006.

[36] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," Foundations and Trends in Networking, vol. 1, no. 1, 2006.

[37] S. Meyn and R. L. Tweedie, Markov Chains and Stochastic Stability. 2nd ed. New York, NY, USA: Cambridge University Press, 2010.

[38] M. J. Neely, "Queue Stability and Probability 1 Convergence via Lyapunov Optimization," ArXiv e-prints, Aug. 2010. [Online]. Available: https://arxiv.org/abs/1008.3519

AUTHORS

Emmanouil Fountoulakis (Graduate Student Member, IEEE) received a Diploma degree in electronic and computer engineering from the Technical University of Crete, Chania, Greece, in 2016.

He is currently pursuing a Ph.D. degree with the Department of Science and Technology, Linköping University, Sweden. His research interests include age-of-information, low-latency communications, energy-efficient resource allocation, applied queueing theory, and stochastic control.

Nikolaos Pappas (Member, IEEE) received a B.Sc. degree in computer science, an M.Sc. degree in computer science, a B.Sc. degree in mathematics, and a Ph.D. degree in computer science from the University of Crete, Greece, in 2005, 2007, 2012, and 2012, respectively.

From 2005 to 2012, he was a Graduate Research Assistant with the Telecommunications and Networks Laboratory, Institute of Computer Science, Foundation for Research and Technology-Hellas, and a Visiting Scholar with the Institute of Systems Research, University of Maryland at College Park, College Park, MD, USA. From 2012 to 2014, he was a Post-Doctoral Researcher with the Department of Telecommunications, Supélec, France.

Since 2014, he has been with Linköping University, Norrköping, Sweden, as a Marie Curie Fellow (IAPP), where he is currently an Associate Professor in mobile telecommunications with the Department of Science and Technology. His main research interests include the field of wireless communication networks with emphasis on the stability analysis, energy harvesting networks, network-level cooperation, age-of-information, network coding, and stochastic geometry.

From 2013 to 2018, he was an Editor of IEEE COMMUNICATIONS LETTERS. He is currently an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE/KICS JOURNAL OF COMMUNICATIONS AND NETWORKS, and IEEE OPEN JOURNAL OF COMMUNICATIONS SOCIETY. He is also a Guest Editor of IEEE INTERNET OF THINGS JOURNAL for the Special Issue Age of Information and Data Semantics for Sensing, Communication and Control Co-Design in IoT.

Anthony Ephremides (Life Fellow, IEEE) received a Ph.D. degree in electrical engineering from Princeton University in 1971. Since 1971, he has been with the University of Maryland, College Park, MD, USA. He holds the Cynthia Kim Professorship of information technology at the Electrical and Computer Engineering Department, University of Maryland, where he is currently a Distinguished University Professor.

He has a joint appointment at the Institute for Systems Research, of which he was among the founding members, in 1986. He has authored several hundred articles, conference presentations, and patents.

His research interests lie in the areas of communication systems and networks and all related disciplines, such as information theory, control and optimization, satellite systems, queueing models, signal processing, and so on. He is especially interested in wireless networks, energy efficient systems, and the new notion of age of information.

RECONFIGURATION ALGORITHMS FOR HIGH PRECISION COMMUNICATIONS IN TIME SENSITIVE NETWORKS: TIME-AWARE SHAPER CONFIGURATION WITH IEEE 802.1QCC

Ahmed Nasrallah[1], Venkatraman Balasubramanian[2], Akhilesh S. Thyagaturu[3], Martin Reisslein[4], Hesham ElBakoury[5]
[1,2,3,4]Arizona State University, School of Electrical, Computer, and Energy Engineering, 650 East Tyler Mall, Tempe, AZ 85287-5706, USA, [3]Intel Corporation, 5000 W. Chandler Blvd., Chandler, AZ 85226, USA, [5]Formerly with Huawei Technologies Co., now Self-employed, 4211 Norwalk Dr., CC-102, San Jose, CA 95129, USA

NOTE: Corresponding author: Martin Reisslein, reisslein@asu.edu

Abstract – *As new networking paradigms emerge for different networking applications, e.g., cyber-physical systems, and different services are handled under a converged data link technology, e.g., Ethernet, certain applications with mission critical traffic cannot coexist on the same physical networking infrastructure using traditional Ethernet packet-switched networking protocols. The IEEE 802.1Q Time Sensitive Networking (TSN) Task Group is developing protocol standards to provide deterministic properties, i.e., eliminates non-deterministic delays, on Ethernet based packet-switched networks. In particular, the IEEE 802.1Qcc, centralized management and control, and the IEEE 802.1Qbv, Time-Aware Shaper (TAS), can be used to manage and control Scheduled Traffic (ST) streams with periodic properties along with Best-Effort (BE) traffic on the same network infrastructure. We investigate the effects of using the IEEE 802.1Qcc management protocol to accurately and precisely configure TAS enabled switches (with transmission windows governed by Gate Control Lists (GCLs) with Gate Control Entries (GCEs)) ensuring ultra-low bounded latency, zero packet loss, and minimal jitter for ST TSN traffic. We examine both a centralized network/distributed user model (hybrid model) and a fully-distributed (decentralized) 802.1Qcc model on a typical industrial control network with the goal of maximizing the number of ST streams.*

Keywords – Cyber-physical systems, low-latency traffic, protocol adaptation, reconfiguration, Time Sensitive Networking (TSN).

1. INTRODUCTION

1.1 Motivation

IEEE 802.1 Time Sensitive Networking (TSN) provides a standardized framework of tools for providing deterministic Ultra-Low Latency (ULL), e.g., for industrial control applications, automotive networking, smart grid applications, and avionics communication systems [11, 22, 30, 34, 57, 63, 88]. In particular, the IEEE 802.1Qbv Time Aware Shaper (TAS) has received extensive attention as a key tool for achieving a deterministic ULL network service. The TAS operation requires careful planning of the synchronized time cycles [79, 85, 91] and the gate times that are allocated to the Scheduled Traffic (ST) and the unscheduled Best-Effort traffic (BE). The TAS parameter settings specifying the timing characteristics (cycle time, gate slot allocations) are also commonly referred to as the Qbv schedule or the TAS schedule. For a given static networking scenario, the TAS operation with a properly configured Qbv schedule can ensure the deterministic ULL required by demanding industrial and automotive applications [8, 27, 40, 59, 62, 77, 84] [1].

Modern network scenarios often involve dynamic changes with varied use cases, such as changes in the network nodes and network topology, or the traffic pattern. For instance, nodes or links may be dynamically added or removed. Or, nodes may inject additional traffic flows or traffic flows may terminate, or the latency requirements of flows may change dynamically. Such dynamic changes have been included in the use cases defined by the IEC/IEEE 802.1 TSN TG [10, 89]. In a typical industrial environment, sensors that periodically or sometimes sporadically send ambient measurements to a local gateway require certain Quality of Service (QoS) guarantees [6, 16, 31, 42, 64]. In such a volatile and dynamic environment, new machinery that requires prioritized execution (e.g., emergency cooling procedures or maintenance tasks for network traffic tests) may be brought onto the factory floor. To deal with such scenarios, the Time-Aware Shaper (TAS) Gate Control Lists (GCLs) in coordination with the Network Management Entities (NMEs), e.g., Centralized Network Configuration (CNC), have to adapt to changing environment conditions by judiciously applying reconfiguration such that stream deadlines and QoS are satisfied.

Generally, in such dynamic networking scenarios, applying only admission control will clearly guarantee (in accordance with a traffic shaper) the QoS metrics of the admitted flows. However, for a given static network configuration, the total number of admissible streams may be well below the number of streams that seek network service. Therefore, adding a dynamic reconfiguration strategy to manage and configure the network appears to be a plausible and attractive solution that intuitively should lower capital and operational expenditures as it mitigates

[1]A preliminary abridged version of this study appeared in the IEEE Globecom 2019 workshop paper [60]. This journal article substantially extends the prior workshop paper, as explained in Section 1.2.

the over-provisioning of network resources. The general idea for such an allocation scheme is to control network access in a timely and orderly fashion such that a maximum number of streams can be effectively serviced.

Our objective therefore is to maximize the number of admitted flows (i.e., tasks or streams) in such a dynamically changing and volatile environment whilst keeping the TSN QoS metric guarantees. In this paper, we focus on the IEEE 802.1Qbv [2] enhancements and design a reconfiguration framework taking inspiration from the IEEE 802.1Qcc [3] standards for managing, configuring, and reconfiguring a TSN network.

In IEEE 802.1Qbv, a TAS time slot (corresponding to a GCE and also referred to as slot time) is defined as the portion of the cycle time (CT, which corresponds to the GCL); TAS time slots are allocated to high-priority ST traffic. In our model, the switch/controller computes the TAS time slot for all admitted streams as follows. Essentially, as streams get registered, we keep track of the available remaining capacity, which we set initially to the maximum available capacity on each egress port until the load (which depends on the ST slot size and the cycle time is negative, i.e., oversubscribed link). Such a link oversubscription invokes a procedure call that increases the slot time (by a step size of 1%, or more fine-grained increments) until the remaining load is positive. This procedure is iteratively called until all registered streams and the new stream are appropriately registered with a sufficient ST slot time to transmit all frames during a single appropriately sized CT.

Our proposed TAS configuration/reconfiguration is designed for the centralized (hybrid) model and for the fully-distributed configuration model. In the "hybrid" model, the CNC is utilized for configuration exchanges and network side management, as explained in more detail in Section 3. In the distributed approach, the GCE slot parameters are configured in a distributed manner by the switches as per the distributed algorithm/procedure explained in Section 4. For brevity we refer to the centralized network/distributed user model (hybrid model) also as the centralized model or the centralized topology. We refer to the fully-distributed (decentralized) model also as the decentralized model or the decentralized topology.

1.2 Related work

We first note that general performance evaluation strategies for TAS have been explored in [39, 50, 73] and we follow these strategies in our study. Raagaard et al. [51, 76] have presented a heuristic scheduling algorithm that reconfigures TAS switches according to runtime network conditions. Feasible schedules are computed and forwarded using a configuration agent (composed of a Centralized User Configuration (CUC) and Centralized Network Configuration (CNC)). Raagaard et al's model places emphasis on the schedule computation complexity for appearing and disappearing synthetic flows in a fog computing platform. Complementary to this approach, we develop comprehensive centralized and distributed reconfiguration frameworks based on firm bandwidth computation strategies that execute at run-time. Further, we conduct a comprehensive performance evaluation of our two frameworks considering common packet flow QoS metrics for both high-priority ST and low-priority BE traffic.

The proposed approach by Nayak et al. [67] exploits the logical centralization paradigm of SDN with real-time traffic to achieve optimal scheduling and routing. Integer Linear Programming (ILP) formulations were used to solve the combined problem of routing and scheduling time triggered traffic. Two main proposals for routing are given, namely $i)$ scheduling and path-sets routing, and $ii)$ scheduling and fixed-path routing whereby the ILP formulations are used to find near optimal flow to time-slot allocations. However, the ILP does not scale well with the number of flows, does not provide schedules at runtime speeds, and does not work well with dynamic flow configuration (or reconfiguration). To enhance the architecture proposed by Nayak et al. [67], an augmentation is proposed in [68] that incrementally adds time sensitive flows to the scheduler making the proposed approach reconfiguration capable. Additionally, Nayak et al. [65,66] provide an analysis and evaluation to the problem of flow-span and routing protocol (Equal Cost Multi-Path, and Shorted Path) on transmission scheduling. Further routing refinements have been studied in [9, 48, 49, 66, 72].

Focusing on in-vehicular networks, Hackel et al. [38] have proposed a SDN based TSN framework that performs reconfiguration using the Stream Reservation Protocol (SRP) as a means to register and allocate resources for TSN streams. The TSN with SDN is evaluated with two TSN switches and two clients (a sources and sink). In contrast, we provide extensive evaluation for larger network topologies and sources. Using OpenFlow and openPowerLink, Herlich et al. [41] have provided a proof-of-concept model that highlights the advantages of TSN with SDN and real-time Ethernet protocol. While the model shows promising advantages in theory, only a coarse-grained evaluation was presented that, in contrast to our evaluation, does not examine stream admission rates and TSN QoS. Focusing on remote monitoring and telemetry, Kobzan et al. [46] have presented a solution concept and implementation of an SDN based TSN architecture using IEEE 802.1Qcc. However, the concept is provided without any empirical evaluation. To the best of our knowledge, there are no prior detailed studies on a fluctuating volatile source or a dynamic stream resource allocation and admission control policy in conjunction with a network reconfiguration policy being executed while flows are carried in a TAS time scheduled network. We provide a comprehensive design and evaluation of an SDN based TSN model that bases the specification on the standardization given by the IEEE 802.1Q standard.

Vlk et al. [87] have proposed a simple hardware enhancement of a switch along with a relaxed scheduling constraint that increases schedulability and throughput of

the time-triggered traffic but maintains the deterministic nature and timeliness guarantees in a TSN network. Several related scheduling refinements that are orthogonal to the reconfiguration studied in this article have been examined in [26,40,43,53,74,93]. We note for completeness that multicast for TSN has been studied in [80, 92], while our focus is on unicast traffic.

This article extends the prior conference paper [60], which provided a brief preliminary overview of the decentralized and centralized reconfiguration models, but did not provide the specific operational details, nor detailed performance evaluations. This present journal article provides the full operational details as well as comprehensive performance evaluations.

1.3 Contributions

We comprehensively evaluate the performance of TAS for reconfigurations in the hybrid and fully distributed models with respect to network deployment parameters, such as the time period for the Gate Control List (GCL) to repeat (whereby the duration of one GCL repetition corresponds to the CT), the gating ratio proportion, i.e., Gate Control Entry (GCE) proportion, to control the delay perceived at the receiving end, the signaling impact on ST and BE classes, and the packet loss rate experienced at the receiving end. In particular, we make the following contributions:

i) We design a CNC interface for a TSN network to globally manage and configure TSN streams, including admission control and resource reservation.

ii) We integrate the CNC in the control plane with TAS in the data plane to centrally manage and shape traffic using the CNC as the central processing entity for flow schedules as more flows are added.

iii) We modify and test the model to operate in a distributed fashion, i.e., the signaling is conducted in-band and the control plane processing is conducted at the individual distributed switches.

iv) We evaluate each design approach for a range of numbers of streams with different TAS parameters. We show results for admission ratios, network signaling overhead, and QoS metrics.

1.4 Organization

This article is organized as follows. Section 2 provides background information and an overview of related work on the 802.1 TSN standardization, focusing on the enhancements to ST as well as centralized management and configuration. Section 3 shows the complete top-down design of the CNC (hybrid model) and the main components that achieve ultra-low latencies and guaranteed QoS for a multitude of ST streams. Similarly, Section 4 shows the approach used in implementing the decentralized (fully distributed) TAS reconfiguration model. The simulation setup as well as main parameters and assumptions are given in Section 5 and results are presented in Section 5.2 and Section 5.3. Finally conclusions and future work are outlined in Section 6.

2. BACKGROUND: IEEE 802.1 TIME SENSITIVE NETWORKING

2.1 IEEE 802.1Qbv: Time Aware Shaper (TAS)

TAS's main operation is to schedule critical traffic streams in reserved time-triggered windows. In order to prevent lower priority traffic, e.g., BE traffic, from interfering with the ST transmissions, ST windows are preceded by a so-called guard band. TAS is applicable for Ultra-Low Latency (ULL) requirements but needs to have all time-triggered windows synchronized, i.e., all bridges from sender to receiver must be synchronized in time [79, 85]. TAS utilizes a gate driver mechanism that opens/closes according to a known and agreed upon time schedule for each port in a bridge. In particular, the Gate Control List (GCL) represents Gate Control Entries (GCEs), i.e., a sequence of on and off time periods that represent whether a queue is eligible to transmit or not.

The frames of a given egress queue are eligible for transmission according to the GCL, which is synchronized in time through the 802.1AS time reference. Frames are transmitted according to the GCL/GCE and transmission selection decisions. Each individual software queue has its own transmission selection algorithm, e.g., strict priority queuing. Whereby, a software queue is the queue before the NIC hardware queue takes ownership of the currently forwarded frame in an 802.1 switch. Overall, the IEEE 802.1Qbv transmission selection transmits a frame from a given queue with an open gate if: (i) The queue contains a frame ready for transmission, (ii) higher priority traffic class queues with an open gate do *not* have a frame to transmit, and (iii) the frame transmission can be completed before the gate closes for the given queue. Note that these transmission selection conditions ensure that low-priority traffic is allowed to *start* transmission only if the transmission will *be completed* by the start of the ST window for high-priority traffic. Thus, this transmission selection effectively enforces a "guard band" to prevent low-priority traffic from interfering with high-priority traffic [30].

2.2 IEEE 802.1Qcc: centralized management and configuration

IEEE 802.1Qcc [3] provides a set of tools to globally manage and control the network. In particular, IEEE 802.1Qcc enhances the existing Stream Reservation Protocol (SRP) with a User Network Interface (UNI) which is supplemented by a Centralized Network Configuration (CNC) node. The UNI provides a common method of requesting layer 2 services. Furthermore, the CNC interacts with the switch UNI to provide a centralized means for perform-

ing resource reservation, scheduling, and other types of configuration via a remote management protocol, such as NETCONF [25] or RESTCONF [12]; hence, 802.1Qcc is compatible with the IETF YANG/NETCONF data modeling language.

The IEEE 802.1Qcc standard specifies three models for configuring the Time-Aware Shaper (TAS) gating schedules (GCL/GCE timing): a fully-centralized model, a centralized network/distributed user model (hybrid model), and a fully-distributed configuration model. The centralized model greatly eases control and configuration messages sent across the network and can precisely configure TAS schedules due to having the complete knowledge of the network and the full capabilities of each bridge. However the centralized model suffers from common disadvantages, such as a single-point of failure, relatively large capital/operational (CapEx/OpEx) expenditures (as the centralized control may be superfluous in a small-scale network [15]), and adding unnecessary complexity to a small-scale network.

Compared to the centralized network/distributed user model (hybrid model), the fully centralized model does not add any benefits for the reconfiguration approach towards enhancing the resource allocation and QoS nor does it allow better deterministic forwarding. The main usage for the CUC is to take into account the application's complex timing and computation requirements for industrial applications which is out of scope for our evaluation. Rather, our focus is on the reconfiguration for proper resource allocation. Therefore, we focus on the centralized network/distributed user model (hybrid model) form of the centralized model in this study.

A fully-distributed configuration model (e.g., SRP over MRP or RAP over LRP) may be attractive for some networks. The fully-distributed configuration model avoids the added complexity and single point of failure of a centralized management entity. Moreover, Chen et al. [15] have argued that the centralized configuration models can be an over-design for real-time applications with relaxed latency requirements (order of magnitude of milliseconds). Chen et al. have also argued that the distributed model is more scalable. (However, studies of the fully distributed model with RAP over LRP targeted typically applications with relatively relaxed latency requirements.)

In the absence of a Centralized Network Configuration (CNC) node, the TSN Task Group (TG) specifies the IEEE 802.1CS (Link-Local Registration Protocol, LRP) [29] standard for registration and distribution of application configuration parameters over point-to-point links targeting newly published TSN features. A legacy protocol, such as the Stream Reservation Protocol (SRP) [1] which is primarily used for Audio-Video Bridging (AVB) applications, is intended to serve as the main resource reservation and admission control protocol. However, extending and porting the SRP to be utilized for bridges that support TAS will not suffice since bandwidth reservation cannot directly apply TAS's time slot reservation natively. Therefore, the Resource Allocation Protocol, IEEE 802.1Qdd (RAP) [15], has been proposed to apply a distributed resource reservation that can exchange TSN features.

3. HYBRID MODEL DESIGN AND FRAMEWORK CONSIDERATIONS

This section presents our design methodology and main signaling framework for the centralized network/distributed user model (hybrid model). Our main goals behind designing the CNC are given by the following constraints. Additionally, the CNC can be logically or physically connected to the data plane with in-band or out-of-band management links. With in-band communication under the hybrid model, only one switch is physically connected to the CNC; thus, signaling packets between the switches and CNC affect data traffic similar to the distributed approach, but the CNC still functions as the centralized configuration. For the hybrid model evaluations in this study, we consider out-of-band communication, i.e., all switches are physically attached to the CNC.

1. Our focus is mainly on stream based network adaptation. By this technique, fluctuating streams (already registered streams and new incoming streams) and their requirements can be accommodated by the network dynamically based on a single remote procedure call to the CNC.
2. We identify and execute flow requirements by populating the registration table. The control plane resource orchestration is purely carried out by monitoring existing flows which have been satisfied.
3. We conduct resource allocation based on the stream network resource utilization.

Our main assumption to accurately apply admission control and, consequently, reconfiguration, is that each source must define a flow in terms of total resources needed (governed by the bandwidth requirements) and the total time needed for the resource to be used (which in our traffic model is termed the resource utilization time). Essentially, the CNC uses this information (which is tagged in the Ethernet frame header) to determine whether a stream (flow) is admitted or rejected.

3.1 Core components

Our design is split into two layers, Control Plane and Data Plane, following the decoupling SDN paradigm, thereby inheriting the benefits of the orthogonality of the two planes, as shown in Fig. 1.

3.1.1 Configuration module

The configuration module is the main component that interacts with the registered flows and network components. It includes the global stream registration table which records all approved streams transmitting in

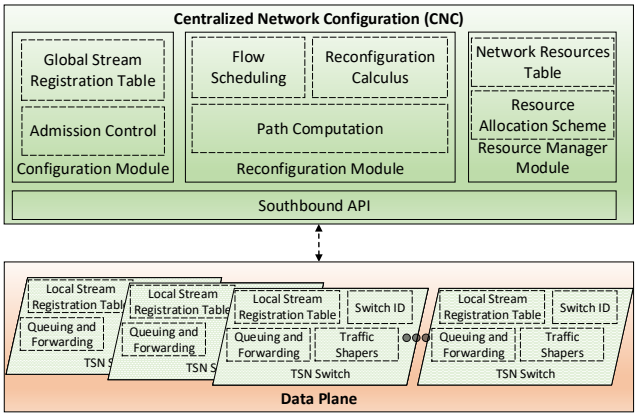


Fig. 1 – Network Management Entity Framework for TSN Switches: Centralized Network Configuration (CNC) is used to send and receive Control Data Traffic (CDT), which we define as the signaling traffic, e.g., the UNI information to and from the CNC and switches/sources or LLDP discovery packets, to configure routing segments and network resources.

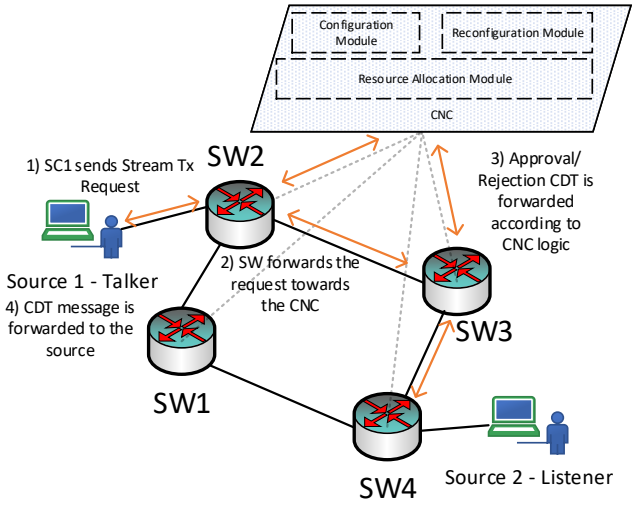


Fig. 2 – Centralized Model Example: Source 1 sends a CDT stream request to its gateway. The gateway forwards it to its governing CNC. The CNC decides if the stream will be serviced according to the source UNI. All switches in the explicit path for the stream are notified if the stream is accepted. Otherwise, the gateway is alerted of the rejection. Lastly, the gateway forwards it back to the source which prompts the source to start sending data ST traffic in the next available cycle (if approved).

the network (i.e., currently utilizing network resources (bandwidth)), and the admission control element that encapsulates and decapsulates CDT headers and forwards the information to the necessary module/element.

Global stream registration table The source streams (devices/users) make a Remote Procedural Call (RPC) via the stream registration interface for providing information that can be mapped as a unique tuple structure identification $< FlowID, BridgeGateway >$. Upon receiving the registration packet, i.e., Control Data Traffic (CDT), the CNC determines whether the new stream can be accepted in its stream table. To guarantee the QoS for all registered streams, admission control principles are applied to all streams according to the stream's path, required network resources, and available resources. Fig. 2 shows an example where the source sends a CDT stream request to the gateway switch, which is then forwarded to the CNC for admission control and resource reservation.

Admission Control The admission control element is the first element that the new streams interacts with. The admission control element in the configuration module globally manages all streams transmitting in the TSN domain governed by the CNC. The admission control element extracts the necessary information from the CDT packet and forwards the information according to the CDT type. The CNC applies several steps to decide whether to accept or reject the stream transmission request.

1. The CNC checks the destination address(es) of the stream and consults its resource manager module for network resources available on the new stream's path, which is computed based on the path computation element within the CNC.

2. According to the bandwidth required for the new stream (calculated at the bridge gateway for the new stream as the stream packet rate multiplied by the packet size and divided by the ST slot time), all links on the path are checked to see if enough bandwidth is available for the new stream.

3. In the event that not enough resources are available, the CNC applies the TAS reconfiguration module to identify the bottleneck link(s) and to check whether the gating ratio can be increased for that specific traffic class whose current resource utilization will not exhaust the resources by being added to the TAS slot reservation.

3.1.2 Reconfiguration module

The reconfiguration module includes the flow scheduling element (for our network model, the Time-Aware Shaper (TAS) is used in the data plane), the reconfiguration calculus element which optimizes flow registration according to each stream's total resource utilization and flow deadlines, and finally the path computation element which defines the path for all streams according to the QoS constraint.

Flow scheduling The flow scheduling element currently takes the Time-Aware Shaper into consideration. Due to the TAS's requirements on time synchronization between network components (switches, hosts, etc.), the CNC follows the same principle of scheduling flows according to a known timescale (initially set to be 50 μs in our network model). The CNC then passes on this time synchronization information to the TSN enabled switches within its domain. Any approved streams will transmit frames according to the time scale specified by the flow scheduler in the CNC.

Reconfiguration calculus In addition to centrally managing resources and providing admission control policies to the network, the CNC can invoke the TAS reconfiguration strategy with the goal of borrowing BE time slots for pending ST traffic streams. This element consults the resource manager module on the bottleneck link and checks whether the added stream will oversubscribe the link. The TAS reconfiguration incrementally (1% of total CT) increases the traffic class slot time and reserves it for the new stream.

Path computation For large scale and complex LAN/MAN topologies, it is often required to supplement streams with equal cost paths in the event of a path disruption (e.g., link failure, stream saturation, and explicit congestion). The CNC's path computation element is tasked with finding such paths as a fail-over approach to avoid any violations to any stream's QoS. Presently, our model has a rudimentary application of path computation, i.e., it is defined statically for all core network components (shortest path), since our main emphasis in this study is on reconfiguration based on stream characteristics as defined by the source.

3.1.3 Resource manager module

The resource manager module centrally manages all network resources within the CNC's domain. It includes the network resource table that records all streams' usage of resources, and the resource allocation scheme element to which we delegate the task of calculating the required network resources for a given stream according to an allocation scheme.

Network resource table To remove certain overheads on the configuration module, the network resource table operates in tandem with the global stream registration table to accurately determine the required network resources (mainly bandwidth for our traffic model). It classifies streams based on periodic stream properties. Any stream that has been approved by the CNC has a record attached to it in the network resource table.

Resource allocation scheme Several allocation schemes can be implemented for all traffic classes defined in the network. For periodic streams, the time slot given by the flow scheduler (according to the TAS Cycle Time and number of traffic classes) and the data rate defined by the source is used to calculate the required bandwidth for each link on the path to the destination (i.e., sink).

3.1.4 Data plane

The data plane contains all core switches. Any TSN switch interfaced by the CNC is given a switch ID and has a local stream registration table. The remaining switch elements compose the forwarding and queuing operation with several traffic shapers (802.1Qbv TAS in our network model).

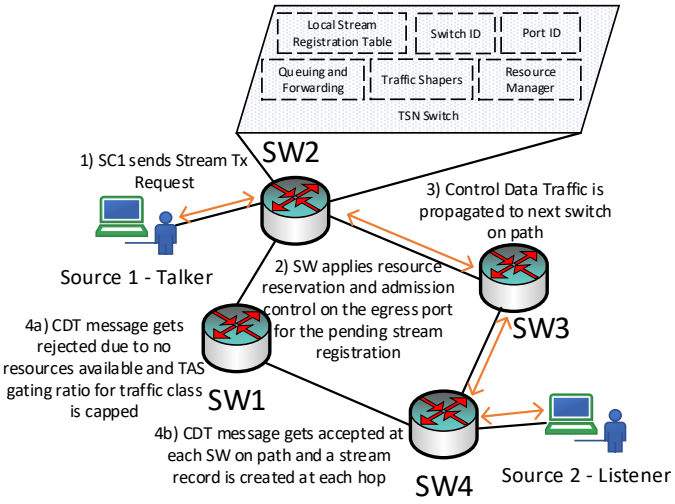


Fig. 3 – A TSN fully distributed configuration model example illustrating the general strategy and logic of each TSN switch with TAS support. In the absence of a CNC to centrally manage network parameters, each switch performs admission control and resource reservation (according to the TAS time slot load) and propagates the information to the next hop on the stream path. A single rejection on one hop terminates the forwarding of the CDT, and sends another CDT on the reverse path indicating the stream rejection outcome. If all switches on the path accept the stream, then the source is notified of the stream acceptance outcome and can begin forwarding in the next TAS cycle. In our model, CDT traffic has higher priority than non-CDT traffic (including ST). The formal definition of the CDT traffic is left for future work.

Local stream registration table This data plane registry contains the subset of source streams that are established for the corresponding bridge gateway and attached sources to each port. The CNC delegates some control to the bridge gateway to instruct and alert sources of any new network conditions and explicit changes.

Traffic shaper — Time-Aware Shaper (TAS) The TAS is the main shaping and scheduling mechanism that controls the gating schedules for all the traffic classes within the TSN domain (which is considered to be equivalent to the CNC domain). All bridges are synchronized to the same gating schedule GCL Cycle Time (CT) given by the CNC's flow schedule element (CT indicates the time period for the GCL to repeat).

4. DECENTRALIZED MODEL DESIGN AND FRAMEWORK CONSIDERATIONS

This section presents our design methodology and framework for the TAS reconfiguration in the decentralized (fully distributed) model. Our current proposed architecture generally follows the steps enumerated below and illustrated in Fig. 3. Our description focuses on the additions to the design of RAP over LRP, e.g., TAS slot computation/reservations.

1. At each egress port (Port Identifier, PID), the TSN switch maintains a local stream registration table that includes information, such as flow ID, gateway (i.e., the first bridge that a talker is connected to), destination address(es), the traffic injection rate per GCL cycle time, and the calculated port bandwidth requirement. The traffic injection rate is not computed, rather the traffic injection rate is reported by the source (talker) to the network devices. It mainly indicates the bandwidth requirements of a stream. Bandwidth for a bridge egress port needed for a stream is computed using the ST injection rate (or ST rate), the average packet size, and the bridge TAS timing configuration (e.g., the CT and current traffic class slot time). This information is carried and communicated between bridges using the CDT packet type identifier (or message type).

2. A source (talker) can send a stream transmission request, i.e., a CDT message of type Stream Transmission Request, to register its stream and to use the TSN service for ST.

3. Each switch maintains a resource manager module for each port. If the newly incoming stream is accepted (due to available resources and TAS slot space). The TAS slot size for a specific traffic class is governed by the CT and traffic class gating ratio (in time). The TAS ST slot can be configured/reconfigured according to stream requests and terminations. The stream registration message is then propagated towards the next switch, and a map is maintained for the stream (and any other streams) pending approval.

4. If accepted by the last switch, then the stream registration record is added to the local stream registration table, and bandwidth resources are allocated for the stream and TAS slot space is modified (if necessary) on the reverse path. The main reason for allocating the resources in the reverse path is as follows. If we allocate the resources in the forward direction but a switch in the next hop rejects the stream (due to lack of resources), then we have to release the resources reserved earlier for the stream. Therefore, we avoid the allocation until all hops provide assurance that the stream will be accommodated.

5. Each switch receiving the pending registration message adds the stream record to its local table, allocates the necessary resources and TAS slot reservation, and propagates the registration message towards the source gateway.

6. The source gateway receives the pending stream registration message and similarly allocates the resources and finally sends an approval granted message towards the source, which prompts the source to start sending data in the next available TAS cycle.

4.1 Core components

This section outlines the main components required to successfully implement stream admission control and resource reservation within switches that support the TAS traffic shaper in a distributed fashion. Fig. 4 illustrates the typical registration/reservation procedure for all streams within the TSN domain.

4.1.1 Admission control

The admission control element extracts the necessary information from the CDT packet and forwards the information according to the CDT type. The switch forwarding mechanism applies several steps to accurately decide whether to accept or reject the stream transmission request. Note that the stream transmission request corresponds to a CDT request. The switch consults the resource manager module to check if enough resources (bandwidth) is available for the new stream. In particular, a given stream's bandwidth requirement is calculated by multiplying the ST injection rate with the average packet size and dividing by the current ST slot size. Note that the traffic class TAS slot time is the time during which the TAS gate is open to transmit frames belonging to the considered class. Also note that all GCEs are executed during each CT. If the CT is smaller than the aggregate of the GCEs, then we need to either increase the CT or reject streams that cause the exceedance of the CT.

4.1.2 Flow scheduling

This element currently takes the Time-Aware Shaper into consideration. Due to the TAS's requirements on time synchronization between network components (switches, hosts, etc.), all switches/hosts follow the TAS GCL timescale cycle time (e.g., 50 μs). Depending on the number of supported traffic classes, the TAS cycle time can be divided into appropriate slots for each traffic class load. The TAS CT is divided among all the traffic classes (in our evaluation model, we consider two traffic classes, BE and ST). Currently, in our evaluations, the CT is initially predefined to 50 microseconds. Note that the CT could be changed/configured dynamically. The dynamic adaptation of the CT with respect to new stream additions, application specifications, or other events is a topic for future research.

4.1.3 Stream registration table

In our evaluations, stream creation follows a Poisson process with a prescribed stream generation rate π. Different scenarios with varying mean stream lifetimes (durations) enable analysis of how reconfiguration works in multiple settings. The stream registration table contains the characteristics of the source streams that are established for the corresponding bridge egress port. Each record gets populated (if accepted) on the reverse path taken by the stream's registration message (after reaching the destina-

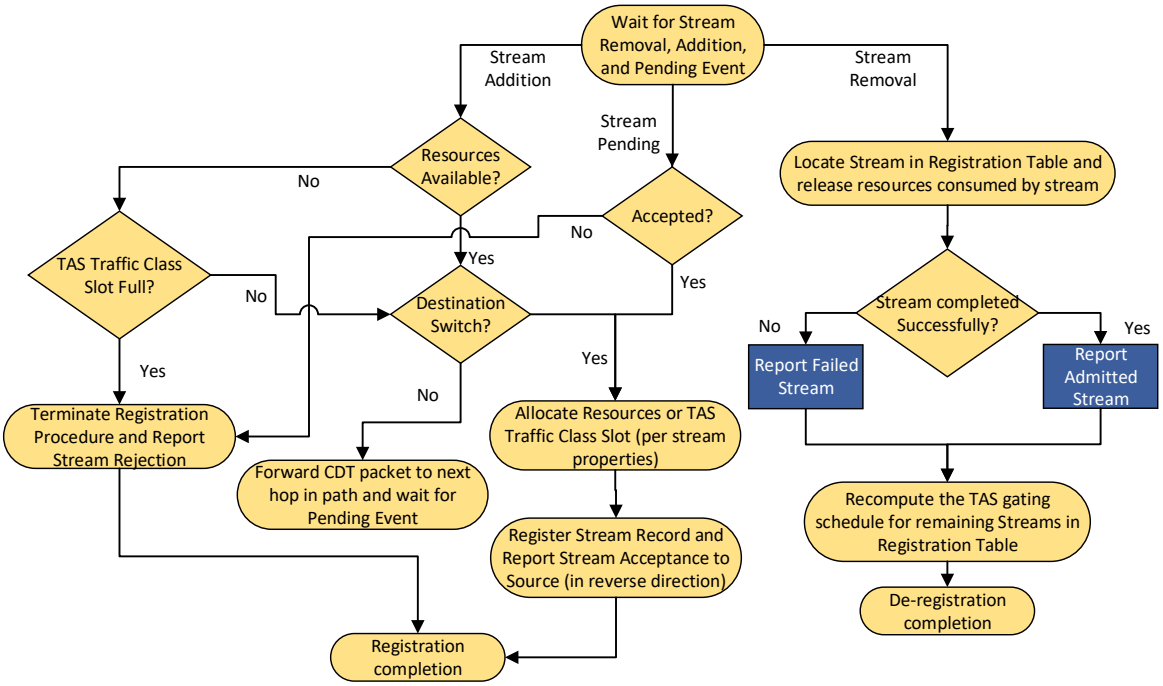


Fig. 4 – The main logical steps performed by each switch along the stream's path are shown to apply stream registration and reservation. Each switch generally waits for an event (addition, removal, or pending) for each stream. For instance, a stream removal is usually based on the resource utilization time (stream lifetime) that was specified at stream establishment. The bridges that allocated resources for the stream can remove the stream after the stream lifetime has expired. For the cases of stream addition or pending, the event is the CDT message received (whether in the forward or reverse direction). Towards completing (finalizing, confirming) a stream reservation (registration), the pending event is the event when a CDT message is received in the reverse direction where each switch (not the last switch) waits for the approval (confirmation of reservation) of the next-hop switch.

tion switch).

4.1.4 Traffic shaper — Time-Aware Shaper (TAS)

The TAS is the main shaping and scheduling mechanism that controls the gating schedules for all the traffic classes within the TSN domain. All bridges are synchronized to the same gating schedule GCL CT that is initially predefined by the network administrator. Ideally, we want the CT to be large enough for all streams from all traffic classes to be accommodated and short enough that all streams meet their delay requirements. In our current evaluations, the CT is predefined at 50 microseconds.

4.1.5 Reconfiguration calculus

The reconfiguration (dynamic configuration) of the TAS schedules (switch GCL/GCE) for each egress port is dynamically invoked according to two principle events, $i)$ adding a new stream, and $ii)$ removing an existing stream. The switch's gating ratio (for a particular stream belonging to a defined traffic class) reports certain parameters (e.g., packet injection rate, maximum packet size, latency requirement, deadline, and application response time) which are then used to check if enough slot time is available (which corresponds to attempting bandwidth reservation). In the event that no slots are available, the GCE slot size is recomputed (according to the registered stream properties within the registration table), generally by allocating more resources from BE Traffic. The stream lifetime is reported by the source to the network as User/Network Information (UNI). Each UNI is propagated by each switch along the path which allows the switch to register the stream and store the stream's resource utilization time (stream lifetime) among other critical information. Any information pertaining to the UNI of a stream is recorded in the stream registration table. In terms of GCEs for TAS with the support of ST and BE traffic classes, only two GCEs within a GCL (1/0 (ST/BE) for the first entry and 0/1 (BE/ST) for the second entry) are necessary with a total of three outbound queues for each egress channel port in a TSN switch; two queues for each traffic class (ST and BE), and another queue for CDT traffic (signaling traffic).

Upon initialization, each switch allocates 20% of the CT to ST traffic, and BE traffic is initially allocated the remaining 80% of the CT. These initial settings are chosen arbitrarily to start up the network system. As streams get registered, the ST slot time is recomputed (according to the stream packet size, ST injection rate, and current slot time). If the stream is the first stream to the switch, then the ST slot size is set at a minimum to 11% (a minimum of 1% step size for the added ST flow plus the minimum ST partition of 10%) of the CT. Thus, as ST streams are admitted and exit the system, the ST vs. BE allocation is dynamically adapted in the reconfiguration scenarios. The minimum step size of 1% of the CT is considered so as to limit the adaption granularity to a reasonable level. Note that the ST to BE slot size (or gating ratio) is limited to

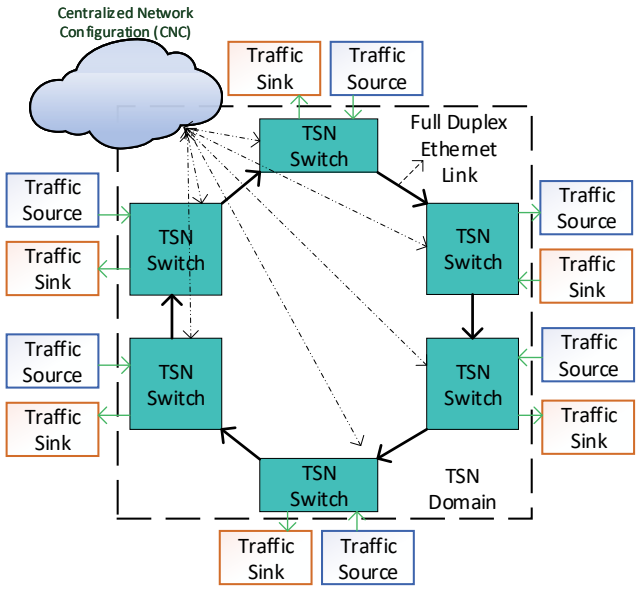


Fig. 5 – Industrial control loop topology [36]: Each source generates stream data with varying hop counts and packet rates unidirectionally or bidirectionally across the six switches ultimately destined to a sink

10% and 90% for the lower and upper limits, respectively. The main reasoning behind this design choice is to avoid any starvation of lower priority traffic.

4.1.6 Path computation

While a path computation module is fundamentally necessary in any switch (in a decentralized/distributed network), we define static shortest path routing tables for destination addresses and associated ports on each switch. Essentially, we assume a procedure to compute paths, i.e., we assume that there is a path computation module, e.g., Path Computation Engine (PCE), that is used in both centralized and distributed configuration models (the path computation can be accelerated with hardware modules [47,70,83], if needed). We make this assumption to simplify operations and place emphasis on the TAS reconfiguration technique.

4.1.7 Network resource table

To remove certain overheads of the configuration procedure, the network resource table operates in tandem with the stream registration table to accurately determine the required network resources (mainly bandwidth for our traffic model) per switch egress port. The network resource table classifies streams based on periodic and sporadic stream properties, though currently our focus is on periodic ST streams. Any stream that has been approved by a switch has an associated record in the network resource table, located within each switch, which can be called to compute and store current and remaining link/port loads for each switch. Each egress port has a network resource table.

Table 1 – Simulation parameters

Key	Symbol	Value
Simulation Duration	Sim_{limit}	100 seconds
Initialized Cycle Time	GCL_{CT}	50 μs
Initialized Gating Ratio	ST^{R}_{init}	20% (i.e., 10 μs)
Average Streams per Second	π	$1-20$
Average stream duration	τ	$2-5$ seconds
BE Traffic Intensity	ρ_L	$0.1, 1.0, 2.0$ Gbps (580 byte packets)
ST sources	S	6
Queue Size	Q_{size}	512 KB

5. PERFORMANCE EVALUATION

5.1 System overview and simulation setup

This section explains the simulation setup and model. Furthermore, the topology and simulation scenarios will be presented. Throughout, we employ the OMNet++ [86] simulation environment. For each evaluation for a given set of parameters, we conduct 5 independent simulation replications; each replication simulates the network for 20 seconds. The widths of the resulting 95% confidence intervals are smaller than 5% of the corresponding sample means and are therefore omitted from the plots to avoid clutter.

5.1.1 Network model

The network topology is modeled around an industrial control loop topology that consists of six core switches in a ring topology. In the case of the centralized model, a CNC is used with out-of-band connections to each of the core switches; while in the distributed approach, the signaling is in-band and can interfere with data traffic within the TSN domain, as shown in Fig. 5. Each switch-to-switch link operates as a full-duplex Ethernet link with a capacity (transmission bitrate) $R = 1$ Gbps. Each switch can act as a gateway for a number of traffic sources and one sink. The distance between two successive switches along the ring is fixed to 100 m and the switch-to-switch propagation delay is set accordingly to 0.5 μs. The out-of-band connections have exactly the same configurations as the normal full-duplex Ethernet links in the data plane, i.e., the same bitrate and propagation delay. All switches are configured to use 802.1Qbv TAS as the traffic shaper for each switch-to-switch egress port whose flow schedule (ST gating ratio and cycle time) is configured by the CNC in the centralized (hybrid) model and independently in the decentralized (fully distributed) model. For all simulation runs, the ST slot size is initialized to 20% of the CT. For the operation without reconfiguration, the ST slot size is kept at 20% of the CT; whereas, for the operation with reconfiguration, the ST slot size is dynamically recomputed when the first stream transmission request arrives.

5.1.2 *Traffic model*

We consider periodic (pre-planned) traffic for the ST traffic and sporadic (random) Poisson traffic for the BE traffic. To emulate dynamic conditions in the network, we employ several distributed ST sources that generate π ST streams according to the network and traffic parameters shown in Table 1. The stream generation follows a Poisson process with a prescribed mean rate of π generated streams per second. We refer to the stream generation rate π also as the stream mean rate π. Each ST stream injects one packet of size 64 bytes per cycle. A destination address is assigned by the number of switch-to-switch hops. A given stream that has been generated at the traffic source attached to a given TSN switch is destined to the traffic sinks at the other five TSN switches with a uniform probability of $1/5$. Furthermore, at stream creation, each stream is given a start time (usually the current runtime), and a finish time according to a stream lifetime (duration) that follows the exponential distribution with mean τ. The exponential stream lifetime is considered as call level dynamics in communication networks often follow Poisson process dynamics, i.e., exponential call lifetimes. As TSN networks become more commonly deployed, it will be important to verify the stream lifetime dynamics through real system measurements.

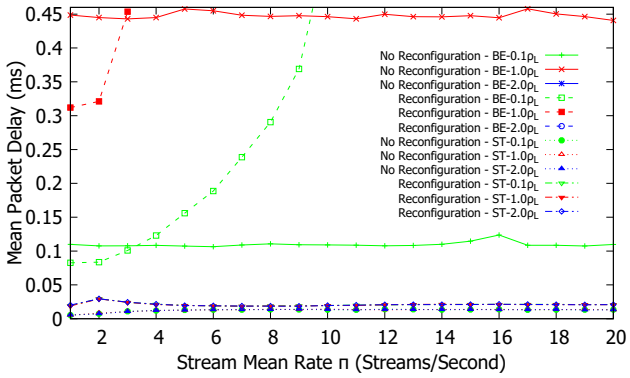
We consider admission as the completion of the reservation of the network resources for the flow from the source node to the destination node. Each source is attached to a core TSN switch gateway (first hop switch). While the TSN switches operate with time synchronization, the ST sources (outside the TSN domain) do not need to be synchronized. However, note that the ST traffic follows an isochronous traffic class, as specified by IEC/IEEE 60802, whereby the sources are synchronized with the network after stream registration is completed. In particular, the ST sources inject the ST traffic in just-in-time fashion, i.e., the transmission of the ST packets out of a source starts at the instant of the start of the ST transmission slot at the switch that is directly attached to the source. Packets are time stamped for the packet delay measurement at the time instant when the packet transmission out of the source commences.

5.2 Centralized (hybrid) model evaluation

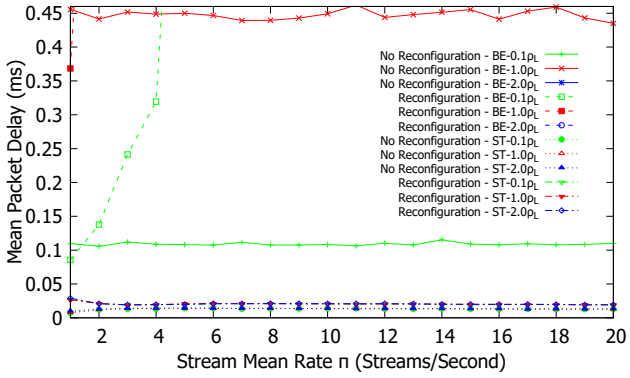
In evaluating the proposed solution described in Section 3, we consider both periodic ST traffic and sporadic BE traffic, as described in Section 5.1.2. We evaluated the CNC with TAS shaper on the industrial control loop for the unidirectional and bidirectional topologies and results are collected for the simulation parameters shown in Table 1.

5.2.1 *Unidirectional ring topology*

Fig. 6 shows the average mean delay for ST traffic and for BE traffic for the centralized unidirectional ring topologies. The average delays are generally short and stable



(a) $\tau = 2$



(b) $\tau = 5$

Fig. 6 – Centralized (Hybrid) Unidirectional Topology: Mean end-to-end delay for ST and BE traffic for mean stream durations $\tau = 2$ and $\tau = 5$ seconds under different BE loads ρ_L, and ST stream rates π.

for both BE and ST traffic. Since the CNC manages the ST traffic streams and therefore guarantees the bandwidth rates needed across the stream's path, the ST delays are less than 100 μs for all average stream durations τ. The ST streams with reconfiguration at the CNC experience higher delays than for the no reconfiguration scenarios since we essentially push more ST traffic into the network which increases the queuing delay. BE traffic experiences much higher delays than ST. With the no reconfiguration approach, the BE traffic delay is nearly constant since the gating ratio is left unchanged. The BE mean delay increases dramatically (up to 21 ms) with reconfiguration since the accepted ST streams tend to consume the full permitted 90% of the CT, leaving only very limited transmission resources for BE traffic.

As mentioned in the introduction section, TSN needs to limit the maximum delay in order to deterministically forward traffic across a TSN domain. Fig. 7 shows the maximum delay for the ST traffic. We observe from Fig. 7 that for the unidirectional ring topology with a maximum of five hop streams, the maximum delay for the reconfiguration approach is below 0.11 ms. On the other hand, for the "no reconfiguration" approach, the maximum delay is below 60μs due to lower frame residence time on each switch; however, reconfiguration increases the admission of ST streams as examined next in Fig 8. TAS in conjunction with the CNC registration and reservation

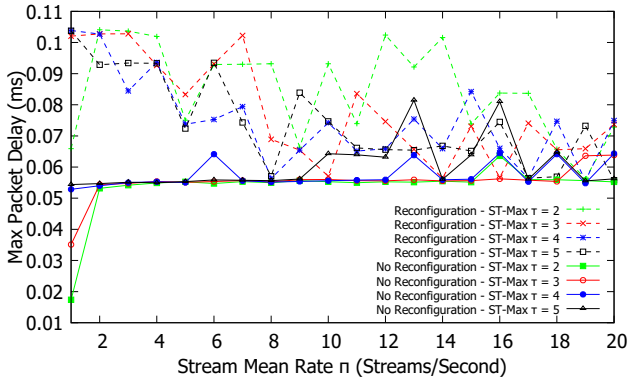


Fig. 7 – Centralized Unidirectional Topology: Maximum packet delay for TAS with centralized configuration (CNC) management.

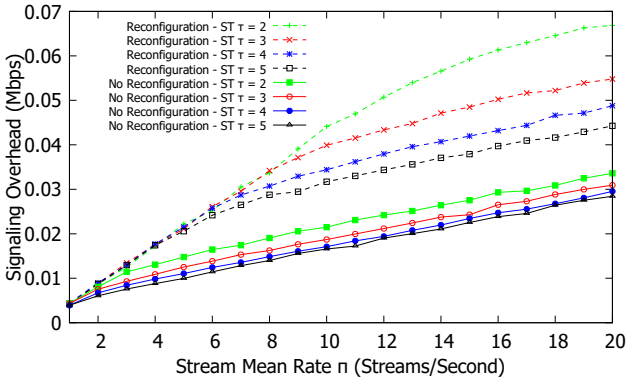


Fig. 9 – Centralized Unidirectional Topology: Stream average signaling Overhead for TAS with centralized configuration (CNC) management.

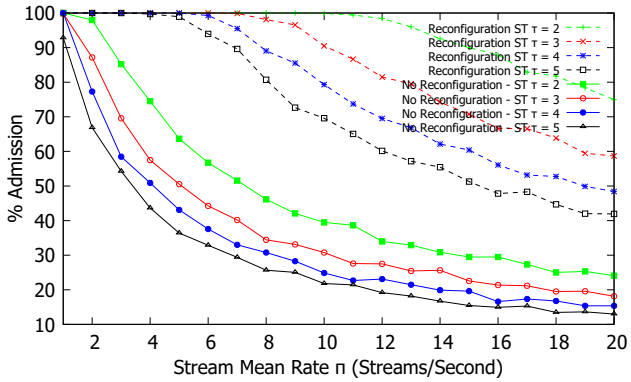


Fig. 8 – Centralized Unidirectional Topology: Stream Admission percentage for TAS with centralized configuration (CNC) management.

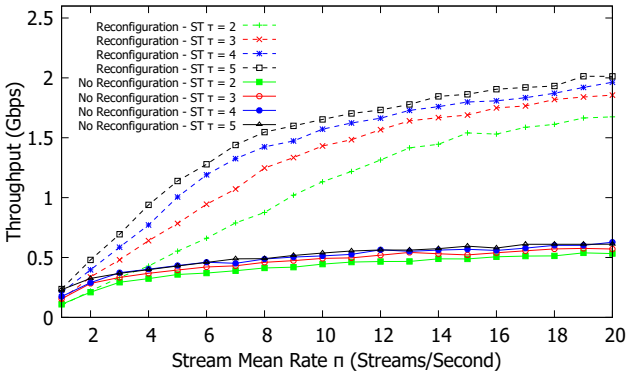


Fig. 10 – Centralized Unidirectional Topology: ST total average throughput measured at the sink for TAS with centralized configuration (CNC) management.

procedure provide a prescribed bandwidth share of the egress port using time division multiplexing. With our empirically chosen parameters, the maximum delays is capped to approximately 100μs which is suitable for the considered topology and time-critical ST traffic that requires less than 1 ms of delay.

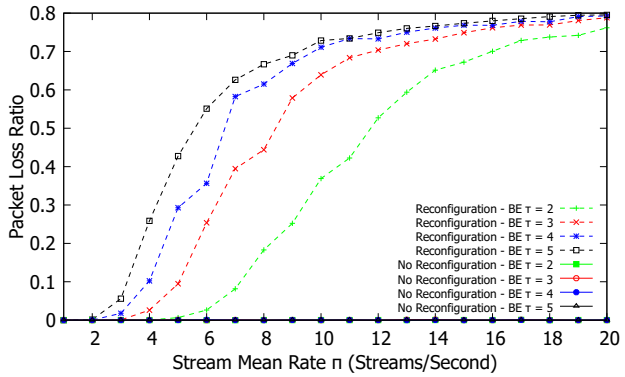
While QoS metrics are important, another factor that determines the performance gains is the admission ratio for the system. Fig. 8 shows the stream admission ratio for both reconfiguration and no reconfiguration. In general, each generated stream needs a data rate of about 11.5 Mbps for a 50μs CT (which corresponds to approximately 45μs of maximum ST slot size since we permit ST traffic to take up at most 90% of the CT) for each egress port on the stream's path with 1 packet injected by an ST per CT and a fixed packet size of 64 B. With an egress port channel capacity of $R = 1$ Gbps, approximately 86 streams can be accommodated. Compared to the "no reconfiguration" approach, the reconfiguration approach significantly improves the admission rates at the expense of higher BE traffic delays, since the ST slot borrows BE time slots to accommodate the ST streams. We also note that increasing the maximum ST allocation above 90% would increase the ST stream admission ratio, at the expense of starving the BE traffic.

CDT traffic that requests transmission guarantees from the CNC experiences some delay before being either admitted or rejected. Since the control plane is out-of band from the data plane within the TSN domain, the delay is constant (around $4\ \mu$s) throughout the simulation run.

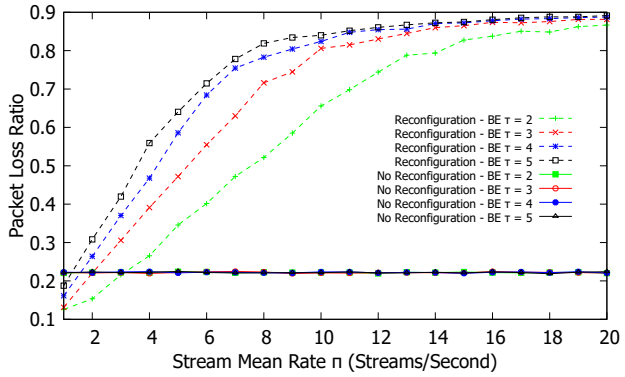
Stream registration and reservation introduce some control plane overhead. Fig. 9 shows the signaling overhead. More specifically, the overhead is measured as the signaling traffic rate in Mbit/s at the CNC for both incoming and outgoing control (CDT) traffic. Generally, the reconfiguration introduces more signaling overhead; however, Ethernet generally has large bandwidths, thus the CDT traffic rates are minuscule compared to the link capacities. Furthermore, when $\tau = 2$, we observe higher signaling overhead due to accepting larger numbers of streams (rejections are inexpensive compared to acceptance) both with and without reconfiguration.

Fig. 10 shows the average throughput measured at the sink for ST traffic. We observe from Fig. 10 that the reconfiguration substantially increases the throughput compared to the no reconfiguration scenario. Typically, the throughput is more than doubled by the reconfiguration.

To examine the reliability performance, Fig. 11 shows the BE packet loss ratio for mid and high BE traffic loads ρ_L; we omitted the low BE traffic load which has negligible losses. Since the CNC manages only ST streams, the TSN guarantees (which include zero packet loss since retransmissions are in general too expansive for ST traffic) are only valid for ST streams. As the ST traffic load increases in the reconfiguration scenario, the BE packet loss in-



(a) Mid BE Traffic Load $\rho_L = 1$ Gbps



(b) High BE Traffic Load $\rho_L = 2$ Gbps

Fig. 11 – Centralized Unidirectional Topology: BE frame loss ratio for TAS with centralized configuration (CNC) management.

creases. For the "no reconfiguration" approach, the BE packet loss is typically constant even for high loads of BE traffic.

For a benchmark comparison of the TSN effectiveness, and specifically TAS, we conducted additional evaluations for the scenario in Fig. 6 without the TSN slot reservation, admission control, and TAS scheduling. Specifically, we considered an ST stream mean generation rate of 1–20 streams per second with a mean lifetime $\tau = 5$ seconds with the mid and high BE traffic loads of $\rho_L = 1.0$ Gbps and 2.0 Gbps. We employed strict priority scheduling at each switch without any TSN slot reservation, i.e., each switch output port schedules and transmits all ST packets before any BE packets. We outline three main observations for the unidirectional ring topology. First, while the mean delays were generally very low for ST traffic (34–55 μs for the low traffic load range $\pi = 1$ to 5 ST streams per second), the priority scheduling of the ST packets can severely starve the low-priority BE traffic (for the high $\rho_L = 2.0$ Gbps BE load, the mean BE packet delays increase from a minimum of 15 ms to a maximum of around 0.1 s as the ST load increases from 1 to 20 streams per second; whereas, with TSN, the mean BE packet delays increase from around 10 ms to 21 ms, which is outside the plotted range of Fig. 6(b)). Additionally, compared to TSN, the maximum delays and jitter increase more strongly as the BE and ST loads increase (the ST maximum packet delays range from 34 μs to 20 ms; while, with the TSN operation, the ST maximum packet delays hover around 55–101 μs, see Fig. 7). This stronger increase of the maximum ST packet delays is a result of the BE packet traffic interfering with the ST packet traffic due to the lack of TAS operation. In particular, ST packets are blocked from transmission during an ongoing transmission of a 580 byte BE packet (as we considered non-preemptive priority scheduling). Second, since no admission control based on TSN slot reservation is used, congestion arises for ST traffic loads of $\pi = 6$ to 20 ST streams per second, causing high mean and maximum delays for both ST and BE traffic. Third, due to the congestion, packet drops occur at high ST loads for both ST and BE packet traffic. We also note that since no signaling traffic is used, the priority scheduling benchmark without TSN operation provides a performance reference for both the centralized and the decentralized TSN model.

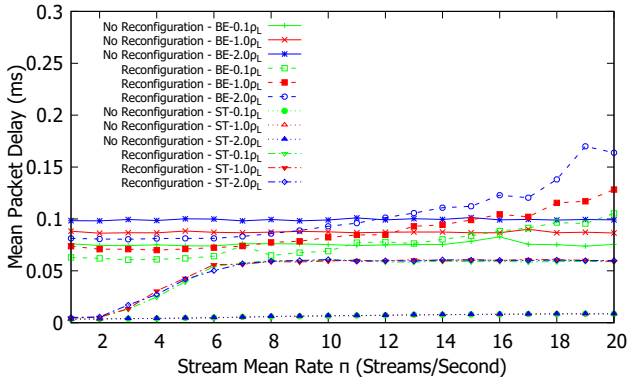
Overall, we conclude that the proposed centralized (hybrid) reconfiguration approach provides a means to ensure that dynamically varying numbers of ST streams are accommodated as permitted by the available link capacity in the unidirectional ring network. However, the unidirectional ring network does not involve any distinct routing choices towards the destination. In order to examine the performance of the proposed centralized reconfiguration in a network with different routing paths, we next consider the operation of the ring network topology as a bidirectional ring network.

5.2.2 Bidirectional ring topology

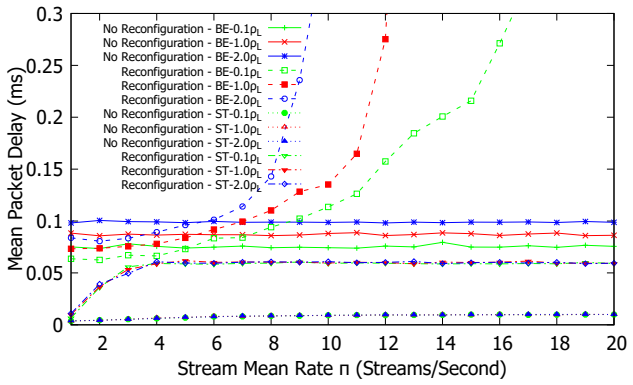
The unidirectional ring topology certainly simplifies the calculation of the ST slot window in the reconfiguration. In order to examine whether the proposed centralized (hybrid) reconfiguration approach can efficiently utilize the higher capacity of a more complex network with multiple routing options, we examine the bidirectional ring network. In the bidirectional ring network, each two-port switch has now two paths to the destination. We employ shortest path routing according to the hop count. We set the edge link (source to first ring switch and last ring switch to sink) capacities to 2 Gbps to avoid congestion on the edge links (which the CNC does not control).

Fig. 12 shows the average mean ST and BE packet delay for different stream lifetimes τ. Compared to the unidirectional topology (see Fig. 6), the bidirectional significantly reduces the packet delay since an extra port with full-duplex link support now provides extra capacity to service streams giving more slot reservations to BE even at high ST stream loads.

Fig. 13 shows the maximum ST packet delays for the bidirectional ring topology with CNC. We observe from Fig. 13 in comparison with the corresponding maximum packet delay plot for the unidirectional ring in Fig. 7, that the bidirectional topology with configuration gives higher maximum packet delays, which is mainly due to the substantially increasing ST stream acceptance, as examined next in Fig 14. The "no reconfiguration" keeps the ST slot size



(a) $\tau = 2$



(b) $\tau = 5$

Fig. 12 – Centralized Bidirectional Topology: Mean end-to-end delay for ST and BE traffic for varied mean stream lifetime τ for different BE loads ρ_L, and ST stream rates π.

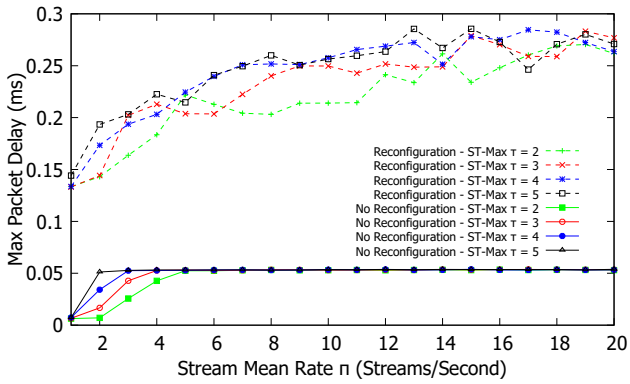


Fig. 13 – Centralized Bidirectional Topology: Maximum ST packet delay for TAS with centralized configuration (CNC) management.

at the initialized value (20% of CT, i.e., $10\ \mu$s), resulting in a constant maximum delay of around $50\ \mu$s, albeit at the expense of rather low admission rates, see Fig 14.

Fig. 14 shows the stream admission ratio (percentage). With the high stream generation rate $\pi = 20$ streams/s and long average stream lifetime $\tau = 5$ s, the admission rate is still slightly above 90% for the bidirectional topology with CNC reconfiguration. The bidirectional ring thus achieves a substantially increased (close to 50% higher) admission rate compared to the unidirectional ring examined in Fig. 8. In contrast, the increases of the admission ratio of the no reconfiguration approach with the bidirectional ring compared to the unidirectional ring are more modest (roughly 20%). This is mainly because the initialized gating ratio is too restrictive and severely underutilizes the links. We found in additional evaluations that are not included due to space constraints that different BE loads ρ_L do not impact the ST stream performance due to the TAS operation, i.e., TAS effectively partitions the traffic at the egress switch/port (BE traffic does not block ST traffic).

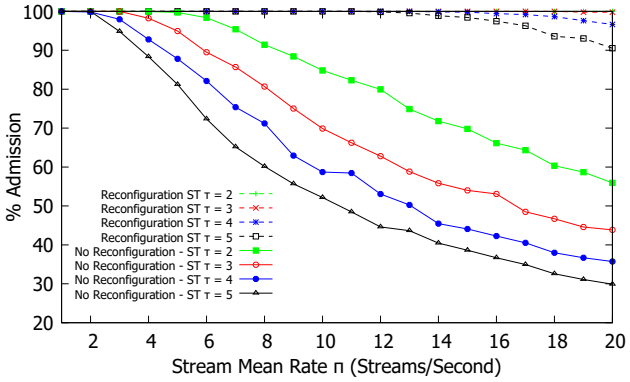


Fig. 14 – Centralized Bidirectional Topology: Stream admission percentage for TAS with centralized configuration (CNC) management.

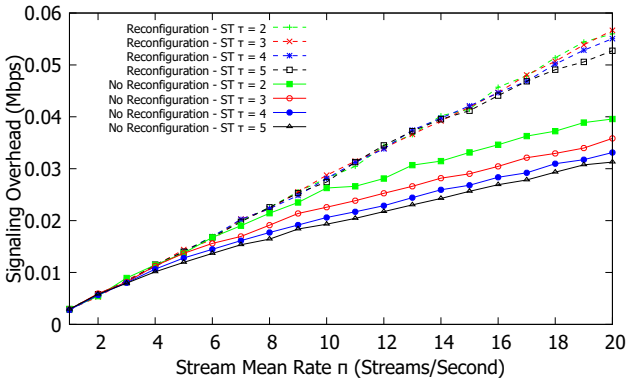


Fig. 15 – Centralized Bidirectional Topology: Average stream signaling overhead for TAS with centralized configuration (CNC) management.

Similar to the unidirectional ring, the bidirectional ring topology provides constant signaling delay (around $3.5\ \mu$s) due to the CNC out-of band signaling channels. The average signaling delay is slightly lower than in the unidirectional ring (which had a signaling delay around $4\ \mu$s), mainly since the signaling hop distances in the bidirectional ring are shorter than in the unidirectional ring.

Fig. 15 shows the signaling overhead. Since the bidirectional ring topology is effectively the same as the unidirectional ring topology (albeit having another port to the switch), the signaling overhead in the bidirectional ring network is in general very similar to the signaling overhead in the unidirectional topology. Note that while the hop traversal is reduced (since the stream can take one of two paths to the destination governed by shortest path, i.e., the smallest hop count), the number of sent and received CDT frames are generally the same. Similar to the unidirectional topology, the reconfiguration approach generates more CDT traffic. Note that admissions

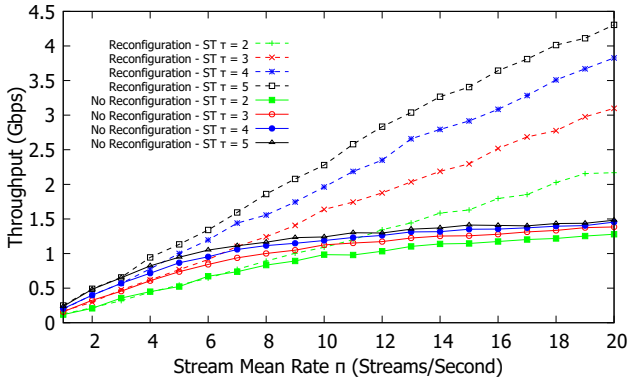
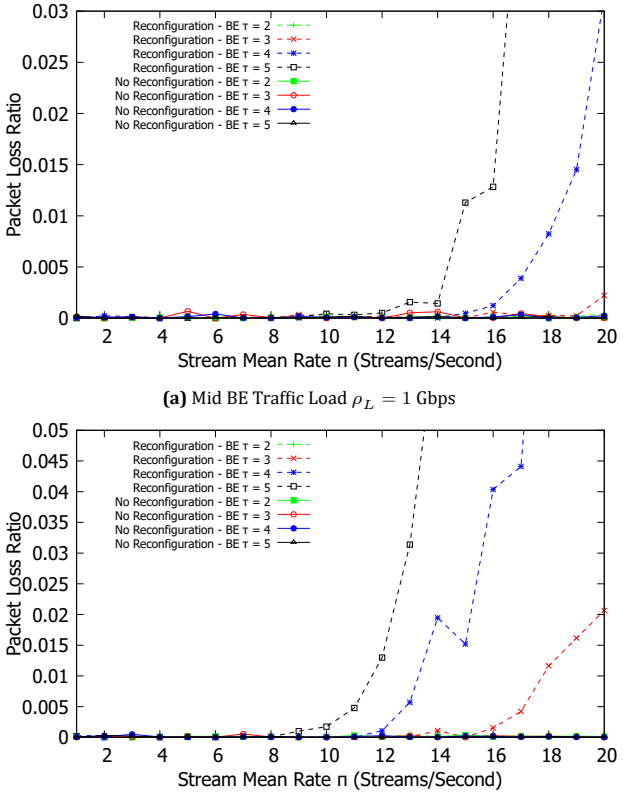


Fig. 16 – Centralized Bidirectional Topology: ST Total average throughput measured at the sink as a results of TAS with centralized configuration (CNC) management.



(a) Mid BE Traffic Load $\rho_L = 1$ Gbps

(b) High BE Traffic Load $\rho_L = 2$ Gbps

Fig. 17 – Centralized Bidirectional Topology: BE frame loss ratio for TAS with centralized configuration (CNC) management.

are in general more costly in terms of sent and received CDT frames in the network. Therefore, the higher the admission rate, the more overhead is observed in the control plane, though based on Fig. 15, the overall overhead is well below 1 Mbps and therefore is minuscule compared to the channel capacity. We also observe from Fig. 15 that the results for different stream lifetimes τ differ only very slightly since for any τ value, almost all the streams are accepted, generating the same total overhead.

Fig. 16 shows the average overall throughput measured at the ST sinks for the bidirectional ring topology. Compared to the unidirectional ring (see Fig. 10), the throughput for the bidirectional ring is much higher, typically increased by a factor of two.

Similar to the unidirectional ring topology, the bidirectional topology achieves zero loss for ST streams while significantly reducing the BE packet loss rate. Fig. 17 shows the BE packet loss ratio for the bidirectional ring network. The maximum BE loss for the high BE traffic intensity $\rho_L = 2.0$ is around 30% which is a significant reduction from the unidirectional topology (of around 90%, see Fig. 11).

In contrast to the unidirectional topology, the bidirectional topology with central (hybrid) CNC reconfiguration achieves improved QoS metrics and admission rates. Overall, the ST traffic throughput is typically doubled in the bidirectional ring network compared to the unidirectional ring network. We can thus conclude that our proposed centralized (hybrid) CNC reconfiguration can effectively utilize the higher capacity provided by the bidirectional ring network for dynamic ST traffic, with random ST flow generations and random ST flow lifetimes.

5.3 Decentralized model evaluation

Analogous to the centralized (hybrid) reconfiguration evaluation, we evaluate our proposed decentralized reconfiguration from Section 4 with both periodic ST traffic and sporadic (random) BE traffic, as specified in Section 5.1.2. As before, we evaluate the network with TAS shaper on the industrial control loop unidirectional and bidirectional topology and collect results for the simulation parameters shown in Table.1.

5.3.1 Unidirectional ring topology

The decentralized model essentially transfers some of the CNC functions (e.g., TAS reconfiguration and resource reservation modules) from the centralized model down to the TAS enabled egress ports of the TSN switches in the data plane. The main difference between the centralized and decentralized models is the signaling performance which is now in-band and can affect data traffic. In additional evaluations that are not included due to space constraints, we have found that with the in-band CDT traffic in the decentralized model, the average ST and BE packet delays are about the same as the centralized model in Fig. 6. Typically, the ST stream's average delay is minimal to near constant for both the reconfiguration and "no reconfiguration" approaches. For BE, the "no reconfiguration" approach produces constant average delay for each BE ρ_L traffic intensity.

Fig. 18 shows the maximum ST packet delay for the unidirectional ring network using the decentralized model. In contrast to the average ST packet delay, the maximum delay is affected by the in-band CDT traffic. In the decentralized model, the CDT traffic is given the highest priority above both ST and BE traffic. Therefore, the maximum delays can reach about 150 μs, which is somewhat higher than for the centralized reconfiguration in Fig. 7, but still well below 1 ms.

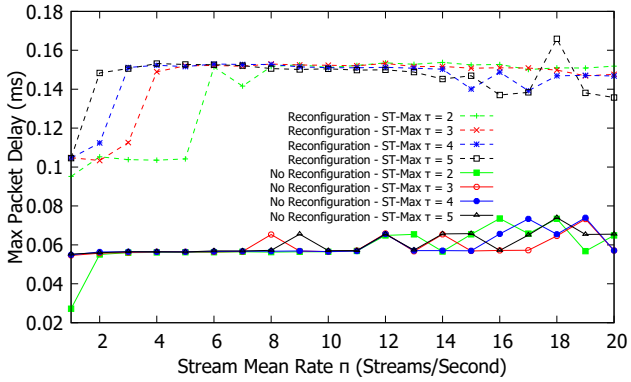


Fig. 18 – Decentralized Unidirectional Topology: Max delay for TAS.

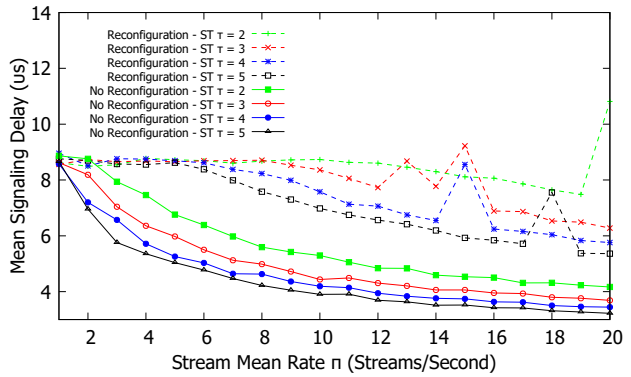


Fig. 19 – Decentralized Unidirectional Topology: Average stream signaling delay for TAS.

The stream admission rate for the decentralized model is very similar to the centralized model (see Fig. 8) and is not displayed in detail due to space constraints. Fig. 19 shows the signaling delay for ST stream registration in the decentralized model. In contrast to the centralized model, the decentralized model's in-band CDT traffic implies varied stream signaling delays. As the streams generation rate π increases, the overall average signaling delay decreases which is due to the increased rejections as more streams attempt to request network resources. In the decentralized model, a rejection by an intermediate bottlenecked switch implies a termination of the reservation attempt and a notification to any previous pending stream records to cancel the potential reservation and eventually notify the source of the rejection. If this rejection happens closer to the source, then the average signaling delay will be shorter compared to a stream acceptance. In general, the average stream signaling delay is on the order of microseconds which is reasonable for most industrial control systems applications.

Generally, the decentralized model produced greater signaling overhead than the centralized model (cf. Fig. 9) since CDT traffic is measured at each data switch traffic port for incoming and outgoing as shown in Fig. 20. Analogous to the signaling delay, the more ST streams are accepted, the more overhead is observed. Therefore, as the stream lifetime τ increases and consequently, the more rejections occur, the lower the overhead. Overall, the comparison of the signaling overhead for the decentralized model in Fig. 20 with the centralized model in Fig. 9 indicates that the decentralization increases the signaling overhead by over two orders of magnitude. However, the aggregate signaling overhead bitrate in the decentralized model is still below 20 Mbps and thus below 2% of the 1 Gbps link capacity.

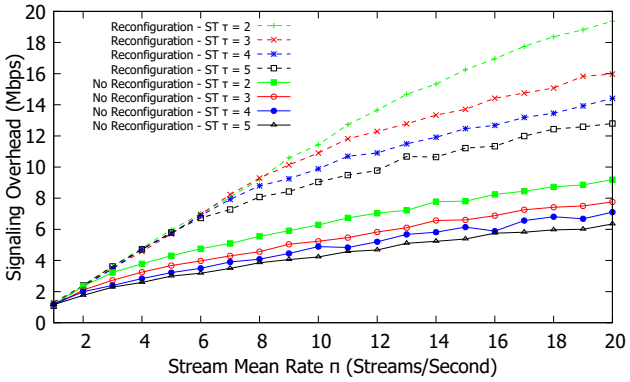


Fig. 20 – Decentralized Unidirectional Topology: Stream Signaling Overhead for TAS.

Throughput results are generally the same when compared to the unidirectional centralized model (cf. Fig. 10) and are therefore omitted. Similarly, the packet loss rate is nearly similar to the unidirectional centralized model (cf. Fig. 11). However, the unidirectional topology with either the centralized or decentralized approach generally gets bottlenecked at lower traffic loads compared to the bidirectional ring network. Therefore, BE traffic suffers as more ST streams request TAS slot reservations. We next examine the bidirectional ring network for decentralized operation to determine how the BE traffic performance can be improved while maintaining the ST traffic performance.

5.3.2 Bidirectional ring topology

For the bidirectional topology using the decentralized model we found that the in-band CDT traffic affects the data traffic similar to the decentralized unidirectional model, i.e., maximum ST packet delay is somewhat increased while the mean ST packet delay is essentially unchanged. As the ST stream lifetime τ is increased, i.e., the number of ST streams at any time increases, the BE slots are reallocated to ST streams which increases the mean BE packet delay which is similar to the centralized model (cf. Fig. 12) and is therefore omitted.

Fig. 21 shows the maximum ST packet delay. While the reconfiguration approach looks very similar to the centralized model (cf. Fig. 13), the no reconfiguration approach is affected by the in-band CDT traffic which raises the maximum ST packet delay in some no reconfiguration scenarios to around $100\ \mu$s.

The admission rate is exactly the same as for the centralized model (cf. Fig. 14). Fig. 22 shows the average signaling delay for ST stream registration. Similar to the unidirectional topology, the mean signaling delay starts to decrease as the load increases due to higher rejections.

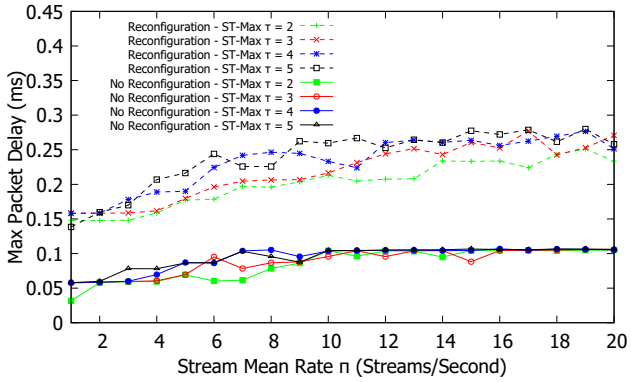


Fig. 21 – Decentralized Bidirectional Topology: Maximum ST packet delay for TAS.

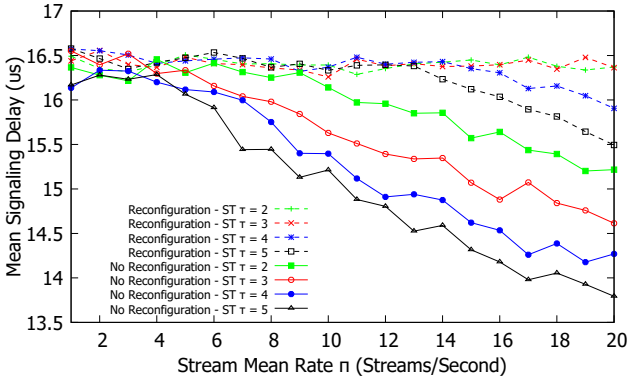


Fig. 22 – Decentralized Bidirectional Topology: Average stream signaling delay for TAS.

We found that the stream signaling overhead with the decentralized bidirectional model is similar to the decentralized unidirectional model (cf. Fig. 20), albeit slightly lower due to the shorter signaling hop counts in the bidirectional ring network.

While omitted for space, additional evaluations have found that the throughput of the bidirectional decentralized model is nearly identical to the centralized model. We observed only very slightly reduced throughput with the decentralized model compared to the centralized model since the decentralized model carries the control traffic in-band, which very slightly reduces the link utilization for data traffic.

Similar to all the preceding models and topologies, ST streams have zero traffic drops. The BE packet loss rates for the decentralized bidirectional model are nearly identical to the centralized bidirectional model. Similarly, the overall performance is largely improved under the bidirectional topology compared to the unidirectional model due to the additional port and path.

The decentralized model was found to operate nearly identically to the centralized model in terms of QoS metrics and overall admission rate. Thus, the segregation of traffic based on the class of service can be accomplished with the proposed decentralized model without the overhead complexities of a CNC node. A main disadvantage of the decentralized model is the in-band CDT traffic which can delay ST streams, particularly affecting the maximum ST packet delays. A potential workaround to explore in future research is to service all the ST streams first, and then service CDT frames before servicing the BE traffic, though this might lead to additional signaling delays depending on the ST load.

6. CONCLUSIONS AND FUTURE WORK

The IEEE 802.1Qcc framework and the 802.1Qbv traffic shaper enable the implementation of a deterministic forwarding plane that provides strict bandwidth guarantees to ST flows without any flow or congestion control mechanism at the source. Using an automated network configuration is an imperative tool set to provide a unified communication platform based on commercial of the shelf (COTS) full-duplex Ethernet with high bandwidth and low complexity compared to Controller Area Networks (CANs), Local Interconnect Networks (LINs), and specialized field-buses in industrial control system applications (e.g., industrial control, automotive, and avionics). Network designs based on the IEEE 802.1Qcc framework and the 802.1Qbv traffic shaper can form a contract with the source to forward mission critical traffic and to automate the network configuration process using 802.1Qcc for the full lifetime of the stream. Additionally, depending on the forwarding plane port traffic shaper (e.g., TAS), the required schedules can be passed to the switch servers using general user/network information protocols (e.g., TLV, NETCONF/Yang, and SNMP).

In this paper, we have investigated the impact of TAS reconfigurations in response to dynamic network conditions, i.e., the addition and removal of transient ST streams (flows) with different lifetimes. We have demonstrated the effectiveness of TAS with and without the CNC, i.e., for centralized (hybrid) vs. decentralized (fully distributed) models. We have examined network QoS traffic characteristics when admitting ST flows based on an iterative heuristic approach that computes TAS schedules for current and newly requested ST streams.

Based on the insights from the present study we outline the following future research directions. First, it would be interesting to judiciously change the GCL time for switches during reconfiguration whilst satisfying QoS requirements. The studied reconfiguration techniques should also be examined in alternate approaches for providing deterministic QoS, e.g., [61,81] as well as in the context of related QoS oriented routing approaches, e.g. [17, 37].

Another interesting future research direction is to adapt the reconfiguration mechanisms that have been developed in this study to the interactions between TSN and fifth generation (5G) wireless communication systems that operate with Ultra-Reliable Low-Latency Communication (URLLC). A few recent studies have begun to explore the use of TSN in the 5G URLLC context, see e.g., [28, 45, 52, 71], indicating significant potential for improving 5G URLLC services by exploiting TSN. The TSN reconfiguration mechanisms developed in this study can poten-

tially help in flexibly providing high-quality 5G URLLC services for varying traffic dynamics. Similarly, TSN reconfiguration may aid low-latency real-time services in future WiFi networks, which may incorporate TSN, see e.g., [5, 14, 35].

In the wider context of QoS networking and related applications, deterministic networking should be examined in the context of emerging multiple-access edge computing (MEC) [21, 33, 58, 82], in particular MEC settings for low-latency applications [23, 90, 94]. As an alternative approach to coordinating the reconfigurations, emerging softwarized control and virtualization paradigms can be explored [7, 13, 19, 20, 44, 56, 75, 78]. Regarding the reliability aspects, a potential future research direction is to explore low-latency network coding mechanisms, e.g., [4, 18, 24, 32, 54, 55, 69], to enhance networking protocols targeting reliable low-latency communication.

REFERENCES

[1] IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP). *IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005)*, pages 1–119, Sept. 2010.

[2] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic. *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q— as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q—/Cor 1-2015)*, pages 1–57, Mar. 2016.

[3] IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. *IEEE P802.1Qcc/D2.0, October 2017*, pages 1–207, Jan. 2017.

[4] J. Acevedo, R. Scheffel, S. Wunderlich, M. Hasler, S. Pandi, J. Cabrera, F. Fitzek, G. Fettweis, and M. Reisslein. Hardware acceleration for RLNC: A case study based on the Xtensa processor with the Tensilica instruction-set extension. *Electronics*, 7(9):180.1–180.22, 2018.

[5] T. Adame, M. Carrascosa, and B. Bellalta. Time-Sensitive Networking in IEEE 802.11be: On the way to low-latency WiFi 7. *arXiv preprint arXiv:1912.06086*, 2019.

[6] V. Addanki and L. Iannone. Moving a step forward in the quest for Deterministic Networks (DetNet). In *Proc. IFIP/IEEE Networking Conference (Networking)*, pages 458–466, 2020.

[7] R. Amin, M. Reisslein, and N. Shah. Hybrid SDN networks: A survey of existing approaches. *IEEE Communications Surveys & Tutorials*, 20(4):3259–3306, 2018.

[8] A. Arestova, K.-S. J. Hielscher, and R. German. Design of a hybrid genetic algorithm for Time-Sensitive Networking. In *Proc. Int. Conf. on Measurement, Modelling and Evaluation of Computing Sys.*, pages 99–117, 2020.

[9] V. Balasubramanian, M. Aloqaily, and M. Reisslein. An SDN architecture for time sensitive industrial IoT. *Computer Networks*, 186(107739):1–12, Feb. 2021.

[10] Belliardi, Rudy et al. Use Cases IEC/IEEE 60802, V1.3, Sept. 2018. Available from http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf; Last accessed Feb. 19, 2019.

[11] L. L. Bello and W. Steiner. A perspective on IEEE time-sensitive networking for industrial communication and automation systems. *Proc. IEEE*, 107(6):1094–1120, 2019.

[12] A. Bierman, M. Bjorklund, and K. Watsen. RESTCONF Protocol. RFC 8040, Jan. 2017.

[13] M. Böhm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser. Dynamic real-time stream reservation for IEEE 802.1 time-sensitive networks with OpenFlow. In *Proc. Int. Conf. on Applied Innovations in IT, (ICAIIT)*, pages 7–12, 2020.

[14] D. Cavalcanti, S. Bush, M. Illouz, G. Kronauer, A. Regev, and G. Venkatesan. Wireless TSN-definitions use cases & standards roadmap. *Avnu Alliance White Paper*, pages 1–16, Mar. 2020.

[15] F. Chen. Resource Allocation Protocol (RAP) based on LRP for Distributed Configuration of Time-Sensitive Streams, 2017. http://ieee802.org/1/files/public/docs2017/tsn-chen-RAP-whitepaper-0917-v01.pdf.

[16] C.-C. Chuang, T.-H. Yu, C.-W. Lin, A.-C. Pang, and T.-J. Hsieh. Online stream-aware routing for TSN-based industrial control systems. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 254–261, 2020.

[17] U. Chunduri, A. Clemm, and R. Li. Preferred Path Routing - a next-generation routing framework beyond Segment Routing. In *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pages 1–7, Dec 2018.

[18] A. Cohen, D. Malak, V. B. Bracha, and M. Medard. Adaptive causal network coding with feedback for delay and throughput guarantees. *arXiv preprint arXiv:1905.02870*, 2019.

[19] N. Deric, A. Varasteh, A. Basta, A. Blenk, R. Pries, M. Jarschel, and W. Kellerer. Coupling VNF orchestration and SDN virtual network reconfiguration. In *Proc. Int. Conf. on Networked Systems (NetSys)*, 2019.

[20] A. Destounis, S. Paris, L. Maggi, G. S. Paschos, and J. Leguay. Minimum cost SDN routing with reconfiguration frequency constraints. *IEEE/ACM Transactions on Networking*, 26(4):1577–1590, 2018.

[21] T. Doan-Van, A. Kropp, G. T. Nguyen, H. Salah, and F. H. Fitzek. Programmable first: Automated orchestration between MEC and NFV platforms. In *Proc. IEEE Consumer Commun. & Netw. Conf. (CCNC)*, pages 1–2, 2019.

[22] T. Docquier, Y.-Q. Song, V. Chevrier, L. Pontnau, and A. Ahmed-Nacer. IEC 61850 over TSN: Traffic mapping and delay analysis of GOOSE traffic. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 246–253, 2020.

[23] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler. Toward low-latency and ultra-reliable virtual reality. *IEEE Network*, 32(2):78–84, 2018.

[24] A. Engelmann, W. Bziuk, A. Jukan, and M. Médard. Exploiting parallelism with random linear network coding in high-speed Ethernet systems. *IEEE/ACM Transactions on Networking (TON)*, 26(6):2829–2842, 2018.

[25] R. Enns, M. Bjorklund, A. Bierman, and J. Schönwälder. Network Configuration Protocol (NETCONF). RFC 6241, June 2011.

[26] J. Falk, F. Dürr, and K. Rothermel. Time-triggered traffic planning for data networks with conflict graphs. In *Proc. IEEE Real-Time and Embedded Techn. and Applications Symp. (RTAS)*, pages 124–136. IEEE, 2020.

[27] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel. NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++. In *Proc. IEEE Int. Conf. on Networked Systems (NetSys)*, pages 1–8, 2019.

[28] J. Farkas, B. Varga, G. Miklos, and J. Sachs. 5G-TSN integration meets networking requirements. *Ericsson Technology Review*, pages 1–11, Aug. 2019.

[29] N. Finn. IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Link-local Registration Protocol. *IEEE P802.1CS/D1.2 December 2017*, Dec. 2017.

[30] N. Finn. Introduction to time-sensitive networking. *IEEE Communications Standards Magazine*, 2(2):22–28, 2018.

[31] F. Fitzek, S.-C. Li, S. Speidel, T. Strufe, M. Simsek, and M. Reisslein. *Tactile Internet with Human-in-the-Loop*. Academic Press, Cambridge, MA, 2021.

[32] F. Gabriel, S. Wunderlich, S. Pandi, F. H. Fitzek, and M. Reisslein. Caterpillar RLNC with feedback (CRLNC-FB): Reducing delay in selective repeat ARQ through coding. *IEEE Access*, 6:44787–44802, 2018.

[33] Y. Gao, W. Tang, M. Wu, P. Yang, and L. Dan. Dynamic social-aware computation offloading for low-latency communications in IoT. *IEEE Internet of Things Journal*, 6(5):7864–7877, Oct. 2019.

[34] V. Gavriluț and P. Pop. Traffic-type assignment for TSN-based mixed-criticality cyber-physical systems. *ACM Transactions on Cyber-physical Systems*, 4(2):1–27, 2020.

[35] E. Genc and L. F. Del Carpio. Wi-Fi QoS enhancements for downlink operations in industrial automation using TSN. In *Proc. IEEE Int. Workshop on Factory Commun. Sys. (WFCS)*, pages 1–6, 2019.

[36] J. W. Guck, M. Reisslein, and W. Kellerer. Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks. *IEEE Transactions on Industrial Informatics*, 12(6):2050–2061, 2016.

[37] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer. Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation. *IEEE Commun. Sur. & Tut.*, 20(1):388–415, First Qu. 2018.

[38] T. Häckel, P. Meyer, F. Korf, and T. C. Schmidt. Software-defined networks supporting time-sensitive in-vehicular communication. *arXiv preprint arXiv:1903.08039*, 2019.

[39] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Dürr. On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN. In *2020 IEEE International Conference on Industrial Technology (ICIT)*, pages 298–303. IEEE, 2020.

[40] D. Hellmanns, A. Glavackij, J. Falk, R. Hummen, S. Kehrer, and F. Dürr. Scaling TSN scheduling for factory automation networks. In *Proc. IEEE Int. Conf. on Factory Commun. Systems (WFCS)*, pages 1–8, 2020.

[41] M. Herlich, J. L. Du, F. Schörghofer, and P. Dorfinger. Proof-of-concept for a software-defined real-time Ethernet. In *Proc. IEEE Int. Conf on Emerg. Techn. and Factory Autom. (ETFA)*, pages 1–4, 2016.

[42] T. Hoeschele, C. Dietzel, D. Kopp, F. Fitzek, and M. Reisslein. Importance of Internet Exchange

Point (IXP) infrastructure for 5G: Estimating the impact of 5G use cases. *Telecommunications Policy*, 45(3):102091.1–102091.18, April 2021.

[43] X. Jin, C. Xia, N. Guan, C. Xu, D. Li, Y. Yin, and P. Zeng. Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries. *IEEE Access*, 8:6751–6767, 2020.

[44] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid. Adaptable and data-driven softwarized networks: Review, opportunities, and challenges. *Proceedings of the IEEE*, 107(4):711–731, April 2019.

[45] M. Khoshnevisan, V. Joseph, P. Gupta, F. Meshkati, R. Prakash, and P. Tinnakornsrisuphap. 5G industrial networks with CoMP for URLLC and time sensitive network architecture. *IEEE Journal on Selected Areas in Communications*, 37(4):947–959, 2019.

[46] T. Kobzan, S. Schriegel, S. Althoff, A. Boschmann, J. Otto, and J. Jasperneite. Secure and time-sensitive communication for remote process control and monitoring. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Autom. (ETFA)*, volume 1, pages 1105–1108, 2018.

[47] M. Kovacevic, V. Skobic, M. Knezic, and Z. Ivanovic. Towards implementation of frame preemption mechanism on FPGA platform. In *Proc. IEEE Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, pages 1–7, 2020.

[48] J. Krolikowski, S. Martin, P. Medagliani, J. Leguay, S. Chen, X. Chang, and X. Geng. Joint routing and scheduling for large-scale deterministic IP networks. *arXiv preprint arXiv:2004.02717*, 2020.

[49] G. N. Kumar, K. Katsalis, and P. Papadimitriou. Coupling source routing with time-sensitive networking. In *Proc. IFIP/IEEE Networking Conference (Networking)*, pages 797–802, 2020.

[50] M. Kumar, M. Boehm, J. Ohms, O. Shulha, and O. Gebauer. Evaluation of the time-aware priority queueing discipline with regard to Time-Sensitive Networking in particular IEEE 802.1Qbv. In *Proc. Int. Conf. on Appl. Innovation in IT*, volume 7, pages 1–6, 2019.

[51] M. Lander, P. Raagaard, M. G. Pop, and W. Steiner. Runtime reconfiguration of time-sensitive networking (TSN) schedules for fog computing. In *Proc. IEEE Fog World Congress (FWC)*, pages 1–6, Oct. 2017.

[52] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez. Analysis of 5G-TSN integration to support Industry 4.0. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 1111–1114, 2020.

[53] Z. Li, H. Wan, Y. Deng, K. Xiong, and X. Song. A resource-efficient priority scheduler for time-sensitive networking switches. *CCF Transactions on Networking*, 3:21–34, 2020.

[54] D. E. Lucani, M. V. Pedersen, D. Ruano, C. W. Sørensen, F. H. Fitzek, J. Heide, O. Geil, V. Nguyen, and M. Reisslein. Fulcrum: Flexible network coding for heterogeneous devices. *IEEE Access*, 6:77890–77910, 2018.

[55] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic. High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals and enabling technologies. *IEEE Internet of Things Journal*, 6(5):7946–7970, 2019.

[56] W. Mandarawi, H. Chahed, and H. de Meer. A framework for virtualizing time-aware shaper using high performance nfv. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 1621–1628, 2020.

[57] L. Martenvormfelde, A. Neumann, L. Wisniewski, and J. Jasperneite. A simulation model for integrating 5G into time sensitive networking as a transparent bridge. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 1103–1106, 2020.

[58] J. Martín-Pérez, L. Cominardi, C. J. Bernardos, A. De la Oliva, and A. Azcorra. Modeling mobile edge computing deployments for low latency multimedia services. *IEEE Transactions on Broadcasting*, 65(2):464–474, June 2019.

[59] M. A. Metaal, R. Guillaume, R. Steinmetz, and A. Rizk. Integrated industrial ethernet networks: Time-sensitive networking over SDN infrastructure for mixed applications. In *Proc. IFIP/IEEE Networking Conference (Networking)*, pages 803–808, 2020.

[60] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury. Reconfiguration algorithms for high precision communications in time sensitive networks. In *Proc. IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2019.

[61] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury. TSN algorithms for large scale networks: A survey and conceptual comparison. *arXiv preprint arXiv:1905.08478*, 2019.

[62] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury. Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access*, 7:44165–44181, 2019.

[63] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury. Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research. *IEEE Comm. S. & T.*, 21(1):88–145, 2019.

[64] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler. A survey on 5G usage scenarios and traffic models. *IEEE Commun. Surveys & Tutorials*, 22(2):905–929, 2020.

[65] N. G. Nayak, F. Duerr, and K. Rothermel. Routing Algorithms for IEEE802. 1Qbv Networks. In *Proc. RTN Workshop, ECRTS*, 2017.

[66] N. G. Nayak, F. Duerr, and K. Rothermel. Routing algorithms for IEEE802.1Qbv networks. *ACM SIGBED Review*, 15(3):13–18, 2018.

[67] N. G. Nayak, F. Dürr, and K. Rothermel. Time-sensitive software-defined network (TSSDN) for real-time applications. In *Proc. ACM Int. Conf. on Real-Time Networks and Systems*, pages 193–202, 2016.

[68] N. G. Nayak, F. Dürr, and K. Rothermel. Incremental Flow Scheduling & Routing in Time-sensitive Software-defined Networks. *IEEE Transactions on Industrial Informatics*, 14(5):2066–2075, May 2017.

[69] V. Nguyen, E. Tasdemir, G. T. Nguyen, D. E. Lucani, F. H. P. Fitzek, and M. Reisslein. DSEP Fulcrum: Dynamic sparsity and expansion packets for Fulcrum network coding. *IEEE Access*, 8:78293–78314, 2020.

[70] G. S. Niemiec, L. M. S. Batista, A. E. Schaeffer-Filho, and G. L. Nazar. A survey on FPGA support for the feasible execution of virtualized network functions. *IEEE Communications Surveys & Tutorials*, 22(1):504–525, 2020.

[71] J. Ohms, M. Böhm, and D. Wermser. Concept of a TSN to real-time wireless gateway in the context of 5G URLLC. In *Proc. IEEE Int. Conf. on Wireless Networks and Mobile Commun. (WINCOM)*, pages 1–6, 2020.

[72] M. A. Ojewale and P. M. Yomsi. Routing heuristics for load-balanced transmission in TSN-based networks. *ACM Sigbed Review*, 16(4):20–25, 2020.

[73] M. A. Ojewale, P. M. Yomsi, and B. Nikolić. Multi-level preemption in TSN: Feasibility and requirements analysis. In *Proc. IEEE Int. Symp. on Real-Time Distributed Computing (ISORC)*, pages 47–55, 2020.

[74] R. S. Oliver, S. S. Craciunas, and W. Steiner. IEEE 802.1Qbv gate control list synthesis using array theory encoding. In *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 13–24, 2018.

[75] Z. Pang, X. Huang, Z. Li, S. Zhang, Y. Xu, H. Wan, and X. Zhao. Flow scheduling for conflict-free network updates in time-sensitive software-defined networks. *IEEE Transactions on Industrial Informatics*, 17(3):1668–1678, Mar. 2021.

[76] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner. Enabling fog computing for industrial automation through time-sensitive networking (TSN). *IEEE Commun. Standards Mag.*, 2(2):55–61, 2018.

[77] W. Quan, W. Fu, J. Yan, and Z. Sun. OpenTSN: An open-source project for time-sensitive networking system development. *CCF Transactions on Networking*, 3(1):51–65, 2020.

[78] H. Sándor, B. Genge, and Z. Szántó. Infrastructure and framework for response and reconfiguration in Industry 4.0. In *Proc. IEEE Int. Symp. on Digital Forensic and Security (ISDFS)*, pages 1–6, 2018.

[79] M. Schüngel, S. Dietrich, D. Ginthör, S.-P. Chen, and M. Kuhn. Analysis of time synchronization for converged wired and wireless networks. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 198–205, 2020.

[80] E. Schweissguth, D. Timmermann, H. Parzyjegla, P. Danielis, and G. Mühl. ILP-based routing and scheduling of multicast realtime traffic in time-sensitive networks. In *Proc. IEEE Int. Conf. on Embedded and Real-Time Computing Systems and Appl. (RTCSA)*, pages 1–11, 2020.

[81] M. Seaman. Paternoster policing and scheduling, Revision 2.1. May 2019. Available from http://www.ieee802.org/1/files/public/docs2019/cr-seaman-paternoster-policing-scheduling-0519-v04.pdf, Last accessed May 25, 2019.

[82] P. Shantharama, A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione. LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing. *IEEE Access*, 6:57545–57561, 2018.

[83] P. Shantharama, A. S. Thyagaturu, and M. Reisslein. Hardware-accelerated platforms and infrastructures for network functions: A survey of enabling technologies and research studies. *IEEE Access*, 8:132021–132085, 2020.

[84] W. Steiner, S. S. Craciunas, and R. S. Oliver. Traffic planning for time-sensitive communication. *IEEE Commun. Standards Mag.*, 2(2):42–47, 2018.

[85] M.-T. Thi, S. B. H. Said, and M. Boc. SDN-based management solution for time synchronization in TSN networks. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 361–368, 2020.

[86] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Proc. ICST Int. Conf. on Simul. Tools and Techn. for Commun., Netw. and Sys. & Workshops*, pages 1–10, 2008.

[87] M. Vlk, Z. Hanzálek, K. Brejchová, S. Tang, S. Bhattacharjee, and S. Fu. Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1 Qbv Time-Sensitive Networks. *IEEE Transactions on Communications*, 68(11):7023–7038, Nov. 2020.

[88] C. von Arnim, M. Drăgan, F. Frick, A. Lechler, O. Riedel, and A. Verl. TSN-based converged industrial networks: Evolutionary steps and migration paths. In *Proc. IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA)*, volume 1, pages 294–301, 2020.

[89] C. von Arnim, A. Lechler, O. Riedel, and A. Verl. Fragmentation in reconfigured real-time production networks. In *Annals of Scientific Society for Assembly, Handling and Industrial Robotics*, pages 105–115. Springer Vieweg, Berlin, Heidelberg, 2020.

[90] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. Fitzek. Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working. *IEEE Journal on Selected Areas in Communications*, 37(5):1098–1116, 2019.

[91] J. Yan, W. Quan, X. Jiang, and Z. Sun. Injection time planning: Making CQF practical in time-sensitive networking. In *Proc. IEEE INFOCOM*, pages 616–625, 2020.

[92] Q. Yu and M. Gu. Adaptive group routing and scheduling in multicast time-sensitive networks. *IEEE Access*, 8:37855–37865, 2020.

[93] C. Zhang, Y. Wang, R. Yao, B. Zhou, L. Cheng, Y. Xu, X. Li, J. Cheng, and B. Liu. Packet-size aware scheduling algorithms in guard band for time sensitive networking. *CCF Transactions on Networking*, 3:4–20, 2020.

[94] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang. Mobile edge computing and networking for green and low-latency Internet of Things. *IEEE Communications Magazine*, 56(5):39–45, 2018.

AUTHORS

Ahmed Nasrallah is currently a researcher funded by Kuwait University to pursue the Ph.D. degree in Computer Engineering at Arizona State University, Tempe. He received his B.Sc. degree in Electrical and Computer Engineering from the University of Dayton, Dayton Ohio, and his M.S. degree in Computer Engineering from Arizona State University. His research is focused on communication and multimedia networking.

Venkatraman Balasubramanian received an MS degree in computer engineering from the University of Ottawa, in 2017. He is currently working towards his PhD degree in School of Electrical and Computer Engineering, Arizona State University. His research interests include edge computing, software defined network, wireless communications, and multimedia sharing over wireless networks.

Akhilesh Thyagaturu is an Engineer at Intel Corporation, Chandler, AZ, USA, and an Adjunct Faculty in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe. He received a Ph.D. in electrical engineering from Arizona State University, Tempe, in 2017. He serves as a reviewer for various journals including the *IEEE Communications Surveys & Tutorials*, *IEEE Transactions of Network and Service Management*, and *Optical Fiber Technology*. He was with Qualcomm Technologies Inc., San Diego, CA, USA, as an Engineer from 2013 to 2015.

Martin Reisslein (S'96-M'98-SM'03-F'14) is a Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe. He received a Ph.D. in systems engineering from the University of Pennsylvania in 1998. He currently serves as Associate Editor for the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Education*, and *IEEE Access* as well as *Computer Networks* and *Optical Switching and Networking*. He is Associate Editor-in-Chief for the *IEEE Communications Surveys & Tutorials* and chairs the steering committee of the *IEEE Transactions on Multimedia*.

Hesham ElBakoury is a thirty five year veteran in the telecommunications and data networking industry with an extensive background and expertise in the architecture, design, and development of Distributed Systems and Broadband Access, Enterprise and Telco Communications Systems. He was a Principal Architect in Futurewei focusing on advanced technology research and standards in the network research Lab. Prior to joining Futurewei, Mr. ElBakoury was Chief Systems Architect in Hitachi-CTA EPON Access Systems Division, and Chief Systems Architect in Nortel and Bell-Northern Research where he led the architecture, design, and development of several very successful Switching/Routing, Security and Carrier Ethernet products. In Nortel/BNR, Mr. ElBakoury initiated and led the Autonomic Network research project in the Enterprise division, and the Software Design and Code Reuse Project in the Data Networking division. Mr. ElBakoury has been active in different standard groups, including IEEE 802, IEEE 1904, IETF, ONF, OIF, MEF, the SCTE Energy 2020 program, and CableLabs where he has been heavily involved in IEEE 802.3/802.1, DPoE/DPoG, DOCSIS 3.1, Full-Duplex DOCSIS 3.1, SDN/NFV, Distributed CCAP Architectures, and Business Services projects. He holds an M.S.C. degree from Waterloo University, ONT, Canada.

AN EVALUATION OF CRYPTOCURRENCY PAYMENT CHANNEL NETWORKS AND THEIR PRIVACY IMPLICATIONS

Enes Erdin[1], Suat Mercan[2], Kemal Akkaya[2]
[1]Department of Computer Science, University of Central Arkansas, Conway, AR 72035, [2]Department of Electrical and Computer Engineering Florida International University Miami, FL 33174

NOTE: Corresponding author: Enes Erdin (eerdin@uca.edu)

Abstract – *Cryptocurrencies redefined how money can be stored and transferred among users. However, public blockchain-based cryptocurrencies suffer from high transaction waiting times and fees. To address these challenges, the payment channel network concept is touted as the most viable solution to be used for micro-payments. The idea is exchanging the ownership of money by keeping the state of the accounts locally which provides transaction approvals in seconds. Such attention on payment channel networks has inspired many recent studies that focus on how to design them and allocate channels such that the transactions will be secure and efficient. However, as payment channel networks are emerging and reaching a large number of users, privacy issues are becoming more relevant, this raises concerns about exposing not only individual habits but also businesses' revenues. In this paper, we first propose a categorization of the existing payment networks formed on top of blockchain-backed cryptocurrencies. After discussing several emerging attacks on user/business privacy in these payment channel networks, we qualitatively evaluate them based on a number of privacy metrics that relate to our case. Based on the discussions on the strengths and weaknesses of the approaches, we offer possible directions for research for the future of privacy based payment channel networks.*

Keywords – Bitcoin, blockchain, lightning network, payment channel network, routing protocols

1. INTRODUCTION

There are many modern money exchange systems such as paper checks, credit/debit cards, Automated Clearing House (ACH) payments, bank transfers, or digital cash which are owned and regulated by financial institutions. Nevertheless, in the evolving world of trade, the movement of money is still going through changes. The last decade witnessed the introduction of *Bitcoin* [1], a new paradigm-shifting innovation where the users control their own money without needing a trusted third party. In this model, the users are governing the system by coming to a consensus for controlling the transfer and the ownership of the money. Following the success of Bitcoin, new cryptocurrencies that offer new capabilities were introduced based on the idea of consensus-based account management [2, 3].

Not so long after, the initial success of cryptocurrencies was hindered due to practicality issues in their daily use. Basically, it was a very limited system in terms of scalability and its wide acceptance for simple daily transactions was quite impossible due to high confirmation waiting times, highly disproportional transaction fees, and low throughput.

Among many solutions an *off-chain payment channel* idea arose as a well-accepted one for solving the above-mentioned problems. The idea is based on establishing off-chain links between parties so that many of the transactions would not be written to the blockchain each time. The payment channel idea later evolved towards the establishment of *payment channel networks* (PCN), where among many participants and channels the participants pay through others by using them as relays, essentially forming a connected network. This is in essence a *Layer-2* network application running on top of a cryptocurrency which covers the *Layer-1* services. A perfect example of PCNs is Lightning Network (LN) [4] which uses Bitcoin and reached many users in a very short amount of time. Raiden [5], based on Ethereum, is another example of a successful PCN.

The emergence of PCNs led to several research challenges. In particular, the security of the off-chain payments is very important as users can lose money or liability can be denied. Besides, the efficiency of payment routing within the PCN with a large number of users is tackled. Such efforts paved the way for introducing many new PCNs in addition to LN. These PCNs rely on various cryptocurrencies and carry several new features. As these newly proposed PCNs become more prominent there will be heavy user and business involvement which will raise issues regarding their privacy just as the user privacy on the Internet. The difference is that in many cases, Internet privacy could be regulated but this will not be the case for PCNs as their very idea is based on decentralization. For instance, a user will naturally want to stay anonymous to the rest of the network while a business would like to keep its revenue private against its competitors.

Therefore, in this paper, we investigate this very emerging issue and provide an analysis of current PCNs along with their privacy implications. We first categorize the PCNs in light of common network architectures and

blockchain types. We then define user and business privacy within the context of PCNs and discuss possible attacks on the privacy of the participants. Specifically, we came up with novel privacy risks specific to PCNs. Utilizing these attack scenarios, we later survey and evaluate thoroughly the existing PCNs in terms of their privacy capabilities based on certain metrics. This is a novel qualitative evaluation to be able to compare what each PCN is offering in terms of its privacy features. Finally, we offer potential future research issues that can be further investigated in the context of PCN privacy. Our work not only is the first to increase awareness regarding privacy issues in the emerging realm of PCNs but also will help practitioners on selecting the best PCN for their needs.

The paper is organized as follows: Section 2 gives an introductory background. Next, Section 3 categorizes the PCNs in light of common network architectures and blockchain types. In Section 4 we define user and business privacy, discuss possible attacks on the privacy of the participants in the PCNs, and present an evaluation of state-of-the-art solutions for what they offer in terms of privacy. Section 5 offers directions about the future research on privacy in PCNs and Section 6 concludes the paper.

2. BACKGROUND

2.1 Blockchain

Blockchain is the underlying technology in cryptocurrency, that brings a new distributed database which is a public, transparent, persistent, and append-only ledger co-hosted by the participants. With various cryptographically verifiable methods, called *Proof-of-X* (PoX), each participant in the network holds the power of moderation of the blockchain [6]. As an example, being the first invented and largest cryptocurrency, Bitcoin and the second largest one, Ethereum, which jointly hold 75% of total market capitalization in the cryptocurrency world, utilize a *proof-of-work* (PoW) mechanism where a participant has to find a "block-hash-value" smaller than a jointly agreed number. A block is an element with a limited size that stores the transaction information. Each block holds the hash of the preceding block which in the long run forms a chain of blocks, called, the blockchain. A block is simply comprised of transactions (data), timestamp, nonce, the hash of the block and the hash of the previous block[1] as shown in Fig. 1. The hash of the transaction is inserted into a Merkle tree which enables users to easily verify whether a transaction is in the block or not. "Who-owns-what" information is embedded in the blockchain as transaction information.

In order for a block to be accepted as valid, the hash of the block should be smaller than a number which is decided by considering the total accumulated computational power in the entire network. By changing the nonce value in the block, the miner aims to find a suitable hash result. Soon after a valid block is found, it gets distributed in the network. After the other nodes validate that block, the next block calculation starts.

Therefore, the cohort of independent participants turns blockchain into a liberated data/asset management technology free of trusted third parties.

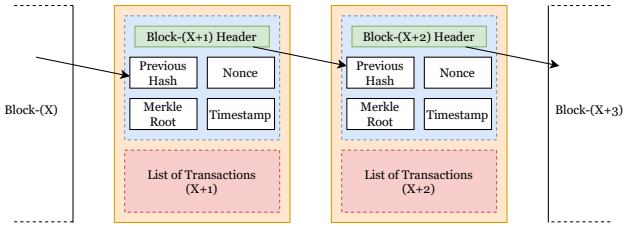


Fig. 1 – Blocks connected with hash.

2.2 Cryptocurrency

Although it finds many areas, the most commonly used application of blockchain technology is cryptocurrencies. A *cryptocurrency* is a cryptographically secure and verifiable currency that can be used to purchase goods and services. In this paper, we will use cryptocurrency and money interchangeably.

Blockchain technology undoubtedly changed the way data can be transferred, stored, and represented. Nonetheless, making a consensus on the final state of a distributed ledger has drawbacks. The first drawback is long transaction confirmation times. For example, in Bitcoin, a block is generated about every 10 minutes. As a heuristic Bitcoin users wait for 6 blocks for the finality of a transaction which yields around 60 minutes of waiting time for finalizing a transaction. In Ethereum, the time between blocks is shorter but users wait 30 consecutive blocks which yield 10-15 minutes of waiting time. Note that, as a block is limited in size, not only the throughput will be limited, but also the total waiting time for the users will be longer during the congested times of the transfer requests. Nevertheless, if a user is in a hurry for approval of a transaction, it will need to pay larger fees to the miners than what its competitors do. This brings us the second drawback of using blockchain for cryptocurrency. The miner nodes, which generate and approve blocks, get fees from the users to include their transactions in blocks. The fee amount is independent of the amount being transacted. During highly congested times, to make a larger profit, miners will be extremely selective in picking the requests from the transaction request pool (*mempool*). So when there is congestion, a payer either has to offer more fees or she/he has to wait more so that a miner picks her/his transaction request.

2.3 Smart contracts

The ability to employ smart contracts is another feature that makes blockchain an unorthodox asset management technology. Smart contracts are scripts or bytecodes, which define how transactions will take place based on the future events defined within the contract. The join-

ing parties will interact under the defined rules to execute the protocol. It provides mechanisms to embed governance rules in a verifiable way that can be audited by the consensus algorithm. It facilitates a complex procedure that involves several third parties. Smart contracts can be utilized in conditional/unconditional peer-to-peer (P2P) transactions, voting, legal testament, etc. As always, the duty of decision-making is on the blockchain. Hence, the blockchain finalizes the transaction outputs when the smart contracts are utilized too.

3. PCNS AND THEIR CATEGORIZATION

3.1 Payment channel networks

Due to scalability issues researchers have always been in the search for solutions to make the cryptocurrency scalable. Among many offered solutions, the *off-chain* payment channel idea has attracted the most interest.

To establish such a channel, two parties agree on depositing some money in a multi-signature (2-of-2 multi-sig) wallet with the designated ownership of their share. The multi-sig wallet is created by a smart contract where both parties sign. The smart contract, mediated by the blockchain, includes the participants' addresses, their share in the wallet, and information on how the contract will be honored. Approval of the funding transaction by the blockchain initiates the channel. Afterward, the idea is simple; the payer side gives ownership of some of his/her money to the other side by mutually updating the contract locally. To close the channel the parties submit the final "commitment transaction" to the blockchain for it to honor the final state of the channel. Thus, each side receives its own share from the multi-sig wallet.

The off-chain mechanism brings a huge advantage such that the peers do not need to publish every transaction on the blockchain. That is, the payments are theoretically instantaneous. Moreover, as there is no need for frequent on-chain transactions, the transactions will be protected from fluctuating, unexpectedly high on-chain transaction fees. In fact, a transaction fee can be zero if the peers agree so.

Payment channels created among many parties make the establishment of *multi-hop payments* from a source to a destination through intermediary nodes possible. As shown in Fig. 2, Alice-Charlie (A-C) and Charlie-Bob (C-B) have channels. Let, A-C and C-B are initialized when `time` is `t`. Although Alice does not have a direct channel to Bob, she can still pay Bob via Charlie. At `time` `t+x1`, Alice initiates a transfer of 10 units to Bob. The money is destined to Bob over Charlie. When Charlie honors this transaction in the C-B channel by giving 10 units to Bob, Alice gives 10 units of her share to Charlie in the A-C channel. When the transfers are over, A-C and C-B channel states get updated. When `time` is `t+x2`, Alice makes another transaction (20 units) to Bob and the shares in the channel states get updated once again.

The multi-hop payment concept enables the establishment of a network of payment channels among users, which is referred to as PCN as shown in Fig. 3. A PCN, in essence, is a collection of payment channels. Going back to the example given in Fig. 2, when Alice wants to transfer X units of money to Bob, they have to find a path between each other in which each channel should have a satisfactory directional deposit so that it can handle the transfer of that amount. For incentivizing the intermediary nodes, the responsible parties can pay forwarding fees to the intermediaries. To prevent intermediaries from stealing the funds a cryptographic hash lock protects the money during the traversal. When an intermediary justifies that it knows the hash of the secret, the channel contract honors the transfer. Hence, when Alice initiates a transfer to Bob she will share the secret with Bob in an out-of-bound communication channel. That will let Bob claim the transfer from the preceding node. In return, the node will learn the secret and like a chain reaction, each node will claim the funds from the preceding node until Alice. Current PCNs vary in terms of what topologies they depend on and which Layer-1 blockchain technology they utilize. We discuss this categorization next. We will then explain each of these PCNs in more detail and categorize them in Section 4.

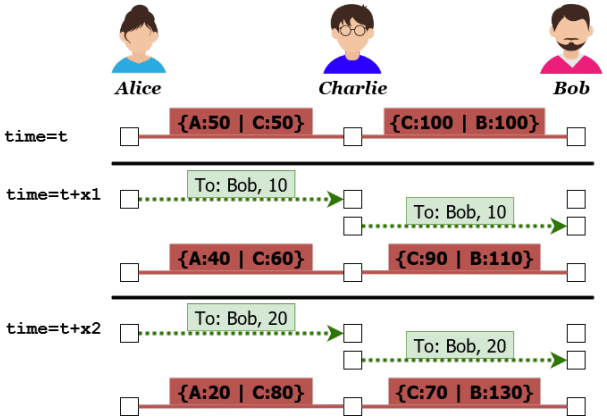


Fig. 2 – A simple multi-hop payment. Alice can initiate a transfer to Bob utilizing channels between Alice-Charlie and Charlie-Bob.

3.2 PCN architectures

In this section, we categorize the types of *network architectures* that can be used in PCNs.

3.2.1 Centralized architecture

In this type of network, there is a central node, and users communicate with each other either over that central node or based on the rules received from the central node as shown in Fig. 4(a). From the governing point of view, if an organization or a company can solely decide on the connections, capacity changes, and flows in the network, then this architecture is called to be a centralized one.

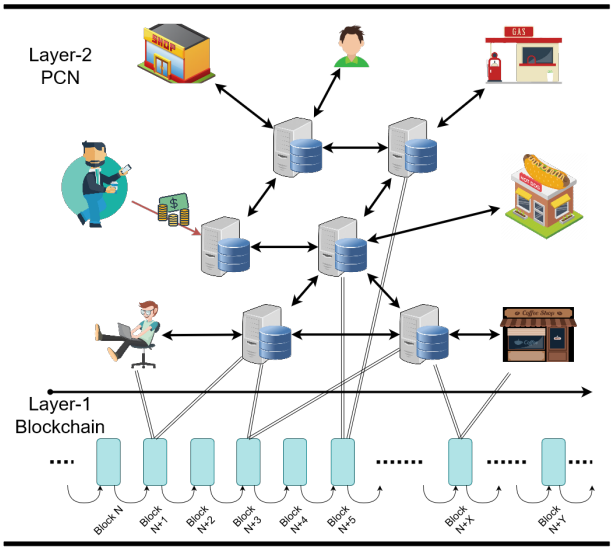


Fig. 3 – The users and businesses independently come together and establish payment channels between each other. Consequently, they form a PCN of end users and relays acting as the backbone. Solid-arrowed lines represent channels between the nodes. Double lines represent how they agreed in the blockchain to establish a channel (only some of these are shown for simplicity).

3.2.2 Distributed architecture

In distributed networks, there is no central node. As opposed to the centralized network, each user has the same connectivity, right to connect, and voice in the network. A sample architecture is shown in Fig. 4(b).

3.2.3 Decentralized architecture

This type of architecture is a combination of the previous two types which is shown in Fig. 4(c). In this architecture, there is no singular central node, but there are independent central nodes. When the child nodes are removed, central nodes' connections look very much like a distributed architecture. However, when the view is concentrated around one of the central nodes, a centralized architecture is observed.

3.2.4 Federated architecture

Federated architecture sounds very much like the federation of the states in the real world and arguably lies somewhere between centralized and decentralized networks. In a federated architecture, there are many central nodes where they are connected in a P2P fashion. Then the remaining nodes (children) strictly communicate with each other over these central nodes which very much looks like a federation of centralized architectures. Moreover, each federation can come up with their local rules in addition to the protocol being used.

3.3 Types of blockchain networks

In this section, we categorize the existing PCNs based on the blockchain type they employ. There are mainly three types of blockchains employed by PCNs.

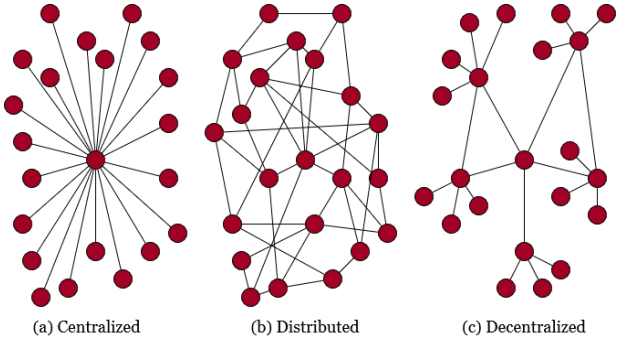


Fig. 4 – Network types

3.3.1 Public blockchain

In a public blockchain, no binding contract or registration is needed to be a part of the network. Users can join or leave the network whenever they want. Consequently, the PCN will be open to anyone who would like to use it.

3.3.2 Permissioned blockchain

Permissioned (i.e., Private) blockchain lays on the opposite side of the public blockchain, where the ledger is managed by a company/organization. Moreover, the roles of the nodes within the network are assigned by the central authority. Not everybody can participate or reach the resources in the permissioned blockchain. PCNs employing permissioned blockchain will be "members-only".

3.3.3 Consortium blockchain

Contrary to the permissioned blockchain, in consortium blockchain, the blockchain is governed by more than one organization. From the centralization point of view, this approach seems more liberal but the governance model of the blockchain slides it to the permissioned side. PCNs utilizing consortium blockchain will be similar to permissioned blockchain in terms of membership but in this case, members will be approved by the consortium.

4. PRIVACY ISSUES IN PCNS: METRICS AND EVALUATION

As PCNs started to emerge within the last few years, a lot of research has been devoted to making them efficient, robust, scalable, and secure. However, as some of these PCNs started to be deployed, they reached a large number of users (e.g., LN has more than 10K users), which is expected to grow further as long as users are satisfied with their services. Such growth brings several privacy issues that are specific to PCNs. In this respect, we observed that strengthening the security in PCNs comes with weaker privacy while strengthening the privacy in PCN makes the network less practical. We argue that very little attention has been paid to these issues and there is a need to identify and understand privacy risks in PCNs from both the users' and businesses' perspectives. Therefore, in this section, we first define these privacy metrics and explain

possible privacy attacks in PCNs. We then summarize the existing PCNs to evaluate their privacy capabilities concerning these metrics for the first time. Our goal is to increase awareness to not only strengthen the privacy features of the existing PCNs but also help designers to consider the privacy-by-design principle when creating new PCNs from scratch. Next, we summarize the state-of-the-art PCN proposals.

4.1 Privacy in PCNs

In its simplest form, *data privacy* or *information privacy* can be defined as the process which answers how storage, access, and disclosure of data take place. For centrally managed systems the central node (or company) is the responsible party for preserving the privacy of the users by defining appropriate policies to manage their data. However, when the system shifts towards a decentralized/distributed one, the privacy of the users should be taken care of by the protocol running beneath the network.

For instance, Bitcoin aims to keep the real identities private utilizing pseudonyms. It is seen that inherited from this philosophy, PCN designers also pay attention to privacy features with different points of view. Nevertheless, we observe that strengthening the security in PCN comes with weaker privacy or strengthening privacy makes the network less practical. The PCN needs to provide services ensuring that the users' data will not be exposed without their authorization. However, the user data travels within the PCN through many other users. Hence, to assess the level of privacy in a particular decentralized system, definitions for privacy within the system are needed. To address these issues, some PCN works aimed to hide the sender (u_s) or the receiver (u_r) identity (i.e., *anonymity*) whereas some others concentrated on strengthening the *relationship anonymity* between the sender and the recipient.

4.2 Attack model and assumptions

There are two types of attackers considered in this paper. The first attacker is an *honest-but-curious* (HBC) where the attacker acts honestly while running the protocols but still collects information passively during operations. The second attacker of interest is the *malicious* attacker that controls more than one node in the network to deviate from the protocols. Hence, it can act based on its own rules, e.g. denial of service or colluding with other nodes in order to learn about the user/payment information. For both of the attacker types, the attacker either tries to learn the origin and the destination of the payment or tries to learn the path of the payment routing. This information can be used for a couple of purposes. The first purpose of trying to get this information is censoring the payment by simply rejecting it. The second purpose is trying to guess the business capacity of a node. The third reason is trying to learn the spending habits of the customers. If a single item is purchased, a persistent attacker will be able to relate the payment to the service or good that has been purchased. The fourth purpose is trying to discredit a particular node simply by slowing down the transaction so that the customers will tend to lose interest in that seller because of a lack of payment usability. These attacker types and how they can situate in the network are shown in Fig. 5 as follows: ① The attacker is on the path of a payment. ② The attacker is not on the path of a particular payment but it can partially observe the changes in the network. ③ The attacker colludes with other nodes, for example, to make packet timing analysis with sophisticated methods.

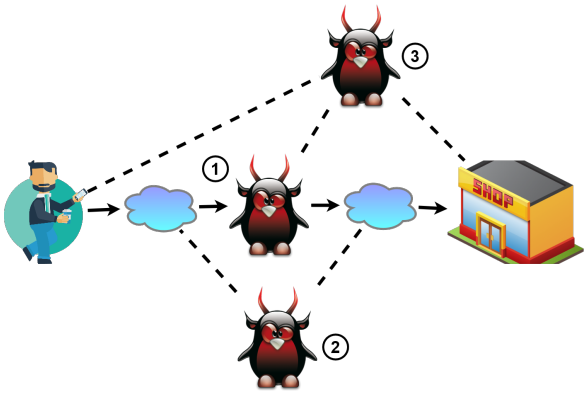


Fig. 5 – Attackers can appear in the network in different places.

Based on these assumptions, we consider the following potential attacks for compromising privacy in PCNs:

- **Attacks on Sender/Recipient Anonymity**: Sender/Recipient anonymity requires that the identity of the sender/recipient (u_s/u_r) should not be known to the others during a payment. This is to protect the privacy of the sender/recipient so that nobody can track their shopping habits. There may be cases where an adversary may successfully guess the identity of the sender/recipient as follows: For case ①, the sender can have a single connection to the network, and the next node is the attacker, hence, the attacker is sure that u_s is the sender. For case ② the attacker may guess the sender/recipient by probing the changes in the channel balances. For case ③ the attacker will learn the sender/recipient if it can carry out a payment timing analysis within the partial network formed by the colluded nodes.
- **Attack on Channel Balance Privacy.** To keep the investment power of a user/business private, the channel capacities should be kept private in PCNs. The investment amount in a channel would give hints about the financial situation of a user or its shopping preferences. Moreover, if the capacity changes in the channels are known, tracing them causes indirect privacy leakages about the senders/recipients. For instance, an attacker can initiate fake transaction requests. After gathering responses from intermediary nodes, it can learn about the channel capaci-

ties. This attack is not necessarily about depleting the channel capacities but guessing the channel capacity of a node. Continuously learning the channel capacities will eventually yield more complicated privacy attacks as discussed in the attack on sender privacy.

- **Relationship Anonymity.** In some cases identities of u_s or u_r may be known. This is a very valid case for retailers because they have to advertise their identities to receive payments. However, if an attacker can relate the payer to the payee, not only the spending habits of the sender but also the business model of the recipient will be learned. In such cases, the privacy of the trade can be preserved by hiding the relationship between the sender and recipient. Specifically, who-pays-to-whom information should be kept private. Some of the PCNs utilize onion-routing to forward the transactions to the destination node. Onion-routing is a source-routing protocol where the source of a message encapsulates the data with the keys of the intermediary nodes like a stacking doll. An intermediary node can remove only one layer from the incoming message to see the next node to which the data is to be forwarded. Hence, in a distributed network, an intermediary node will not confidently be aware of who talks to whom.

- **Business Volume Privacy.** For a retailer, publicly disclosed revenue will yield the trade secrets of its business, which must be protected by the PCN. In that sense, the privacy of every payment is important. Such payment privacy can be attacked as follows: In a scenario where two or more nodes collude, the amount of a transaction can be known to the attacker. In another scenario, if the recipient is connected to the network via a single channel through the attacker, then it will track all of the flows towards the recipient.

4.3 State-of-the-art PCNs and their privacy evaluation

In this section, we briefly describe current studies that either present a complete PCN or propose revisions to the current ones, then analyze their privacy capabilities based on our threat model. We provide a summary of the assessment of the current PCNs' categorizations and privacy features in Table 1. Although our main interest in this paper is specifically payment channel networks, for privacy in permissionless blockchains, the readers are advised to have a look at [7].

4.3.1 Lightning network (LN)

LN [4] is the first deployed PCN that utilizes Bitcoin. It started in 2017 and by June 2020 serves with more than 12.000 nodes and 36.000 channels. Nodes in LN utilize *"Hashed Time-Locked Contracts"* (HTLC) for multi-hop transfer. The directional capacities in the payment channels are not advertised but the total capacity in the channel is known for a sender to calculate a path. This provides a partial channel balance privacy. The sender encrypts the path by using the public keys of the intermediary nodes by utilizing *"onion-routing"* so that the intermediary nodes only know the addresses of the preceding and the following nodes. None of the intermediary nodes can guess the origin or the destination of the message by looking at the network packet.

4.3.2 Raiden network

Shortly after LN, Ethereum foundation announced Raiden Network [5]. Raiden is the equivalent of LN designed for transferring Ethereum ERC20 tokens and provides the same privacy features. Although Ethereum is the second-largest cryptocurrency, that popularity is not reflected well in the Raiden Network. As of June 2020, Raiden serves with 25 nodes and 54 channels. The advantage of Raiden over LN is, due to tokenization, users can generate their own tokens to create a more flexible trading environment.

4.3.3 Spider network

Spider network [16] is a PCN that proposes applying packet-switching based routing idea which is seen in traditional networks (e.g., TCP/IP). However, it is known that in packet-switching the source and the destination of the message should be embedded in the network packet. The payment is split into many micro-payments so that the channel depletion problem gets eliminated. The authors also aimed to have better-balanced channels. In this PCN, there are *spider routers* with special functionalities that communicate with each other and know the capacities of the channels in the network. The sender sends the payment to a router. When the packet arrives at a router, it is queued up until the funds on candidate paths are satisfactory to resume the transaction. The authors do not mention privacy and plan to utilize onion-routing as a future work. The micro-payments might follow separate paths, which would help to keep business volume private if the recipients were kept private. Additionally, the hijack of a router will let an attacker learn everything in the network.

4.3.4 SilentWhispers

SilentWhispers [9] utilizes landmark routing where landmarks are at the center of the payments. In their attack model, either the attacker is not on the payment path or a landmark is HBC. Here, landmarks know the topology but they do not know all of the channel balances. When the sender wants to send money to a recipient, she/he communicates with the landmarks for her/his intent. Then landmarks start communicating with the possible nodes from "sender-to-landmark" to the "landmark-to-recipient" to form a payment path. Each node in the

Table 1 – Qualitative evaluation of privacy features of existing PCNs.

	Network Type	Blockchain Type	Sender Anonymity	Recipient Anonymity	Channel Balance Privacy	Relationship Anonymity	Business Volume Privacy
Lightning Network (HTLC) [4]	Decentralized/ Distributed	Public	◑	◑	◑	◑	●
Raiden Network [5]	Decentralized/ Distributed	All	◑	◑	◑	◑	●
Spider [8]	Decentralized/ Centralized	All	◑	◑	○	◑	◑
SilentWhispers [9]	Decentralized/ Centralized	All	◑	◑	◑	◑	◑
SpeedyMurmurs [10]	Decentralized/ Centralized	Public	◑	◑	◑	◑	◑
PrivPay [11]	Decentralized/ Centralized	Permissioned	◑	◑	◑	◑	◑
Bolt [12]	Centralized	Public	●	●	●	●	●
Fulgor [13]	Decentralized/ Distributed	Public	●	●	◑	●	●
Rayo [13]	Decentralized/ Distributed	Public	◑	◑	◑	◑	●
Erdin et al. [14]	Distributed/ Federated	All	◑	●	◑	◑	●
Anonymous Multi-Hop Locks (AMHL) [15]	Decentralized/ Distributed	Public	○	●	◑	●	●

◑: Partially satisfies OR can not defend against all mentioned attacks.
●: Fully satisfies.
○: Does not satisfy.

path discloses the channel balance availability for the requested transfer amount to the landmarks. Then landmarks decide on the feasibility of the transaction by doing multi-party computation. During the transfer phase, when an intermediary node realizes the transaction to the next node, it informs the landmark. Landmarks acknowledge the transactions and when all of the transactions are executed on the intended path, the transaction is marked successful. In SilentWhispers, the sender and the receiver are kept private but the landmarks know the sender-recipient pair. The payment amount is also private for the nodes who do not take part in the transaction. Moreover, the balances of the channels within the network are kept private. Although centralization is possible, the approach is decentralized and landmarks are trusted parties.

4.3.5 SpeedyMurmurs

SpeedyMurmurs [10] is a routing protocol, specifically an improvement for LN. In SpeedyMurmurs, there are well-known landmarks like in SilentWhispers. The difference of this approach is that the nodes on a candidate path exchange their neighbors' information anonymously. So if a node is aware of a path closer to the recipient, it forwards the payment in that direction, called "shortcut path". In a shortcut path, an intermediary node does not necessarily know the recipient but knows a neighbor close to the recipient. SpeedyMurmurs hides the identities of the sender and the recipient by generating anonymous addresses for them. Intermediary nodes also hide the identities of their neighbors by generating anonymous addresses. Although it may be complex, applying de-anonymization attacks on the network will turn it into SilentWhispers. This is because, while the algorithm is a decentralized approach, with unfair role distribution, it may turn into a centralized approach.

4.3.6 PrivPay

PrivPay [11] is a hardware-oriented version of SilentWhispers. The calculations in the landmark are done in tamper-proof trusted hardware. Hence, the security and privacy of the network are directly related to the soundness of the trusted hardware which may also bring centralization. In PrivPay, sender privacy is not considered. Receiver privacy and business volume privacy is achieved by misinformation. When an attacker constantly tries to query data from other nodes the framework starts to produce probabilistic results.

4.3.7 Rayo and Fulgor

: Rayo and Fulgor [13] are two multi-hop routing protocols for PCNs (Fulgor is suitable for LN only). They develop these protocols against the security flaw coming from hash distribution in LN. Specifically, the same hash of the pre-image is distributed on the path when a payment takes place so the authors argue that this creates a problem for the privacy and relationship anonymity between the sender and the recipient. To solve that problem they introduce multi-hop HTLC contracts. In their non-blocking approach, Rayo a non-blocking payment routing, there is a global payment identifier system that helps the nodes to order the payments with respect to their identifier number. For that reason, Rayo is prone to relationship anonymity attacks if the attacker is located on the payment path. Fulgor aims for guaranteed privacy. The multi-hop HTLC contract offered in Fulgor is fully compliant with the Bitcoin scripts. Thus, it is only usable in LN or Bitcoin-like cryptocurrency backed PCNs. Fulgor's motivation is that in LN the same hash of the pre-image is distributed on the payment path. This creates a privacy problem which by comparing the collected hashes colluding nodes can learn about the path of the payment. Fulgor introduces one more phase of messaging with zero-knowledge proof-based communication. The sender distributes unique hashes to the intermediary nodes. It satisfies balance privacy, business privacy, sender and recipient anonymity.

4.3.8 Bolt

Bolt [12] is a hub-based payment system. That is, there is only one intermediary node between the sender and recipient. Bolt assumes *zero-knowledge proof* -based cryptocurrencies. It does not satisfy privacy in multi-hop payments, however, it satisfies very strong relationship anonymity if the intermediary node is honest. On the other hand, being dependent on a single node makes this approach a centralized one.

4.3.9 Permissioned Bitcoin PCN

In PCNs, if the network topology is not ideal, e.g., star topology, some of the nodes may learn about the users and payments. To this end, the authors in [14] propose a new topological design for a permissioned PCN such that the channels' depletion can be prevented. They come up with a real use case where a consortium of merchants create a full P2P topology and the customers connect to this PCN through merchants which undertakes the financial load of the network to earn money. The privacy of the users in the PCN is satisfied by LN-like mechanisms. The authors also investigate how initial channel balances change while the sender/receiver privacy and the relationship anonymity can be satisfied by enforcing at least 3-hops in a multi-hop payment.

4.3.10 Anonymous Multi-Hop Locks (AMHL)

In the AMHL proposal [15], the authors offer a new HTLC mechanism for PCNs. On a payment path, the sender agrees to pay a service fee to each of the intermediaries for their service. However, if two of these intermediaries maliciously collude they can eliminate honest users in the path and consequently steal their fees. In order to solve this, they introduce another communication phase

in which the sender distributes a one-time-key to the intermediary nodes. Although the HTLC mechanism is improved for the security of the users the sender's privacy is not protected; each of the intermediaries learns the sender. However, relationship anonymity can still be satisfied.

5. FUTURE RESEARCH ISSUES IN PCNS

Privacy in PCNs is an understudied topic and many open issues need to be addressed as future research. In this section, we summarize these issues:

Abuse of the PCN protocols. Most of the PCNs rely on public cryptocurrencies, whose protocol implementations are public. This freedom can be abused such that by changing some parameters and algorithms in the design, an attacker can behave differently than what is expected. This will bring privacy leakages and censorship to the network. A topological reordering of the network will help solve this problem. If a sender gets suspicious about an intermediary node, it can look for alternatives instead of using that node.

PCN topologies. The most widely accepted and readily available solution, Lightning Network, has a user base of more than 12 thousand nodes as of today. Furthermore, if the channels are observed it creates an impression that most of the nodes are experimental to discover the capabilities of LN. Even the trust in the protocol becomes perfect, assuming that ordinary users will put hundreds of dollars in their channels as collateral does not make perfect sense. This reality reminds us that PCNs are inclined to slide towards centrally managed networks. In that case, topology formation comes into the scene. Right now, the `autopilot` feature of `lnd` (an LN client) highlights a scale-free Barabasi-Albert network formation method. However, this method does not take the financial strength of the attendees but only their existence.

Discovery of Colluding Nodes. When the nodes collude in a PCN, they can extract more information about the users. To prevent this, the protocols should be enriched to discover the colluding nodes or by adding redundancy to the protocols, colluding nodes can be confused.

Policy Development. The cryptocurrency and PCN idea is still in the early phases of their lives. Hence, policy and regulation for not only the security of the participants but also for the privacy of them are highly needed in this domain. This will also create a quantitative metric for the researchers to measure the success of their proposals.

Impact of Scalability on Privacy. One of the aims for introducing PCNs was making the cryptocurrencies more scalable. For example, LN advises running the Barabasi-Albert scale-free network model while establishing new connections [17]. Thus, the final state of the network can impose centralization which will have adverse effects on the privacy of the nodes in the network.

Integration of IoTs with PCNs. Use of IoT devices for payments are inevitable. Aside from the fact that most IoT devices are not powerful to run a full node, the security and privacy of the payments and the device identities within the IoT ecosystem need to be studied. These devices are anticipated to be able to participate in the network through gateways. The revelation of device ownership will reveal the real identity of the users to the public which is a big threat to privacy.

Privacy in Permissioned PCNs. While establishing a network of merchants in permissioned PCNs, the merchants should at least disclose their expected trade volume in order to establish a dependable network. This will, however, yield trade secrets of the merchants. To prevent this, zero-knowledge proof based multi-party communication can be explored.

6. CONCLUSION

PCN is a promising solution to make cryptocurrency-based payments scalable. This idea aimed to fix two major shortcomings of cryptocurrencies: long confirmation times and high transaction fees. There are many studies on the design of payment channels and PCNs to make the transfers secure and efficient. However, these studies do not mention the possible privacy leakages of these methods in case of a wide adaptation of proposed ideas. In this paper, we first made the categorization of PCNs based on the type of blockchain being used and the topological behavior of the network. After clearly defining possible privacy leakages in a PCN, we compared and contrasted the state-of-the-art PCN approaches from the privacy point of view.

REFERENCES

[1] Satoshi Nakamoto et al. "Bitcoin: A peer-to-peer electronic cash system". In: (2008).

[2] Ujan Mukhopadhyay, Anthony Skjellum, Oluwakemi Hambolu, Jon Oakley, Lu Yu, and Richard Brooks. "A brief survey of cryptocurrency systems". In: *2016 14th annual conference on privacy, security and trust (PST)*. IEEE. 2016, pp. 745–752.

[3] Florian Tschorsch and Björn Scheuermann. "Bitcoin and beyond: A technical survey on decentralized digital currencies". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2084–2123.

[4] Joseph Poon and Thaddeus Dryja. *The bitcoin lightning network: Scalable off-chain instant payments*. 2016.

[5] *Raiden Network: Fast, cheap, scalable token transfers for Ethereum*. https://raiden.network/. Accessed: 2020-06-06.

[6] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. "SoK: Consensus in the age of blockchains". In: *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 2019, pp. 183–198.

[7] Li Peng, Wei Feng, Zheng Yan, Yafeng Li, Xiaokang Zhou, and Shohei Shimizu. "Privacy preservation in permissionless blockchain: A survey". In: *Digital Communications and Networks* (2020).

[8] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, Giulia Fanti, and Pramod Viswanath. "Routing Cryptocurrency with the Spider Network". In: *Proceedings of the 17th ACM Workshop on Hot Topics in Networks* (2018), pp. 29–35.

[9] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. "SilentWhispers: Enforcing Security and Privacy in Decentralized Credit Networks." In: *The Network and Distributed System Security (NDSS)*. 2017.

[10] Stefanie Roos, Pedro Moreno-Sanchez, Aniket Kate, and Ian Goldberg. "Settling payments fast and private: Efficient decentralized routing for path-based transactions". In: *arXiv preprint arXiv:1709.05748* (2017).

[11] Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Kim Pecina. "Privacy preserving payments in credit networks". In: *Network and Distributed Security Symposium*. 2015.

[12] Matthew Green and Ian Miers. "Bolt: Anonymous payment channels for decentralized currencies". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 473–489.

[13] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. "Concurrency and privacy with payment-channel networks". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 455–471.

[14] Enes Erdin, Mumin Cebe, Kemal Akkaya, Senay Solak, Eyuphan Bulut, and Selcuk Uluagac. "A Bitcoin Payment Network with Reduced Transaction Fees and Confirmation Times". In: *Computer Networks* (2020), p. 107098.

[15] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. "Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability." In: *NDSS*. 2019.

[16] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, Giulia Fanti, and Pramod Viswanath. "Routing cryptocurrency with the spider network". In: *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*. 2018, pp. 29–35.

[17] Stefano Martinazzi and Andrea Flori. "The evolving topology of the Lightning Network: Centralization, efficiency, robustness, synchronization, and anonymity". In: *Plos one* 15.1 (2020), e0225966.

AUTHORS

Enes Erdin is an Assistant Professor at the University of Central Arkansas. He got his Ph.D. in the Department of Electrical and Computer Engineering at Florida International University and he was an NSF CyberCorps Fellow. He conducts research in the areas of hardware security, blockchain technology, and Cyber-physical Systems.

Suat Mercan is a postdoctoral researcher at Florida International University. He received his Ph.D. degree in Computer Science at the University of Nevada, Reno in 2011 and his M.S degree in Electrical and Computer Engineering from the University of South Alabama in 2007. His main research interests are blockchain, payment channel and peer-to-peer networks, cybersecurity, digital forensic, and content delivery.

Kemal Akkaya is a professor in the Department of Electrical and Computer Engineering at Florida International University. He leads the Advanced Wireless and Security Lab and is an area editor of the Elsevier Ad Hoc Networks Journal. His current research interests include security and privacy, and protocol design. He has published over 120 papers in peer-reviewed journals and conferences. He received the "Top Cited" article award from Elsevier in 2010.

CONTROLLER PLACEMENT OPTIMIZATION FOR SOFTWARE DEFINED WIDE AREA NETWORKS (SDWAN)

Lusani Mamushiane[1,2], Joyce Mwangama[2], Albert Lysko[1,2]
[1]Council for Scientific and Industrial Research (CSIR), [2]University of Cape Town, South Africa

NOTE: Corresponding author: Lusani Mamushiane, Lravhuanzwo@csir.co.za

Abstract *– Software Defined Networking (SDN) has emerged as a promising solution to revolutionize network deployment, operations and economic growth. This paradigm aims to address management and configuration complexities in legacy networks so as to reduce the total cost associated with deploying and running telecommunication infrastructures. At the heart of SDN is a controller which oversees orchestration of resources. An important problem that must be addressed during the initial design of an SDN-based network deployment is to find the optimal number of controllers and their locations, to achieve desired operational efficiency. This problem constitutes competing objectives such as latency, load balancing, and reliability. We apply Silhouette Analysis, Gap Statistics and the Partition Around Medoids (PAM) algorithms and, unlike previous work, we add a new method for solving the controller placement problem using an emulation orchestration platform. Our approach aims to optimize controller-to-node latency, alleviate control-plane signalling overhead and ensure control-plane resiliency. Our results for South African national research network (SANReN) reveal that deploying two controllers yields the lowest latency, reduces control-plane signalling overhead and guarantees control-plane resiliency. Our approach can be used by network operators as a guideline to start integrating SDN or plan a new SDN deployment, by helping them make quick automatic decisions regarding optimal controller placement.*

Keywords – Controller placement, Gap Statistics, Partition Around Medoids, Silhouette, Software Defined Networks, South African National Research and Education Network

1. INTRODUCTION

Over the past decade, the use of information and communication technology has reached the upper bounds of Internet penetration [1]. According to a Cisco White paper [2], Internet usage is anticipated to continue on an upward trajectory in the foreseeable future. This strong appetite for Internet access is causing a high demand for bandwidth and putting significant pressure on the existing telecommunication infrastructure. There is a consensus that the current infrastructure will not suffice to cater for these exploding demands [3]. This is primarily attributed to the rigidity of the legacy infrastructure, especially because of vendor lock-in (the use of proprietary silicon hardware) which stifles innovation and makes it difficult to scale the network on the fly. As a result of vendor lock-in, the cost associated with upgrading the infrastructure to cater for the changing traffic patterns is very high, meaning adding new features ad hoc is virtually impossible [4]. Therefore, network operators desiring new features to address their market needs end up beholden to a vendor's upgrade timelines and costs. To cater for the increase in Internet demand, the infrastructure has to evolve from its current monolithic nature to a vendor-agnostic, programmable, cost-effective (in terms of deployment (CapEx) and operational costs (OpEx)) and more flexible infrastructure.

Software Defined Networking (SDN) has emerged as a promising candidate to revolutionize future telecommunication landscapes. Contrary to the traditional network architecture where the control and data-plane of packet processing devices are tightly coupled, SDN presents a paradigm shift in networking by decoupling the control-plane logic from the underlying physical infrastructure [5]. The control-plane is then logically centralized in an external entity called a controller and interacts with the physical infrastructure via its southbound interface. By decoupling the control logic from the physical hardware, operators can programme new traffic engineering policies (such as bandwidth management, security, protection and restoration policies) without worrying about the constraints of closed proprietary hardware and firmware. Moreover, the abstraction of lower level functionality provided by SDN enables convergence of heterogeneous hardware thereby fostering a vendor-neutral ecosystem. In addition to enabling centralized network provisioning and holistic network management, SDN promises benefits such as security granularity (by providing a central point of control to holistically and consistently disseminate security information), savings in operational costs (by automating network administrative tasks), savings in capital expenditures (by capitalizing on commodity hardware) and cloud abstraction (which is critical to consolidate and facilitate the management of massive data centers) [6]. According to [7] a huge portion of operational expenditure is from costs related to the

management and configuration of the telecommunication infrastructure. Therefore, leveraging SDN to automate management and configuration tasks is likely to improve the return on investment (ROI).

This work presents a framework which can be used by network operators to optimize their SDN controller placement during the deployment phase.

The paper is organized as follows: Section 2 presents the problem being addressed by this work, Section 3 describes related work and their drawbacks, and highlights our contributions, Section 4 describes the algorithms (based on mathematical modelling) used to solve the controller placement problem, Section 5 provides implementation details of the algorithms, Section 6 presents results from our mathematical modelling, Section 7 describes the emulation experiments conducted to verify the outcome of the mathematical modelling, Section 9 concludes the paper. Lastly, Section 10 describes future work.

2. PROBLEM STATEMENT

Although Local Area Networks (LANs) like Data Center Networks (DCNs) have already benefited from SDN, deploying SDN in real Wide Area Networks (WANs) still poses several design challenges. As the centralized brain of the network, an SDN controller must be able to respond to control requests promptly. Moreover, control tasks such as data-plane monitoring, must be performed as efficiently as possible to maintain up-to-date state information. This requires optimization on the southbound interface. Due to the significant influence of propagation latency (switch-to-controller latency) on WAN performance, controller placement has emerged as a crucial design problem that influences SDN's southbound performance. Controller placement defines the location of SDN controllers relative to the data-plane elements, that yields better network perfomance.

Another aspect to controller placement has to do with the number of controllers deployed in a given WAN. Deploying a certain number of controllers has an impact on several objectives such as propagation latency and reliability. Even though the number of controllers may be known in advance, the location of these controllers usually needs to be optimized to meet user requirements and contraints.

Therefore, the overall problem that must be addressed is: given a real SDN-enabled WAN, how many SDN controllers are needed and where should they go to optimize user-defined requirements and constraints while maintaining an acceptable runtime and accuracy. This is a multi-objective optimization problem and constitutes competing objectives. It is necessary to address this problem during the early stages of SDN planning.

3. STATE OF THE ART

This section presents an analysis of state–of–the–art controller placement solutions.

3.1 Related work

To date, there has been numerous research studies directed towards addressing the controller placement problem in SDN. These can be broadly classified into two categories: (i) studies that implemented exhaustive algorithms, as exampled by [8] –[9] and (ii) studies that implemented heuristic algorithms, as exampled by [16] –[30].

The controller placement problem was first introduced by Heller et al. [8] in 2012. The authors study the controller placement problem by investigating the impact of uncapacitated controller location on average and worst–case latency. The algorithm used in this study is k–center. To maintain realism, the authors tested their algorithm on the Internet2 OS3E topology [10]. Their results indicate that increasing the number of controllers decreases the overall network latency with a significant trade-off between worst–case and average latency. The authors conclude that deploying one controller often suffices to meet existing latency requirements in campus networks. Expectantly, they also argue that one controller is not sufficient for large-scale deployments with fault tolerant requirements.

Hu et al. [11] proposes the use of multiple controllers to ensure reliability in the control-plane. To optimize controller placement, the authors carry out a comparative evaluation of optimization algorithms, namely random placement, l–w greedy and brute force. They focus their reliability metric on the "expected percentage of valid control paths", where a control path is defined as the interface between the switch and controller (southbound interface) as well as the connection between controllers (east/westbound interface). The algorithms were evaluated on Internet2 topology as well as various ISP topologies from the Rockefuel database [12]. From their simulations, random placement produced the least optimal results, while brute force produced optimal results after a significantly long runtime. As a result, the authors recommend the l–w greedy as the most optimal solution. This work is similar to Gong et al. [13] in that they both aim to optimize reliability in the event of node or link failure. However, latency (both switch–to–controller and inter–controller latency) and load balancing are not considered in these research works. Moreover, the number of controllers is assumed to be known in advance.

Tanha et al. [9] study the controller placement problem to optimize network resilience in the event of controller

failure while considering network deployment costs and satisfying switch–to–controller latency. In order to mimic a production scenario, the authors take into account the capacity of the controller and assume a varying switch load. To maintain realism, they assessed their algorithms on real tier–1 service provider topologies. The outcome of their experiments demonstrated that controller resiliency is topology dependent. The drawback of this solution is that it is resource intensive and only ideal for small and medium network instances. The algorithm used in this study is the capacitated k–center algorithm.

The research work of Yao et al. [14] proposes a heuristic algorithm for capacitated controller placement in consideration of the switch–to–controller latency and traffic load of switches. The main objective of this work is to optimize controller load balancing under heterogeneous data-plane load while minimizing switch–to–controller latency. Resiliency is handled by deploying additional controllers in the network. The main shortcoming of this solution is that it is less accurate in larger deployments and therefore applicable only for small–scale networks.

Jimenez et al. [15], also proposes a capacitated controller placement solution to optimize load balancing. Contrary to Yao et al., this work is not limited to the size of the network and propose a divide and conquer philosophy to achieve scalability and robustness. Moreover, authors assume homogeneous traffic load on the data-plane. The solutions proposed by Jimenez et al. and Yao et al. optimize controller placement based on fixed traffic observed initially, but do not adapt to the changing traffic load. This shortcoming is addressed by Bari et al. [16] and Jourjon et al. [17] who propose a heuristic algorithm for dynamic controller placement i.e. controller placement based on current data-plane load. The metrics considered are switch–to–controller latency and controller processing load. The solutions proposed rely on trial and error and are not as reliable. Sanner et al. [18] propose a genetic algorithm leveraging the Non-dominated Sorting Genetic Algorithm (NSGA) II framework to optimize load balancing and inter–controller latency. Authors conclude that their solution consumes a lot of CPU resources and is only ideal for small and medium–sized networks.

Rath et al. [19] propose a Non–Zero–Sum game theory approach to optimize controllers' utilization. In this solution, controllers can be added or removed dynamically and can also go to sleep mode occasionally based on the traffic load on the controllers. This solution is intended to reduce network deployment costs (by minimizing the number of controllers deployed) and operational costs (by optimizing energy consumption through on–demand controller deployment). This solution ignores controller placement in the network. Sallahi et al. [20] propose a mathematical formulation to find the optimum number of controllers to deploy. However, their approach suffers the same shortcoming as that proposed by Rath et al. in that it does not determine the optimal controller placement. Furthermore, both these research works are limited to small–scale networks.

Wendong et al. [21] study the trade-off between reliability and latency using random placement, l–w greedy and simulated annealing. The results suggest that simulated annealing yields the most optimal solution in comparison with the other approaches. The outcome of the trade-off analysis indicate a significant trade-off between reliability and latency. Authors argue that the number of controllers must be chosen carefully. They demonstrate that using too few controllers has an adverse effect on reliability while using too many controllers can result in a broadcast storm on the east/westbound interface.

Hock et al. [22] and Lange et al. [23] advocate for careful consideration of latency (controller–to–controller) and reliability (defined as resiliency in the event of a node or link failure and control-plane load balancing) during controller placement. This work proposes a resilient Pareto–based Optimal Controller placement framework to achieve optimal controller placement. The authors use load imbalance as the key metric, which is the difference between the controller having more switch assignments and the controller having fewer switches under its supervision. The results from this work indicate that the optimal solution is achieved when 20% of all network nodes are controllers. The downside of this solution is that, instead of partitioning the network into small administrative domains, the authors treat the network as a whole with controllers working collaboratively. This means the controllers frequently share their network state information with their peers to maintain an accurate global view. This increases the probability of incurring a network broadcast storm which increases inter–controller latency. Therefore, this solution is restricted to small and medium–scale network instances. Furthermore, this solution ignores the average switch–to–controller latency which is a critical parameter in SDN.

Ksentini et al. [24] consider three objectives in optimizing controller placement: (i) switch–to–controller latency, (ii) inter–controller latency and (iii) control-plane load balancing simultaneously. The authors propose a bargaining game–based algorithm to optimize controller placement. Authors claim that their results outperform other mono–objective–based controller placement results. However, their algorithm is only suitable for small–scale networks and is less accurate for larger network instances.

Last but certainly not least, He et al. [26] formulate a controller placement model to optimize flow setup time,

where flow setup time is the total amount of time taken by the controller to install a flow instruction on the switch's flow table. The authors argue that dynamic controller placement is necessary to help reduce flow setup time. The results from this work reveal that, for low flow densities, dynamic controller placement can reduce the flow setup time by up to 50% in comparison with static controller placement. However, for high flow densities, static controller placement produced better results.

As demonstrated by Heller et al. [8], Hock et al. [22] and Wendong et al. [21], there exists a significant trade-off between load balancing, reliability (also known as resiliency) and latency. Therefore it is almost impossible to optimize one objective without sacrificing the other. This study attempts to address the controller placement problem in consideration of switch-to-controller latency metric. This metric has emerged as an important QoS determinant in SDN. This is primarily because the communication between the controller and data-plane has to be seamless to ensure an accurate view of the network state and prompt data-plane flow installations.

Table 1 provides a summary of the state of the art in research pertaining to SDN controller placement.

3.2 Contribution

From the state-of-the-art review, it is apparent that most studies (with the exception of the work by Sallahi et al. [20]) assume the number of controllers to be known in advance. However, the model proposed by Sallahi et al. is ideal to plan a small–scale SDN and runs out of memory when solving larger instances. Moreover, most studies relied on heuristic algorithms to reduce algorithm runtime. However, this is achieved at the expense of solution accuracy. To the best of our knowledge, the only research studies that implement exhaustive algorithms are by Heller et al. [8] and Tanha et al. [9]. Both Heller et al. and Tanha et al. propose the use of *k*-center to solve the controller placement problem. However, *k*-center is sensitive to outliers and does not always consistently yield accurate results [27]. Perhaps more importantly, there is currently no analysis of the controller placement problem purely using an emulation platform to mimic a real SDN deployment. Most studies relied on mathematical modelling to address the controller placement problem, making it difficult to verify validity and reliability of the results.

Controller placement is a network planning problem, and is normally not time sensitive. Consequently, this study proposes exhaustive algorithms to optimize solution accuracy. In order to find the best locations

Table 1 – Classification of existing controller placement solutions

Solution	Topology(s)	Scale of Network	Environment	Algorithm(s)	Placement Metric(s)	Network Partitioning
Heller et al. [8]	Internet2 OS3E	Large–scale	Static	k–center	average switch–to–controller latency worst–case latency	No
Hu et al.[11]	Internet2 OS3E	Small and medium-sized	Static	l–w greedy	Reliability	No
Tanha et al. [9]	Sprint ATT NA PSINET UUNET	Large-scale	Static	Capacitated k-center	switch–to–controller latency Reliability	No
Yao et al. [14]	Internet Zoo	Large–scale	Dynamic	Linear relaxation	switch–to–controller latency Load balancing	No
Jimenez et al. [15]	Sparse Medium Dense	Large–scale	Dynamic	k–critical	Load balancing	Yes
Bari et al. [16]	RF-I	Large–scale	Dynamic	DCP-GK	switch–to–controller latency Load balancing	Yes
Jourjon et al. [17]	Not discussed	Large–scale	Dynamic	LiDy+	switch–to–controller latency Load balancing	Yes
Sanner et al. [18]	Internet2 OS3E	Large–scale	Dynamic	NSGA	inter–controller latency load balancing	Yes
Rath et al. [19]	Random network with 28 switches	small-scale	Dynamic	Non–zero–Sum Game	Load balancing	No
Sallahi et al. [20]	Random network with 10, 20, 30, 40, 50, 75, 100, 150 switches	small-scale	Dynamic	CPLEX	Load balancing	No
Wendong et al. [21]	Internet2 OS3E	Large–scale	Static	l–w greedy	switch–to–controller latency Reliability	No
Hock et al. [22]	Internet2 OS3E	Small and medium–sized	Static	POCO	switch–to–controller latency Reliability Load balancing	No
Lange et al. [23]	Internet2 OS3E Internet Zoo	Large–scale	Dynamic	Simulated Annealing	switch–to–controller latency Reliability Load balancing	No
Ksentini et al.[24]	Ring Binary Tree	Large–scale	Static	No specific name	switch–to–controller latency Inter–controller latency Load balancing	Yes
Mamushiane et al.[25]	SANReN	Small-scale	Static	Partition Around Medoids (PAM) Gap Statistics Silhouette Analysis Johnson's Algorithm Emulation	average switch–to–controller latency worst–case latency switch-to-controller balancing propagation +queuing + processing latency signalling overhead	Yes

to place SDN controllers, this study proposes the use of a classical machine learning algorithm called Partition Around Medoids (PAM) [28]. To determine the optimal number of controllers to deploy given a wide area network, this study proposes the use of Silhouette [29] and Gap Statistics [30] algorithms. To mimic a real SDN deployment, the controller placement problem is studied using an emulation orchestration platform. This is something that to the best of our knowledge has not been done, and we consider it necessary to verify the outcome of the mathematical modelling. Finally, a mechanism to manage control-plane overhead is proposed.

The key performance indicators used to gauge network performance are: (i) network latency (propagation + queuing + processing latency), (ii) reliability (in the event of link and/or node failure) and (iii) control-plane signalling overhead.

4. MATHEMATICAL MODELLING

In order to compute the optimal number of controllers, we propose two "unsupervised" machine learning approaches, namely Silhouette and Gap Statistics. Unsupervised algorithms learn from input data that has no labeled responses [31]. These algorithms are classically used to analyze cluster quality through the metric of minimum distances between data points. In the context of controller placement, we leverage these algorithms to find the number of controllers that minimizes overall network propagation latency (i.e. switch-to-switch latency). To find the best locations for these controllers, we extend and apply a facility location algorithm called Partition Around Medoids algorithm (PAM), with propagation latency (i.e. controller-to-switch latency) as our main objective function. For realism, we use the South African National Research Network (SANReN) as a case study. The choice of this topology was mainly motivated by the fact that it represents the emerging market case study which is the key use case of this study.

Since the links between SANReN's switches are known to be fibre where speed is approximately the speed of light in fibre (i.e. $2 \times 10^8\ m/s$), we compute propagation latency by taking the ratio of average distance (between nodes) to speed of light in fibre. The distances are calculated using the Harvesine approach [32]. The results from our simulations and discussions are also presented in this section.

4.1 Assumptions

The following assumptions apply to the proposed algorithms:

- Switch-to-controller communication is assumed to happen out-of-band (control and regular traffic do not share the same links) ;
- The bandwidth for all connection links is constant;
- Control path security and reliability has been perfectly solved;
- Controllers are co-located with some of the switches;
- Switches incur a fixed load.

4.2 Optimal number of controllers

This section introduces the algorithms used to find the optimal number of controllers to deploy given a wide area network. Table 2 defines some of the notation used in this section.

Table 2 – Mathematical symbols

Symbol	Definition
C_k	k_{th} cluster
$L(C_k)$	Intra-cluster propagation latency variation
$G(V,E,X)$	Network topology graph
V	Data-plane nodes
E	Links between nodes
X	Geographic locations of nodes
φ	Latitude of a node
λ	Longitude of a node
r	Radius of the earth
k	Number of clusters
B	Randomly generated reference data set of network topology
s	Standard deviation

4.2.1 Silhouette analysis

Silhouette Analysis is a method of interpretation within existing clusters, used to measure the quality of a cluster (how close each point in a cluster is to its adjacent clusters) for a varying number of partitions [33]. In the context of the controller placement problem, we adopt and extend this algorithm to answer this question: given a wide area network topology, how many controllers are needed to achieve minimum intra-cluster propagation latency variation? Eq. *(1)* shows our objective function.

$$X = min \sum_{k=2}^{n} L(C_k) \qquad (1)$$

Algorithm 1 outlines the Silhouette approach. The algorithm requires three input parameters, namely a clustering algorithm (clustAlg) to cluster network data-plane nodes, distance function handle (disfun), network topology graph ($G(V,E,X)$), where V denotes data-plane nodes (switches), E denotes edges (links between nodes), and X denotes the geographic locations (longitude, latitude) of nodes), and maximum number of controllers (maxNumControllers). The clustering algorithm used is called Partition Around Medoids (PAM)

Algorithm 1 Silhouette Analysis

Require: $G(V,E,X)$, $maxNumControllers$, $disfun$, $clustAlg$
1: $totalNodes \leftarrow G(V,E,X).size()$
2: $k \leftarrow 2$
3: **for** $k \leftarrow 2$ *to* $maxNumControllers$ **do**
4: $clusters \leftarrow Cluster.train(G(V,E,X), k, disfun, clustAlg)$
5: **for** $j \in G(V,E,X)$ **do**
6: $intraClustVar \leftarrow clusters.computeCost(j)/totalNodes$
7: $centroides \leftarrow sc.parallelize(clusters.clusterCenters)$
8: $clusterCentroids \leftarrow Cluster.train(centroides)$
9: $interClustVar \leftarrow clusterCentroids.computeCost(centroides)/k$
10: **end for**
11: $Silhouette \leftarrow (interClustVar - intraClustVar)/max(intraClustVar, interClustVar)$
12: **end for**

described in Section 4.3.2 [33]. The Harvesine distance approach was used to compute the great circle distances between pairs of switches [34]. The great circle distance is the shortest distance between two locations on a sphere, measured along the surface of the sphere (as opposed to the ordinary Euclidean distance)[35] [36] . An alternative method to compute geographic distances is the Law of Cosines, which is optimal for shorter distances and is not as accurate for longer distances [37]. To compute the great circle distance, Eq. *(2)* which defines the Harvesine approach is used, where φ_1 and φ_2 are the latitudes of points 1 and 2 respectively, λ_1 and λ_2 is the longitudes of point 1 and 2 respectively and r denotes the radius of the earth, a constant equal to 6371 km.

$$Distance = 2(r)arcsin\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + cos(\varphi_1)cos(\varphi_2)sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \quad (2)$$

The procedure to compute the optimal number of controllers using Silhouette (with steps/instructions enumerated from 1 to 12 in Algorithm 1) is as follows: First, a cluster model is created from input network data using PAM and Harvesine approach (Instruction 4). Next, the average propagation latency from each switch to its cluster centroid is calculated (Instruction 6), to find the intra-cluster propagation latency variation (intraClustVar). To this end, a model from the centroids is created (Instruction 7). Next, the average propagation latency between each centroid to the global centre (Instruction 8-9) is calculated. In this way we obtain the inter-cluster propagation latency variation (interClustVar). The last step is to calculate the silhouette coefficient (Instruction 11). This procedure is repeated as specified by the maxNumControllers input parameter in order to calculate the silhouette coefficient for each number of controllers. Moreover, for each number of controllers (Instruction 3), the number of iterations was set to 20 to maximize accuracy of the results.

The optimal number of controllers is one that yields the maximum silhouette coefficient. This coefficient has a range of [-1,1]. Therefore a value closer to +1 is preferred as it indicates better cluster configuration.

4.2.2 Gap statistics

Similar to Silhouette Analysis, Gap Statistics is a partition algorithm typically used in neural networks, to measure the quality of clustering measure based on average intra-cluster variation [38] [30]. We adopt and enhance this algorithm to verify the results from our Silhouette Analysis. Therefore our goal is to determine the optimal number of SDN controllers to deploy given a network topology, and compare the outcome of the simulation with the results from the Silhouette Analysis.

The Gap Statistics algorithm constitutes the following steps (enumerated by instructions from 1 to 12 in Algorithm 2): First the network topology is partitioned (using the PAM algorithm), by varying the number of controllers k (which corresponds to the number of clusters) from 2 to the maximum user-defined value (Instruction 3).

Algorithm 2 Gap Statistics

Require: $G(V,E,L)$, $maxNumControllers$, $disfun$, $clustAlg$, $nrefs$
1: $gaps \leftarrow [\,]$ {Intialize empty array}
2: $k \leftarrow 2$
3: **for** $k \leftarrow 2$ *to* $maxNumControllers$ **do**
4: $intraClusVar \leftarrow clustAlg(G(V,E,L), maxNumControllers, disfun)$
5: **for** $i \in nrefs$ **do**
6: $rRef \leftarrow random(G(V,E,L)$
7: $intraClusVarRef \leftarrow clustAlg(rRef, disfun)$
8: $gap \leftarrow log(intraClusVarRef - intraClusVar)$
9: **end for**
10: $s_k \leftarrow standardDev(rRef, k, disfun)$
11: **return** $gap \leftarrow gap.argmax$ {Take maximum gap value}
12: **end for**

This is followed by the computation of the average intra-cluster propagation latency variation (intraClusVar denoted by $L(C_k)$ in Eq. *(3)*) between the switches (Instruction 4). Next a reference data set ($rRef$ denoted by B in Eq. *(4)*) of the network topology is randomly generated (Instruction 6). The average intra-cluster latency variation of the reference data set (intraClusVarRef denoted by $L^*(C_{kb})$ in Eq. *(4)*) is computed (Instruction 7). The Gap Statistics is calculated using Eq. *(3)* and *(4)*. Finally, the standard deviation of B Monte Carlo replicates is calculated [30]. The optimal number of controllers is one that meets the condition in Eq. *(5)*, where s_{k+1} denotes the standard deviation of B Monte Carlo replicates.

$$gap_n(k) = E_n^* log(L^*(C_k)) - log(L(C_k)) \qquad (3)$$

where

$$E_n^* log(L^*(C_k)) = \frac{1}{B}\sum_b log(L^*(C_{kb})) \qquad (4)$$

$$gap(k) \geq gap(k+1) - s_{k+1} \qquad (5)$$

4.3 Optimal controller location

This section describes the algorithms used to find the best locations to place SDN controllers.

4.3.1 Johnson's algorithm

In order to determine the best locations to place SDN controllers in a WAN, the shortest paths between each pair of switches must be known. Johnson's algorithm [39] provides a means to find the shortest paths between node pairs and has become a popular method for addressing SDN optimization problems [40]. Therefore, we used the results from this algorithm alongside the PAM algorithm to determine the best places to deploy controllers. A pseudocode of this algorithm is as shown in Algorithm 3 and consists of the following steps: First a new arbitrary switch (denoted by q) is added to the network graph, connected by zero-weight links to all other switches (denoted by v) in the network graph (Instructions 1-5). If this step detects a negative weight-cycle (i.e. a cycle whose weight sums to a negative number), the algorithm is terminated (Instruction 6-7). Second, a single source shortest path algorithm called Bellman-Ford algorithm is evoked, to find the shortest path $h(v)$ from each switch v in the network to the new switch (Instructions 9-11). Next, the graph is reweighted to find new link weights w_{new} (Instruction 12-14). Finally, the new switch is removed, and Dijkstra's algorithm is used to compute the shortest paths $p(u, v)$ from each each node to every other node in the reweighted graph (Instructions 15-22).

Algorithm 3 Johnson's Algorithm

Require: $G(V, E)$ {undirected weighted network graph}
1: Compute G' $where$ $V[G'] \leftarrow V[G] \cup q$ {G' is a new graph containing q}
2: **for** $v \in V[G]$ {for all switches (v) in the original graph} **do**
3: $E[G'] \leftarrow E[G] \cup (q, v) : v \in V[G]$
4: $z(q, v) \leftarrow 0$
5: **end for**
6: **if** $BELLMAN - FORD(G', w) == False$ **then**
7: **print** Error! Negative cycle detected.
8: **else**
9: **for** $v \in V[G]$ **do**
10: set $h(v) \leftarrow \delta(q, v)$ compute shortest path using Bellman-Ford
11: **end for**
12: **for** $(u, v) \in E[G']$ **do**
13: $w_{new} \leftarrow w(u, v) + h(u) - h(v)$
14: **end for**
15: **for** $u \in V[G]$ **do**
16: execute $Dijkstra(G, w_{new}, u)$ to compute $\delta_{new}(u, v)$ for all $v \in V[G]$
17: **for** $v \in V[G]$ **do**
18: $p(u, v) \leftarrow \delta_{new}(u, v) + h(u) - h(v)$
19: **end for**
20: **end for**
21: **end if**
22: **return** shortest path matrix

4.3.2 Partition around medoids (PAM)

After determining the optimal number of controllers to use given a WAN topology, the next step is to find the best locations to place the controllers such that the QoS is maximized. This can be achieved by leveraging "unsupervised" machine learning heuristic algorithms (such as Simulated Annealing [41] and Clustering LARge Applications (CLARA)[42]) or exhaustive algorithms (such as k-means [43] [44] and PAM [45] [46][47]. However, heuristic algorithms are suboptimal in the sense that they are primarily focused on optimizing runtime over solution accuracy. Therefore, heuristic algorithms are more ideal for scenarios requiring dynamic controller placement. However, this study explores static controller placement, where the controller placement problem is addressed during network planning. Therefore, the accuracy of the optimization algorithm is significantly more important than the speed of computation. From the available exhaustive algorithms, we opted for the PAM algorithm. This is mainly because the k-means algorithm is very sensitive to outliers which can lead to solution inaccuracy [48]. Unlike k-means, PAM is more stable and more accurate [49].

Algorithm 4 describes the steps we followed to compute the optimal locations of SDN controllers. Our approach assumes co-location of controllers and switches. First, k

arbitrary switches (where k is the number of controllers to place)) are selected as the potential controller locations (Instruction 3). This is followed by the association of each switch to the closest controller (Instructions 4-6). While the cost of configuration (the overall propagation latency) decreases, the controller location R_i and switch S_o are swapped (Instructions 7-9), and each switch is reassigned to their closest controller location (Instructions 4-6).

Algorithm 4 Partition Around Medoids (PAM)

Require: $G(V,E)$, $NumControllers$, $disfun$, $clustAlg$, $edgWeigts$

1: Compute shortest path matrix using Johnson's algorithm
2: $k \leftarrow NumControllers$
3: $R_{i\ (i\in[1..k])} \leftarrow$ randomly select k objects from $G(V,E,)$
4: **for** $S_o \in G(V,E)$ **do**
5: Compute similarity score of S_o with each $R_{i\ (i\in[1..k])}$ using $disfun$
6: Associate S_o to the most similar R_i
7: **end for**
8: **for** S_o and R_i **do**
9: $swapCost \leftarrow computeCost(S_o, R_i)$
10: **end for**
11: **if** $swapCost \leq 0$ **then**
12: $S_o \leftrightarrows R_i$
Go back to step 4
13: **else**
14: **for** $S_o \in G(V,E)$ **do**
15: Compute similarity score of S_o with each $R_{i\ (i\in[1..k])}$
16: Assign S_o to the most similar R_i
17: **end for**
18: **end if**
19: **return** cl

If an increase in configuration cost is detected, the swap is undone and the optimal controller locations that optimize QoS are found (Instructions 12-18). Two QoS parameters are considered in our solution, that is the average propagation latency (which is the overall propagation latency) and the worst-case propagation latency (which is the maximum network latency). Eq. *(6)* and *(7)* define how these parameters are defined, where $L_{avg}(Z')$ is the average latency, $L_{wc}(Z')$ is the worst-case latency, $d(v,z)$ is the shortest distance from the switch (node $v \in V$) to the controller (node $z \in Z$), $N = |V|$ denotes the number of nodes and 2×10^8 is the speed of light in fibre.

$$L_{avg}(Z') = \frac{1}{(2X10^8)N} \sum_{v\in V} \min_{z\in Z'} d(v,z) \qquad (6)$$

$$L_{wc}(Z') = \max_{v\in V} \min_{z\in Z'} d(v,z) \qquad (7)$$

5. IMPLEMENTATION OF MATHEMATICAL MODELLING

This section explains our implementation for solving the controller placement problem using the algorithms described in sections 4.2 and 4.3. These algorithms are implemented in MATLAB 2018b. The primary objective is to establish the number of controllers for the achievement of minimum propagation latency and to determine the best locations to place these controllers in a WAN topology. The results from our simulation experiments are also presented in this section.

5.1 Topologies

To maintain realism, our proposed solution is applied on a real-world WAN called South African Research Network (SANReN), operated by CSIR'S Next Generation Enterprises and Institutions (NGEI) cluster [50]. The reason for choosing the SANReN network was so that we could demonstrate our proposed solution on an emerging market use case. However, it may be noted that our solution is topology-agnostic and can easily be used to test other networks of different configurations and sizes.

The SANReN network constitutes a core national backbone, with each Point of presence (PoP) connecting a metropolitan network. This work only focuses on the PoP-level instead of the router-level view of the SANReN network. This is because the router-level view is proprietary and not publicly available. Moreover, the PoP-level view has been deemed more useful [51] for several points: it provides a larger scale view of network links, which are most interesting for network optimization; it shows end users where they can connect to the network and it's the level where resiliency and redundancy are critical. The PoP-level geographical map of the SANReN topology comprises 7 nodes and 7 fibre links configured in a ring topology. The data set of this topology was downloaded from a repository called The Internet Topology Zoo [52]. The format of the data set is in Geography Markup Language (GML) and includes geographic locations (longitude, latitude) of nodes and topological configuration of the SANReN network.

5.2 Hardware and software used for modelling

All the experiments have been executed under an Ubuntu Desktop 16.04 LTS-64 bit on a PC with the following specification: Intel(R) Core(TM) i7-5600U CPU, with 4 cores (8 threads), a clock speed of 2.60 GHz, RAM amount of 8 GB and a storage capacity of 250 GB.

5.3 Flowchart of proposed solution

The flowchart depicted in Fig. 1 summarizes the steps in our proposed controller placement solution. First, network graph modelling is used to model the network

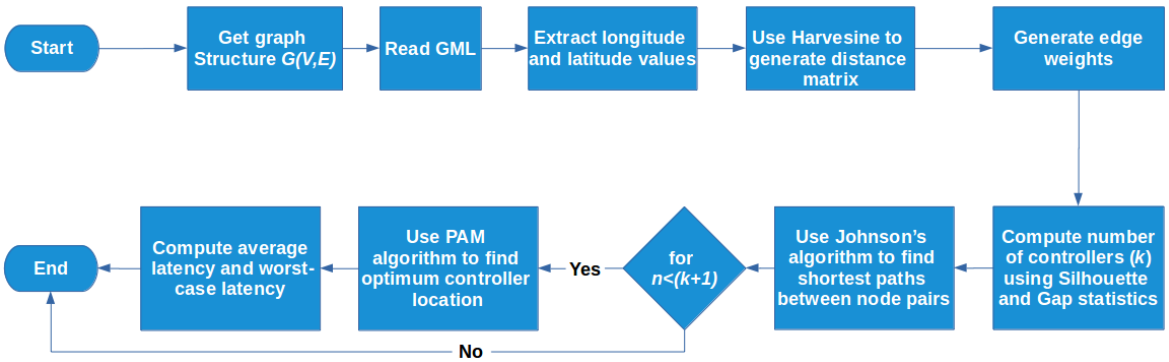


Fig. 1 – Flow chart of proposed method.

topology as an undirected graph $G(V,E)$, where V denotes network switches and E represents fibre links (edges) connecting the switches. This is followed by the extraction of the geographic location data using the input data set. Next, the Harvesine approach is applied on the location data to generate the distance matrix. To determine edge weights, an adjacency matrix is implemented between all connected switches. Then, computation of the number of controllers that minimize intra-cluster latency is carried out using the Silhouette algorithm as described in Section 4.2.1, Algorithm 1. To verify the results from Silhouette, Gap Statistics is implemented as described in Section 4.2.2, Algorithm 2. This is followed by computation of the shortest path matrix by applying Johnson's algorithm outlined in Algorithm 3. The results from Silhouette, Gap Statistics and Johnson's algorithm, are used as inputs to the PAM algorithm discussed in Section 4.3.2, Algorithm 4, which is used to find the best locations that minimize propagation latencies, namely the average latency and worst-case latency defined in Section 4.3.2 (Eq *(6)* and *(7)*). The key factor in our mathematical formulation is the distance (under the assumption of constant bandwidth across all fibre links). Therefore under constant bandwidth, propagation latency is directly proportional to distance.

6. RESULTS FOR MATHEMATICAL MODELLING

This section presents and discusses the results obtained after applying the approaches described in Section 4.

6.1 Optimal number of controllers

6.1.1 Silhouette analysis

In order to determine the optimal number of controllers to deploy on the SANReN backbone, we applied our enhanced Silhouette algorithm with propagation latency as our key performance indicator. The results from our Silhouette analysis are as depicted in Fig. 2.

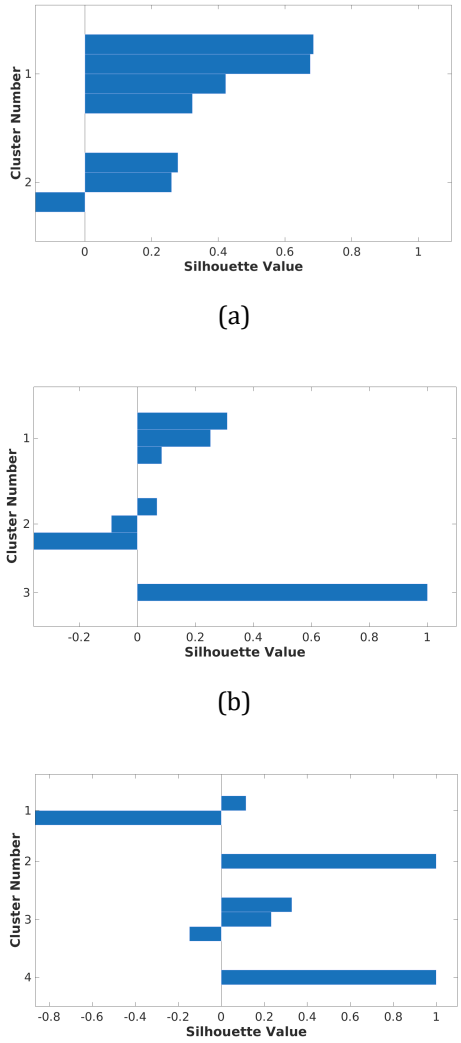


Fig. 2 – Silhouette analysis to determine optimal number of controllers for (a) $k = 2$(b) $k = 3$(c) $k = 4$.

These plots show the clustering quality when a different number of SDN controllers are deployed. For instance, Fig. 2 (a) illustrates the clustering quality when 2 controllers are deployed. The metric used to measure

clustering quality is the average intra-cluster propagation latency. Each blue horizontal bar in the plots represent a switch and its corresponding silhouette score. A silhouette score reveals the proximity of a switch to all other switches outside and within its cluster. Silhouette scores lie in the range of [-1,1]. The desired score is one that is closer to +1 as it indicates high proximity of switches within the same cluster. On the other hand, silhouette scores near -1 indicate high dissimilarity within a cluster and is a sign of poor clustering quality. A value of 0 shows that the switch is on or very close to the decision boundary between two adjacent clusters [53].

Our results indicate that deploying 3 or 4 controllers (shown in Fig. 2 (*b*) and (*c*), respectively) would result in poor clustering quality due to the presence of clusters with very low silhouette scores and the high fluctuations in the size of the silhouette plots. The increase in average silhouette score from 3 to 4 controllers (as shown in Fig. 3(a)) is caused by the high proximity of nodes in the same cluster. Given that SANReN constitutes 7 nodes, deploying 4 controllers would result in the following network partitions:

- 2 clusters with 1 node per cluster ;
- 1 cluster with 2 nodes; and
- 1 cluster with 3 nodes.

On the other hand, deploying 3 controllers would result in the following network partitions:

- 2 clusters with 3 nodes per cluster ; and
- 1 cluster with 1 node.

Given the sparse locations of the SANRen topology, it only makes sense that partitioning the network into 4 clusters would yield a higher average silhouette score than 3 clusters since there are fewer nodes per cluster and fewer outliers. Therefore, 2 controllers are the ideal number of controllers to deploy on the SANReN network as this will ensure lower propagation latency and a fair switch-to-controller distribution. This is seen from the high silhouette score obtained when the number of controllers is set to 2. Although deploying 4 controllers would yield a fairly good clustering quality and improve network reliability, it is likely to result in high inter-controller latency (due to the frequent state information exchange between controllers) and require high CapEx. However, if latency and cost are topmost priority, then 2 controllers are recommended. Moreover, 2 controllers would still suffice to meet reliability requirements unless the network has stringent requirements. However, different results are observed for different topologies.

6.1.2 Gap statistics

To verify the results from our Silhouette algorithm, we applied the Gap Statistics algorithm on the SANReN topology. With Gap Statistics the optimal number of controllers corresponds to the highest gap value with the statistical deviation, as it reflects a low intra-cluster propagation latency. Fig. 3(b) indicates that the optimal number to deploy on SANReN is 2 controllers. These results match the outcome of our Silhouette analysis.

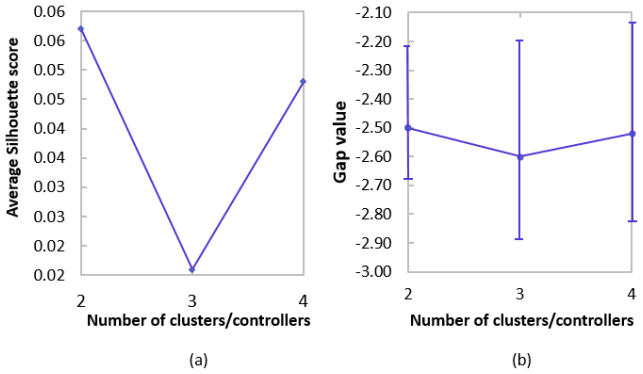


Fig. 3 – (a) Silhouette and (b) Gap Statistics evaluation summary.

6.1.3 Cost-latency trade-off analysis

Another factor that influences the decision regarding the number of controllers to deploy, is the cost associated with installing new controllers in a given network. This metric is critical as it contributes to the overall CapEx and determines how much return on investment (ROI) network operators generate. However, there exists a considerable trade-off between cost and the QoS delivered by the network. Our intention here is to quantify this trade-off so as to provide a practical guideline to network operators, regarding the ideal number of controllers to use taking into account cost and latency. This trade-off is termed "cost factor" and is defined in Eq. *(8)*, where k is the number of controllers, CPX_k is the normalised cost of deploying a single controller and L_{avg} is the average latency when k controllers are deployed. The normalised cost of deploying a single controller was kept at a constant value of 1$ per controller (assuming that a company plans to install the same model of an SDN controller).

$$cost\ factor = \frac{k * CPX_k}{L_{avg}} \quad \left[\frac{\$}{ms}\right] \qquad (8)$$

The average latency is the overall propagation latency computed using the PAM algorithm described in Section 4.3.2 for a varying number of controllers. Fig. 4 shows our results from analyzing the trade-off between cost and network performance. As expected, the results indicate that deploying 1 controller is an ideal choice to ensure minimal trade-off between cost and network performance. However, to ensure network scalability and failover, we recommend using 2 controllers. This is primarily because 2 controllers are the second best option that provides the least trade-off, and our Silhouette and Gap Statistics analysis recommend 2 controllers as the optimal number to deploy on SANReN.

It is important to note that our proposed approach does not provide a comprehensive cost analysis, but only provides a basis for one.

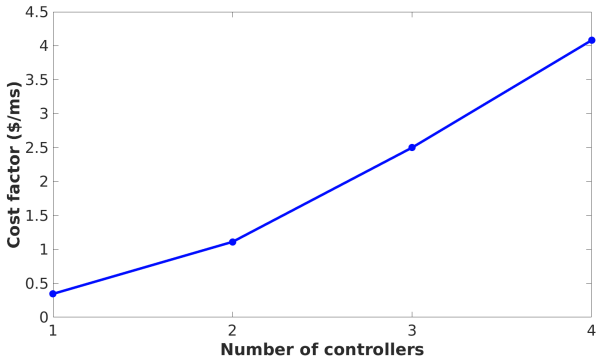


Fig. 4 – Trade-off between cost and latency for varying number of controllers.

6.2 Optimal controller locations

After determining the optimal number of controllers using the Silhouette analysis and Gap Statistics, the next step is to determine the best locations to place the recommended two SDN controllers. To find these locations, we use our proposed PAM algorithm described in Section 4.3.2. The results (depicted in Fig. 5) indicate that the optimal locations to place two controllers are Pretoria and East London with the average propagation latency of L_{avg} = 1.81. The selection of these locations guarantees the best network performance with respect to the southbound communication in the SANReN network. In contrast, deploying the controllers in Port Elizabeth and Bloemfontein would result in poor network performance, with the worst-case propagation latency being L_{wc} = 3.92.

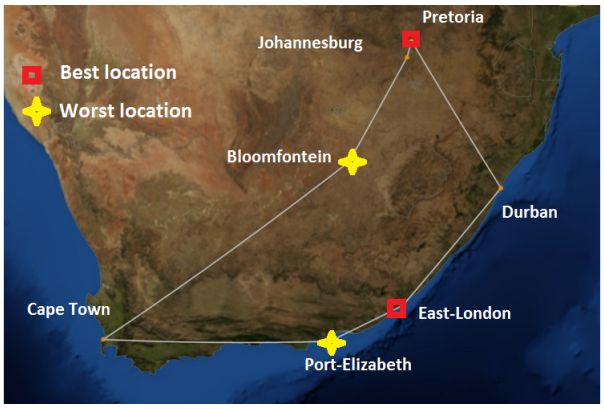


Fig. 5 – Best and worst placements of two controllers on SANReN backbone.

Table 3 presents the effect of increasing the number of controllers (k) on average and worst-case latency. These results were obtained by applying the PAM algorithm. The results indicate that, varying the number of controllers from k=1 to k=2 significantly reduces propagation latency (approximately 38% reduction of average latency and 42% reduction of worst-case latency). A further reduction is observed when the number of controllers is set to k=3. However, increasing the number of controllers beyond 3 controllers has a much less significant impact on latency (as depicted in Fig. 6).

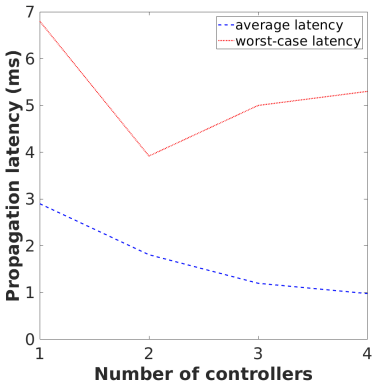


Fig. 6 – Relation between number of controllers and latency.

7. CONTROLLER PLACEMENT ON EMULATED WAN

The controller placement results presented in Section 6 relied strictly on mathematical modelling. In this section, we describe a method for finding optimal and worst locations of SDN controllers using an emulation orchestration platform called Mininet, which is able to include many of the practical implementation effects and so critical to mimic a real SDN deployment. We use controller-to-node latency (propagation + queuing +processing latency) as a key performance indicator. Our main goal is to match and verify the outcome from our mathematical formulation regarding the best locations to place the controller in a wide area network (WAN). To further optimize network performance, we also consider control-plane resiliency, as well as propose a means to alleviate signalling overhead on the control channel.

For the control-plane, we implement an ONOS controller (version 1.14) because of its distributed core which improves the robustness of the control-plane, by providing backup control in the event of network failure [54]. Moreover, ONOS' distributed core is self-coordinating and enables load sharing through fragmentation of the data-plane. This controller has an advanced east/westbound interface to ensure high inter-controller communication efficiency. Finally, employing a geographically distributed core reduces the node-to-controller latency, thus improving the controller reactivity as perceived by the network nodes. Last but not least, our decision to choose ONOS is influenced by the results from our ealier controller benchmarking experiments in [55] which confirm ONOS scalability features making it ideal for carrier grade deployments.

The evaluation of the proposed emulation approach is carried out on a model of a local national backbone called

Table 3 – Average (L_{avg}) and worst-case (L_{wc}) latency for varying number of controllers

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
L_{avg} (ms)	2.9	1.81	1.2	0.98
L_{wc} (ms)	6.8	3.92	5	5.3
Names of locations for L_{avg}	Durban	Pretoria East London	Pretoria Johannesburg Port Elizabeth	Johannesburg Durban East London Port Elizabeth
Names of locations for L_{wc}	Bloemfontein	Port Elizabeth Bloemfontein	Cape Town East London Durban	Pretoria Cape Town Bloemfontein Port Elizabeth

SANReN, the same network we used in Section 5. It may however be noted that our approach is generic and can be used to optimize any other network.

7.1 Experimental setup

The experiment setup is as illustrated by Fig. 7 and Fig. 8 (captured from Miniedit). Node c0 and c1 are ONOS SDN controller instances running on a dedicated remote machine (with 8 CPUs, 16 GB RAM and 1 TB HDD and no swap partition), and h0-h6 are hosts attached to SDN Open Virtual Switches (OVS 2.9.90) running OpenFlow version 1.3. A built-in application for reactive flow instantiation is activated to set the ONOS controller to reactive operational mode. The red dash-dotted lines show connection (over WiFi) between switches and controllers and the blue solid lines are links between the switches. The switch-to-controller communication is assumed to happen out-of-band. Since the links between the switches are known to be fibre, where speed is approximately the speed of light in fibre i.e. about $2 \times 10^8 m/s$, we use the latency formula Eq. *(9)* to configure the link properties.

$$propagation\ latency\ (sec) = \frac{distance\ (m)}{speed\ \left(\frac{m}{sec}\right)} \qquad (9)$$

The distances between nodes are calculated using the Harvesine great circle approach and the actual GPS coordinates of the nodes.

The data-plane emulated on Mininet version 2.2.2 (with default settings, for all experiments) is running on a separate machine (with 8 CPUs, 16 GB RAM and 1 TB HDD). Each switch in the data-plane has a unique data path ID (DPID). The connection between the control-plane and data-plane is via port 6633 of the controller over a slow WiFi router. The control link parameters are configured using the Linux Traffic Control (TC) utility (installed on the machine used for data-plane emulation) under the assumption that the optimal controller placement is co-located with one of the switches. The programming language used to develop the software is Python 2.7.14.

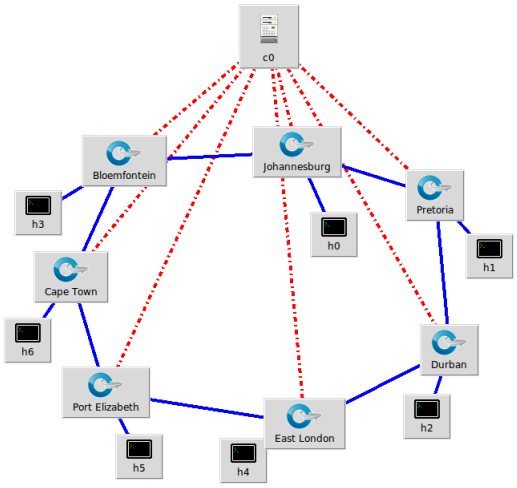


Fig. 7 – Experiment setup with one ONOS controller

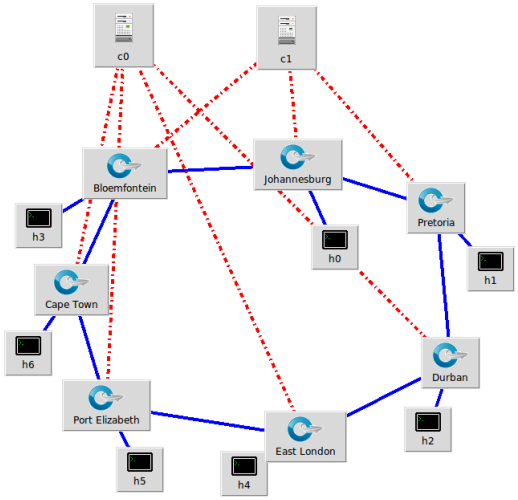


Fig. 8 – Experiment setup with two self-coordinating ONOS controllers

7.2 Methodology

This work constitutes two independent experiments. The first experiment is carried out with the intention to address the controller placement problem leveraging emulation. The first experiment analyses two scenarios: (1) when only one controller is used and (2) when two controllers are used. The second experiment presents different approaches through which signalling overhead on the control channel can be reduced, in consideration of control-plane resiliency.

7.2.1 Controller placement

On example of one controller case, Fig. 9 summarizes our approach in a flow chart (where n is the total potential controller placement locations, i.e. the total number of nodes in a given topology). For the SANReN network, n is 7, meaning there are a total of 7 potential controller placement locations in the network.

The following procedure (outlined in Fig. 9) is used for each node to determine average latency: To find optimal controller locations, first we install the ONOS controller in the same geographic location as the first OpenFlow switch node (using the Harvesine great circle approach and the Linux TC utility). The next step is to trigger a packet-In message to the controller. This is done by generating traffic flows between all pairs, i.e. between this node and all other nodes in the SANReN topology. To do this we generate an ICMP packet using the ping utility for each pair. This is followed by computation of the ICMP pinging results to obtain the total average latency (round-trip time) from the node to all other nodes in the network. This step is repeated for all nodes in the SANReN topology. To ensure valid and reliable results, we repeat the above procedure 5 times under a soft idle timeout for the controller entry of 5 seconds (the soft idle timeout defines the expiry time of a controller flow rule when there is no flow activity) and compute the average results. The soft idle timeout is set to ensure generation of control traffic upon pinging reiterations.

For the case of two controllers (see Fig. 10), the network is partitioned into two smaller administrative domains, namely cluster one and cluster two, each supervised by a dedicated ONOS instance. The parameters $n1$ and $n2$ denote the total number of switches in cluster one and two, respectively. After executing the mastership module, the partition results are as follows: The first ONOS instance ($c1$) is assigned three switch nodes in region Pretoria, Bloemfontein and Durban, while the other ONOS instance constitutes switches located in Johannesburg, Cape Town, East London and Port Elizabeth. In order to optimize the placement of these two controllers, an exhaustive search is carried out by iterating through all possible combinations (within the limits defined by each controller domain). In other words, $c1$ is placed at different regions within its domain. For each placement of $c1$, $c2$ is then placed at different regions within its

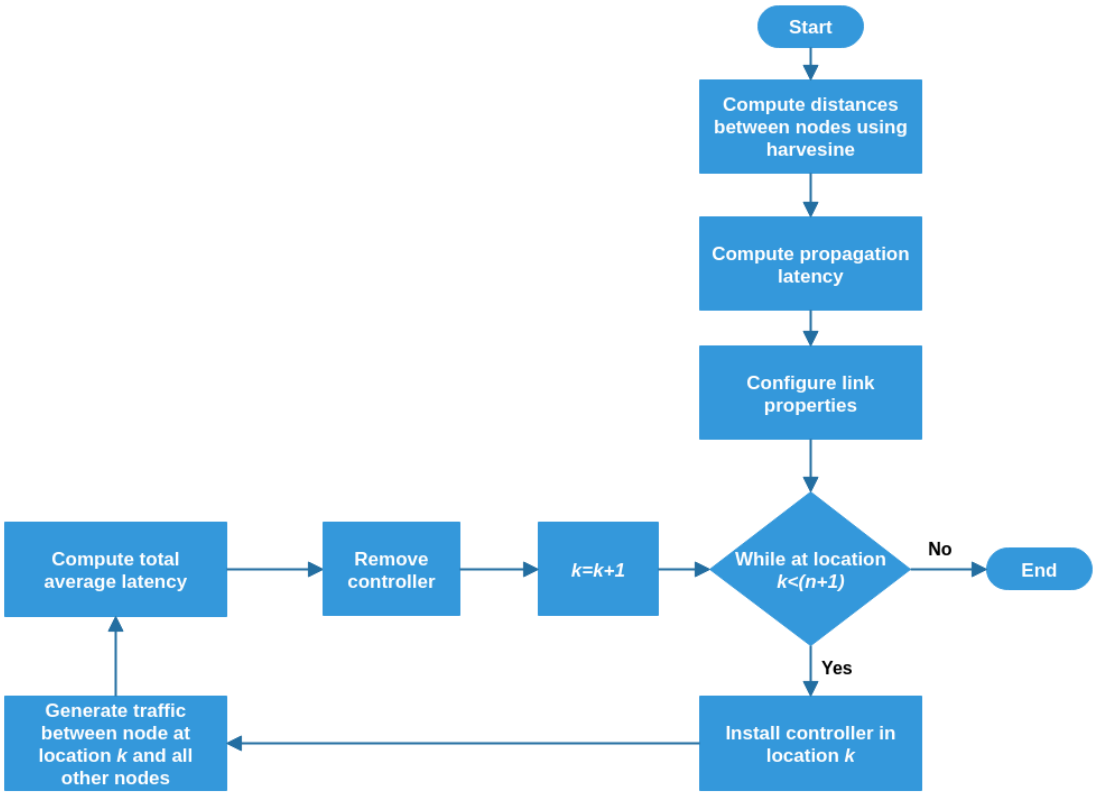


Fig. 9 – Flow chart of proposed method for one controller[56].

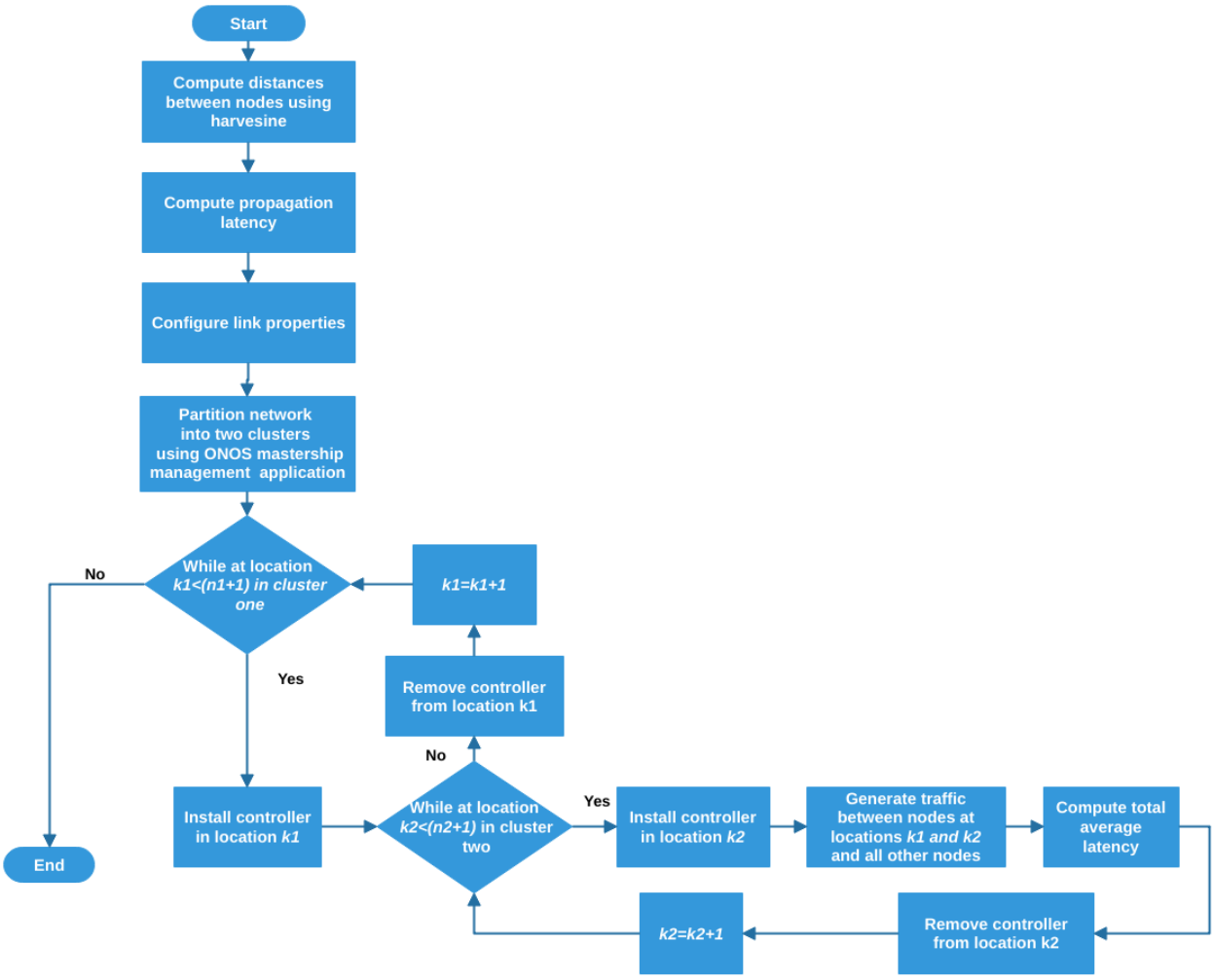


Fig. 10 – Flow chart of proposed method for two controllers.

domain. For each set of placement, the average latency is computed following the same procedure as described above.

7.2.2 Control-plane signalling overhead and failover

The centralized control scheme adopted by SDN puts the control channel at risk of incurring very high signalling overhead generated during data-plane monitoring (e.g. Stats-Request and Stats-Reply) and reactive flow instantiation (such as packet-In, packet-Out and Flow-Removed). In order to manage this rapid influx of traffic on the control channel, the following procedure is used: First we configure a cluster of two ONOS instances each managing a segment of the network. The cluster is configured using the REST API of each separate ONOS. Upon data-plane instantiation, the switch-to-controller placement (in terms of the number of switches per cluster) is imbalanced. This is because switch-to-controller placement is based solely on best effort (meaning the controller that completes the handshake with the switch first, gets mastership of the switch). By partitioning the data-plane into two clusters, the traffic induced by data-plane monitoring (code-named polling) is reduced. Specifically, after clustering, the controller sends and receives monitoring data from just a fraction of data-plane nodes. To balance the switch-to-controller placement, we activate the ONOS mastership management module. This results in a more balanced monitoring load which we expect to further decrease control-plane overhead.

To quantify the impact of switch-to-controller placement, we generate variable traffic between two virtual hosts (the client connected to Johannesburg and the server connected to Cape Town), a distance of 1399 km from each other. This is carried out using the Distributed Internet Traffic Generator (D-ITG) tool. The transport protocol is set to UDP and the number of packets per second is varied from 50 000 to 200 000 in increments of 50 000. The packet size is set to 512 bytes. The link bandwidth was kept at 10 Mb/s. The duration for the generation process is set to 5 minutes. The key performance indicators are delay, jitter and packet loss all monitored at the server end.

This procedure is carried out for two scenarios: 1) when the switch-to-controller placement is imbalanced (switch-to-controller assignment is two and five switches for controller one and two respectively) and 2) for the scenario where switch-to-controller placement is balanced (switch-to-controller assignment is three and four for controller one and two respectively).

In addition to switch-to-controller balancing, the control-plane has several tuneable parameters in the control-plane, such as polling frequency and soft idle timeout [57]. Polling frequency is a parameter that specifies how frequently statistics requests are sent to the data-plane. Soft idle timeout specifies the total time an inactive flow entry is stored in the flow tables before deletion. Tuning these parameters impacts control-plane overhead. In other words, increasing polling frequency is likely to decrease the control-plane overhead (of course at the expense of data-plane protection and restoration) while increasing the soft idle timeout results in more flow rules in the flow tables and reduces control-plane overhead (with the switch resource (e.g. memory and storage) exhaustion as a trade-off). In an operational environment, OpenFlow switches with TCAM (Ternary Content Addressable Memory) support are typically preferred for fast processing [58]. However, TCAM is very expensive with very limited memory space [59]. Therefore, the soft idle timeout can only be increased up to a certain threshold to maintain the switch memory utilization around acceptable levels.

To determine how the soft idle timeout affects control-plane overhead, we gradually increase the soft idle timeout and polling frequency (from 5 s to 40 s in increments of 5 s) and measure the number of packets (i.e. Packet-In, Packet-Out, Flow-Mod, Stats-Request and Stats-Reply). In order to evoke control traffic we generate 200 000 packets between two hosts (one connected to the node in Johannesburg and the other connected to a node in Cape Town). The duration, packet size and bandwidth are the same as for the switch-to-controller placement experiment. This experiment leveraged the results from the controller placement experiment (for the case when two control instances are deployed). In other words, two control instances were deployed at optimal locations to minimize propagation latency. Additionally, the ONOS mastership management module was activated to balance the switch-to-controller placement.

Failover is evaluated by shutting down one controller in the cluster and calling the "pingall" function. If no packet loss is observed, then it means all hosts can reach each other and switch reassignment to the active controller was successful. We also take note of the time it takes for the controller to take mastership of the "controller-less" switches.

7.3 Results and discussion

This section presents and discusses the results obtained from following the procedures described above.

7.3.1 Controller placement

Fig. 11 and Fig. 12 present the results obtained from our analysis of the SANReN network. As per Fig. 11, our results show that the optimum controller location when one controller is deployed is Cape Town since this node has the lowest average latency (L_{avg}=88.78 ms). Similarly, the worst location to place the controller when one controller is deployed is Bloemfontein since this location yields the highest average latency (L_{avg}=164.4 ms).

Fig. 12 presents the results obtained when two controllers are deployed. These results are intepreted as follows: the blue bars indicate a scenario where one controller is placed in Pretoria (a region belonging to cluster one as described in Section 7.2.1), while the other controller's location is iterated between Johannesburg, East London, Port Elizabeth and Cape Town (regions belonging to cluster two). Similarly, the red and green bars indicate controller placement in Durban and Bloemfontein (regions belonging to cluster one) while the other controller is placed in all regions within cluster two.

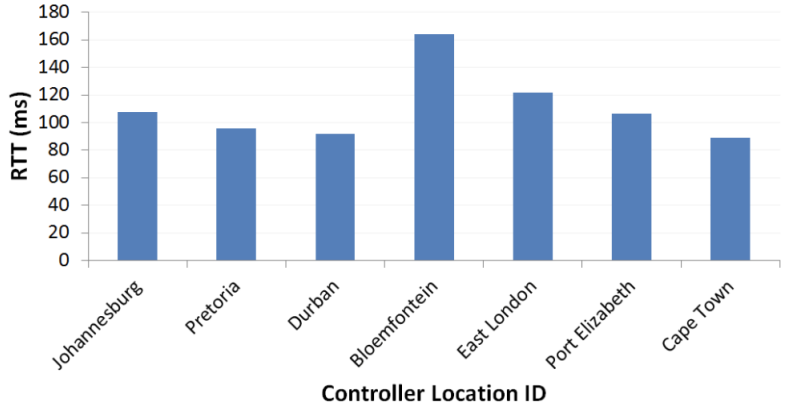


Fig. 11 – Total average latency for the ONOS controller without clustering.

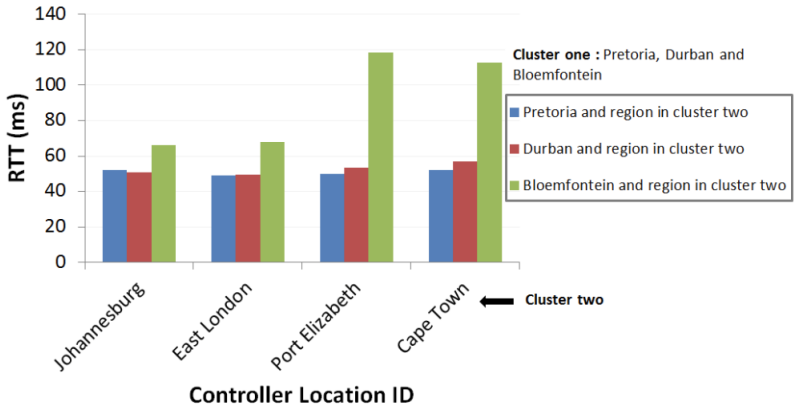


Fig. 12 – Total average latency for ONOS controller when network is partitioned into two clusters.

Our results shows that when two controllers are deployed and the mastership management module is activated, the optimum controller locations are Pretoria for cluster one and East London for cluster two, with L_{avg}=48.9 ms. The worst locations are Bloemfontein and Port Elizabeth for cluster one and cluster two respectively, with L_{avg}=118.4 ms. These results coincide with the results from our mathematical formulation in Section 6.2.

7.3.2 Control-plane failover and signalling overhead

The outcome of our failover tests was positive in that all nodes could reach each other regardless of the failed control node. This means that the switch nodes under the supervision of the failed controller were automatically reassigned to the active controller in the other cluster. The reassignment took approximately 0.5 seconds. The reassignment time was measured by carrying out the ONOS performance benchmark test case provided in [60]. The failure recovery time increases significantly with the number of disconnected switches [61]. The recovery time can potentially be improved by using more powerful servers with more RAM and CPU resources [62].

Fig. 13 and and Fig. 14 depict the results we obtained both before and after switch-to-controller placement balancing. As expected (see Fig. 13), the average delay is in overall lower after switch-to-controller placement balancing compared to the case of imbalance. We believe this is primarily because, after balancing the switch-to-controller placement, data-plane monitoring traffic is fairly divided between the controller nodes thus improving overall network performance.

The decline in average delay (both before and after switch-to-controller balancing) is a result of an increased matching probability of preserved flow rules with newly arriving packets, which reduces the number of packet-In messages to the controller, resulting in a reduction in network delay. When the number of packets is increased to 150 000 packets, an increase in average delay is

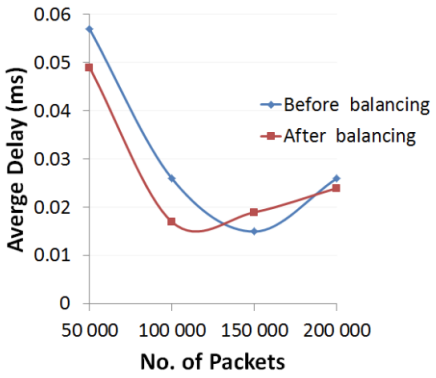


Fig. 13 – Average latency.

observed. This is likely because during the transmission of the first 150 000 packets, the switch has matching entries in its flow tables on how to route traffic, which eliminates the need to forward incoming packets to the control-plane for routing decisions. After a certain time (from 150 000 packets upwards), the switch reaches a hard timeout and clears its flow tables leading to an additional processing delay. The additional processing delay is likely the cause of the increase in average delay. Similar results are observed with regards to network jitter (as shown in Fig. 14). Last but certainly not least, when the number of packets is increased to 150 000 and 200 000, we observe a percentage

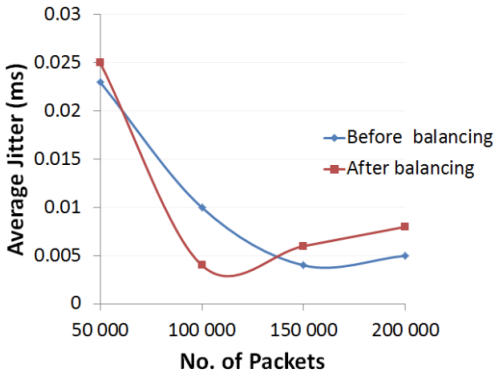


Fig. 14 – Average jitter.

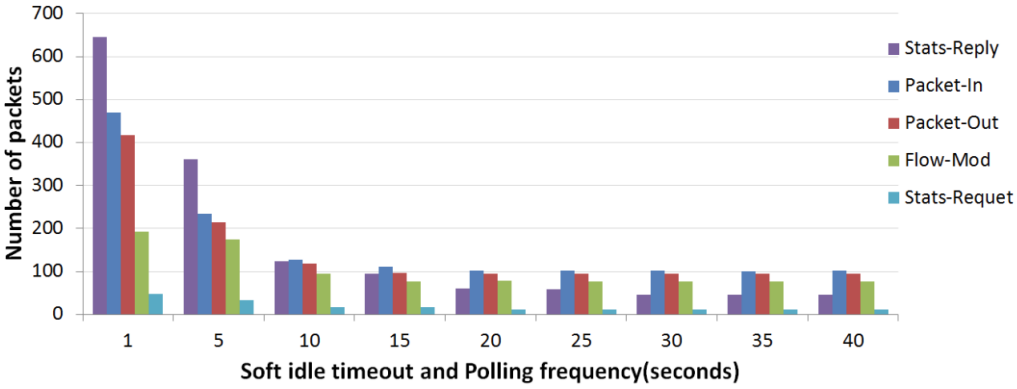


Fig. 15 – Impact of soft idle timeout on control-plane overhead.

packet drop of 0.19% and 0.53% (respectively) before switch-to-controller placement balancing, and 0.14% and 0.07% (respectively) after switch-to-controller placement balancing.

Fig. 15 depicts the impact of tuning the soft idle timeout and polling frequency on control-plane overhead. The results indicate that, increasing the polling frequency and soft idle timeout decreases the number of control packets (synonymous with control-plane overhead) generated during reactive flow instantiation. However, from 20 seconds forward, the number of control packets remains constant. Therefore, we can conclude that configuring the polling interval and idle timeout to 20 seconds would be the sufficient choice to achieve an acceptable control overhead and a potentially lower switch memory utilization. Increasing the timeout beyond 20 seconds would not change the load on the control channel but will potentially lead to a higher memory utilization.

8. SOURCE CODES

The source codes for the proposed solution have been made publicly available on Github, a world's leading code repository. The source codes can be downloaded from this link: https://github.com/Lusani/SDN-Controller-Placement

9. CONCLUSION

This study considers determining the number and location of SDN controllers in a wide area network, and associated performance and cost implications and is intended to be used to address the SDN controller placement problem. The work is applied to a national network from a developing country, SANReN. The work includes mathematical modelling and a method for obtaining the results through emulation on a popular controller suitable for real world deployments. The emulation confirmed the modelling and is also used to derive important practical limits. The modelling included Silhouette, Gap and PAM approaches. Using graph modelling, two "unsupervised" machine learning algorithms, namely Silhouette and Gap Statistics algorithms were applied to optimize the number of controllers to deploy in a given topology. Given the fact that network operators are more concerned about the cost associated with network deployment, this study also takes into consideration the trade-off between the cost of installing a new SDN controller and performance. This is necessary to facilitate decision making regarding the number of controllers to deploy, based on performance requirements and cost constraints. To determine the optimal locations to install the controllers, a classical algorithm called PAM was used. The applied algorithms are exhaustive making them ideal for static controller placement with minimal to no time constraints. In order to mimic a real SDN deployment and also to verify the outcome from our mathematical model, we use exhaustive search on an emulator to address the controller placement problem. This approach also takes into account resiliency and control-plane overhead metrics. We use the ONOS SDN controller due to its inherent self-coordinating distributed core. Our emulation results show that running a single controller yields high reaction times as some switches are located too far away from the controller. Moreover, running a single controller is not enough to meet resiliency requirements. When the number of controllers was increased to two, the reaction time was reduced considerably since the network was subdivided into two administrative domains. Moreover, the two controllers worked collaboratively to alleviate control overhead and ensure resiliency in the network. Leveraging our controller placement results as well as balancing the switch-to-controller placement, we also investigated the impact of soft idle timeout and polling frequency on control-plane overhead. Our findings suggested that a large soft idle timeout and polling frequency reduces the overall control-plane overhead. In reading the above conclusions, it should be noted that the solution to the controller placement problem is topology dependent and thus the results presented by this work only apply to the SANReN topology studied. However, the proposed approach is "protocol" agnostic and can be adopted

to solve controller placement in any SDN-enabled data-plane (such as data-planes that support protocols such as P4 Runtime, a successor to OpenFlow). We believe that our method and analysis would be beneficial for operators and service providers, not only during the initial design, but also during the incremental design of the SDN-enabled networks.

10. FUTURE WORK

In future we intend to extend our work to address dynamic controller placement which is necessary to meet 5G requirements such as ultra-reliable low latency communications achievable through dynamic placement of the mobile edge computing node. We also plan to evaluate the security aspect of SDN controllers. This is motivated by the fact that centralizing the network control intelligence presents a single point of attack/failure.

The assumption that the bandwidth for all connection links is constant (see Section 4.1) is not valid for the actual SANReN network. This assumption was made to simplify the mathematical model formulation. In future, a more complicated scenario with different link bandwidths will be considered.

In this work, traffic load was set to 512 bytes which does not reflect the actual traffic exchanges between SANReN nodes. As future work, a characterization of the actual traffic profiles combined with a rerun of the evaluation in Section 7 will be considered. We also intend to use the proposed approach to optimize data-planes running P4 Runtime protocol. Last but not least, we intend to develop a dynamic control traffic load balancing application based on current switch resource (RAM, CPU, and storage) utilization.

REFERENCES

[1] Menzie D Chinn and Robert W Fairlie. “ICT use in the developing world: an analysis of differences in computer and Internet penetration”. In: *Review of International Economics* 18.1 (2010), pp. 153–167.

[2] Cisco. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. Tech. rep. 2017.

[3] Marc Bourreau and Tommaso Valletti. “Enabling digital financial inclusion through improvements in competition and interoperability: What works and what doesn’t”. In: *CGD Policy Paper* 65 (2015), pp. 1–30.

[4] *Why SDN or NFV Now?* Last accessed June 2018. URL: https://www.sdxcentral.com/sdn/definitions/why-sdn-software-defined-networking-or-nfv-network-functions-virtualization-now/.

[5] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. “Software-defined networking: A comprehensive survey”. In: *Proceedings of the IEEE* 103.1 (2015), pp. 14–76.

[6] Slavica Tomovic, Milica Pejanovic-Djurisic, and Igor Radusinovic. “SDN based mobile networks: Concepts and benefits”. In: *Wireless Personal Communications* 78.3 (2014), pp. 1629–1644.

[7] Enrique Hernandez-Valencia, Steven Izzo, and Beth Polonsky. “How will NFV/SDN transform service provider opex?” In: *IEEE Network* 29.3 (2015), pp. 60–67.

[8] Brandon Heller, Rob Sherwood, and Nick McKeown. “The controller placement problem”. In: *ACM SIGCOMM Computer Communication Review* 42.4 (2012), pp. 473–478.

[9] Maryam Tanha, Dawood Sajjadi, and Jianping Pan. “Enduring node failures through resilient controller placement for software defined networks”. In: *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE. 2016, pp. 1–7.

[10] *Internet2 Network Infrastructure Topology*. Last accessed January 2018. URL: https://www.internet2.edu/media/medialibrary/2018/07/16/Internet2-Network-Infrastructure-Topology-All-legendtitle.pdf.

[11] Yan-Nan Hu, Wen-Dong Wang, Xiang-Yang Gong, Xi-Rong Que, and Shi-Duan Cheng. “On the placement of controllers in software-defined networks”. In: *The Journal of China Universities of Posts and Telecommunications* 19 (2012), pp. 92–171.

[12] *Rocketfuel*. accessed March 2018. URL: https://github.com/CS236340/RocketFuel.

[13] Yannan Hu, Wang Wendong, Xiangyang Gong, Xirong Que, and Cheng Shiduan. “Reliability-aware controller placement for software-defined networks”. In: *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. IEEE. 2013, pp. 672–675.

[14] Guang Yao, Jun Bi, Yuliang Li, and Luyi Guo. “On the capacitated controller placement problem in software defined networks”. In: *IEEE Communications Letters* 18.8 (2014), pp. 1339–1342.

[15] Yury Jimenez, Cristina Cervello-Pastor, and Aurelio J Garcia. “On the controller placement for designing a distributed SDN control layer”. In: *Networking Conference, 2014 IFIP*. IEEE. 2014, pp. 1–9.

[16] Md Faizul Bari, Arup Raton Roy, Shihabur Rahman Chowdhury, Qi Zhang, Mohamed Faten Zhani, Reaz Ahmed, and Raouf Boutaba. "Dynamic controller provisioning in software defined networks". In: *Network and Service Management (CNSM), 2013 9th International Conference on*. IEEE. 2013, pp. 18–25.

[17] Md Tanvir Ishtaique ul Huque, Weisheng Si, Guillaume Jourjon, and Vincent Gramoli. "Large-scale dynamic controller placement". In: *IEEE Transactions on Network and Service Management* 14.1 (2017), pp. 63–76.

[18] Jean-Michel Sanner, Yassine Hadjadj-Aoul, Meryem Ouzzif, and Gerardo Rubino. "An evolutionary controllers' placement algorithm for reliable SDN networks". In: *IFIP International Workshop on Management of SDN and NFV Systems (ManSDNNFV'2017)*. 2017.

[19] Hemant Kumar Rath, Vishvesh Revoori, Shameemraj M Nadaf, and Anantha Simha. "Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game". In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*. IEEE. 2014, pp. 1–6.

[20] Afrim Sallahi and Marc St-Hilaire. "Optimal model for the controller placement problem in software defined networks". In: *IEEE communications letters* 19.1 (2015), pp. 30–33.

[21] Yannan Hu, Wendong Wang, Xiangyang Gong, Xirong Que, and Shiduan Cheng. "On reliability-optimized controller placement for software-defined networks". In: *China Communications* 11.2 (2014), pp. 38–54.

[22] David Hock, Matthias Hartmann, Steffen Gebert, Thomas Zinner, and Phuoc Tran-Gia. "POCO-PLC: Enabling dynamic pareto-optimal resilient controller placement in SDN networks". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE. 2014, pp. 115–116.

[23] Stanislav Lange, Steffen Gebert, Thomas Zinner, Phuoc Tran-Gia, David Hock, Michael Jarschel, and Marco Hoffmann. "Heuristic approaches to the controller placement problem in large scale SDN networks". In: *IEEE Transactions on Network and Service Management* 12.1 (2015), pp. 4–17.

[24] Adlen Ksentini, Miloud Bagaa, and Tarik Taleb. "On using SDN in 5G: the controller placement problem". In: *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE. 2016, pp. 1–6.

[25] Lusani Mamushiane, Albert A Lysko, and Joyce Mwangama. "Controller placement optimization for Software Defined Networks". In: ITU Journal. 2021.

[26] Mu He, Arsany Basta, Andreas Blenk, and Wolfgang Kellerer. "Modeling flow setup time for controller placement in sdn: Evaluation for dynamic flows". In: *IEEE International Conference on Communications (ICC)*. 2017.

[27] Peter O Olukanmi and Bhekisipho Twala. "Sensitivity analysis of an outlier-aware k-means clustering algorithm". In: *Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), 2017*. IEEE. 2017, pp. 68–73.

[28] Aruna Bhat. "K-medoids clustering using partitioning around medoids for performing face recognition". In: *International Journal of Soft Computing, Mathematics and Control* 3.3 (2014), pp. 1–12.

[29] Trupti M Kodinariya and Prashant R Makwana. "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6 (2013), pp. 90–95.

[30] Robert Tibshirani, Guenther Walther, and Trevor Hastie. "Estimating the number of clusters in a data set via the gap statistic". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2 (2001), pp. 411–423.

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. "Unsupervised learning". In: *The elements of statistical learning*. Springer, 2009, pp. 485–585.

[32] Gregory Saint Vincent. *The Haversine Formula*. Last accessed September 2018). URL: `http://www.longitudestore.com/haversine-formula.html`.

[33] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

[34] Jovin J Mwemezi and Youfang Huang. "Optimal facility location on spherical surfaces: algorithm and application". In: *New York Science Journal* 4.7 (2011), pp. 21–28.

[35] Agarwal Prakhar. *Haversine formula to find distance between two points on a sphere*. Last accessed February 2019. URL: `https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/`.

[36] Lusani Mamushiane, Albert Lysko, and Joyce Mwangama. "Optimum Placement of SDN Controllers in African Backbones: SANREN and ZAMREN as a Case Study". In: *Southern Africa Telecommunication Networks and Applications Conference (SATNAC), 2018*. 2-5 September 2018.

[37] Joseph D Donnay. *Spherical trigonometry*. Read Books Ltd, 2013.

[38] Mojgan Mohajer, Karl-Hans Englmeier, and Volker J Schmid. "A comparison of Gap statistic definitions with and without logarithm function". In: *arXiv preprint arXiv:1103.4767* (2011).

[39] N Damak. "Distance problems in networks—theory and practice". In: *Journal of Classical shortest-path algorithms* (2010), pp. 1–9.

[40] M Bindhu and GP Ramesh. "Load balancing and congestion control in software defined networking using the extended Johnson algorithm for data centre". In: *International Journal of Applied Engineering Research* 10.17 (2015), p. 2015.

[41] KLEIN Bouleimen and HOUSNI Lecocq. "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version". In: *European Journal of Operational Research* 149.2 (2003), pp. 268–281.

[42] AM Fahim, AM Salem, F Af Torkey, and MA Ramadan. "An efficient enhanced k-means clustering algorithm". In: *Journal of Zhejiang University-Science A* 7.10 (2006), pp. 1626–1633.

[43] K Krishna and M Narasimha Murty. "Genetic K-means algorithm". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29.3 (1999), pp. 433–439.

[44] Michael Steinbach, George Karypis, Vipin Kumar, et al. "A comparison of document clustering techniques". In: *KDD workshop on text mining*. Vol. 400. 1. Boston. 2000, pp. 525–526.

[45] Qiaoping Zhang and Isabelle Couloigner. "A new and efficient k-medoid algorithm for spatial clustering". In: *International conference on computational science and its applications*. Springer. 2005, pp. 181–189.

[46] Li Sheng Zhang, Mei Jie Yang, and Da Jiang Lei. "An improved PAM clustering algorithm based on initial clustering centers". In: *Applied Mechanics and Materials*. Vol. 135. Trans Tech Publ. 2012, pp. 244–249.

[47] Peter O Olukanmi, Fulufhelo Nelwamondo, and Tshilidzi Marwala. "k-Means-Lite: Real Time Clustering for Large Datasets". In: *2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE. 2018, pp. 54–59.

[48] Shalini S Singh and NC Chauhan. "K-means v/s K-medoids: A Comparative Study". In: *National Conference on Recent Trends in Engineering & Technology*. Vol. 13. 2011.

[49] Erich Schubert and Peter J Rousseeuw. "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms". In: *arXiv preprint arXiv:1810.05691* (2018).

[50] *South African National Research Network*. Last accessed October 2018. URL: https://www.sanren.ac.za/backbone/.

[51] Simon Knight, Hung X Nguyen, Nick Falkner, Rhys Bowden, and Matthew Roughan. "The Internet topology zoo". In: *IEEE Journal on Selected Areas in Communications* 29.9 (2011), pp. 1765–1775.

[52] The University of Adelaide. *The Internet Topology Zoo*. accessed October 2018. URL: http://www.topology-zoo.org/dataset.html.

[53] Lovisa Lovmar, Annika Ahlford, Mats Jonsson, and Ann-Christine Syvänen. "Silhouette scores for assessment of SNP genotype clusters". In: *BMC genomics* 6.1 (2005), p. 35.

[54] Hlabishi I Kobo, Adnan M Abu-Mahfouz, and Gerhard P Hancke. "A survey on software-defined wireless sensor networks: Challenges and design requirements". In: *IEEE access* 5 (2017), pp. 1872–1899.

[55] Lusani Mamushiane, Albert Lysko, and Sabelo Dlamini. "A comparative evaluation of the performance of popular SDN controllers". In: *Wireless Days (WD), 2018*. IEEE. 2018, pp. 54–59.

[56] Lusani Mamushiane, Albert A Lysko, and Joyce Mwangama. "Resilient SDN Controller Placement Optimization applied to and emulated on South African National Research Network (SANReN)". In: *submitted for Wireless Communications and Networking Conference*. IEEE, 2019.

[57] Jian Li, Jae-Hyoung Yoo, and James Won-Ki Hong. "Dynamic control plane management for software-defined networks". In: *International Journal of Network Management* 26.2 (2016), pp. 111–130.

[58] Shizhong Xu, Xiong Wang, Guangxu Yang, Jing Ren, and Sheng Wang. "Routing optimization for cloud services in SDN-based Internet of Things with TCAM capacity constraint". In: *Journal of Communications and Networks* 22.2 (2020), pp. 145–158.

[59] Jhih-Yu Huang and Pi-Chung Wang. "TCAM-based IP address lookup using longest suffix split". In: *IEEE/ACM Transactions on Networking* 26.2 (2018), pp. 976–989.

[60] *Test Case on The Role Intimation Time for Controller Cluster At Setup*. Last accessed January 2019. URL: https://wiki.onosproject.org/pages/viewpage.action?pageId=11864465.

[61] *The Role Intimation Time for Controller Cluster After Master Failover*. Last accessed September 2018). URL: https://wiki.onosproject.org/pages/viewpage.action?pageId=11864470.

[62] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, et al. "ONOS: towards an open, distributed SDN OS". In: *Proceedings of the third workshop on Hot topics in software defined networking*. ACM. 2014, pp. 1–6.

AUTHORS

Lusani Mamushiane received BEng degree in Electrical and Electronic Engineering from the University of Johannesburg and MSc in Electrical Engineering from the University of Cape Town (UCT), South Africa, in 2014 and 2019 respectively. In 2015, she joined Microsoft where she worked as an Electronic and Telecoms Engineer for Intelligent Traffic Systems. In 2016 she then joined Advanced and Network Architecture Systems research group in Next Generation Enterprises and Institutions (NGEI) cluster, CSIR as a Researcher. Ms Mamushiane's research interests include, but not limited to, Optical Networks, Software Defined Networks (SDN), Network Function Virtualization (NFV), Intelligent Traffic Systems and 5G Mobile Architectures and Services.

Joyce B. Mwangama received BSc degree in Electrical and Computer Engineering and MSc in Electrical Engineering from the University of Cape Town (UCT), South Africa, in 2008 and 2011 respectively. In 2015, she joined the Department of Electrical Engineering as a Lecturer and received a PhD degree in Electrical Engineering from the University of Cape in 2017. Her current research interests include Future Internet Technologies, 5G Mobile Network Architectures and Services, Software Defined Networks, Network Function Virtualization, Machine-to-Machine Communications and the Internet of Things. Dr Mwangama is a member of the IEEE and the South African Institute of Electrical Engineers (SAIEE).

Albert A. Lysko received an MSc in Radiophysics from St-Petersburg State Technical University, Russia in 1998 and a PhD in ICT from Norwegian University of Science and Technology (NTNU), Norway in 2010. Dr Lysko started working before his first degree and has over 25 years of combined industrial and academic experience gained in Russia, Norway and South Africa. From 2007, he is with the Council for Scientific and Industrial Research (CSIR) in South Africa, where he is a Principal Researcher. He is also an Adjunct Professor with the University of Cape Town and Extraordinary Researcher with the North-West University, South Africa. His research interests include networking, especially software-defined networking; numerical modelling; wireless communications and especially television white spaces (TVWS). He authored and co-authored a book, two book chapters, three patents, several technology demonstrators, numerous technical reports, and over 100 research, popular science and news publications. He has made contributions to national regulations in South Africa. His work affected TVWS regulations in the USA. The TVWS work was recognised with a national-level award. Dr Lysko has organised over 100 events and 3 international conferences. He has shared three best paper awards, given numerous invited talks including a plenary and keynote talks, and served on multiple national and international funding and research/standardization panels/committees. He is an award-winning volunteer and a Fellow of South African Institute of Electrical Engineers (SAIEE) and a Senior member of IEEE. He also represented South Africa in two EU Cooperation in Science and Technology (COST) Actions IRACON and VISTA.

ECONOMIC EFFICIENCY OF SPECTRUM ALLOCATION

Vadim Nozdrin, PhD.,
International Telecommunication Union*, Place des Nations, Geneva, Switzerland, vadim.nozdrin@itu.int

Abstract – *Spectrum demand is rapidly expanding, driven by the developments of incumbent radio systems, as well as with the requirements of new technologies and market players to get spectrum access. In this regard, one of the fundamental functions of the state spectrum regulator is to create an enabling administrative and legal environment to ensure efficient spectrum use and to mitigate scarcity of this valuable natural resource. This paper analyses the economic value of spectrum use and suggests an extension of the spectrum's traditional technical boundaries, in order to take into consideration energy, environmental, sanitary and bio-geotechnical limits. The study reviews the phenomena of increasing transaction costs and negative externalities of spectrum access requiring economic criteria and assessment to achieve optimum resource allocation with a spectrum's demand growth. Results develop an input-output matrix as an allocation instrument to achieve spectrum efficiency and optimum equilibrium based on analysis of the economic effects from different band utilization scenarios, supplemented by currently used technical criteria. Practical application of proposed economic methods should improve the existing spectrum management system.*

Keywords – Input-output matrix, spectrum efficiency, spectrum externalities, spectrum management

1. INTRODUCTION

Satisfying the ever-growing material, physical and social needs of humanity under conditions of limited resources, places the implementation of an overall strategy for sustainable development at the heart of contemporary economic, political, environmental, social, and technical research. Invention of data transmission by means of electromagnetic waves gave humanity a new invisible resource: radiofrequency spectrum. Every single radio system uses artificially-produced or naturally-emitted electromagnetic waves as a carrier of data. Radiofrequency spectrum is defined as the electromagnetic frequency continuum ranging from 3 kHz to 3 000 GHz. Originating from maritime applications in the past, nowadays spectrum plays an essential role in all areas of human activity. For the time being, spectrum is primarily utilized based on an exhaustive approach, relying predominantly on technical limits defined by Shannon theorem and absence of mutual harmful interference between the growing number of space and terrestrial radio systems.

The growing problem of spectrum scarcity requires spectrum efficiency evaluation and introduction of modern methods for its improvement. Nobel laureate Prof. R. Coase emphasized that spectrum allocation should be determined by market forces rather than by administrative decisions, leading to zero transaction costs and coping with any externalities [1]. Later on his idea was developed in more detail, advocating for the implementation of alternative spectrum management instruments including spectrum privatization, secondary trading and auctioning [2]. Several fundamental studies designated the introduction of a spectrum fee, deregulation of management systems, and delegation of some administrative functions to the private sector as the basic economic methods to improve spectrum management systems [3]. In 1996, International Telecommunication Union (ITU) initiated studies aimed to develop a guidance for Member States in applying economic methods to improve their spectrum management systems [4]. In 2001, 37 world famous American economists, including four Nobel laureates, addressed the US Federal Communications Commission with the need to implement radical changes in the principles governing spectrum usage that would be required to ensure the harmonious development of telecommunications [5]. Their proposals included modifying users' rights to allow spectrum trading and, accordingly, implementing auctions and market mechanisms for the resource allocation.

However, the problem of spectrum efficiency is still vital. Conflicts of interests between different spectrum users, newcomers and incumbents, public

* The ideas expressed in this paper are those of its author and they do not necessarily represent those of the International Telecommunication Union, the Radiocommunication Bureau or any known institution.

and private operators, national and regional telecommunication groups provoke a lot of controversial technical discussions leading to non-optimal allocation decisions. The latter are producing negative externalities, innovation delays, late comer price, increases in transaction costs and non-harmonized spectrum exploitation. The key reason for the inefficient decision-making mentioned above is the choice of solely depending on technical criteria in spectrum management such as electromagnetic compatibility with incumbent applications. In some cases, the principle of "protection of existing radio services" could be seen as somewhat obsolete and the introduction of new technologies would benefit from the reallocation of spectrum from incumbent operators to newcomers. The traditional approach requires further improvement in international and national spectrum management systems by applying economic criteria and valuations to accommodate booming spectrum demand.

This paper defines the place and role of the spectrum in the digital economy, specifies major spectrum users and considers the economic value of spectrum use. The author embraces multivariable direct and indirect boundaries, related to energy, environmental, sanitary and biogeotechnical restrictions. The study covers the phenomena of increasing transaction costs and negative externalities of spectrum access. This work proposes input-output Leontief matrix as an allocation instrument to achieve spectrum efficiency and optimum output based on an analysis of the economic effect from different band utilization scenarios. The arguments of the present article deal with the problem of spectrum efficiency, but comments received seem equally applicable for post-pandemic economic development in order to choose an optimal scenario of natural resources use on a global and national level.

2. ECONOMIC VALUE OF SPECTRUM USE

Table 1 shows that in recent decades spectrum is directly or indirectly used practically in all industry sectors, standing as one of the key factors promoting economic, social and cultural development, increasing consumer and producer surplus, enhancing public welfare, protecting safety-of-life and improving living standards.

Some facts illustrating the economic value of spectrum use for the modern economy and society are as follows:

- the contribution of the mobile industry to global GDP in 2018 achieved nearly USD 4 trillion [6];
- the economic benefits of accurate weather forecasts obtained from satellite Earth monitoring systems that can mitigate the economic consequences of natural disasters in EU countries is estimated at 61 billion Euros annually [7];
- reallocation of 500 MHz spectrum band from TV companies and federal agencies to mobile operators created 250 000 new jobs in the United States of America [8];
- as a part of the US federal plan to combat the coronavirus, 600 MHz of the spectrum has been reallocated from satellite to mobile operators for 60 days, "supporting social life, to keep economy going and to connect remotely to health care professionals during this crisis" [9].

Earth observation and meteorological systems contribute towards achieving all the United Nations Strategic Development Goals (SDGs), without a single exception. In addition, data obtained by using relevant radio systems are essential for monitoring the results of SDG initiatives. Around 30 of the 232 indicators developed for monitoring progress towards the SDGs can be verified only based on data obtained from Earth observation satellites.

Additional important areas where spectrum use has a significant impact are radio astronomy and space research, which are crucial to studies of classical physics and Universe evolution. Research in these areas, based on data obtained with the aid of the relevant radio systems, have an enormous impact on the development of other sectors of the economy which, at first glance, would appear to have no relation to those spectrum-based systems. For example, the success of algorithms for processing large data volumes in radio astronomy and cosmic research have been a foundation of the digital economy.

Spectrum use has considerable social importance from the point of view of a State's obligations to protect the basic rights of its citizens in democratic society. It is not coincidental that basic provisions concerning spectrum management in France are set by legislation on freedom of speech and information. Radiocommunication is critical in all phases of disaster management. Aspects of radiocommunication associated with disasters

Table 1 – Main spectrum users

Users	Radio systems used	Applications	End use output	Intersectoral use output
Defence	Mobile and satellite communication systems, microwave links, radiolocation, radars, navigation	Operational control, management	Safeguarding national security, border control, antiterrorist operation	Development of radio-electronic industries and science
Security and law-enforcement agencies	Mobile and satellite communication systems, microwave links, radars	Operational control of subordinate units	Maintenance of internal security and law enforcement, government communications	Radio-electronic industries and science
Communications and informatization	Mobile and satellite communication systems, microwave links, sensor systems	Provision of access to data transmission systems and public communication networks	Public communication and informatization services, maintenance of economic security	Radio-electronic industries and trade, satellite industry, social development, disaster relief
Broadcasting	Satellite and radio-relay communication systems, broadcasting networks	Broadcasting of TV and radio programmes	Freedom of thought, expressions and public information	Commerce, advertising, security, disaster relief
Earth exploration and monitoring	Remote Earth sensing satellites	Collection of data on the state of the Earth's natural characteristics	Cartography, geoinformation, data on the state of the climate and natural resources	Construction, natural resources management, disaster relief
Radionavigation	Satellite and land-based navigation systems	Transmission of time signals and frequency standard emissions	Precise determination of the location, shift and speed of objects	All branches of industry
Meteorology	Remote earth sensing and data collection satellites, radars, meteorology probes and sensors	Collection, transmission and distribution of meteorological data	Weather forecasting, prediction of natural disasters, monitoring of climate change and SDGs indicators	Defence, emergency agencies, transport, agriculture and forestry, green energy
Transport	Mobile and satellite communication and navigation systems, sensors, radars	Remote monitoring and control of transport, broadband access	Improving safety of passenger and cargo transport, optimization of traffic, unmanned transport	World trade, hotel and tourism sector, postal services
Radio astronomy	Ground and space radio astronomy stations	Collection of data on the Universe	Development of fundamental science	All branches of industry
Space research	International Space Station, piloted and robotic space missions	Studies on the near-Earth space environment and space objects	Development of fundamental science, search for natural resources, protection from potential hazards from space	All branches of industry
Production industry	Office and corporate networks, control, automation system	Improving production efficiency and safety	Optimizing use of resources	All branches of industry

include disaster prediction, detection, alerting, and relief. In certain cases, when the wired infrastructure is significantly or completely destroyed by a disaster, only radio systems can be employed for disaster relief operation in a timely manner.

With technology evolution, spectrum appears to play an even more critical role and is becoming an essential element of the numerous governmental *e*-programs, such as *e*-medicine, *e*-government, *e*-education etc. A future digital economy implies a transition from current intuitive manual forms of management to an algorithm-based one that relies on the development and use of coherent models for political, economic, technological and behavioural analysis, as well as decision-making and artificial intelligence. It will depend critically on a powerful complex of different types of radio systems to gather, deliver and disseminate a variety of data, boosting spectrum demand. Information collected by different sensors and receivers on-board satellites, aircraft, and ships, fitted to buoys,

livestock, and so on, once disseminated, will help humanity to fight against bureaucracy and mismanagement by means of better understanding of the relevant technocratic, natural and socio-economic processes and their interactions and impact.

In terms of importance, the spectrum ranks alongside national energy resources or reserves of raw minerals. The efficient use of the spectrum in the modern world has a major impact on economic development and is a critical factor of sustainable development.

3. LIMITATION OF SPECTRUM USE

The spectrum as a natural resource can be classified as follows [10]:

- a category of energy resources, subcategory artificially activated energy sources, along with atomic and thermonuclear energy;
- a renewable resource;
- spectrum sharing by multiple radio systems is possible using a combination of factors relating to time/phase, space, frequency dimensions and performance characteristics.

Complex technological processes of spectrum exploitation define interdependent vectors of interfaced limitation:

- Physical limits. The choice of spectrum for most radio technologies is complicated by the varying quality of the resource in terms of radiowave attenuation and propagation characteristic. As an example, only a few specific bands are suitable for monitoring particular natural features, depending on the unique resonance frequencies, and they must be protected from all man-made radiation in order to ensure measurement quality. Physical limits have a major strong impact on economic indicators of spectrum use projects. Sometimes a radio application can utilize a less occupied, higher frequency range alternative, but it will require more investment/operational costs to provide the same quality of service, challenging project efficiency;
- Technological limits. With industry evolution, the upper boundary of the usable spectrum is constantly raised. Currently the spectrum between 3 kHz and 100 GHz is actively used, while the range from 100 to 275 GHz has a certain potential for use, although at present it is not heavily occupied. Existing wireless technology cannot meet the demands of future ultra-wideband networks, since data rates beyond 10 Gbps are not achievable by current millimeter wave systems, which are typically operating below 60 GHz. The frequency band of 275~3 000 GHz is especially of interest for future wireless systems with 100 Gbps data rates. Although radio equipment for this range is still in a very early stage of development, some studies show its potential applications [11]. Any future use of the range will be extremely constrained by the propagation characteristics, and in particular by high levels of rain attenuation and transmission path loss. Such quality of spectrum will make ultra-high frequencies unsuitable for the majority of conventional applications;
- Technical limits. Spectrum use is limited by the extent to which it can be shared by different radio systems, given the potential for harmful interference which can have a major impact on the quality of service. In other words, technical conditions for a given system to use spectrum is determined by electromagnetic compatibility conditions imposed by other radio systems, operating in the same band and in the same location. From the economic theory, unintentional interference could be regarded as a negative externality. Waves emitted by a radio system constitute a wanted signal for the receiver within that system and enable it to operate, but may cause harmful interference to other radio systems, degrading their operating quality or preventing operation altogether. Mutual interference is a main reason for international and national spectrum management systems, to ensure the possibility of a globally coordinated use of the resource. The potential of a radio system in terms of the creation of and sensitivity to interference is determined by the performance of the radio equipment used, which is dependent on the operational quality and ultimately the cost of the radio systems;
- Administrative limits. International and national use of the spectrum is based on block allocation of the resource among different users/radio services in technical terms. Any radio system may use a frequency band that is administratively defined in an international

and/or national allocation table to the radio service in which the radio system in question refers. At the international level, questions pertaining to the international frequency allocation table are decided at the ITU World Radiocommunication Conference (WRC);

- Health-related limits. The harmful effects of artificially generated electromagnetic fields on human health is a matter of growing concern and is being actively studied by a number of medical institutions under the umbrella of the World Health Organization. Further uncontrolled increases in the intensity of use of the spectrum may prove harmful to human health [12]. At present, two types of limits have been established, namely, basic restrictions and reference levels. Basic restrictions are standards relating to the effect on human beings of time-varying electrical, magnetic and electromagnetic fields. These are based directly on their proven exposures on human health, although measuring them is difficult (for example, induced current density in the human body). Reference levels are derived from the basic restrictions and are expressed in easily measurable units. They set maximum permissible emission power for specific types of radio system. It is considered that, when applying reference levels, the overall effect on the body should not exceed the established basic restrictions. Increasing the density of radio systems especially in urban areas could call into question the viability of the current approach, as it would be necessary to improve current WHO and ICNIRP regulation taking into account the aggregated field strength in order to protect public health;
- Environmental limits. The large-scale use of different radio systems creates the problem of electronic waste (e-waste), in other words, the necessity to recycle large quantities of obsolescent or defective radiocommunication hardware. ITU reported that the total global volume of e-waste had reached around 53.6 million tonnes in 2019 with a forecast that this figure may be tripled by 2021. Of this total amount, 44.5 million tonnes of e-waste were discarded in landfills, burned, or illegally traded, causing so called "export of the environmental crisis" and leading to air and water pollution, soil contamination, and biodiversity loss. Environmental limits for spectrum use should consider space debris also. Under current practice and regulations, a geostationary satellite that has come to the end of its operational life is to be transferred to a higher altitude orbit, but it stays in space. There are plans to launch huge constellations into non-geostationary orbits in order to provide direct broadband access with global coverage requiring thousands of satellites. Near-Earth space is already polluted by artificial junk as a result of satellite collisions and explosions, including around 20 000 fragments larger than 10 cm in size and around 500 000 between 1 and 10 cm in size [13]. The problem of space debris is expected to be left for future generations.

  Delivery of all planned satellites and space missions into space will dramatically increase demand for rocket launches. Up to now rocket launches occur irregularly and annually do not exceed 100 globally so that concerns about the damage done to the ozone layer by rocket emissions have not elicited regulation. ITU study predicts, more than 1 000 returning launches only for suborbital vehicles in 2025, raising a question on appropriate mitigation measures at the international level against potential negative impacts on Earth's atmosphere, carbon emissions increase, ozone depletion etc [14].

  Booming of radio equipment production causes the growth in CO_2 emissions. By 2023, total carbon emissions resulting from the growth in production of mobile equipment alone could reach 235 megatons CO_2 per year [15].
- Power limits. The use of higher bands and increasing demands for transmission speed, as well as a growth in the number of radio systems of different types result in a considerable increase in power consumption, which exacerbates the global energy shortage. ITU estimates that the information and communication technology (ICT) sector's share of global electricity consumption in 2015 was 3.9 per cent. A typical 5G base station consumes up to twice or more the power of a 4G base station. To address this challenge, some countries plan further development of coal power plants to support an intensive introduction of new generation radio networks in spite of international carbon emission limitations;

– Bio-geotechnical limits. According to the "1 per cent" law, an artificial change of 1 per cent in the energy of a natural system as a rule causes it to depart from a Chatelier's equilibrium [16]. In other words, our biota cannot compensate the volume of artificially generated energy that exceed 1 per cent of its natural level. Experience of using other natural resources suggests that violating that rule may lead to catastrophic change in the natural environment, such as global warming. A part of solar energy transforms into a natural electromagnetic field of the Earth forming a 1 per cent limit. An aggregate assessment of man-made emissions is required when the strategy of spectrum use is defined to avoid transgressing this threshold.

4. INPUT-OUTPUT ANALYSIS FOR SPECTRUM MANAGEMENT

A definition of limits is just the first step of any economic analysis of efficient resource management. Regulators should establish an accurate economic model, enabling them to find an optimum scenario of development under conditions of resource scarcity and negative externalities. Nobel laureate W. Leontief pointed out that "the essence of input-output analysis in study of the resources problem consists in constructing several alternative scenarios with different combinations of input-output vectors describing the technological structure of the different methods of production and usage". The input-output matrix enables a resolution to the problem of a shortage of resources, taking into account externalities, through planning and redistribution. From the analysis of the technological process, of interdependencies between industry sectors and of the economic impacts of spectrum usage, the inter-sector input-output matrix for spectrum can be used as the economic model for assessing spectrum efficiency. The proposed matrix for spectrum use is presented in Table 2.

Explanation of the values used in Table 2:

n — number of users (i.e. industry sectors) or radio services using the spectrum (see Table 1 for example);

k — spectrum allocation scenario;

s_{nk} — aggregate spectrum allocation for the n-th user (radio service) in the k-th scenario. Here it must be understood that account needs to be taken not only of the bandwidth directly accessible for the user, but also of other technical conditions and restrictions on its utilization stemming from shared use. In some cases, analysis of this component is simplified thanks to the decisions on global or regional spectrum harmonization for a specific standard;

S_k — aggregate spectrum allocations in the k-th scenario. Analysis of this value is complicated by the possibility of shared use of the spectrum, i.e. $S_k \neq \Sigma s_{nk}$;

Table 2 – Input-output matrix for spectrum use

		Main spectrum users				**Aggregate bandwidth for *k*-th scenario**	**Economic impact**		**Overall impact for *k*-th scenario**	**Overall cost for *k*-th scenario**
		1	**2**	**…**	***n***		**Output for end user for *k*-th scenario**	**Inter-industry output for *k*-th scenario**		
Spectrum allocation scenarios	1	s_{11}	s_{12}	…	s_{n1}	S_1	y_{int1}	y_{ext1}	Y_1	W_1
	2	s_{21}	s_{22}	…	s_{n2}	S_2	y_{int2}	y_{ext2}	Y_2	W_2
	…	…	…	…	…	…	…	…	…	…
	k	s_{n1}	s_2	…	s_{nk}	S_k	y_{intk}	y_{extk}	Y_k	W_k
Revenue		p_1	p_2	…	p_n					
Capital costs		cap_1	cap_2	…	cap_n					
Operating costs		op_1	op_2	…	op_n					
Transaction costs		tc_1	tc_2		tc_n					
Externalities costs		ec_1	ec_2		ec_n					
Spectrum user costs		w_1	w_2	…	w_n					

y_{intk} — economic impact of spectrum use in the k-th scenario, as the sum of the impacts of the production of the n-th spectrum user for the end user;

y_{extk} — economic impact of spectrum use in the k-th scenario, as the sum of the impacts of the production of the n-th spectrum user for inter-sector use of output $Y_k = y_{intk} + y_{extk}$ — Overall contribution of use of the resource to the country's GDP for the k-th allocation scenario. Given the complexity of the inter-sector relationships and technological processes involved in the use of a resource, of direct and indirect assessments of a resource's economic impact and of the different manifestations thereof (e.g. the creation of new jobs), this is a multifaceted task that calls for the accumulating and processing of Big Data. The simplest approach of direct evaluation is spectrum auction. One of the latest example is underway C-band auctioning in the US already bidding USD 80 billions [17];

p_n — revenue of the n-th user. State operators use spectrum in order to fulfil political, economic and social purposes, so normally $p_n = 0$. Where commercial operators are concerned, taking this component into account is very important for establishing a favorable investment climate in the relevant segment of the market;

cap_n — capital costs of the n-th spectrum user in the k-th scenario. The costs of a radio system depend on the available bandwidth and the technical conditions of the spectrum use, e.g. propagation [4]. The wider and less occupied the required band, the lower the capital costs for network development, both in terms of coverage of the service area and cost of the equipment itself. Therefore, when analysing the input-output matrix, it is necessary to examine the relationship $cap_n = f(s_{nk})$. In the analysis of a spectrum reallocation scenario, this item has to take into account the associated expenses and of compensation of losers in accordance with potential Pareto criteria;

op_n — operating costs of the n-th spectrum user in the k-th scenario. These are also a function of bandwidth since, first, according to Shannon, bandwidth shortfalls for a given quality of service can be offset by increasing transmission power, and, secondly, in congested parts of the spectrum achieving interference-free operation of radio networks calls for highly qualified technical staff and the implementation of special technical measures requiring specialized hardware and software. On the basis of these considerations, it is necessary to examine the relationship $op_n = f(s_{nk})$. This item may also take into account an annual spectrum fee if such a fee is levied on users in the band in question;

tc_n — transaction costs for the n-th spectrum user in the k-th scenario. These can include the license and spectrum/auction fees, as well as reallocation costs;

ec_n — externality costs of the n-th spectrum user in the k-th scenario, defined by spectrum limitations. Inclusion of this contributor in economic analyses allows the principle on internalization to be met and to define real social cost of resource use resulting in a Pareto-optimal plan.

The overall costs for the n-th spectrum user in the k-th scenario may be expressed as follows:

$$w_n = p_n + cap_n(s_{nk}) + op_n(s_{nk}) + tc_n(s_{nk}) + ec_n(s_{nk}) \quad (1)$$

The basis of the input-output method lies in assessing the ratio of impact to costs, such that efficiency in use of the spectrum for the n-th user may be expressed as:

$$e_n = y_n / w_n \quad (2)$$

In cases where it proves impossible in practice to place an objective monetary value on the impact generated by use of the specific spectrum use, the cost ratio a_k may be used as an economic criterion to study allocation scenarios:

$$a_k = s_{nk} / w_n \quad (3)$$

The proposed approach offers an economic mechanism to objectively assess the overall impact of a decision in regard to the use of a band at any stage of the management cycle, and to compare alternative options with a view to selecting the optimum one in terms of the regulator priorities. A full scale introduction of the proposed method in order to improve efficiency of the spectrum allocation system requires development of big data system to support the processing of all required information. However, even simplified economic criteria (3) could provide essential results to be taken into account. For example, WRC-19 allocated

two possible bands for 5G introduction – 3.4-3.8 GHz and 4.8-4.99 GHz. A study of propagation model [18] and required base station density [19] in two bands shows that cost ratio a_k for the second frequency band will be almost twice as much, making low band utilisation more efficient under all other equal conditions (penetration rate, spectrum availability, economies of scale, border coordination, etc.).

5. CONCLUSIONS

Transition to a sustainable model of development through the digital economy is the guiding force driving the current stage of human technological progress. This is of immense importance, not so much for generating economic profit, as for identifying ways of ensuring humanity's continued survival. Humanity on board "Noah's Ark" consumes resources in an uncontrolled way through constantly increasing consumption that "bores holes in the hull", and thus unthinkingly destroys its critical life-support systems. We should hope that artificial intelligence will help us to understand that our current irrational devouring of resources has no future, and that even our immediate descendants would encounter catastrophic problems bequeathed to them by their grandparents.

Successful worldwide implementation of a digital economy initiatives must be based on the optimization of all available resources use with respect of available resources boundaries. One of these essential resources is the spectrum, playing a role as an integral element of the digital economy ecosystem. In terms of economic value, the spectrum could be ranked alongside national reserves of energy and raw materials. Multisided limits of spectrum use require international and national actions to avoid them to be transgressed. The digital economy urgently requires a digital spectrum management system to accommodate new radio technologies and to facilitate economic growth.

The input-output model can be considered as the proper mechanism for appropriate assessment and optimization of the use of resources at the macroeconomic level, providing a complex evaluation of the consequences of all possible scenarios and to assist decision makers in selection of those which best serve their strategic goals. Our "Noah's Ark" needs not only the wind of A. Smith and R. Coase market forces to fill its sails, but a steering wheel of W. Leontief and L. Kantorovich is required also, heating and bearing to true at all times in order to avoid reefs and shoals and to overcome storms. As the current situation shows, the market economy in some cases cannot resolve problems of negative externality and protect public interest. Resource management policy should consider social costs and internalization of negative externalities to ensure optimal social welfare, including environmental quality and protection of future generation interests.

REFERENCES

[1] R.H. Coase, "The Federal Communications Commission", The Journal Law and Economics, Volume 2, Oct. 1959.

[2] H.J. Levin, "The Invisible Resource: Use and Regulation of the Radio Spectrum". The John Hopkins University Press, 1971.

[3] CSP International Ltd., "Deregulation of the Radio Spectrum in the UK". H.M. Stationery Office, 1987.

[4] Report ITU-R SM.2012, "Economic aspects of spectrum management". Geneva, 2018. Available: https://www.itu.int/pub/R-REP-SM.2012.

[5] WT Docket No. 00-230, "Comments of 37 concerned economists. Promoting efficient use of spectrum", Feb. 2001.

[6] World Economic Forum. Statista, Feb. 2019. Available: https://www.weforum.org/.

[7] A. Ratier, "Socio-Economic Benefit of Polar Satellite Data", presented at the EUMETSAT Meteorological Satellite Conference. Geneva, 23 Sept. 2014.

[8] US Congress, "Middle Class Tax Relief and Job Creation Act of 2012", Public Law 112–96, Feb. 22, 2012.

[9] FCC News. "Additional spectrum to help keep Americans connected during coronavirus pandemic". Washington D.C., March 15, 2020.

[10] J. Conrad, "Resource Economics", Cambridge University Press, 2010.

[11] Report ITU-R SM.2352, "Technology trends of active services in the band above 275 GHz",

Geneva, 2015. Available: https://www.itu.int/pub/R-REP-SM.2352.

[12] Report ITU-R SM.2452, "Electromagnetic field measurements to assess human exposure", Geneva, 2019. Available: https://www.itu.int/pub/R-REP-SM.2452.

[13] Secure World Foundation, "Space Sustainability", 2017.

[14] Report ITU-R M.2477, "Radiocommuncations for suborbital vehicles", pp. 5-6. Geneva, 2019. Available: https://www.itu.int/pub/R-REP-M.2477.

[15] Earth Project, "Economic and Ecological Impact of ICT", 30 April 2011.

[16] V.G. Gorshkov, "Physical and Biological Bases of Life Stability", Man. Biota, Environment, Berlin, Springer-Verlag. 1995.

AUTHOR

Vadim Nozdrin received a Telecommunication engineer degree in 1988 and Ph.D. in spectrum management in 1999 from Moscow Technical University of Telecommunication and Informatization and Dr. habil. in telecommunication management from St. Petersburg State University of Economics in 2019.

From 1988 to 2000, he has devoted his career to Radio Research & Development Institute (NIIR) in Moscow in positions with increasing responsibility, ultimately attaining the position of Deputy Director on International Spectrum Management Affairs. In 2000, he joined Space Service Department of Radiocommunication Bureau in ITU, dealing with preparation for World Radiocommunication Conferences, technical and regulatory applications of Radio Regulations and expert support for international radiocommunications worldwide. Since 2010, he is Counsellor for ITU-R Study Group 7 and Working Group 5B.

He has published more than 60 scientific articles on radiocommunication and spectrum management issues as well as co-authoring three monographs.

[17] FCC News. Major 5G Spectrum Auction. Available: https://www.fcc.gov/document/fcc-begins-major-5g-spectrum-aucton.

[18] Recommendation ITU-R P.1410, "Propagation data and prediction methods required for the design of terrestrial broadband radio access systems operating in a frequency range from 3 to 60 GHz", Annex 1. Geneva, 2012. Available: https://www.itu.int/rec/R-REC-P.1410/en.

[19] Report ITU-R M.2292, "Characteristics of terrestrial IMT-Advanced systems for frequency sharing/interference analyses", Annex 1, Table 2.1. Geneva, 2013. Available: https://www.itu.int/pub/R-REP-M.2292.

PERFORMANCE OF A PARALLEL HAMMING CODING IN SHORT-FRAME OFDM SENSOR'S NETWORK

Raouia Masmoudi Ghodhbane[1] and Jorge Fernandez-Mayoralas[1,2]
[1]Safran Tech, Safran Sensing Systems $(\mathcal{S}^3)$, Rue des Jeunes Bois, Châteaufort , 78114 Magny-les-Hameaux, France,
[2]CentraleSupelec, Integrated Circuit and Electronic Systems, rue Joliot-Curie, F-91192 Gif-sur-Yvette Cedex, France.

NOTE: Corresponding author: Raouia Masmoudi Ghodhbane, raouia.masmoudi@gmail.com

Abstract *– In this paper, we focus on the most relevant Error Correcting Codes (ECCs): the Hamming code and the Reed-Solomon code in order to meet the trade-off between the low implementation complexity and the high error correction capacity in a short-frame OFDM communication system. Moreover, we discuss and validate via simulations this trade-off between complexity (Hamming is the easiest to code) and error correction capability (Reed-Solomon being the most effective). Therefore, we have to either improve the correction capacity of the Hamming code, or decrease the complexity cost for the Reed-Solomon code. Based on this analysis, we propose a new design of parallel Hamming coding. On the one hand, we validate this new model of parallel Hamming coding with numerical results using MATLAB-Simulink tools and BERTool Application which makes easier the Bit Error Rate (BER) performance simulations. On the other hand, we implement the design of this new model on an FPGA mock-up and we show that this solution of a parallel Hamming encoder/decoder uses a few resources (LUTs) and has a higher capability of correcting when compared to the simple Hamming code.*

Keywords – error correcting codes, FPGA, Hamming code, parallel Hamming coding, OFDM communication systems, Reed-Solomon code, short-frame, sensor's network, VHDL simulation

1. INTRODUCTION

Nowadays, Orthogonal Frequency Division Multiplexing (OFDM) systems are increasingly used in several applications such as: digital radio, television broadcasting systems, mobile communication systems and Power-Line Communications (PLC). OFDM is a convenient modulation scheme which combines the advantages of high data rates and easy implementation.

However, the main challenge associated with OFDM communication systems today remains related to Error Correcting Codes (ECCs) implementation [2]. Encoding and decoding are very important blocks in OFDM systems in which redundant information is added to the signal to allow the receiver to detect and correct errors that may have occurred in transmission. There are many techniques for error detection and correction such as: the Hamming code [3], Turbo code [4], Bose, Chaudhuri, and Hocquenghem (BCH) code [5], Reed-Solomon code [6], Convolutional code/ Viterbi [7], and LDPC [8].

One way to compare the efficiency of several ECCs is to compare their performance in terms of their complexity of implementation and their error detection or/and correction capability. Hamming and Reed-Solomon codes have proved to be a good compromise between efficiency and complexity. Hamming is very easy to implement and does not consume many resources, and it is a robust ECC, but the Bit Error Rate (BER) performance is not the most effective. The Reed-Solomon is more optimal to eliminate errors (especially for burst errors), but it is also more complex to implement.

Our domain of application is PLC based on OFDM systems for aeronautic sensor's network [9, 10]. According to [10], a PLC channel has major limitations especially in terms of bandwidth, impedance mismatches and noise.

Since power consumption has been a critical factor in the design of sensor's network, we consider sensor's network with short periods of activity, in which some sensors' devices go into standby mode. This consideration and the need for low latency access [11] to sensor data underscores the need for a short-frame OFDM and fast communication protocol.

In this paper, we focus on a low-powered short-frame OFDM communication system where the sensors' devices send only one OFDM symbol per frame on the PLC channel. We simulate and model the design of this low-powered short-frame OFDM communication system in terms of error correction. As mentioned previously, the complexity of the chosen ECC is one of the studied criteria: the ECC has to be effective and fairly easy to implement at the same time.

Usually, to have better performance in terms of error correction and/or error detection, we have to add several error correcting codes (ECCs) in serial either at the encoder or on the decoder level. Our case of applications requires a very simple ECC in order to implement it on a High Temperature (HT) ASIC whose number of cabled flip-flops and multipliers are limited by the HT technology.

Therefore, the novelty of this paper lies in the performance analysis of a new design based on the parallel Hamming coding which meets the trade-off between the low implementation complexity and the high error correction capacity. In the literature, the closest work to ours

is [12] in which the authors used parallel filters with only one Hamming code as an error correcting code. Their scheme allows more efficient protection when the number of parallel filters is large.
Our contributions can be summarized as follows:

- The design of this low-powered short-frame OFDM communication system in terms of error correction is modelled by MATLAB-Simulink tools.
- We analyse and choose the adapted Error Correcting Codes (ECCs); such as Hamming and Reed-Solomon for this low-powered short-frame OFDM communication model.
- We discuss the trade-off between the low implementation complexity and the high error correction capacity.
- Moreover, we propose a new model of parallel Hamming coding in order to increase the error correction capability of our model and we illustrate its performance in terms of Bit Error Rate (BER) vs. EB/N0.
- Finally, we validate the performance of parallel Hamming encoder/decoder in terms of complexity of implementation on an FPGA board using VHDL simulations.

The remainder of this paper is organized as follows: First, a brief description of the communication model in Section 2. Then, we study the trade-off between the two ECCs (Hamming code and Reed-Solomon code) which are more adapted to our communication model. We propose a new design of a parallel Hamming coding in the case of a short-frame OFDM sensor network in Section 3. Moreover, we illustrate the performance of these ECCs in terms of Bit Error Rate (BER) for different scenarios using BERTool application in Section 4. In Section 5, we validate the performance of parallel Hamming encoder/decoder in terms of complexity of implementation on an FPGA board using VHDL simulations. Finally, we conclude this paper in Section 6.

2. DESCRIPTION OF THE COMMUNICATION MODEL

Here, we consider a sensor's network which is composed of one master device and several slave-sensor node devices $(S_1, S_2, ..., S_{20})$ using Time Division Multiple Access (TDMA) to share the PLC channel illustrated in Fig. 1.

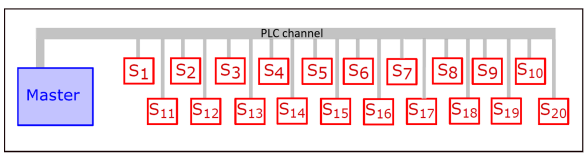


Fig. 1 – Shared bus architecture for our sensor's network

Since low-power consumption is a critical factor in the design of sensor's network, we consider a low-powered short-frame OFDM communication system where sensors' devices send only one OFDM symbol per frame on the PLC channel. Each time slot is composed of an OFDM symbol which represents a communication between the master device and each slave-sensor node device.
In Table 1, we denote our short-frame OFDM parameters that are considered later in our MATLAB-Simulink simulations:

Table 1 – Scenarios and short-frame OFDM parameters used in our MATLAB-Simulink simulations

Parameters	Value
N_{FFT}: FFT size	64
N_{SC}: Used sub-carriers	30/32
f_s: Sampling frequency	1.6 MHz
Δ_f: Sub-carrier spacing	25 KHz
T_s: Total Symbol duration	$45\mu s$
M: Modulation size	4 (QPSK)
L: Packets of L bits	50 bits
ECC: Error correcting codes	Hamming, Reed-Solomon
CR: Coding rates	$CR_1 = (4/7), CR_2 = (11/15),$ $CR_3 = (23/31), CR_4 = (26/31),$ $CR_5 = (57/63), CR_6 = (56/64)$
$BERTool$: BER Performance Analyser GUI	$N_{errors} = 10^6$; $N_{bits} = 10^{10}$; $\frac{E_b}{N_0} = 0 : 15$ dB

In this paper, we use MATLAB-Simulink tools in order to model this low-powered short-frame OFDM communication system as shown in Fig. 2.

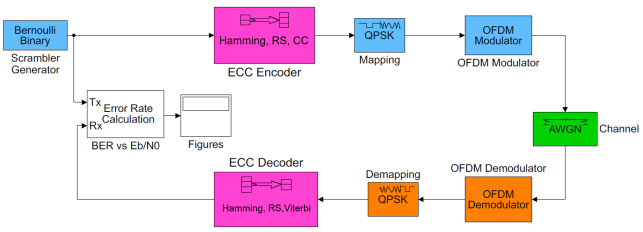


Fig. 2 – Short-frame OFDM communication model using MATLAB-Simulink tools: General model

Next, we will describe each block represented in Fig. 2 representing our general low-powered short-frame OFDM communication model:

2.1 Random input generation

In this block, the input data is randomized by a "Bernouilli Binary" block in order to spread the energy over all the bits before being encoded by the following block.

2.2 Error correcting code (ECC)

The purpose of this block is to add enough redundancy to the data packets being sent, so that even if some of the received data includes errors, there will be enough infor-

mation provided to the receiver to reconstruct the data from its origin. In our communication model, we suppose that the ECC will be in charge of correcting all the errors created by the OFDM communication model.

- ECC encoder: This block is the transmitter side of the ECC. The communication channel alters the signal, which introduces errors. We can use error correcting codes to add redundancy in the transmitted data. Many encoding methods have been studied in the literature, such as the Hamming and the Reed-Solomon codes, which are detailed in Section 3.
- ECC decoder: This block is on the reception side of the model, receiving the information that has passed through the channel and that has already been demodulated. The decoder has to decode the message and translate it back into its' original form by exploiting the redundant bits that were added previously by the encoder and that allows the decoder to detect an error and correct it. In general, the decoder is more complex than the encoder, especially concerning its implementation (for example Viterbi decoder [7]).

2.3 Mapping/De-Mapping

This block takes the coded messages and builds a constellation in an I/Q plane, in order to transmit our digital stream of bits through the analog channel, under the form of frequencies.

- Mapping: This block takes a binary stream and outputs a point in the I/Q plane. There are many types of modulation: such as shifting the phase of the signal 'Phase-Shift Keying (PSK)', modulate only the amplitude of the signal 'Amplitude Modulation (AM)', or both 'Quadrature amplitude modulation (QAM)'. In our communication model, we use a Quadrature phase-shift keying (QPSK) modulation. There are $M = 4$ possible points in the constellation and it can encode $n_b = \log_2(M) = 2$ bits per symbol.
- De-Mapping: This block does the inverse of the previous mapping block. It takes the signal in the time domain after the Fast Fourier transform (FFT) block (which is inside the OFDM modulator block), and with the constellation reference point it restores the original binary message corresponding to a given point.

2.4 Orthogonal Frequency Division Multiplexing (OFDM) MoDem

OFDM is a type of digital transmission and a method of encoding digital data on multiple carrier frequencies. In OFDM, multiple closely spaced orthogonal sub-carrier signals with overlapping spectra are transmitted to carry data in parallel.

- OFDM Modulator: This block introduces the pilot signals (which are used later in the channel estimation block to correct the effects of the channel), the guard bands, the cyclic prefix and a raised cosine filter (which serves as windowing to focus our study on the used frequencies). Then, it includes also an Inverse Fast Fourier transform (IFFT) module which allows the information to be transmitted in orthogonal frequencies through the communication channel.
- OFDM Demodulator: This block does the reverse operation of the previous one. An FFT module transforms the information back into the time domain, the pilot signals are used for the channel estimation, then they are removed to restore the original message (the guard bands and cyclic prefix are removed as well).

2.5 The communication channel

Here, the communication channel is modelized by an Additive White Gaussian Noise (AWGN) block. We induce noise by using the $\frac{Eb}{No}$ which represents a normalized Signal to Noise Ratio (SNR) by bit, and is well adapted to compare the Bit Error Rate (BER) of different ECCs. The relation between these two parameters is given by:

$$SNR = \frac{Eb}{No} * \frac{Bit\ Rate}{Channel\ Bandwidth} \qquad (1)$$

The deviation (which is equal to $10mV$ in our case) represents the thermal noise in our real Power Line Communication (PLC) channel [9, 10]. We have estimated this deviation on LT Spice tools by adding the quantification noise and converting the sine wave by the ADC/DAC block. In order to make the channel block more realistic, we fixed these parameters of the AWGN block the closest to reality.

In the following Section, we will focus only on the ECC blocks which are described in Section 2.2 and adapted to our short-frame OFDM communication system.

3. TRADE-OFF BETWEEN SIMPLE HAMMING CODE AND REED-SOLOMON CODE

In this Section, we will focus on two error correcting codes (ECCs) that meet a trade-off between the low implementation complexity and the high error correction capacity. Since we have considered a short-frame OFDM communication model, we will study Hamming code and Reed-Solomon code which are the most adapted in our case.

3.1 The Hamming code

The Hamming code is one of the error correcting codes that can be used to detect and correct bit errors that can occur when data is moved or stored. Like other error correcting codes, the Hamming code makes use of the concept of parity bits which are bits that are added to data so that the validity of the data can be checked when it is read or after it has been received in a data transmission.
The Hamming code method is based on two methods (even parity, odd parity) for generating redundancy bits.

The number of redundancy bits are generated using the following formula:

$$2^r = D + r + 1, \qquad (2)$$

where, r represents the number of redundancy bits and D the number of information data bits.
For example, if we calculate the number of redundancy bits for a $D = 11$ bits then it comes to add $r = 4$ redundancy bits. These parity/redundancy bits (P_1, P_2, P_4, P_8) are added to the information bits $(D_1, ..., D_{11})$ at the transmitter (Hamming encoder) and then removed at the receiver (Hamming decoder) which is able to detect and correct errors.

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits	P_1	P_2	D_1	P_4	D_2	D_3	D_4	P_8	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}
P_1	x		x		x		x		x		x		x		x
P_2		x	x			x	x			x	x			x	x
P_4				x	x	x	x				x	x	x	x	x
P_8								x	x	x	x	x	x	x	x

Table 2 – The encoded bits for Hamming code $[15, 11]$

The Hamming encoder calculates these parity bits according to Table 2, and outputs a 15 bits message. The Hamming Decoder calculates the parity bits:

- If each parity bit is equal to zero, i.e., $(P_1, P_2, P_4, P_8) = 0000$, there are no errors in this OFDM communication model.
- If not, the position of the error is displayed in the four parity bits (P_1, P_2, P_4, P_8). The decoder flips then the concerned bit and returns the 11 bits message $(D_1, ..., D_{11})$.

Next, we will simulate the Hamming code $[n, k]$ for several coding rates: $CR_1 = (4/7), CR_2 = (11/15), CR_4 = (26/31)$, where each coding rate $CR = k/n$ is the ratio between the code dimension k and the code length n, in order to study its influence on the system's performance.

The theoretical point of view "the longer the code, the better the error performance" is proved in Fig. 3 and Fig. 4. Fig. 3 shows that the coding rate is crucial to obtain a good performance. When $\frac{E_b}{N_0} \leq 9$ dB, the Hamming code curve with the coding rate CR_4 is very close to the Hamming code curve with the coding rate CR_2 in terms of BER.

3.2 The Reed-Solomon code

The Reed-Solomon code operates on a block of data treated as a set of finite-field elements called symbols. Reed–Solomon code is able to detect and correct multiple symbol errors especially burst errors. Since the Reed-Solomon code is a non-binary code, the code has symbols from $\mathbb{F}_q$ with parameters $(q-1, k)$, which is used to make a mapping of primitive polynomial with binary coefficients,

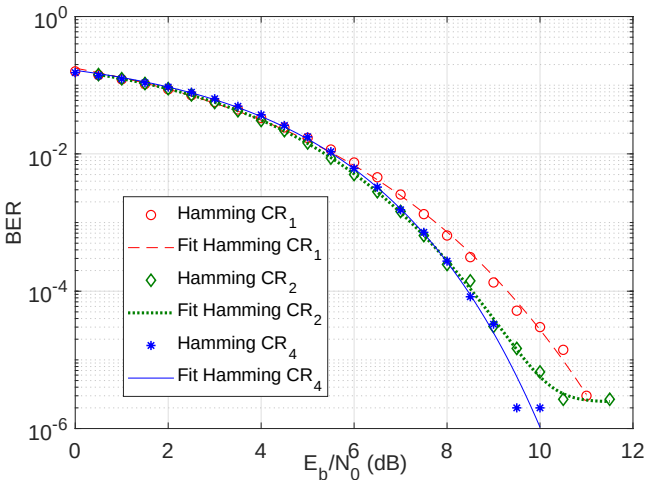


Fig. 3 – BER performance of the Hamming code for different coding rates CR_1, CR_2, CR_4

frequently we have $q = 2^m$ to use them as binary codes, each element being represented as a binary m-tuple.
In terms of complexity, the Reed–Solomon encoder is fairly simple in terms of blocks and only involves multipliers and adders in the Galois Field. We can either create a multiplication module or use RAM slots and create multiplication tables, However, the Reed–Solomon decoder includes several algorithms that consume a lot in resources, especially the Berlekamp algorithm.
Next, we will simulate the Reed-Solomon code (n, k) for several coding rates: $CR_1 = (4/7)$, $CR_2 = (11/15)$, $CR_3 = (23/31)$, where each coding rate $CR = k/n$ represents the ratio between k the code dimension and the code length $n = q - 1$, in order to study its influence on the system's performance.

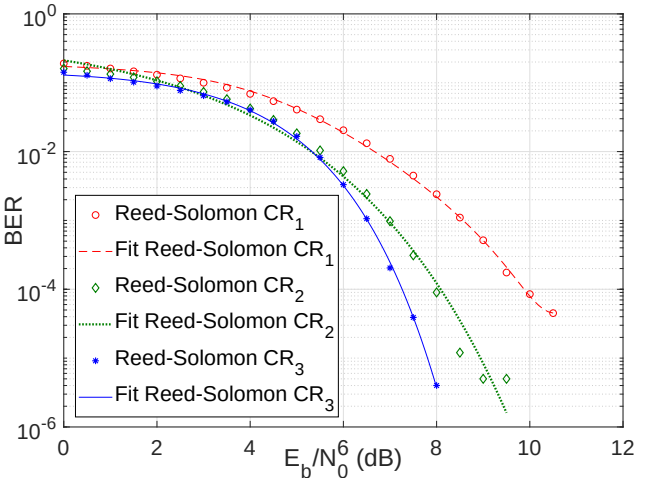


Fig. 4 – BER performance of the Reed-Solomon code for different coding rates CR_1, CR_2, CR_3

Once again, it is a trade-off between performance and complexity with different coding rates: as the most effective coding is the highest, but also the most complex. Fig. 4 shows that the three curves converge faster to zero when compared to the hamming code curves in Fig. 3. In fact, the Reed-Solomon curve with the highest coding rate

CR_3 manages to converge to no errors for $\frac{E_b}{N_0} = 8$dB, which represents the noise level that we will find in the real PLC channel.

3.3 Discussion and comparison

As we discussed before, the complexity and the capacity of error correction of the chosen ECC are the most important criteria: the ECC has to be effective and fairly easy to implement at the same time.

The capacity of error correction of each ECC is known and depends on its structure and their way of coding the information. Table 3 summarizes the detection and correction capability of each ECC scheme. We denoted here by SED/DED/MED: the Single/Double/Multiple Error Detection, and by SEC/DEC/MEC: the Single/Double/Multiple Error Correction.

Table 3 – The capability of error detection or/and error correction for each ECC.

ECC	SED	SEC	DED	DEC	MED	MEC
Hamming code	x	x				
Extended Hamming	x	x	x			
Reed-Solomon Code	x	x	x	x	x	x
BCH Code	x	x	x	x	x	x

While the Hamming code can be implemented creating just two fairly simple modules (encoder/decoder) based on XOR gates, the Reed-Solomon code requires a higher number of blocks. In fact, the Reed Solomon encoder needs: Adder in Galois Field, Multiplier in Galois Field, Multiplex and Registers [13]. The Reed-Solomon decoder needs algorithms to decode the code word such as: Syndrome calculator; Berlekamp-Massey algorithm (or Euclid algorithm)which finds the location of the errors by creating an error locator polynomial; Chien Search Algorithm which finds the roots of the previous polynomial; Forney's algorithm where the symbol's error values are found and corrected. Thus, these blocks are complex and use multiplications (or RAM memory if we use tables).

Here, we have simulated in Fig. 5 the Hamming code and the Reed-Solomon code for the same coding rate CR_2 in order to compare their BER performance.

The Hamming and Reed-Solomon codes have proved to be a good compromise between efficiency and complexity. Hamming is very easy to implement and does not consume many resources, and it is a robust ECC, but the Bit Error Rate (BER) performance (in Fig. 5) shows that it is not the most effective ECC. Reed-Solomon is more optimal to eliminate errors, but it is also more complex than the Hamming code.

In the following Section, we will propose a new model of parallel Hamming coding in order to increase the error correction capability of our model.

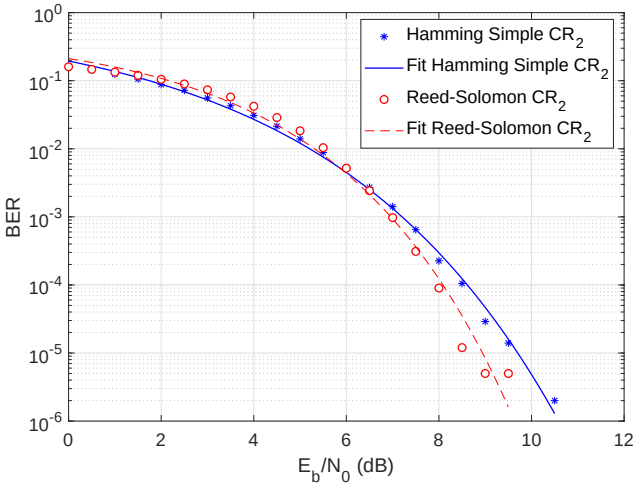


Fig. 5 – Comparing the simple Hamming code to the Reed-Solomon code

4. PARALLEL HAMMING CODING RESULTS USING MATLAB-SIMULINK AND BERTOOL

In this Section, we have selected only a few of the most illustrative and interesting scenarios to be presented here. In order to plot the Bit Error Rate (BER) function of the $\frac{E_b}{N_0}$ which is given in equation *(1)*, we have used the Matlab tools called "BERTOOL".

4.1 BER performance analyser for MATLAB-Simulink models

Here, we use the Bit Error Rate (BER) analysis GUI (called BERTool) from Matlab application. The BERTool application is able to analyze the BER performance of different communications systems as a function of signal-to-noise ratio $\frac{E_b}{N_0}$ given in equation *(1)*. It analyzes performance either with Monte-Carlo simulations of MATLAB functions and MATLAB-Simulink models or with theoretical closed-form expressions for selected types of communication systems[14].

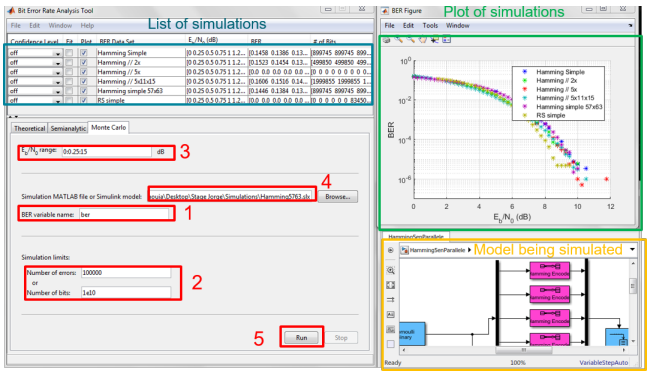


Fig. 6 – Bertool Interface: steps to plot the BER vs EB/No simulations

On the Monte Carlo window in Fig. 6, we have specified the BERTool parameters (which are detailed in Table 1) based on our scenarios. In order to generate BER data for each communication system using the Simulink models, we follow five steps based on the following BERTool process [15]:

1. We calculate Bit Error Rate (BER) as a function of the energy per bit to noise power spectral density ratio ($\frac{E_b}{N_0}$).

2. We fix the number of errors ($N_{errors} = 10^6$) and the number of bits ($N_{bits} = 10^{10}$) in order to make accurate error rate. We have chosen this number of bits value to prevent the simulation from running too long, especially at large values of $\frac{E_b}{N_0}$.

3. We have specified the $\frac{E_b}{N_0}$ range based on our PLC channel model: $\frac{E_b}{N_0} = 0 : 15$ dB.

4. We generate the BER data for a chosen Simulink model. This Simulink Model is displayed and run in real time on the models being simulated window (as shown in Fig. 6) for each value of the energy per bit to noise ratio $\frac{E_b}{N_0}$. BERTool iterates over our choice of the energy per bit to noise ratio $\frac{E_b}{N_0}$ value and collects the results on a list of simulations.

5. Finally, we run the simulation in order to plot the estimated BER values function of the previous steps. The plot of simulation window is displayed and shows each curve for each Simulink model. We save later in the list of simulations each curve in order to compare graphically the different models.

In the following paragraphs, we will describe the most interesting models that we have simulated with the BERTool application interface as shown in Fig. 6.
As we have discussed in the previous Section, Hamming seems to be the most appropriate ECC for our scenario, but the small correction capacity could be an obstacle if there are several errors in the 50 bit package. For instance, we consider the Hamming code $[n, k]$ with coding rate $CR = k/n$ where n represents the code length and k the code dimension.

- Model 1: We cut the message on 1 time when we are coding with Hamming coding rate $CR_5 = 57/63$; Since in Model 1, the Hamming code has length 63 and dimension 57, we have to append 7 zeros to 50 information bits and then do encoding.

- Model 2: We cut the message on 2 times when we are coding with Hamming coding rate $CR_4 = 26/31$; Since in Model 2, the Hamming code has length 62 and dimension 52, we have to append only 2 zeros to 50 information bits before encoding.

- Model 3: We cut the message on 5 times when we are coding with Hamming coding rate $CR_2 = 11/15$. Since in Model 3, the Hamming code has length 75 and dimension 55, we have to append 5 zeros to 50 information bits before encoding.

Model 1: Simple Hamming code $[63, 57]$

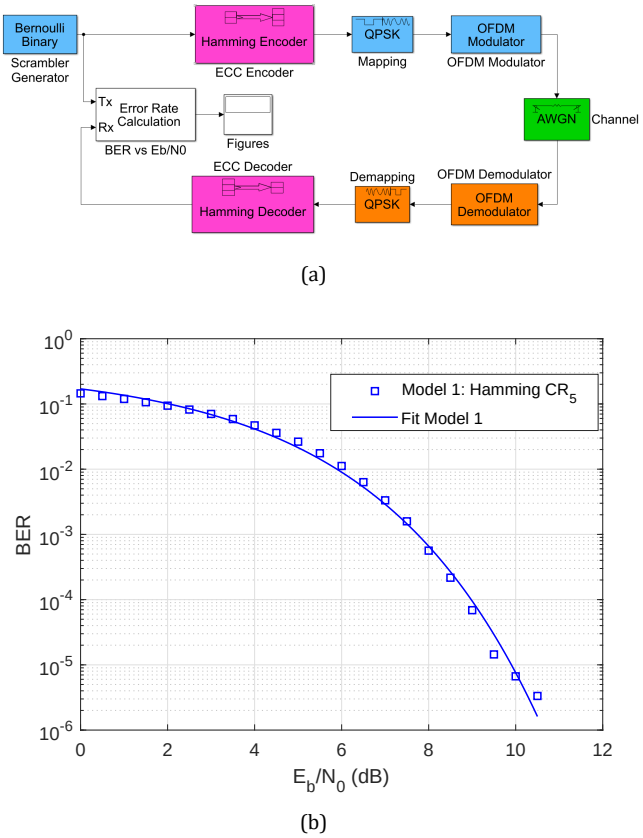


Fig. 7 – Model 1: (a)Short-frame OFDM communication model using MATLAB-Simulink tools (b) BER performance plotted on BERTool application

We use MATLAB-Simulink tools in order to model the simple Hamming code communication system $[63, 57]$ as shown in Fig. 7.

We generate the BER data for a simple Hamming code communication system model $[63, 57]$. This MATLAB-Simulink model (see Fig. 7 (a)) is displayed and run in real time on the models being simulated window (as shown in Fig. 6) for each value of the energy per bit to noise ratio $\frac{E_b}{N_0}$. Then, we obtain the plot in Fig. 7 (b) as the BER performance vs. $\frac{E_b}{N_0}$ for this model.

For Model 1, we have a very long coding rate (around 50 bits for the input message). In the following, we will compare to the concatenation of several encoders/decoders with shorter coding rates in order to prove that parallel Hamming has better performances than Hamming simple, while keeping the same simplicity of implementation.

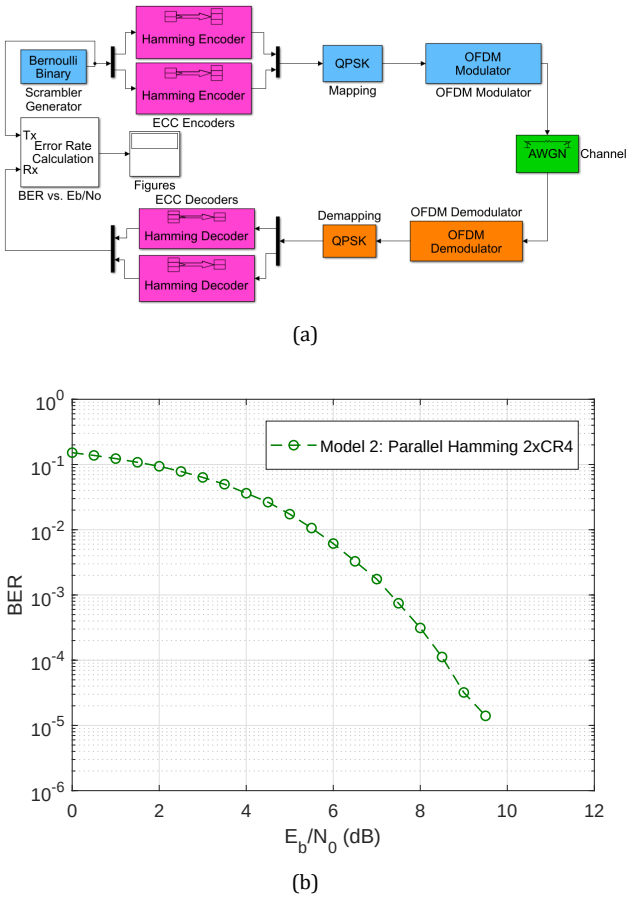


Fig. 8 – Model 2: (a)Short-frame OFDM communication model using MATLAB-Simulink tools (b) BER performance plotted on BERTool application

Model 2: Parallel Hamming code 2*[31,26]

In order to improve the correction capacity of Model 1: Simple Hamming code Model $[63, 57]$. Here, we propose to implement 2 couples of encoders/decoders in parallel respectively, so we can improve the correction capability of the Hamming code; which will be 2 bits out of 52 bits (20% of the total message), and detection would be 4 out of 52, while the first model could only detect 2 and correct 1. We use MATLAB-Simulink tools in order to model the parallel Hamming code communication system $2\times[31, 26]$ as shown in Fig. 8.

Model 3: Parallel Hamming code 5*[15,11]

For Model 3, we can cut the message on 5 times, each of $N = 11$ bits, which would be coded by 5 encoders Hamming $5 \times [15, 11]$. Thus, the capacity of correction in this case would be of 5 bits out of 55 (10% of the total message), and detection would be 10 out of 55, while the first model could only detect 2 and correct 1. We use MATLAB-Simulink tools in order to model the parallel Hamming code communication system $5 \times [15, 11]$ as shown in Fig. 9.

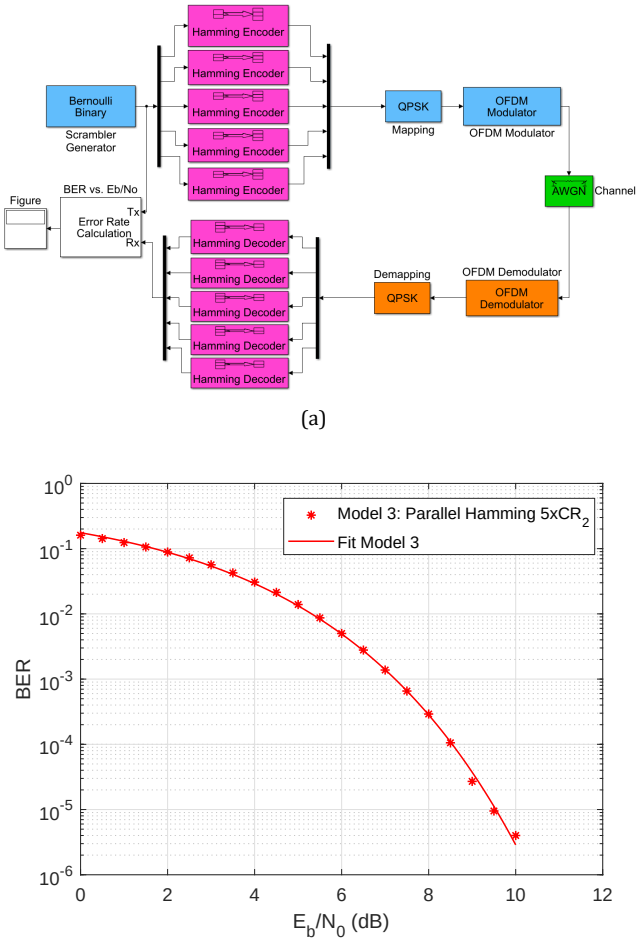


Fig. 9 – Model 3: (a)Short-frame OFDM communication model using MATLAB-Simulink tools (b) BER performance plotted on BERTool application

4.2 Comparison between parallel Hamming and simple Reed-Solomon

In this subsection, we will compare the previous models that are adapted to our technical requirements: Hamming (Model 1, Model 2, Model 3) to Reed-Solomon CR_6 (Model 4). We want to verify if there is a significant difference in BER performance: since parallel Hamming coding is very interesting in terms of simplicity and robustness, however Reed-Solomon is very interesting in terms of error correction capability.

In this scenario, we choose Model 4 where $CR_6 = 56/64$ is the simple Reed-Solomon coding rate (around 50 bits). Then we repeat the same simulations done for Model 1, Model 2 and Model 3 and for Reed-Solomon (Model 4) several times for each model and we calculate the average points for each value of $\frac{E_b}{N_0}$. We plot then the results in Fig. 10.

As we can see in Fig. 10, the parallel coding represents a gain in correction capacity, and both Model 2 and Model 3 converge faster to no errors than its equivalent with simple Hamming (Model 1). In our application case, and es-

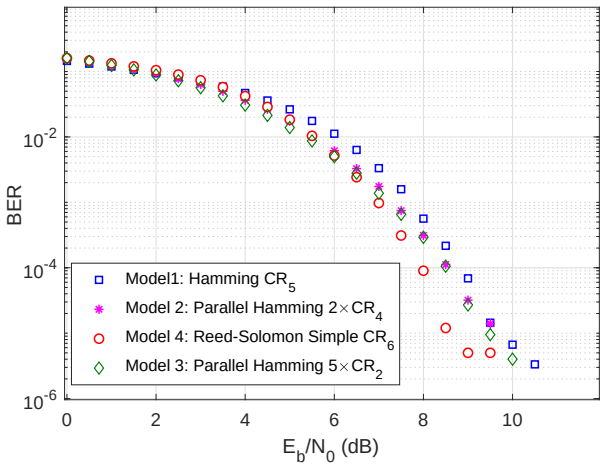


Fig. 10 – Comparing simple, parallel Hamming and Reed-Solomon codes for the same total size (around 50 bits).

pecially through our PLC channel, we have an average error probability almost equal to 3 errors for a packet of 50 bits. That 's why, Model 3 seems to be slightly better, especially towards the end, which can seem logical given the fact that Model 3 can have a correction of up to 5 errors, while Model 2 can only correct 2 errors.

Even if the Reed-Solomon code is apparently more efficient, we can see at the end of the Reed-Solomon simulation curve that the BER converges suddenly to zero: this is due to the complexity of the Reed-Solomon algorithm, MATLAB-Simulink tools couldn't simulate for long periods of time and we could only send a finite amount of bits before it made the simulation stop. The two last points of the Reed-Solomon simulation curve are stagnant because it's the limit of the simulation given its complexity.

However, Hamming curves (for Model 1, Model 2 and Model 3) can go to lower values of BER $\leq 10^{-6}$, which proves that it's easier and robust.

To give an order of magnitude: for Model 1 (simple Hamming code), we simulated 2 million bits (for each value of $\frac{E_b}{N_0}$), while for Model 4 (simple Reed-Solomon) we could only simulate 700 thousand bits which shows the complexity of the Reed-Solomon code when compared to the Hamming code.

Concerning their performance, in Fig. 10, we remark that Reed-Solomon is better, especially for $\frac{E_b}{N_0} \geq 8$ dB and with a high value of $\frac{E_b}{N_0}$ there are few errors to be corrected. Nevertheless, this gain in effectiveness for Reed-Solomon has a cost in complexity when compared to Model 3 of the Parallel Hamming coding.

Based on the previous analysis, we have discussed and validated via simulations the trade-off between complexity (Hamming is the easiest to code) and error correction capability (Reed-Solomon being the most effective). Table 4 summarizes the advantages and disadvantages of each ECC. Therefore, we have chosen to improve the correction capacity of the Hamming code instead of decreasing the complexity cost for the Reed-Solomon code since we have

Table 4 – ECCs comparison in terms of complexity of implementation and capacity of correction

ECCs	Advantages	Disadvantages
Simple Reed-Solomon	Very effective especially with burst errors; High correction capacity: can correct multiple errors simultaneously.	Complex and need more resources (LUT) than Hamming code.
Parallel Hamming $5 \times [15, 11]$	Easy to implement; Correction capacity: correct 5 of 10 detected errors for a 50 bits packet.	Not the most effective in terms of BER.

shown in Fig. 10, the curve of Model 3, which represents a new design of parallel Hamming coding, is closer to Reed-Solomon than the other models of Hamming coding.
For these reasons, we have chosen parallel Hamming $5\times$encoder/decoder CR_2 (Model 3) to be implemented next in VHDL in order to show that this solution uses a few resources and has a higher capability of correcting compared to the simple Hamming code.

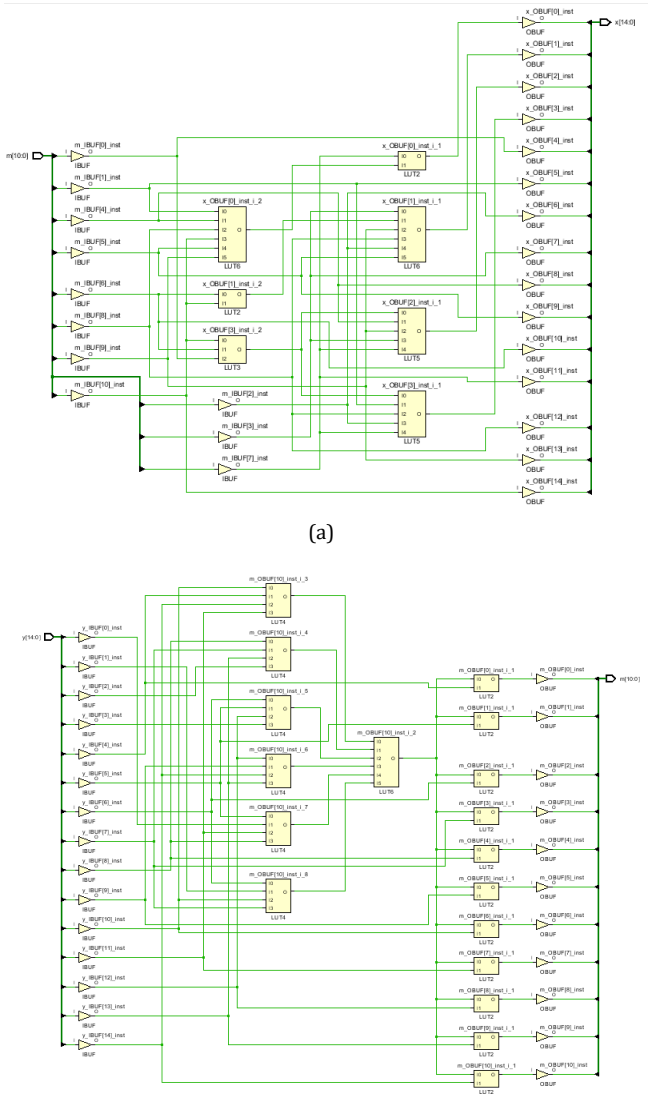
5. IMPLEMENTATION OF PARALLEL HAMMING ENCODER/DECODER

In this Section, we will analyse and validate the low complexity of Model 3 by implementing the design of the parallel enCOde/DECoder (CODEC) on an FPGA mock-up and simulating this design on VHDL code.

5.1 Parallel Hamming CODEC design

As we discussed before, the idea here is to make a trade-off between Hamming simplicity of implementation and Reed Solomon's capacity of correction and performance. In fact, with parallel Hamming encoder/decoder (Model 3), we will consume five times more resources than with simple one Hamming encoder/decoder, but we will have a notable improvement in terms of BER performance.

In Fig. 11, Hamming encoder/decoder $[15, 11]$ module is the base module to create our parallel Hamming encoder/decoder (Model 3), which is composed of $5\times$ Hamming encoders/decoders $[15, 11]$ added to Demux/Mux to concatenate the messages respectively.
The encoder/decoder circuit to compute the parity bits of the Hamming encoder/decoder $(11, 15)$ is shown in Fig. 11. These parity bits (P_1, P_2, P_4, P_8) are added to the in-



(a)

(b)

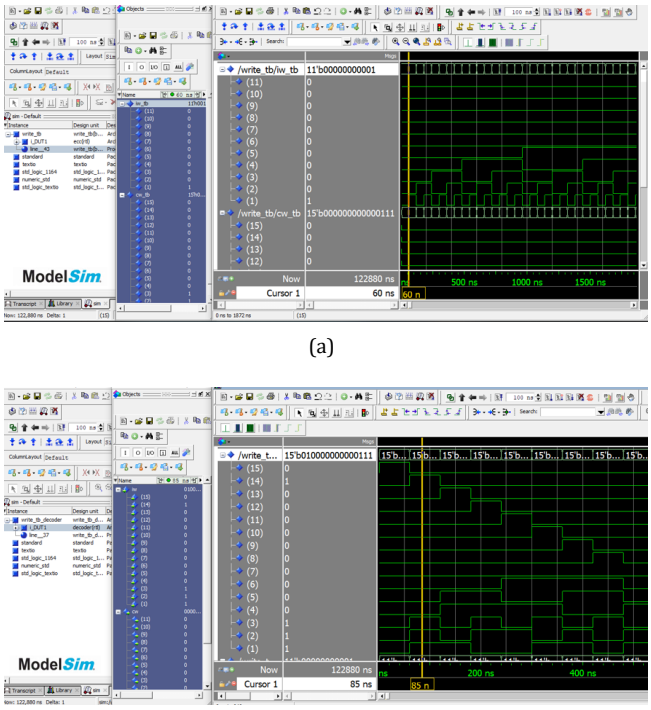
Fig. 11 – Design in terms of logic ports of: (a) One Hamming encoder $[15, 11]$; (b) One Hamming decoder (15,11)

formation bits $(D_1, ..., D_{11})$ at the transmitter (Hamming encoder) and then removed at the receiver (Hamming decoder) which is able to detect and correct errors.

5.2 VHDL functional simulations

Next, we will create the VHDL functional simulation of the Hamming encoder/decoder $[15, 11]$ module based on two VHDL files, one for the encoder which calculates parity bits and outputs the original message with the parity bits added in specific positions and one for the decoder which recalculates the parity bits to locate the error, correct it, and outputs the original 11 bit message.

Therefore, we make a test bench for each file in order to test their encoding and correction capability. In the Fig. 12, we can see the simulation in ModelSim of the test benches of both the encoder and the decoder. The input is the 2048 values that the 11 bit message can take, the output is this same file so our encoder $[15, 11]$ works correctly.



(a)

(b)

Fig. 12 – VHDL functional simulation output of: (a) Hamming encoder $[15, 11]$; (b) Hamming decoder $(15, 11)$

Next, once this step has been simulated successfully in ModelSim, we we will implement and synthesize the code in Vivado which represent the Xilinx software for this FPGA mock-up.

5.3 Performance and experimental results

As we have seen before, we conclude our study of the Hamming encoder/decoder $[15, 11]$ by synthesizing and implementing the VHDL code using Vivado. We have used a Xilinx Spartan 7 mock-up for the implementation. Using the same code that was validated previously by ModelSim, we create one project for each module (encoder and decoder) and we seek to verify the consumed and utilized resources in the FPGA mock-up.

As we can see in Fig. 11, the synthesis was successful and we manage to create a logic port scheme of both encoder and decoder.

We denoted by LUT: a Look-up table which represents a small asynchronous SRAM used to implement combinational logic and by IO: an Input/Output Buffer.

Table 5 shows the amount of look-up tables that are taken by our encoder and decoder. With a total of $6 + 10 = 16$ $LUTs$ for the couple encoder/decoder, we confirm Hamming's code simplicity that was supposed in our analysis in Section 3.

Furthermore, knowing how much one Hamming encoder/decoder module would consume, we can deduce that our parallel coding with five couples of encoders/decoders would consume $5 \times 16 = 80$ $LUTs$, which is

still fewer than Reed-Solomon's and without recurring to multiplication or RAM-consuming tables. This amount of $LUTs$ consumed is a trivial and negligible amount compared to the available $LUTs$ in our board.

Table 5 – Parallel Hamming and Reed-Solomon encoder/decoder resources

ECC	Resource type	Utilisation
Hamming Encoder $5 \times [15, 11]$	LUT	30
	IO	130
Hamming Decoder $5 \times [15, 11]$	LUT	50
	IO	130
Reed-Solomon Encoder $(64, 56)$	LUT	90
	FDRE	20
Reed-Solomon Decoder $(64, 56)$	LUT	260
	FDRE	230

While our parallel Hamming code $5\times[15, 11]$ can be implemented using only 30 LUTs for the encoders and 50 LUTs for the decoders, Reed-Solomon code requires a higher number of blocks which are complex and uses more resources for the Reed-Solomon encoder/decoder up to $3\times$ parallel Hamming encoder and $5\times$parallel Hamming decoder. For more details, we have made the comparison between Hamming code and Reed-Solomon coding based on the aspects of memory occupation and running time in our work [13].

6. CONCLUSION

This paper deals with the design of an Error Correcting Code in a short-frame OFDM communication system. In order to respond favourably to the requirements of the low-powered sensor network, we have analysed the performance of several Error Correcting Codes (ECCs): such as Hamming code and Reed-Solomon code based on different parameters. Moreover, we have discussed the trade-off between the low implementation complexity (Hamming is the easiest to implement) and the high error correction capacity (Reed-Solomon being the most effective). Therefore, we have to: either improve the correction capacity of Hamming code, or decrease the complexity cost for Reed-Solomon code. That 's why, we propose a new design of parallel Hamming Coding. The parallel Hamming Code is chosen as five blocks of simple Hamming Code $[15, 11]$. Each encoder takes 11 bytes date block and generate 15 byte code block to be transmitted on the communication channel. After an implementation of this solution on an FPGA mock-up, we have shown that this parallel hamming encoder/decoder uses a few LUTs and has the capability of correcting up to five errors per message (packet with 55 bits). The encoder and decoder coding is done in VHDL on Xilinx tool. This process is implemented on Xilinx Spartan 7 FPGA.

Future work will include modelling the analog part of the PLC channel by the Matlab toolbox Simscape [16] instead of the AWGN channel estimation. Moreover, we will study also the possibility of decreasing the complexity cost [13] for the Reed-Solomon code by removing the multiplications on an FPGA processes and simplifying the complexity of the Reed-Solomon encoder/decoder design [17]. We will include then the implementation of the complete chain OFDM Tx-Rx on an FPGA.

REFERENCES

[1] S. Tilkioğlu, "OFDM Transmitter and Receiver Implementation on FPGA," in *Master Thesis dissertation*, Graduate School of Natural and Applied Sciences Electronic and Communication Engineering, 2018.

[2] Y. J. Qazi, J. A. Malik, and S. Muhammad, "Performance Evaluation of Error Correcting Techniques for OFDM Systems," in *Book*, 2014.

[3] A. Skylar *et al.*, "Optimization of Error Correcting Codes in FPGA Fabric Onboard Cube Satellites," in *Ph.D Thesis dissertation*, Montana State University-Bozeman Norm Asbjornson College of Engineering, 2019.

[4] A. M. Cruz *et al.*, "Low Complexity Turbo code Specification for Power-Line Communication (PLC)," in *2011 IEEE Electronics, Robotics and Automotive Mechanics Conference*. IEEE, 2011, pp. 349–354.

[5] L. M. Ionescu, C. Anton *et al.*, "Hardware implementation of BCH Error-correcting codes on a FPGA," in *International Journal of Intelligent Computing Research (IJICR)*, vol. 1, no. 3, pp. 148–153, 2010.

[6] S. Mahajan *et al.*, "BER Performance of Reed-Solomon Code Using M-ary FSK Modulation in AWGN Channel," in *International Journal of Advances in Science and Technology*, vol. 3, no. 1, pp. 7–15, 2011.

[7] V. Kavinilavu, S. Salivahanan *et al.*, "Implementation of Convolutional Encoder and Viterbi Decoder using Verilog HDL," in *International Conference on Electronics Computer Technology*, vol. 1. IEEE, 2011, pp. 297–300.

[8] Z. He, S. Roy *et al.*, "FPGA implementation of LDPC decoders based on joint row-column decoding algorithm," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2007, pp. 1653–1656.

[9] C. H. Jones, "Communications over Aircraft Power Lines," in *IEEE International Symposium on Power Line Communications and Its Applications*. IEEE, 2006, pp. 149–154.

[10] V. Degardin, E. Simon *et al.*, "On the Possibility of Using PLC in Aircraft," in *IEEE International Symposium on Power Line Communications and Its Applications (ISPLC)*. IEEE, 2010, pp. 337–340.

[11] M. Shirvanimoghaddam *et al.*, "Short Block-length Codes for Ultra-Reliable Low Latency Communications," in *IEEE Communications Magazine*, vol. 57, no. 2, pp. 1–8, 2019.

[12] Z. Gao, P. Reviriego, *et al.*, "Fault tolerant parallel filters based on error correction codes," in *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 23, no. 2, pp. 384–387, 2014.

[13] J. Fernandez-Mayoralas, R. Masmoudi Ghodhbane, "Performance of Error Correcting Codes (ECCs) for Short-frame OFDM Sensors' Network," in *IoT & Sensor Networks Symposium of IEEE International Conference on Communications (ICC)*, 2021.

[14] James E. Gilley, "Bit-Error-Rate Simulation Using MatLab," in *Transcrypt International, Inc.*, August 19, 2003, pp. 1-7.

[15] T. Youssef and E. Abdelfattah, "Performance evaluation of different QAM techniques using Matlab/Simulink," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2013, pp. 1–5.

[16] P. Degauque, V. Degardin, *et al.* , "Power-line communication: Channel characterization and modeling for transportation systems," in *IEEE Vehicular Technology Magazine*, 2015, 10(2), pp.28–37.

[17] P. Parvathi and P. R. Prasad, "FPGA based design and implementation of Reed-Solomon Encoder & Decoder for error detection and correction," in *Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG)*. IEEE, 2015, pp. 261–266.

AUTHORS

Dr. Raouia Masmoudi Ghodhbane (BSc.'10, MSc.'11, PhD.'15) received a Ph.D thesis in telecommunications from the University of Paris Cergy CY on the topic "Home automation telecommunications efficient in terms of energy consumption" in 2015. Currently, Raouia holds the position of research engineer in electronic interface and sensors' telemetry at the Safran Sensing Systems research department in Safran Tech. She has worked on different thematics: connected sensors in harsh aeronautic environments, low latency and low power sensor network communication, IoT networks, wireless networks, cognitive radio systems, opportunistic access to spectrum and green communications. Raouia is a member of the Technical Program Committees of the IEEE International Conference on Wireless for Space and Extreme Environments (WISEE).

Jorge Fernández-Mayoralas (MSc.'20) received an engineer degree in Integrated Circuit and Electronic Systems from CentraleSupelec in 2020. Since then, Jorge holds the position of Consultant on Energy, Utilities & Environment at Sia Partners. Jorge worked in his engineering intern-ship on the topic of "Integration of an error correcting code over a new aeronautic communication on an FPGA board" in Safran Tech.

RODENT: A FLEXIBLE TOPSIS BASED ROUTING PROTOCOL FOR MULTI-TECHNOLOGY DEVICES IN WIRELESS SENSOR NETWORKS

Brandon Foubert[1] and Nathalie Mitton[1]
[1]Inria, Lille, France

NOTE: Corresponding author: Brandon Foubert, brandon.foubert@inria.fr

Abstract – *Wireless Sensor Networks (WSN) are efficient tools for many use cases, such as environmental monitoring. However WSN deployment is sometimes limited by the characteristics of the Radio Access Technologies (RATs) they use. To overcome some of these limitations, we propose to leverage the use of a Multiple Technologies Network (MTN). What we refer to as MTN is a network composed of nodes which are able to use several RAT and communicating wirelessly through multi-hop paths. The management of the RAT and routes must be handled by the nodes themselves, in a local and distributed way, with a suitable communication protocol stack. Nodes may reach multiple neighbors over multiple RAT. Therefore, each stack's layer has to take the technologies' heterogeneity of the devices into account. In this article, we introduce our custom Routing Over Different Existing Network Technologies protocol (RODENT), designed for MTN. It enables dynamic (re)selection of the best route and RAT based on the data type and requirements that may evolve over time, potentially mixing each technology over a single path. RODENT relies on a multi-criteria route selection performed with a custom lightweight TOPSIS method. To assess RODENT's performances, we implemented a functional prototype on real WSN hardware, Pycom FiPy devices. Unlike related prototypes, ours has the advantage not to rely on specific infrastructure on the operator's side. Results show that RODENT enables energy savings, an increased coverage as well as multiple data requirements support.*

Keywords – heterogeneous, Pycom FiPy, routing, TOPSIS, WSN

1. INTRODUCTION

Wireless Sensor Networks (WSN) enable a remote monitoring of various metrics and many more use cases [1]. Such networks usually rely either on a medium distance Radio Access Technology (RAT) (*e.g.,* IEEE 802.15.4) and a multi-hop path routing or on a long distance RAT (*e.g.,* LoRaWAN) and a star topology. The latter simplifies the network structure and enables a wider coverage. When deployed, WSN usually use a single RAT shared by all nodes. Deployments are thus constrained by the limits of the chosen RAT, in terms of coverage and performance (throughput, energy consumption, costs, etc). For instance, the network of Sigfox, an operator-based RAT, provides long range communication (up to km) but is not available worldwide. Some RATs are even so constrained that they may not be able to comply with specific data requirements such as delay-intolerant data, high throughput or firmware over-the-air upgrade. Additionally, outdoor nodes have to bear the weather changes (*e.g.,* rain) which greatly impact the wireless links' quality.

Traditional WSN lack flexibility to support multiple use-cases. Many different RATs are available for WSN nowadays [2]. Different RATs come with different performances and capabilities. Multiple Technologies Networks (MTN) could overcome the aforementioned issues [3]. With several RATs built-in, the nodes' range of deployment would be extended, as nodes could switch from one RAT to another at each hop and relay data through multi-hop. An MTN's nodes would be able to select the best technology and route available. The choice would be based on the routes' availability and costs, in terms of energy, money, etc. If the environment changes, and the selected route's quality decreases, a node can dynamically select a better route and RAT. Nodes that support several data requirements (*e.g.,* temperature and video monitoring) can follow several paths accordingly. Network resiliency is increased, as in case of a RAT failure, a node can switch to an alternative technology.

Thus, nodes have to use specific methods to autonomously and dynamically choose which technology is the best suited depending on the data requirements and current context. This issue is known as Network Interface Selection (NIS). Several tools are available in the literature to tackle the NIS problem. Among them are the Multiple Attribute Decision-Making (MADM) methods. MADM methods provide a ranking of different alternatives based on their attributes and their associated weights. One of the most used and studied MADM methods is Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Said simply, TOPSIS compares candidates based on their mathematical distances to two ideal positive and negative alternatives.

However, TOPSIS suffers from an issue known as rank reversal. A rank reversal happens when the ranking is modified following the removing of one of the alternatives under study. This can alter the quality of the ranking and lead to a sub-optimal NIS. In our case, this could outcome in too many useless and costly technology switches. Moreover, considering hardware constrained WSN nodes, TOPSIS computation is resource-intensive. This would

decrease the devices' lifetimes and may overload the devices' limited memories which leads to hardware failure. We address those issues in this paper, by proposing a lightweight TOPSIS-based NIS method optimized for WSN devices. Furthermore, our method simplifies TOPSIS computations and completely eliminates rank reversal by modifying the TOPSIS normalization algorithm. This results in less complexity and provides time and energy savings.

Currently available routing protocols are not suited for MTN. In this article, we introduce a novel Routing Over Different Existing Network Technologies protocol (RODENT) designed for MTN leveraging our custom TOPSIS method. Our contribution takes every RAT of each node into account for the route selection. Every node has a list of available links between itself and its neighbors. Links have associated costs and performances, in terms of delay, energy consumption etc. A node constructs its routes based on its links' values and the routes' values shared by its neighbors. Criteria for the best route depend on the use case and the requirements data has to meet (*e.g.,* data size, deadline).

RODENT is implemented and its performances are assessed through experimental evaluation. Results show that RODENT increases network flexibility and reliability, decreases energy consumption and enables better consideration of the data requirements while maintaining a good Packet Delivery Ratio (PDR). Compared to related work, RODENT offers a flexible and dynamic way to overcome WSN's limitations without the need of a dedicated infrastructure other than multi-RAT nodes.

The contributions of this paper can be summarized as follows:

- We designed a lightweight selection method for WSN based on TOPSIS, free of rank reversal which shows an improvement in the computation time of around 38%, which in turn results in energy savings, while the technology selection is equivalent to using the classic TOPSIS method in 82% of the experiments.
- We designed a multi-technology routing protocol for WSN based on our custom selection method. It is capable of handling multi-technology devices and selecting the best route and technologies for specific data requirements.
- We designed and developed an MTN prototype composed of Pycom FiPy devices running a custom implementation of RODENT.

The rest of this paper is organized as follows: Section 2 introduces the work related to MTN and TOPSIS. Section 3 presents the background about MADM and TOPSIS. Section 4 explains what issues have to be faced with TOPSIS. Section 5 details our lightweight TOPSIS method. Section 6 presents our experiments on the selection method and the results we have obtained. Section 7 exposes the network model and assumptions RODENT is based on. Section 8 details RODENT's inner workings. Section 9 presents the hardware used and firmware implemented for our MTN prototype. Section 10 introduces the experimental setup and scenario. Section 11 details the experiments' results. Section 12 concludes this article and lists future work.

2. RELATED WORK

Several works have been conducted to mitigate rank reversal in TOPSIS or to apply TOPSIS to NIS. To the best of our knowledge, only few works exist in the literature about multi-technology network. This section presents related work about TOPSIS and multi-RAT devices.

2.1 TOPSIS method

[4] proposes an iterative TOPSIS method, where TOPSIS is executed, then the worst alternative is removed from the ranking, and TOPSIS is re-executed, as long as there is more than one alternative in the ranking. The remaining one is selected as a communication technology. [5] combines TOPSIS with fuzzy logic, in order to improve how uncertain attributes are taken into account. [6] introduces alternative methods based on TOPSIS, but with different normalization algorithms using maximum and minimum values of the attributes. [7] compares several NIS methods applied to heterogeneous WSN. [8] introduces an original MADM method along with an in-depth analysis of TOPSIS. [9] proposes a new Service-based Interface Selection Scheme algorithm based on TOPSIS to enable NIS applied to vehicle-to-vehicle communications scenarios. [10] details a fast TOPSIS-based NIS technique for vertical handover in heterogeneous emergency communication systems.

Overall, those propositions reduce the probability of occurrence of rank reversal, but does not nullify it because the euclidean normalization is still used. Furthermore, some of the proposed modifications tend to increase the complexity of the TOPSIS method. This would increase the execution time of TOPSIS and in turn the energy consumption of the nodes, thus reducing their lifetime.

To the extent of the authors knowledge, no works has been conducted to propose a rank reversal free TOPSIS-based method for NIS specifically for energy constrained devices. Thus, in this in paper we introduce a lightweight TOPSIS-based NIS method that aims not only to reduce the complexity and energy consumption of TOPSIS, but also to completely eliminate rank reversal.

2.2 Multi-technology networks

The authors of [11] propose an IoT architecture for multi-RAT devices. This architecture is based on a network convergence layer in charge of the multi-RAT management in nodes, and a heterogeneous network controller on the network operator side. It also proposes a hardware platform for the nodes, a polling scheme as well as a compression scheme based on Static Context Header Compres-

Table 1 – MADM decision matrix.

	P_1 w_1	P_2 w_2	...	P_m w_m
A_1	x_{11}	x_{12}	...	x_{1m}
A_2	x_{21}	x_{22}	...	x_{2m}
...	...	...	...	...
A_n	x_{n1}	x_{n2}	...	x_{nm}

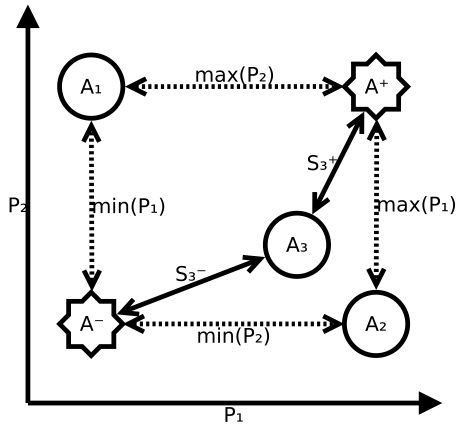


Fig. 1 – Representation of TOPSIS with three alternatives and two attributes.

sion (SCHC). This increases network flexibility, however it requires a specific virtual network operator. While the precedent work focuses on the device side, in [12] the authors propose a cloud-based virtual network operator for multi-modal LPWA networks. This operator takes care of the configuration and management of heterogeneous LPWAN equipment. Again this requires specific infrastructure on the operator side.

In [13], a green path selection inter-MAC selection protocol is detailed. This protocol allows path selection at the MAC layer while focusing on energy consumption and radio frequency minimization. However, it does not give any information about the routing layer. The article [14] presents the ORCHESTRA framework which manages real-time inter-technology handovers. It is based on a virtual MAC layer which coordinates the different layers from different technology with a unique MAC address. This work also focus on the link layer and not on the routing layer.

The aforementioned works increase WSN's flexibility. However several limitations are still present, such as the need of a dedicated infrastructure. In this article we propose a routing protocol adapted to MTN, which greatly increase WSN's capabilities while requesting only multi-RAT nodes.

3. TECHNOLOGY SELECTION BACKGROUND

Multi-technology devices have to autonomously select the best communication technology based on many factors. In the literature, several tools are available to perform this Network Interface Selection (NIS): utility and cost functions, Markov chains, fuzzy logic, game theory, data mining, Dempster-Shafer theory, to name a few. Particularly, Multi-Attribute Decision Making (MADM) methods [15] are commonly used for NIS. MADM methods are interesting as they rank several alternatives, based on their attributes as well as the relative importance associated to those attributes.

The problem can be modelled with a decision matrix as shown in Table 1. It is composed of $A = \{A_i \mid i = 1, 2, ..., n\}$ the set of the alternatives, $P = \{P_j \mid j = 1, 2, ..., m\}$ the set of the attributes and $W = \{w_j \mid j = 1, 2, ..., m\}$ the set of the weights associated to each attribute. Applied to NIS, A is the set of technologies, P the set of attributes associated to those and W the data requirements. The MADM methods take as input a decision matrix and output a ranking of the alternatives. Several MADM methods exist, the most known being: Simple Additive Weighting (SAW), Weighting Product (WP), Analytical Hierarchy Process (AHP) and Gray Relational Analysis (GRA).

One of the most used and studied methods is Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [16]. TOPSIS ranks alternatives depending on their relative mathematical distance to the ideal solution. The TOPSIS method runs the following steps:

1. The values x_{ij} of each attribute from the decision matrix (cf. Table 1) are normalized according to Equation *(1)*.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \qquad (1)$$

2. The normalized values r_{ij} are weighted according to Equation *(2)*.

$$v_{ij} = w_j r_{ij}, \ \sum_{j=1}^{m} w_j = 1 \qquad (2)$$

3. The positive and negative ideal alternatives A^+ and A^- are constructed according to Equation *(3)*.

$$\begin{aligned} A^+ &= [v_1^+ ... v_m^+] \\ A^- &= [v_1^- ... v_m^-] \end{aligned} \qquad (3)$$

4. The attribute values of the ideal alternatives are determined according to Equation *(4)* for upward attributes (*e.g.* range) or Equation *(5)* for downward attributes (*e.g.* latency).

$$\begin{aligned} v_j^+ &= Argmax\{v_{ij}, i = 1, ..., n\} \\ v_j^- &= Argmin\{v_{ij}, i = 1, ..., n\} \end{aligned} \qquad (4)$$

$$\begin{aligned} v_j^+ &= Argmin\{v_{ij}, i = 1, ..., n\} \\ v_j^- &= Argmax\{v_{ij}, i = 1, ..., n\} \end{aligned} \qquad (5)$$

Table 2 – Simple decision matrix.

	P_1	P_2	P_3
A_1	1.024537	7.828443	8.650221
A_2	4.226149	0.09865402	4.673396
A_3	8.026353	5.455392	2.536936
A_4	1.700537	1.398855	0.7656412

5. The distances between each alternative and the positive and negative ideal alternatives A^+ and A^- are computed according to Equation *(6)*.

$$S_i^+ = \sqrt{\sum_{j=1}^{m}(v_j^+ - v_{ij})^2} \qquad S_i^- = \sqrt{\sum_{j=1}^{m}(v_j^- - v_{ij})^2} \tag{6}$$

6. Finally, the relative closeness to the ideal solution is computed for each alternative according to Equation *(7)* and a ranking is established based on those values.

$$C_{TOPSIS} = \frac{S_i^-}{S_i^- + S_i^+} \tag{7}$$

When using TOPSIS for NIS, the technology with the highest value of C_{TOPSIS} is selected. A graphical representation of the TOPSIS method with three alternatives and two attributes is depicted in Fig. 1.

4. TOPSIS PROBLEM STATEMENT

TOPSIS is particularly interesting, as it grades alternatives based not only on the closeness from the best alternative but also on the distance from the worst one. However, TOPSIS suffers from an issue known as rank reversal that can happen when a non-optimal alternative is removed from the ranking. This can alter the quality and pertinence of the ranking. Rank reversal is an issue common to several MADM methods. With an ideal method, the ranking of alternatives should not be altered when another alternative is removed. The cause of rank reversal is the normalization algorithm. Indeed, the TOPSIS normalization (a.k.a. euclidean normalization) computes the normalized values for an attribute based on the values of all the other alternatives for that same attribute. Thus if set A changes, the result of Equation *(1)* also changes, which may modify the final ranking.

To clarify rank reversal let us consider an example. Table 2 represents a simple decision matrix randomly filled. Running TOPSIS on it outputs a ranking order corresponding to $[A_1, A_3, A_2, A_4]$. If the alternative A_4 was to be removed from the ranking (*e.g.* because of a broken link for example), it is expected that the ranking of the remaining alternatives should not be altered and therefore should correspond to $[A_1, A_3, A_2]$. However, running TOPSIS on Table 2 after removal of alternative A_4 outputs a ranking corresponding to $[A_3, A_1, A_2]$. This corresponds to a rank reversal. Applied to NIS, it means that the loss of the wireless link of technology A_4 would change the selected technology from A_1 to A_3. This would cause a technology switch which will require energy and does not bring any overall improvement.

It is to be noted that rank reversal is not a theoretical issue for multi-technology WSN devices. Actually, the wireless technologies' links' quality depends on many factors such as atmospheric and environmental conditions, which vary heavily across the year. This may results in broken links, thus removing a technology from the set of alternatives and potentially resulting in rank reversal, as seen in the previous example. The frequency of such events is entirely dependent on external factors and cannot be anticipated, thus links' quality has to be considered in the NIS process. Rank reversal could lead to the selection of a sub-optimal technology, on top of spending energy for switching between technologies.

A second issue posed by TOPSIS-based NIS on constrained devices is the complex computations that are required. The TOPSIS method as seen in Section 3 is based on computations that use numerous operations and memory accesses. WSN devices are generally hardware constrained, energy-limited and a repetitive execution of the TOPSIS method will have a considerable impact on the energy consumption of nodes. As an example, the Pycom FiPy's CPU [17] holds two cores that can go up to 240 MHz. A classic laptop CPU, *e.g.,* the Intel® Core™ i7-8650U, holds four cores that can go up to 4.20 GHz.

5. LIGHTWEIGHT TOPSIS FOR WSN

As stated in Section 4, the rank reversal issue is due to TOPSIS' normalization which computes normalized values based on all the other alternatives' values. Moreover, this normalization method is rather complex, and may increase the energy consumption of nodes.

Thus, we propose to use a simplified normalization method, which will not cause rank reversal and simplify the computations. Rank reversal happens because other alternatives are taken into account when computing normalized values. Thus, our proposition is to compute those values without taking into account other alternatives' values. Therefore, we need a stable normalization referential to measure our values against. We know that multi-technology devices have a fixed set of technologies available. Those are not supposed to change after deployment, and they have fixed maximum and minimum capabilities. We propose to use those maximum and minimum bounds as referential for our normalization.

5.1 Algorithm

That simplification takes the form of Algorithm 1, which replaces Equation *(1)* in the steps of our lightweight TOPSIS. Each value x_{ij} is normalized by being divided with the upper or the lower bound of its attribute j. Upward attributes' values are divided by their upper bound, while downward attributes divide their lower bound. The set

Algorithm 1 Lightweight normalization

Require: x_{ij} the raw value of each attribute j for each candidate i
  for each attribute P_j **do**
    if P_j is an upward attribute **then**
      B_j^+ is the upper bound of P_j
      $r_{ij} = \frac{x_{ij}}{B_j^+}$
    else if P_j is a downward attribute **then**
      B_j^- is the lower bound of P_j
      $r_{ij} = \frac{B_j^-}{x_{ij}}$
    end if
  end for
  return r_{ij} the normalized value of x_{ij}

$B = \{B_j^+, B_j^- \mid j = 1, 2, ..., m\}$ is composed of the upper and lower bounds of each attribute j, such that $\forall\, x \in B,\ 0 < x < +\infty$. B is stable, thus normalized values from the alternatives will not be altered by the removing of any other alternative. This completely eliminates rank reversal and reduces algorithmic complexity at once. Indeed, Equation *(1)* requires the computation of the denominator $\sqrt{\sum_{i=1}^{n} x_{ij}^2}$ for each value of j (for m attributes). This is not required with our bounded normalization and only the division between the bound and the value is computed. Knowing the fixed bounds allows us to simplify TOPSIS further: Equation *(3)* is used to establish the ideal positive and negative alternatives. Extreme values are found according to Equation *(4)* for upward attributes or Equation *(5)* for downward ones. Those operations require many comparisons. With bounded normalization, we can simplify the determination of the ideal alternatives: determination of A^+ and A^- is trivial, as the normalized maximum and minimum bounds of the attributes are respectively equal to 1 and 0. Thus, Equation *(4)* and Equation *(5)* can be simplified by Equation *(9)*. In turn, determination of the ideal alternatives according Equation *(3)* shows that these are static and shown in Equation *(8)*. Finally, distances computation according to Equation *(6)* can be simplified by Equation *(10)*. Indeed, as the ideal alternatives are known and static, we thus know that $v_j^+ = 1$ and $v_j^- = 0$.

Those simplifications reduce the complexity of the TOPSIS method. Moreover, as the normalization uses a stable referential, rank reversal probability is eliminated. Those modifications thus reduce the time required for execution, as we will see in Section 10.

$$\begin{aligned} A^+ &= [1...1] \\ A^- &= [0...0] \end{aligned} \tag{8}$$

$$\begin{aligned} v_j^+ &= 1 \\ v_j^- &= 0 \end{aligned} \tag{9}$$

Fig. 2 – FiPy board from Pycom [18].

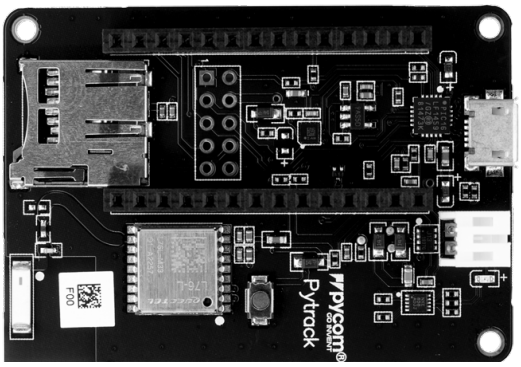


Fig. 3 – Pytrack sensor shield from Pycom [18].

$$\begin{aligned} S_i^+ &= \sqrt{\sum_{j=1}^{m} (1 - v_{ij})^2} \\ S_i^- &= \sqrt{\sum_{j=1}^{m} v_{ij}^2} \end{aligned} \tag{10}$$

5.2 Complexity

The reduced complexity of our algorithm can be assessed with an algorithmic complexity comparison. As big O notation is only pertinent for large inputs, we choose to quantify the number of operations spared with our method instead of classic TOPSIS. We consider one operation as one of the four basic arithmetic operations: addition, subtraction, multiplication and division. We also consider square root and value comparison as a single operation. This is just an estimation and is not exact as a square root is decomposed into multiple simpler operations when computed. However, as the exact decomposition is dependent on the hardware, it is irrelevant to assign a precise operation cost to a square root. Hereafter we consider n and m to be the dimensions of the decision matrix.

Firstly, Equation *(1)* requires at least $3nm$ operations, while using Algorithm 1 instead reduces it to nm operations. Replacing equations *(3)*, *(4)* and *(5)* by Equations *(8)* and *(9)* spares the cost of the min-max algorithm, thus $2(mn - 1)$ operations. Finally, using Equation *(10)* instead of Equation *(6)* spares nm operations. Our proposition thus spares a total of $5mn - 2$ operations.

6. SELECTION EXPERIMENTS & RESULTS

We implemented both algorithms in MicroPython on FiPy modules from Pycom, coupled with Pytrack expansion

Table 3 – Attributes' weights.

	Energy	Delay	Cost
$W_{monitoring}$	0.6	0.1	0.3
W_{alarm}	0.1	0.8	0.1

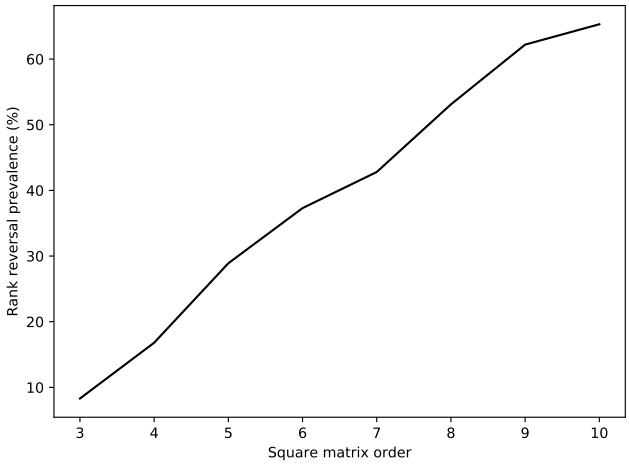


Fig. 4 – Rank reversal prevalence as a function of the decision matrix' size.

boards. Both are depicted in Fig. 2 and Fig. 3. Those devices offer five different wireless communication technologies, and provide hardware close to the one used in WSN. The available technologies on FiPy platform are WiFi, LoRa, Sigfox, LTE-M, NB-IoT and Bluetooth Low Energy. Each one comes with different performances, based on different metrics such as: energy consumption, economical cost, throughput, delay, loss rate, etc. Attributes of each technology are used to fill the decision matrix values x_{ij} used as input for the NIS algorithms. Weights associated to attributes are determined based on the data requirements. Table 3 shows an example set of weights that could be used: for regular monitoring data the weight and thus importance of the energy consumption will be higher. This would probably lead to an NIS of the best energy-efficient technology (*e.g.,* Sigfox). On the contrary, for an alarm the weight of delay will be higher, leading to an NIS of the fastest technology (*e.g.,* WiFi).

6.1 Rank reversal prevalence

We wanted to know how painful can be a rank reversal using TOPSIS for NIS. We ran experiments to quantify the prevalence of rank reversal using TOPSIS. The nodes execute the following steps: *i)* create a random matrix, *ii)* run TOPSIS on it and *iii)* compute the resulting ranking. Then we randomly remove one of the potential alternatives and the new ranking was computed. TOPSIS was run again on the matrix without the alternative removed from the ranking, and the resulting ranking was compared with the previous ranking. If the order of remaining options was different, then a rank reversal happened.
Results are highly dependent on the size of the matrices. Generally, the bigger the decision matrix, the more rank reversals as we can see in Fig. 4. Large matrices are not

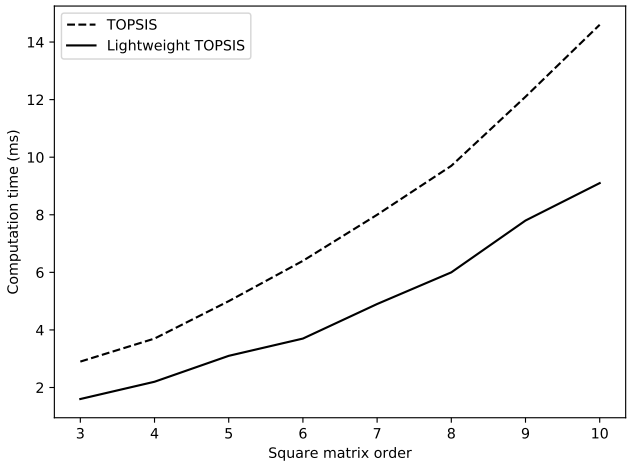


Fig. 5 – Classic and lightweight TOPSIS run times.

a current realistic representation of NIS in WSN. Multi-technology WSN nodes have several technologies available, but it is very unlikely that plain nodes carry hundreds of technologies. Similarly, technologies can have tens of attributes compared, but it is unlikely to be hundreds. Nonetheless, later, hardware will integrate more and more computing resources and communication technologies so our proposition will be able to scale with them. Still, we can see that even with small (5×5) matrices as we can obtain with FiPy modules, rank reversal happens approximately in 30% of the experiments. Rank reversal may cause useless technology switches, that are costly energy-wise. Larger matrices imply more frequent rank reversal, which emphasizes the need for a solution as ours. This is considerable if we assume TOPSIS to be run periodically to select the best technology after attributes or data requirements change.

6.2 Computation time

We compare the performance of a classic TOPSIS with our lightweight TOPSIS. We measure the time needed for the algorithms completion with the Timer library available for the FiPy as well as the similarity between the resulting NIS. It is worth noting that TOPSIS does not embed an objective comparison referential to estimate the quality of a ranking. However, TOPSIS is considered to produce a good quality ranking and is thus commonly used as a point of comparison. The obtained results are visible in Fig. 5.
We obtain a mean speed up of the computing time of 38%. At the same time, we still maintain a similarity with TOPSIS ranking in 82% of the experiments. Note that the ranking in the remaining 18% of the experiments cannot be qualified as worse for all cases since it mainly depends on the application and of what is expected or required. The ranking is only different from TOPSIS' ranking, which we used as a reference, but is not a ground truth. If we look at what we obtained when using a (5×5) matrix for a population of 7000 experiments with the results rounded to two decimal places, the mean execution time of the

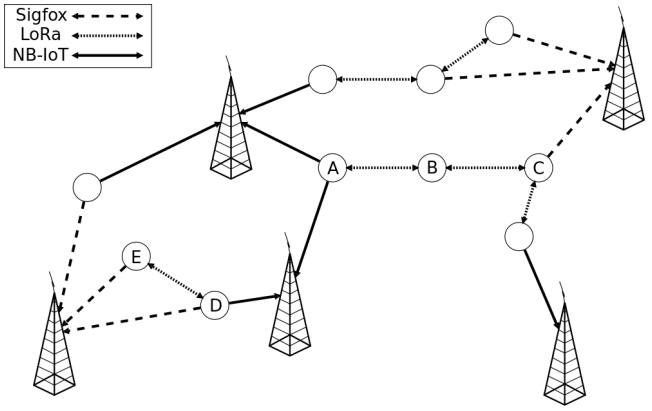


Fig. 6 – MTN example.

Table 4 – Example link matrix LM_D.

	Energy	*Money*	*Bit rate*
Sigfox BS	12	102	22
NB-IoT BS	151	87	174
Node E (LoRa)	37	0	72

classic TOPSIS is 4.79 ms, while the mean execution time of our lightweight TOPSIS is 2.96 ms. This means that a node could benefit from a mean time of 1.83 ms longer sleep periods between two TOPSIS executions. Based on the FiPy CPU data sheet [17], with a maximum CPU consumption of 68 mA and a power supply of 3.6 V, it would save up to approximately 448 µJ per TOPSIS run. Data sheets are notoriously optimistic, so in practice the energy savings could be even more significant. The standard deviation is of 0.05 ms, and the confidence intervals are $+/- 2.76 * 10^{-3}$ ms and $+/- 2.48 * 10^{-3}$ respectively for classic TOPSIS and for our lightweight TOPSIS, with a confidence level of 99.999%. Larger matrices offer similar results.

7. NETWORK MODEL & ASSUMPTIONS

We based the design of RODENT on a specific network model and assumptions made on the lower layers of the communication stack. In this section we describe this model and assumptions.

7.1 Network model

In WSN, the nodes usually follow one or multiple traffic patterns [19]. In this work, we assume that the nodes communicate in a convergecast pattern. Nodes forward packets exclusively to sink nodes. The nodes taking part in an MTN are heterogeneous in terms of RAT. We assume that the network is a connected graph where we consider every link from every node independently of their RAT *i.e.,* there can be several links between a single pair of nodes. Nodes can meet several data requirements (*e.g.,* monitoring, alarm, etc.), as long as those requirements are known by every node in the MTN. An MTN is depicted in Fig. 6. In this example, node B (N_B) measures temperature and is not in range of a Sigfox or NB-IoT base station. However,

Table 5 – Example route matrix RM_D.

	Energy	*Money*	*Bit rate*	*Hops*
Sigfox BS	12	102	22	1
NB-IoT BS	151	87	174	1
Node E (LoRa)	49	102	94	2

Table 6 – Requirements vectors.

	Energy	Money	Bit rate
$RV_{monitoring}$	0.6	0.3	0.1
RV_{alarm}	0.1	0.1	0.8

N_B can forward its data to N_A or N_C using LoRa. The latter can then offload N_B's data to a base station with a different RAT.

7.2 Data requirements

RODENT aims to support multiple use cases. Nodes can have multiple purposes (*e.g.,* monitoring temperature, video recording). The data requirements differ depending on the use case. For instance, for video data, we need a RAT with a high bit rate to ensure low delay and jitter. For an alarm, we need a very short delay but not necessary high bandwidth. For regular and small monitoring data, the focus is on saving the nodes' energy. A single node can have multiple data requirements *e.g.,* sending regular monitoring data of a rainfall and an alarm in case of a flood. Thus the route selection must satisfy as best as possible all nodes' data requirements.

7.3 Assumptions on communication stack

This article focuses on the network layer, specifically routing. We assume that the other communication stack's layers are comprised of protocols suited to MTN and that the physical and link layers are able to assess the availability and quality of links toward the nodes' neighbors *i.e.,* nodes or base stations. We assume that this process is possible for every RAT. We consider that those layers are able to gather or estimate information about the cost and performances of each link *i.e.,* energy cost, bit rate, etc. Radio link quality estimation in WSN is a well studied subject [20].

RODENT takes a link matrix as input, to which we refer to as LM_x for node x. LM_x's size depends on multiple factors: the number of characteristics, the number of RAT available, and the number of x's neighbors. For example, N_D in Fig. 6 could have a link matrix LM_D such as the one in Table 4. LM_D is comprised of every available link between N_D and its neighbors, and the characteristics of those links.

We refer to the route matrix of node x as RM_x. For route selection, RM_x is composed of all the routes available for node x. RM_x's attributes are relative to the routes *e.g.,* the number of hops, expected transmission count or the total energy consumption. For example, N_D in Fig. 6 could have a route matrix RM_D such as the one in Table 5. TOPSIS takes as input a set of weights for each attribute. The

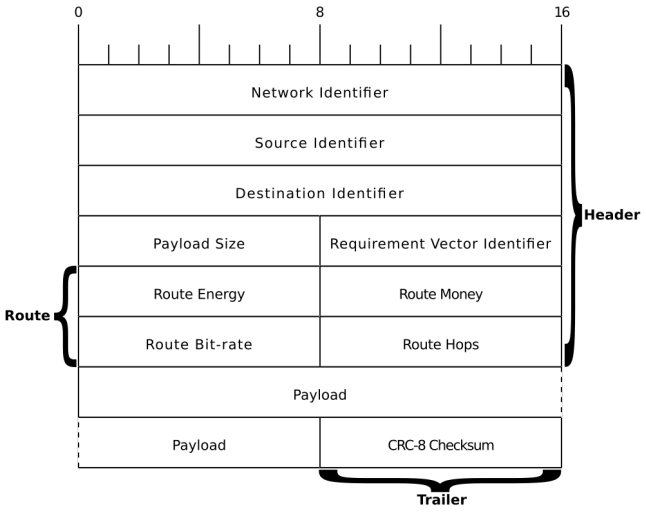


Fig. 7 – RODENT packet structure.

weights represent the importance of each attribute in the ranking process. We refer to a set of weights as a Requirements Vector (RV). RV_x is the requirements vector for use case x *e.g.,* $RV_{monitoring}$. For route selection *RV*'s values are set based on the data requirements that the node have to meet *e.g.,* prioritize speed over energy consumption, and such that $RV\,\{e_n \in RV \mid \sum_{n=1}^{|RV|} e_n = 1\}$. Example requirements vectors are depicted in Table 6.

8. ROUTING OPERATIONS

The distinctive feature of RODENT is to enable multi-RAT routes. Each route offers different cost and performances. In this section we detail RODENT's routing operations. The following notations are used further. Node i is referred to as N_i. Nodes that are in the vicinity of N_i are called neighbors. The set of N_i's neighbors is referred to as $NBR(i)$ and $NBR(i)_j$ is the node j such that $N_j \in NBR(i)$. A neighbor N_j of N_i has at least one link with N_i. For RAT x, such a link is referred to as L_{ij}^x. Consequently, the route from N_i that follows link L_{ij}^x is referred to as R_{ij}^x.

8.1 Overview

Let's consider the operations of N_D and N_E from Fig. 6 as an example. N_D boots without any knowledge of its surroundings. N_D's link layer scans the environment for every RAT and builds its link matrix LM_D as in Table 4. Based on LM_D, the network layer starts to build the route matrix RM_D. The direct links between N_D and the base stations are registered in RM_D as single-hop routes. N_E meanwhile does the same, and selects its only available route toward the Sigfox base station. N_E advertises its route which is received by N_D through their LoRa link L_{ED}^{LoRa}. N_D constructs its third route by adding the route's and link's costs. Here, we assume that the links' values between the Sigfox base station and N_D and N_E are similar. RM_D is then similar to Table 5. N_D then selects a best route for each of its *RV*. The selection is made independently of the RAT and based only on the routes' costs and performances. In this case and considering Table 6, the best route for $RV_{monitoring}$ is the one toward the Sigfox base station because low energy consumption is favored. The best route for RV_{alarm} is the one toward the NB-IoT base station because high bit rate is favored. N_D then starts to advertise and use its best routes.

8.2 Packet structure

RODENT packets' structure is depicted in Fig. 7. A packet is composed of three parts: *(i)* the header *(ii)* the payload *(iii)* the trailer. The header contains the required control data for RODENT. The *Network Identifier* is a two byte value shared by all nodes and is used to differentiate RODENT's communication. The *Source Identifier* is a two byte value corresponding to the packet's source node's unique ID. The *Destination Identifier* is a two byte value corresponding to the packet's destination node's unique ID. The *Payload Size* is a one byte value equal to the payload's size in bytes. The *Requirement Vector Identifier* is a one byte value which indicates the type (*i.e.,* use case) of the payload's data. The *Route* is a four byte array with $N_{Source\ Identifier}$'s best route's values *i.e.,* energy, money, bit rate and number of hops. The payload contains the data shared by the source. It is a series of n bytes with n equal to the header's *Payload Size* field. The trailer is a single byte carrying the *CRC8 Checksum* of the header and payload parts.

8.3 Route construction

Let's consider the operations of node i. N_i boots up and starts the construction of its route matrix RM_i. RODENT accesses two sets of data: the link matrix LM_i and the set of route shared by N_i's neighbors. N_i's first step is to check LM_i for any link toward a base station *e.g.,* a Sigfox antenna or a LoRaWAN gateway. Such links are turned to single hop routes based on the links values from LM_i. Routes are stored in RM_i. N_i's second step is to construct the routes passing through $NBR(i)$'s nodes. Let's consider the reception of a route from $NBR(i)_j$. N_i adds the received route's attributes to the attributes' values of the link L_{ij}^x. The resulting route R_{ij}^x is stored in RM_i.

8.4 Route selection

In classic WSN, route selection is trivial as the route with the lowest cost or rank is selected. In MTN, a route is a set of successive links, where each link may use a different RAT. Different RATs offer various performances and route selection in MTN has to take account of multiple criteria. We aim to support multiple use cases with different data requirements. Section 5 introduced RODENT's selection method. For node N_i, our lightweight TOPSIS takes as input the route matrix RM_i and a requirement vector RV_x relative to use case x. The selection outputs a ranking of the routes. The route coming out on top best fulfills the data requirements of use case x. For N_i, a best route BR_i^x

is selected for every use case x.

8.5 Route propagation

Route propagation occurs through two mechanisms: piggybacking and control packets. Piggybacking allows routes to be shared without dedicated transmissions. Considering a RODENT packet carrying a data payload of use case x, the header contains RV_x's id number and the best route BR_i^x. Wireless communications share a common medium. Thus, N_i overhears every packet from $NBR(i)$, which allows N_i to update RM_i opportunistically. If N_i stops overhearing route R_{ij}^x from its neighbor $NBR(i)_j$ *e.g.,* because N_j is down, R_{ij}^x will time out and will be removed from RM_i. To keep alive unused routes, $NBR(i)_j$ will send dedicated control packets. Control packets are regular packets with an empty payload.

9. PROTOCOL IMPLEMENTATION

Our implementation of RODENT is performed on Pycom FiPy devices [21]. The specificity of FiPy devices is that they offer five different RATs. These nodes take part in the MTN and offload data to WiFi and LoRa base stations (BS). The hardware and firmware used are detailed in this section.

9.1 Hardware

Pycom FiPy nodes are composed of WSN hardware: wireless RAT, ESP32 CPU, little memory available which allows ultra–low power usage. The available RATs are WiFi, LoRa, Sigfox, LTE-M, NB-IoT and Bluetooth Low Energy (BLE). Each RAT comes with different performances in terms of energy consumption, economical cost, bit rate, etc. RODENT performs route selection based on these characteristics. FiPy nodes are coupled with Pytrack sensor shields which provide an accelerometer, a GPS and a micro-USB port.
The LoRa BS is a B-L072Z-LRWAN1 board [22]. The WiFi BS is an Edimax EW-7811Un dongle [23] connected to the main computer. A Trip Lite U223-007 (7-Port USB Hub) is used to connect every device. The main computer is a Dell Latitude 5590. It powers devices, collects and analyses results.

9.2 Firmware

A port of MicroPython available as firmware for the FiPy allowed us to implement RODENT in Python. Upon boot, a node N_i computes its unique ID. Based on LM_i it boots up the needed RAT and constructs routes. The node is then locked up in the main loop: *i)* select best route for each RV_x, *ii)* add next payload to transmission buffer *iii)* send every payload in buffer. Neighbor's routes are added in RM_i upon reception. Neighbor's payloads are appended in the transmission buffer. Nodes print on the serial port the characteristics of the packets sent. Upon the Pytrack's button press, nodes switch between the two *RV* implemented: $RV_{monitoring}$ and RV_{alarm}.
The LoRa BS's firmware is implemented in C. It listens constantly for LoRa transmissions. Upon reception of a RODENT packet, it is unpacked and its characteristics are printed on the serial port. The WiFi BS is coded in Python. It listens for RODENT WiFi transmissions, unpacks them and prints characteristics on stdout.

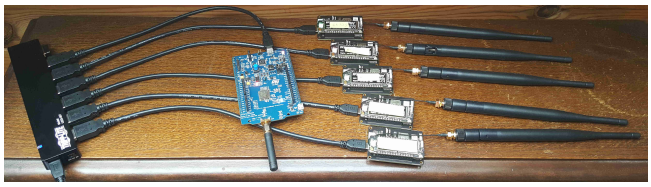


Fig. 8 – Experimental setup.

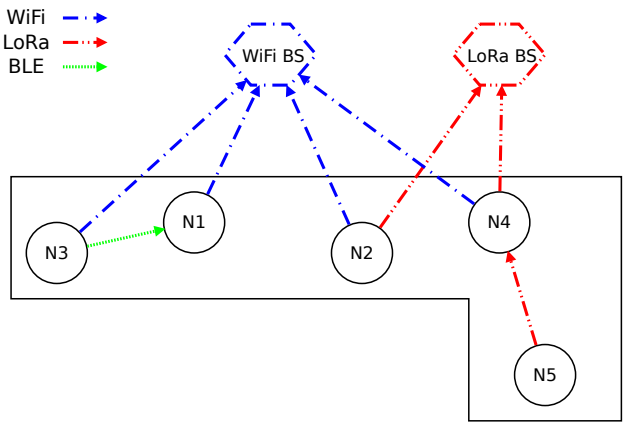


Fig. 9 – Farm monitoring scenario.

10. RODENT EXPERIMENTS

To assess the performances of RODENT, we run experiments on real hardware. We configured the nodes to follow a specific scenario and measured the results. The experimental setup and scenario are presented in this section.

10.1 Setup

The aforementioned devices in Section 9 are connected to the main computer through the USB hub. Every node and BS are powered at the same time and boot up immediately. As we can see in Fig. 8, every device is laying very close to each other. The main computer reads the stdout of the WiFi BS and the serial ports of the nodes and LoRa BS. Results are then computed offline, post-experiment.

10.2 Scenario

We simulate a farm monitoring use case. Smart agriculture can help farmers in their everyday life, but farms are often an unfriendly environment for wireless sensors (large rural areas, tall crops...). MTN eases the technical difficulties by offering nodes multiple possibilities of communication (operator based networks, personal net-

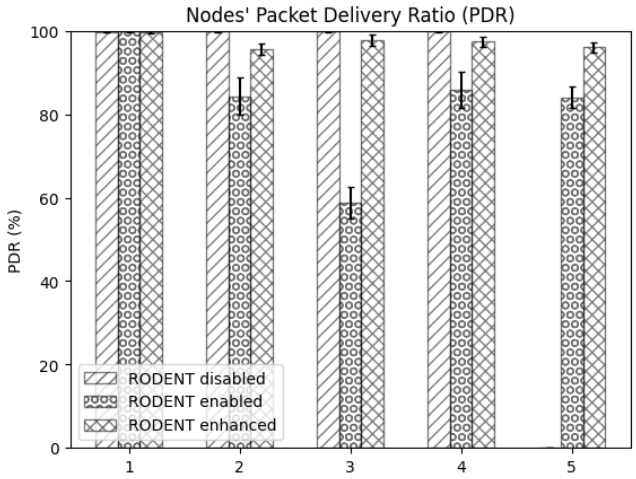


Fig. 10 – Packet Delivery Ratio per node.

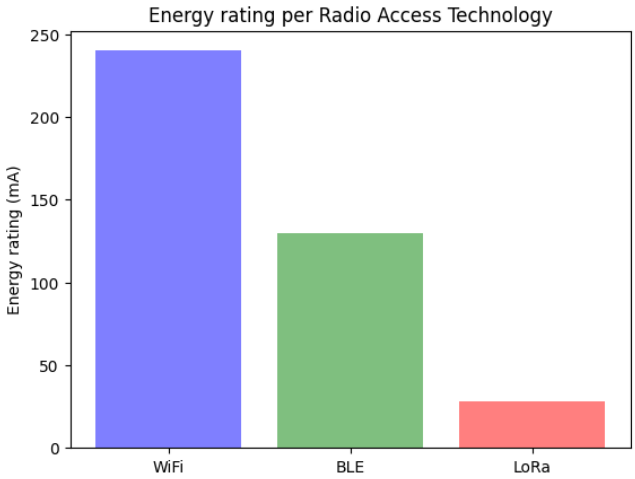


Fig. 11 – Energy consumption per RAT.

works, multi-hop networks...). In our scenario, nodes are deployed throughout a field used for cultivating crops. The simulated setup is illustrated in Fig. 9. Five nodes monitor environmental metrics useful for farmers. Nodes have to offload numerical data on a regular basis while saving up power. They may have to send an alarm if a metric becomes off chart, putting the crops at risk (*e.g.,* temperature).

Out of the five RATs available on FiPy, we are using WiFi, LoRa and BLE in this scenario. Sigfox and LTE-M/NB-IoT are not open technologies, so we could not use them directly. LoRa and BLE links are more interesting in terms of energetic savings than WiFi. Each node (Nx) is in a different situation. $N1$ is the control node, it only has a WiFi link with the WiFi BS. $N2$ can reach the WiFi BS and benefits from the LoRa link when RODENT is active. $N3$ has to choose between reaching the WiFi BS directly at a high energy cost or forwarding its data to its neighbor $N1$ via a BLE link. $N4$ needs to be able to send regular monitoring data as well as alarms, via WiFi or LoRa. $N5$ is an isolated node, deployed too far away to directly communicate with the WiFi BS. Farms are usually located in wide rural environments, unfriendly to wireless waves because of tall crops (*e.g.,* corn). Thus white zones and isolated nodes are common. Using RODENT, $N5$ can forward its data to its neighbor $N4$ using LoRa.

We run three types of experiments. First, RODENT is not active and nodes only use WiFi links, depicted in blue in Fig. 9. Second, RODENT is active, which allows nodes to switch to LoRa and BLE links, depicted in red and green in Fig. 9. Third, RODENT is active, each LoRa message is sent two times and each BLE message is sent three times, which increases the network's reliability. A video of an experiment running is available online[1].

[1] `http://chercheurs.lille.inria.fr/bfoubert/ressources/rodent.mp4`

11. EXPERIMENTAL RESULTS

Topology and Packet Delivery Ratio (PDR) are measured. Nodes transmit at an interval randomly picked in $[2; 4]$ seconds. We consider a population of 20 experiments lasting 10 minutes each. A small population is sufficient because of the low standard deviation. Longer experiments are not relevant because the network stabilizes after a few messages have been exchanged. We do not directly compare RODENT's results to related works as the heavy difference between proposals makes it irrelevant, and increased flexibility cannot be measured. In this section we present the results obtained.

11.1 Topology

With the use of RODENT, the MTN's topology changes. $N1$ does not change its link because it can only reach the WiFi BS. $N2$ uses the LoRa link instead of the WiFi link, because it costs less energy. $N3$ decides to use the BLE link to offload its data to $N1$, which in turn forwards it to the WiFi BS. $N4$ offloads its monitoring data to the LoRa BS, to reduce energy consumption compared to WiFi. It can still use the WiFi link to forward alarms that needs a quicker RAT at the expense of a higher energy cost. $N5$ is not isolated anymore, as it forwards its data to $N4$ through LoRa which will offload it to the LoRa BS in turn.

11.2 Packet Delivery Ratio

The Packet Delivery Ratio (PDR) is the ratio between the total packets received and the total packets sent. The PDR of every node taking part in the MTN is depicted in Fig. 10 along with its standard deviation. $N1$'s PDR does not change, as its route remains the same. Without RODENT, $N5$'s PDR is null as the node is isolated and cannot offload a single data packet. The PDR of $N2$, $N4$ and $N5$ is around 80% with RODENT which allows them to use LoRa. It is not the same as WiFi because of frequent collisions, as nodes do not use a proper MAC. $N3$'s PDR is around 60% with RODENT. The node forwards its data

through BLE to $N1$. We achieved BLE raw transmissions through the use of single BLE advertisements, hence the packet losses. With the enhanced RODENT, we can see a better PDR for all nodes, close to the one obtained with only WiFi.

11.3 Energy consumption

Physical measurement of the Pycom FiPy's energy consumption is hazardous since it suffers from design problems which lead to erroneous measurements [24]. We choose to stick to the energy ratings given in the components data sheets [17, 21] to get a general idea; these are shown in Fig. 11. Compared to WiFi, BLE needs approximately half-less current and LoRa a tenth. With the Pycom FiPy's CPU, WiFi and BLE offers the same bit rate. LoRa's bit rate is much slower leading to longer transmission for the same amount of data. WiFi and BLE require a heavier traffic control than LoRa does, which allows LoRa to consume less energy. Thus, we can assume that RODENT enables significant energy savings.

12. CONCLUSION

WSN deployments are limited by the coverage and performances of the devices' RAT. The use of several RATs in an MTN allows these shortcomings to be overcome. In this article, we introduced the novel Routing Over Different Existing Network Technologies protocol (RODENT). RODENT is based on a lightweight TOPSIS method that reduces the complexity of the computations and eliminates rank reversal issues, lessening computation time of about 38%. The resulting network interface selection is still similar to the one obtained using classic TOPSIS in 82% of the experiments.

RODENT is designed for routing in MTN and enables the use of multi-technology routes. We demonstrate the feasibility and utility of MTN with a prototype network based on a custom implementation of RODENT. Results show that RODENT increase flexibility, reliability, energy savings and maintains a good PDR.

For future work, we plan to precisely measure energy savings and extend RODENT to support downlink communications. This will further increase nodes' flexibility and possibilities (*e.g.,* firmware over the air upgrade). We plan to conceive an efficient link layer protocol for precise link costs and performance assessment for multi-RAT. In addition, we plan to combine RODENT with an efficient data reduction scheme to reduce even more energy consumption. We intend to run a larger simulation and experimentation, where end-to-end delay will be measured to assure that RODENT's usage does not slow down the network, and we will show how this satisfies QoS in terms of delay.

ACKNOWLEDGMENTS

This work was partially supported by a grant from CPER DATA, Sencrop, FEDER, I-SITE and Lirima Agrinet project.

REFERENCES

[1] D. Ye, D. Gong, and W. Wang. Application of wireless sensor networks in environmental monitoring. In *International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, 2009.

[2] B. Foubert and N. Mitton. Long-range wireless radio technologies: a survey. *Future Internet*, 12(1), 2020.

[3] B. Foubert and N. Mitton. Autonomous collaborative wireless weather stations: a helping hand for farmers. *ERCIM News*, (119):37–38, 2019.

[4] F. Bari and V. C. M. Leung. Multi-attribute network selection by iterative topsis for heterogeneous wireless access. In *IEEE Consumer Communications and Networking Conference (CCNC)*, 2007.

[5] W. Zhang. Handover decision using fuzzy madm in heterogeneous networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2004.

[6] M. A. Senouci, M. S. Mushtaq, S. Hoceini, and A. Mellouk. Topsis-based dynamic approach for mobile network interface selection. *Computer Networks*, 107:304 – 314, 2016.

[7] E. Stevens-Navarro and V. W. S. Wong. Comparison between vertical handoff decision algorithms for heterogeneous wireless networks. In *IEEE Vehicular Technology Conference (VTC)*, 2006.

[8] F. Bari and V. C. M. Leung. Automated network selection in a heterogeneous wireless network environment. *IEEE Network*, 21(1):34–40, 2007.

[9] L. H. Teixeira and A. Huszak. Preemptive network selection for v2v communication. In *International Conference on Telecommunications and Signal Processing (TSP)*, 2019.

[10] I. Bisio and A. Sciarrone. Fast multiattribute network selection technique for vertical handover in heterogeneous emergency communication systems. *Wireless Communications and Mobile Computing*, 2019:1–17, 2019.

[11] J. Famaey, R. Berkvens, G. Ergeerts, E. D. Poorter, F. V. d. Abeele, T. Bolckmans, J. Hoebeke, and M. Weyn. Flexible multimodal sub-gigahertz communication for heterogeneous internet of things applications. *IEEE Communications Magazine*, 56(7):146–153, 2018.

[12] J. Hoebeke, J. Haxhibeqiri, B. Moons, M. Van Eeghem, J. Rossey, A. Karagaac, and J. Famaey. A cloud-based virtual network operator for managing multimodal lpwa networks and devices. In *Cloudification of the Internet of Things (CIoT)*, 2018.

[13] O. Bouchet, A. Kortebi, and M. Boucher. Inter-mac green path selection for heterogeneous networks. In *IEEE Globecom Workshops*, 2012.

[14] T. De Schepper, P. Bosch, E. Zeljković, F. Mahfoudhi, J. Haxhibeqiri, J. Hoebeke, J. Famaey, and S. Latré. Orchestra: enabling inter-technology network management in heterogeneous wireless networks. *IEEE Transactions on Network and Service Management*, 15(4):1733–1746, 2018.

[15] R. V. Rao. *Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods*, chapter Introduction to multiple attribute decision-making (madm) methods, pages 27–41. Springer, 2007.

[16] C. L. Hwang and K. Yoon. *Multiple attribute decision making: methods and applications*. New York: Springer-Verlag, 1981.

[17] Espressif Systems. *ESP32 series datasheet*, 2020. Version 3.3.

[18] Pycom website. `https://pycom.io/`. [Online; accessed 22-Jan-2021].

[19] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados. Energy-efficient routing protocols in wireless sensor networks: a survey. *IEEE Communications Surveys Tutorials*, 15(2), 2013.

[20] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Trans. Sen. Netw.*, 8(4), 2012.

[21] Pycom. *FiPy specsheet*, 2018. Version 1.0.

[22] STMicroelectronics. *UM2115 user manual*, 2018. Rev 5.

[23] Edimax. *EW-7811Un datasheet*, 2015.

[24] T. Andersén. Energy-efficient adaptive sensing in low power wide area networks. Master's thesis, Norwegian University of Science and Technology, 2018.

AUTHORS

Brandon Foubert received a Master's degree in Computer Networks and Embedded Systems from the University of Strasbourg in 2018. He investigated "Cooperation between multiple RPL networks" under the direction of Julien Montavont in the Network research group from the ICube laboratory. He is currently pursuing a PhD degree under the supervision of Nathalie Mitton in the FUN team at Inria Lille - Nord Europe since September 2018 and until September 2021. He is studying "Polymorphic wireless communication for connected agriculture".

Nathalie Mitton received the MSc and PhD. degrees in Computer Science from INSA Lyon in 2003 and 2006 respectively. She received her Habilitation à diriger des recherches (HDR) in 2011 from Université Lille 1. She is currently an Inria full researcher since 2006 and from 2012, she is the scientific head of the Inria FUN team . Her research interests focus on self-organization from PHY to routing for wireless constrained networks. She has published her research in more than 50 international journals and more than 120 international conferences. She is involved in the set up of the FIT IoT LAB platform (`http://fit-equipex.fr`, `https://www.iot-lab.info`), the H2020 CyberSANE and VESSEDIA projects and in several program and organization committees such as Infocom (since 2019), PerCom 2020&2019, DCOSS (since 2019), Adhocnow (since 2015), ICC (since 2015), Globecom (since 2017), Pe-Wasun 2017, VTC (since 2016), etc. She also supervises several PhD students and engineers.

A MULTI-LINK COMMUNICATION CONNECTIVITY GAME UNDER HOSTILE INTERFERENCE

Andrey Garnaev[1], Wade Trappe[1], Narayan B. Mandayam[2], H.Vincent Poor[3]
[1]WINLAB, Rutgers University, North Brunswick, NJ, USA (e-mail: andrey.garmaev@gmail.com, trappe@winlab.rutgers.edu), [2]Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA, (e-mail: narayan@winlab.rutgers.edu), [3]Department of Electrical Engineering, Princeton University, Princeton, NJ USA (e-mail: poor@princeton.edu)

NOTE: Corresponding author: A. Garnaev, e-mail: andrey.garmaev@gmail.com

Abstract *– In this paper, we consider a communication connectivity problem involving a primary user (transmitter, for example, a Ground Control Station (GCS)) servicing a group of secondary users (receivers, for example, drones) under hostile interference. We formulate this multi-link communication connectivity problem, where the channels are affected by Rayleigh fading, as a zero-sum power resource allocation game between a transmitter and an adversary (jammer). The transmitter's objective is to maximize the probability of communication connectivity with all the receivers. It is proven that the problem has unique equilibrium in power allocation strategies, and the equilibrium is derived in closed form. Moreover, we reduce the problem of designing the equilibrium in power resource allocation strategies to the problem of finding a fixed point of a real-valued function. An algorithm based on the bisection method to find the fixed point (and so equilibrium strategies) is developed, and its convergence is proven.*

Keywords – communication connectivity, interference, jamming, Nash equilibrium, resource allocation

1. INTRODUCTION

Communication between a transmitter and a receiver under hostile interference is a well-studied problem in the wireless literature (see, for example, the survey in [1]). Such problems are multi-objective problems since they deal with different agents (say, a transmitter and an adversary (jammer)), and each of these agents has its own objective. Game theory supplies concepts for analyzing and solving such multi-objective problems [2], and, thus, game theory has been widely used to model jamming problems. Typically, jamming problems can be categorized according to two frameworks: (i) maintaining communication reliability and (ii) maintaining communication connectivity. In communication reliability problems, the transmitter's payoff is a function of throughput or Signal-to-Interference-plus-Noise Ratio (SINR) at the receiver, and the transmitter intends to maximize such payoff [3, 4, 5, 6, 7, 8, 9, 10, 11]. Meanwhile, in communication connectivity problems the transmitter must keep its SINR greater than or equal to a threshold value to ensure a connection can be sustained [12, 13, 14, 15, 16, 17, 18, 19]. In this paper, we consider the communication connectivity problem under hostile interference involving a transmitter servicing a group of secondary users (receivers) with the channels affected by Rayleigh fading. We formulate and solve this *Multi-Link Communication Connectivity* (MLCC) problem as a zero-sum power allocation problem between the transmitter and the jammer, where the transmitter wants to maximize the *Probability of Communication Connectivity* (PCC) with all the receivers.

One of the core difficulties in communication connectivity problems is that the transmitter has to keep its SINR of the transmitted signal at the receiver greater or equal to a threshold value, which we refer to as the *Threshold of Communication Connectivity* (TCC). This leads generally to to the non-existence of equilibria in power-level assignment strategies, i.e., pure strategies, even in a *Single-Link Communication Connectivity* (SLCC) problem [14, 17], and it may cause destabilization of communication. One of the ways to stabilize such systems, i.e., to make them have equilibrium, is to extend the set of feasible strategies to mixed strategies (in other words, to assign the equilibrium strategy via randomization (lottery) over pure strategies). Although using a lottery introduces a factor of uncertainty for the decision maker it allows it to find an equilibrium in such mixed strategies for the MLCC problem via the Colonel Blotto game approach [12, 13, 16, 18, 19, 17] and for the SLCC problem via a war of attrition game approach [17].

To avoid introducing a random factor in decision making, in [14, 15], another approach was suggested to stabilize such communication. In [14, 15], it was proven that if channels are affected by Rayleigh fading and background noise at the receiver is negligible, then an equilibrium in power-level assignment strategies (i.e., pure strategies) exists and is unique for an SLCC problem.

The goal of this paper is to prove that if channels are affected by Rayleigh fading and *any* background noise, a random factor can be eliminated in decision making for communication stabilization even in MLCC systems. To the best of the authors' knowledge this problem has not been considered before.

The main contributions of this paper are as follows:

(1) A problem of multi-link communication connectivity

under jamming of a transmitter with a group of receivers when the channels are affected by Rayleigh fading is modeled as a zero-sum power power allocation game.

(2) Existence and uniqueness of the equilibrium in power allocation strategies are proven. Thus, in contrast to Colonel Blotto games, if channels are affected by Rayleigh fading, then stability of communication connectivity in a multi-link system can be maintained without introducing a random factor for a decision maker.

(3) We reduce the problem of finding the equilibrium in power resource allocation strategies to the problem of solving a fixed point equation in a scalar variable. An algorithm based on the bisection method to find the fixed point (and so equilibrium strategies) is developed, and its convergence is proven. This algorithm can be considered to be a *learning* algorithm since it allows one to reduce the zone of uncertainty for the equilibrium by a factor of two on iteration.

The organization of this paper is as follows. In Section 2, a short summary of an SLCC model is given. In Section 3, its generalization for the MLCC scenario is suggested as a zero-sum game between the transmitter communicating with a group of receivers and the jammer. In Section 4, the equilibrium strategies are designed and the uniqueness of the equilibrium is proven. In Section 5, the equilibrium strategies are established in closed form for the boundary cases of network parameters: (a) for small or large TCC, (b) for small or large total jamming and transmission power budgets and (c) for small background noise at the receivers. In Section 6, an algorithm to design the equilibrium in the general case is presented and its convergence is proven. Finally, in Section 7, illustrations of the results are given, and, in Section 8, conclusions are offered. All proofs are provided in the appendix.

2. SHORT OVERVIEW OF A SINGLE-LINK COMMUNICATION CONNECTIVITY MODEL

In this section following [14, 15] we give a short overview of the SLCC model with one transmitter communicating directly to a single receiver. This communication is affected by hostile interference from a jammer. Let G and H be the channel power gains from the transmitter to the receiver and the jammer to the receiver, respectively. In practice, the channel power fading gains depend on distances, fading and antenna characteristics. In general, the channel power gains are random variables (e.g., representing channel fading) with means $\boldsymbol{E}[G] = g$ and $\boldsymbol{E}[H] = h$. Let p and q be the power levels used by the transmitter and the jammer, respectively. Thus, $\mathbb{R}_+$ is the set of feasible strategies for the transmitter as well as for the jammer. The receiver also is affected by noise power N. Thus, the SINR at the receiver is given by

$$\mathrm{SINR}(p,q) = \frac{Hp}{N + Gq}. \tag{1}$$

We say that the communication from the transmitter to the receiver is maintained if and only if the SINR at the receiver is greater than or equal to a given TCC ϵ, i.e., the following condition holds:

$$\mathrm{SINR}(p,q) \geq \epsilon. \tag{2}$$

This TCC ϵ depends on the system's requirements, such as bit rate and bit error rate (BER).
Then, since G and H are random variables, the probability that the link between the transmitter and the receiver is maintained, i.e., PCC, is given by

$$\boldsymbol{P}(p,q) = \boldsymbol{P}\left(\mathrm{SINR}(p,q) \geq \epsilon\right). \tag{3}$$

For the case when the channels are affected by Rayleigh fading (i.e., G and H are exponential random variables with means $\boldsymbol{E}[G] = g$ and $\boldsymbol{E}[H] = h$, respectively), by [14, 15], the probability *(3)* can be represented as follows:

$$\boldsymbol{P}(p,q) = \frac{e^{-\epsilon\frac{N}{hp}}}{1 + \epsilon\frac{gq}{hp}}. \tag{4}$$

Note that $\boldsymbol{P}(p,q)$ is continuous in $p \geq 0$ and $q \geq 0$ and $\boldsymbol{P}(0,q) = 0$ for $q \geq 0$.

3. A MULTI-LINK COMMUNICATION CONNECTIVITY MODEL

In this section, we generalize the SLCC problem to the MLCC problem involving a transmitter with n receivers as follows:

- We assume that the transmitter is equipped with n directed antennas to communicate with n receivers.
- Let $p = (p_1, \ldots, p_n)$ be the strategy of the transmitter where p_i is the power assigned to communicate with the receiver i and

$$\sum_{i \in \mathcal{N}} p_i = \overline{P} \text{ and } p_i \geq 0, i \in \mathcal{N} \tag{5}$$

  with $\overline{P}$ is the transmitter's total power budget and $\mathcal{N} \triangleq \{1, \ldots, n\}$. Denote by $\mathcal{P}$ the set of all feasible strategies for the transmitter.
- Let the jammer also be equipped with n directed antennas.
- Let $q = (q_1, \ldots, q_n)$ be strategy of the jammer, where q_i is the power assigned to jam the communication between the transmitter with receiver i, and

$$\sum_{i \in \mathcal{N}} q_i = \overline{Q} \text{ and } q_i \geq 0, i \in \mathcal{N} \tag{6}$$

  with $\overline{Q}$ is the total jamming power budget. Denote by $\mathcal{Q}$ the set of all feasible strategies for the jammer.

- By *(4)*, the PCC with the receiver i is given as follows:

$$\boldsymbol{P}_i(p_i, q_i) = \frac{e^{-\epsilon \frac{N_i}{h_i p_i}}}{1 + \epsilon \frac{g_i q_i}{h_i p_i}}, \tag{7}$$

  where h_i and g_i are channel gains and N_i is the background noise.

- The PCC with all the n receivers is given as follows:

$$\pi(p, q) = \prod_{i \in \mathcal{N}} \boldsymbol{P}_i(p_i, q_i) = \prod_{i \in \mathcal{N}} \frac{e^{-\epsilon \frac{N_i}{h_i p_i}}}{1 + \epsilon \frac{g_i q_i}{h_i p_i}}. \tag{8}$$

The goal of the transmitter is to maximize this PCC, while the jammer wants to minimize this probability. Such a problem could arise in military operations where one radio station (say, GCS) must transmit data to n military units (e.g., drones) under hostile interference. Thus, $\pi(p, q)$ is the payoff to the transmitter, while for the jammer $\pi(p, q)$ is the cost function. Thus, here we deal with a zero sum game. We look for the *Nash equilibrium* [2]. Recall that (p_*, q_*) is a Nash equilibrium in a zero-sum game if and only if the following inequalities hold:

$$\pi(p, q_*) \le \pi(p_*, q_*) \le \pi(p_*, q) \text{ for all } (p, q) \in \mathcal{P} \times \mathcal{Q}. \tag{9}$$

Let

$$\begin{aligned} v(p, q) &= \ln(\pi(p, q)) \\ &= \sum_{i \in \mathcal{N}} \ln\left(\frac{h_i p_i}{h_i p_i + \epsilon g_i q_i}\right) - \epsilon \sum_{i \in \mathcal{N}} \frac{N_i}{h_i p_i}. \end{aligned} \tag{10}$$

Since $\ln(\cdot)$ is an increasing function, the problem to find the Nash equilibrium with payoff $\pi(p, q)$ to the transmitter is equivalent to finding the Nash equilibrium with payoff $v(p, q)$ to the transmitter, i.e., such (p_*, q_*) that

$$v(p, q_*) \le v(p_*, q_*) \le v(p_*, q) \text{ for all } (p, q) \in \mathcal{P} \times \mathcal{Q}. \tag{11}$$

Denote this game by $\Gamma = \Gamma(v, \mathcal{P}, \mathcal{Q})$.
Note that the transmitter's equilibrium strategy also reflects the most fair power resource allocation to maintain communication with all the receivers under the worst hostile interference since the utility v given by *(10)* can also be considered as a proportional fairness utility [20, 21].

Theorem 1 *The game $\Gamma(v, \mathcal{P}, \mathcal{Q})$ has at least one Nash equilibrium.*

The proof can be found in Appendix 9.1.
Note that, generally in resource allocation problems even if the payoffs are concave the game might have multiple equilibria (see, for example, [22]). In this paper we establish uniqueness of the equilibrium as a side effect of solving the best response equations associated with *(11)*.

4. SOLUTION OF THE GAME

In this section we find equilibrium strategies in closed form using a constructive approach via finding all solutions of the best response equations. Recall that, by *(11)*, (p, q) is a Nash equilibrium if and only if each of these strategies is the best response to the other, i.e., (p, q) is a solution of the best response equations:

$$p = \operatorname{argmax}\{v(p, q) : p \in \mathcal{P}\}, \tag{12}$$

$$q = \operatorname{argmin}\{v(p, q) : q \in \mathcal{Q}\}. \tag{13}$$

Note that *(12)* and *(13)* are Non-Linear Programming (NLP) problems.

4.1 Explicit form for the equilibrium strategies

In this section we find in closed form all of the possible solutions of the best response equations, i.e., equilibrium strategies, as functions of two auxiliary parameters ω and ν (Lagrange multipliers for the NLP problems *(12)* and *(13)* correspondingly).

Proposition 1 *Each equilibrium (p, q) of the game $\Gamma(v, \mathcal{P}, \mathcal{Q})$ has to have the following form*

$$\begin{aligned} p_i &= p_i(\omega, \nu) \\ &\triangleq \begin{cases} \sqrt{\frac{\epsilon N_i}{h_i \omega}}, & \frac{\epsilon g_i^2}{N_i h_i} \omega \le \nu^2, \\ \frac{1 + \sqrt{1 + 4 \frac{\epsilon N_i}{h_i} \left(\omega + \frac{h_i}{\epsilon g_i} \nu\right)}}{2\left(\omega + \frac{h_i}{\epsilon g_i} \nu\right)}, & \frac{\epsilon g_i^2}{N_i h_i} \omega > \nu^2 \end{cases} \end{aligned} \tag{14}$$

and

$$\begin{aligned} q_i &= q_i(\omega, \nu) \\ &\triangleq \left\lfloor \frac{1}{\nu} - \frac{h_i}{\epsilon g_i} \frac{1 + \sqrt{1 + 4 \frac{\epsilon N_i}{h_i} \left(\omega + \frac{h_i}{\epsilon g_i} \nu\right)}}{2\left(\omega + \frac{h_i}{\epsilon g_i} \nu\right)} \right\rfloor_+, \end{aligned} \tag{15}$$

where $i \in \mathcal{N}$ and $\lfloor \xi \rfloor_+ \triangleq \max\{\xi, 0\}$. Thus, $\lfloor \xi \rfloor_+ = \xi$ if $\xi \ge 0$ and $\lfloor \xi \rfloor_+ = 0$ otherwise. Moreover, $\omega > 0$ and $\nu > 0$ are solutions of the following equations:

$$P(\omega, \nu) \triangleq \sum_{i \in \mathcal{N}} p_i(\omega, \nu) = \overline{P}, \tag{16}$$

$$Q(\omega, \nu) \triangleq \sum_{i \in \mathcal{N}} q_i(\omega, \nu) = \overline{Q}. \tag{17}$$

The proof can be found in Appendix 9.2.
Note that, by Theorem 1 and Proposition 1, the non-linear equations *(16)* and *(17)* have at least one solution. To find this solution and to establish its uniqueness we cannot

apply a non-linear modification for the Gaussian elimination method suggested in [23] of a search game, and, further, in [24], applied for an Orthogonal Frequency-Division Multiplexing (OFDM) jamming game and, in [22], for a multi-user OFDM game. The issue is that although we can establish monotonicity in one direction by both variables for one of the functions (in our case, P), the other function (in our case, Q) generally is not monotonic in opposite directions of its variables. Instead of that approach, we reduce the non-linear equations *(16)* and *(17)* to a fixed point equation on ν and prove that it has a unique solution and then develop an algorithm to find this fixed point.

4.2 Auxiliary monotonicity properties

In this section we establish monotonicity properties of $P(\omega,\nu)$ with respect to its parameters and derive a bijective relation between ω and ν based on *(16)*.

Proposition 2 *Function $P(\omega,\nu)$ has the following properties:*

(a) *$P(\omega,\nu)$ is continuous in $\omega>0$ and $\nu\geq 0$;*[1]

(b) *for a fixed $\nu>0$, $P(\omega,\nu)$ is decreasing in ω from infinity for $\omega\downarrow 0$ to zero for $\omega\uparrow\infty$;*

(c) *for a fixed $\omega>0$, $P(\omega,\nu)$ is decreasing in ν from*

$$P(\omega,0)=\frac{1}{2\omega}\sum_{i\in\mathcal{N}}\left(1+\sqrt{1+4\frac{\epsilon N_i}{h_i}\omega}\right) \text{ for } \nu=0 \tag{18}$$

to

$$P(\omega,\infty)=\sum_{i\in\mathcal{N}}\sqrt{\frac{\epsilon N_i}{h_i\omega}} \text{ for } \nu\uparrow\infty; \tag{19}$$

(d) *for a fixed $\nu\geq 0$ there exists the unique $\omega=\Omega(\nu)$ such that*

$$P(\Omega(\nu),\nu)=\overline{P}. \tag{20}$$

Moreover, $\Omega(\nu)\in[\Omega_\infty,\Omega_0]$ and $\Omega(\nu)$ can be found via the bisection method;

(e) *$\Omega(\nu)$ is continuous and decreasing from Ω_0 for $\nu=0$ to Ω_∞ for $\nu\uparrow\infty$, where Ω_0 is the unique root of the equation:*

$$\frac{1}{2\Omega_0}\sum_{i\in\mathcal{N}}\left(1+\sqrt{1+4\frac{\epsilon N_i}{h_i}\Omega_0}\right)=\overline{P} \tag{21}$$

and

$$\Omega_\infty=\frac{\epsilon}{\overline{P}^2}\left(\sum_{i\in\mathcal{N}}\sqrt{\frac{N_i}{h_i}}\right)^2. \tag{22}$$

The proof can be found in Appendix 9.3.

[1] $\xi\downarrow a$ denotes that ξ tends to a decreasingly. Similarly, $\xi\uparrow a$ denotes that ξ tends to a increasingly.

4.3 An auxiliary fixed point equation

In this section we reduce Equation *(16)* and Equation *(17)* to a fixed point equation. First note that multiplying both sides of Equation *(17)* by ν implies that Equation *(17)* is equivalent to

$$\tilde{Q}(\omega,\nu)=\overline{Q}\nu, \tag{23}$$

where

$$\tilde{Q}(\omega,\nu)\triangleq\nu Q(\omega,\nu). \tag{24}$$

Moreover, by *(15)* and *(24)*, we have that

$$\tilde{Q}(\omega,\nu)\triangleq\sum_{i\in\mathcal{N}}\left[1-\frac{\frac{h_i}{\epsilon g_i}\nu\left(1+\sqrt{1+4\frac{\epsilon N_i}{h_i}\left(\omega+\frac{h_i}{\epsilon g_i}\nu\right)}\right)}{2\left(\omega+\frac{h_i}{\epsilon g_i}\nu\right)}\right]_+. \tag{25}$$

In the following proposition we establish monotonicity of $\tilde{Q}(\omega,\nu)$ on parameters ω and ν, and reduce Equation *(16)* and Equation *(17)* to a fixed point equation.

Proposition 3 *Function $\tilde{Q}(\omega,\nu)$ has the following properties:*

(a) *$\tilde{Q}(\omega,\nu)$ is continuous in $\omega>0$ and $\nu\geq 0$;*

(b) *for a fixed $\omega>0$, $\tilde{Q}(\omega,\nu)$ is decreasing in ν from n for $\nu=0$ to zero for $\nu\geq\overline{\nu}$ where $\overline{\nu}$ is the unique positive root of the equation*

$$A+B\overline{\nu}=C\overline{\nu}^{-3/2} \tag{26}$$

with

$$A=2\Omega_0,\ B=2\max_{i\in\mathcal{N}}\frac{h_i}{\epsilon g_i} \text{ and } C=\min_{i\in\mathcal{N}}\frac{h_i N_i^{1/2}}{\epsilon g_i^{3/2}}; \tag{27}$$

(c) *for a fixed $\nu>0$, $\tilde{Q}(\omega,\nu)$ is increasing in ω and tends to n for $\omega\uparrow\infty$;*

(d) *function*

$$\varphi(\nu)\triangleq\tilde{Q}(\Omega(\nu),\nu) \tag{28}$$

is decreasing in ν from n for $\nu=0$ to zero for $\nu\geq\overline{\nu}$,

(e) *The following fixed point equation has the unique positive root ν_*:*

$$\varphi(\nu_*)/\overline{Q}=\nu_*. \tag{29}$$

This root can be found via the bisection method with $[0,\overline{\nu}]$ as the initial localization interval for such ν_*.

The proof can be found in Appendix 9.4.

4.4 Equilibrium and its uniqueness

In this section we find the equilibrium and establish its uniqueness.

Theorem 2 *In the game $\Gamma(v, \mathcal{P}, \mathcal{Q})$, Nash equilibrium (p, q) is unique. Moreover, this Nash equilibrium is $(p(\omega, v), q(\omega, v))$ given by (14) and (15), where $v = v_*$ uniquely given by (29) and $\omega = \Omega(v_*)$ uniquely given by (20).*

The proof can be found in Appendix 9.5.

5. THE BOUNDARY CASES

In this section we find the equilibrium strategies in closed form for boundary cases of network parameters such as the TCC, the total jamming/transmission power resources and background noise at the receivers.

5.1 Negligible background noise at the receivers

In this section we consider the scenario with negligible background noise at the receivers.

Proposition 4 *Let the background noise at the receivers be negligible, i.e.,*

$$N_i = 0 \text{ for } i \in \mathcal{N}. \tag{30}$$

Then the unique Nash equilibrium (p, q) is given as follows:

$$p_i = \frac{g_i \overline{Q}/(\epsilon g_i \overline{Q} + h_i \overline{P})}{\sum_{j\in\mathcal{N}} g_j \overline{Q}/(\epsilon g_j \overline{Q} + h_j \overline{P})} \overline{P}, \tag{31}$$

$$q_i = \frac{g_i \overline{Q}/(\epsilon g_i \overline{Q} + h_i \overline{P})}{\sum_{j\in\mathcal{N}} g_j \overline{Q}/(\epsilon g_j \overline{Q} + h_j \overline{P})} \overline{Q}, \quad i \in \mathcal{N}. \tag{32}$$

The proof can be found in Appendix 9.6.
Proposition 4 implies that, for negligible background noise at the receivers, the equilibrium strategies of the transmitter and the jammer are proportional to ratio $\overline{P}/\overline{Q}$. Note that, in the SLCC problem solved in [14, 15] for negligible background noise at the receiver, equilibrium strategies are given in closed form. Proposition 4 also supplies the equilibrium strategies in closed form for the MLCC problem. Thus, an increase in the number of communication links does not lead to an increase in the complexity involved in designing the equilibrium strategies.

5.2 Large and small total transmission power

In this section we consider the cases where total transmission power is either large or small.

Proposition 5 *Let the total transmission power $\overline{P}$ be large. Then the Nash equilibrium (p, q) can be approximated as follows:*

$$q_i \approx \frac{g_i}{h_i}\left[\tau - \frac{N_i h_i}{g_i^2}\right]_+, \tag{33}$$

$$p_i \approx \frac{\overline{P}}{\overline{T}} \begin{cases} \sqrt{\frac{N_i}{h_i}}, & \tau \le \frac{N_i h_i}{g_i^2}, \\ \sqrt{\tau}\frac{g_i}{h_i}, & \tau > \frac{N_i h_i}{g_i^2}, \end{cases} \tag{34}$$

where τ is the unique positive root of the equation:

$$\sum_{i\in\mathcal{N}} \frac{g_i}{h_i}\left[\tau - \frac{N_i h_i}{g_i^2}\right]_+ = \overline{Q} \tag{35}$$

and

$$T = \sum_{\tau \le N_i h_i/g_i^2} \sqrt{\frac{N_i}{h_i}} + \sqrt{\tau} \sum_{\tau > N_i h_i/g_i^2} \frac{g_i}{h_i}. \tag{36}$$

The proof can be found in Appendix 9.7.

Proposition 6 *Let the total transmission power $\overline{P}$ be small. Then the Nash equilibrium (p, q) can be approximated as follows:*

$$p_i \approx \frac{\sqrt{N_i/h_i}}{\sum_{j\in\mathcal{N}} \sqrt{N_j/h_j}} \overline{P}, \tag{37}$$

$$q_i \approx \overline{Q}/n \text{ for } i \in \mathcal{N}. \tag{38}$$

The proof can be found in Appendix 9.8.
In particular, Proposition 5 and Proposition 6 imply that for large or small total transmission power the transmitter's strategies and the jammer strategy are insensitive to the TCC.

5.3 Large or small total jamming power

In this section we consider the cases where the total jamming power is either large or small.

Proposition 7 *Let the total jamming power $\overline{Q}$ be large. Then, the Nash equilibrium (p, q) can be approximated as follows:*

$$p_i \approx p_i(\Omega_0, 0) = \frac{1}{2\Omega_0} \sum_{i\in\mathcal{N}} \left(1 + \sqrt{1 + 4\frac{\epsilon N_i}{h_i}\Omega_0}\right), \tag{39}$$

$$q_i \approx \overline{Q}/n \text{ for } i \in \mathcal{N}, \tag{40}$$

where Ω_0 is given by (21).

The proof can be found in Appendix 9.9.

Proposition 8 *Let the total jamming power $\overline{Q}$ be small. Then, the Nash equilibrium (p, q) can be approximated as follows:*

$$p_i \approx \frac{\sqrt{N_i/h_i}}{\sum_{j\in\mathcal{N}} \sqrt{N_j/h_j}} \overline{P} \text{ for } i \in \mathcal{N} \tag{41}$$

$$q \in Q \text{ such that } supp(q) \subset \mathcal{I}, \tag{42}$$

where Ω_0 is given by (21), and

$$supp(q) \triangleq \{i \in \mathcal{N} : q_i > 0\}, \quad (43)$$

$$\mathcal{I} \triangleq \{i \in \mathcal{N} : N_i h_i / g_i^2 = \min_{j \in \mathcal{N}} N_j h_j / g_j^2\}. \quad (44)$$

The proof can be found in Appendix 9.10.
In particular, Proposition 7 and Proposition 8 imply that with large or small total jamming power resources, the jammer's strategy and the transmitter's strategy are insensitive to the TCC.

5.4 Large or small threshold of communication connectivity

In this section we consider the cases where the TCC is either small or large.

Proposition 9 (a) *Let the TCC ϵ be large. Then the Nash equilibrium (p, q) can be approximated by (37) and (38) of Proposition 6.*

(b) *Let the TCC ϵ be small. Then, the Nash equilibrium (p, q) can be approximated by (34) and (33) of Proposition 5.*

The proof can be found in Appendix 9.11.

6. ALGORITHM TO ARRIVE AT THE EQUILIBRIUM

In this section, an algorithm based on superposition of two bisection methods to arrive at equilibrium strategies is given.

Algorithm 1 The algorithm for deriving the equilibrium strategies $p(\omega, \nu)$ and $q(\omega, \nu)$, where $\delta > 0$ is a tolerance for the algorithm.

Procedure *Strategies()*
let $\nu_L = 0$ **and** $\nu_R = \overline{\nu}$
repeat
  let $F_L = \tilde{Q}(\Omega(\nu_L), \nu_L) - \overline{Q}\nu_L$
  let $F_R = \tilde{Q}(\Omega(\nu_R), \nu_R) - \overline{Q}\nu_R$
  let $\nu_C = (\nu_L + \nu_R)/2$
  let $F_C = \tilde{Q}(\Omega(\nu_C), \nu_C) - \overline{Q}\nu_C$
  if $F_C F_L < 0$ **then**
    let $\nu_R = \nu_C$
  else
    let $\nu_L = \nu_C$
  end if
until $\nu_R - \nu_L > \delta$
return strategies $p(\Omega(\nu_C), \nu_C)$ **and** $q(\Omega(\nu_C), \nu_C)$
End Procedure
Procedure $\Omega(\nu)$
let $\omega_L = \Omega_\infty$
let $\omega_R = \Omega_0$
repeat
  let $F_L = P(\omega_L, \nu)$
  let $F_R = P(\omega_R, \nu)$
  let $\omega_C = (\nu_L + \nu_R)/2$
  let $F_C = P(\omega_C, \nu)$
  if $(F_C - \overline{P})(F_L - \overline{P}) < 0$ **then**
    let $\omega_R = \omega_C$
  else
    let $\omega_L = \omega_C$
  end if
until $\omega_R - \omega_L > \delta$
return ω_C
End Procedure

Proposition 10 *Algorithm 1 converges to an equilibrium.*

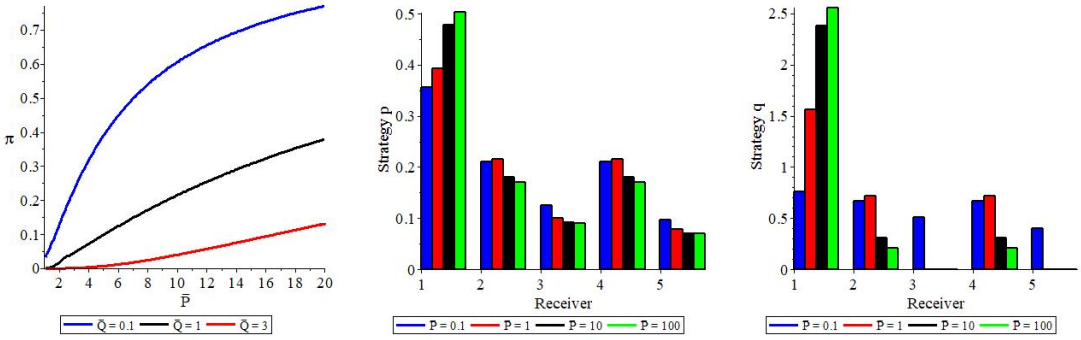
The proof can be found in Appendix 9.12.
The complexity involved in designing the equilibrium strategies by Algorithm 1 is $\log_2(\overline{\nu}/\delta) \log_2((\Omega_0 - \Omega_\infty)/\delta)$.

7. DISCUSSION OF THE RESULTS

In this section we illustrate Algorithm 1 using a system consisting of $n = 5$ receivers, with fading gains from the transmitter to the receivers $h = (1, 2, 3, 4, 5)$, fading gains from the jammer to the receivers $g = (3, 2, 1, 4, 1)$, the background noises at the receivers $N = (3, 2, 1, 4, 1)$, the total transmitter power budget $\overline{P} = 2$ and the total jamming power budget $\overline{Q} = 3$. Fig. 1(a) illustrates that an increase in the total transmission power leads to an increase in the PCC (i.e., in π), while an increase in the jamming power reduces the PCC. Fig. 1(b) illustrates the transmitter's normalized strategies, i.e., $p/\overline{P}$, while Fig. 1(c) illustrates the jammer's strategies for the total power transmitter budget $\overline{P} \in \{0.1, 1, 10, 100\}$. It shows that the jammer's strategy for a small total transmitter power budget $\overline{P}$ tends to a uniform strategy (Proposition 6), while for a large total transmitter power budget $\overline{P}$ the jammer's strategy tends to waterfilling-form strategies given by *(33)*. Due to the waterfilling form of Equation *(33)*, smaller $N_i h_i / g_i^2$ calls for applying larger jamming efforts. Here we have that

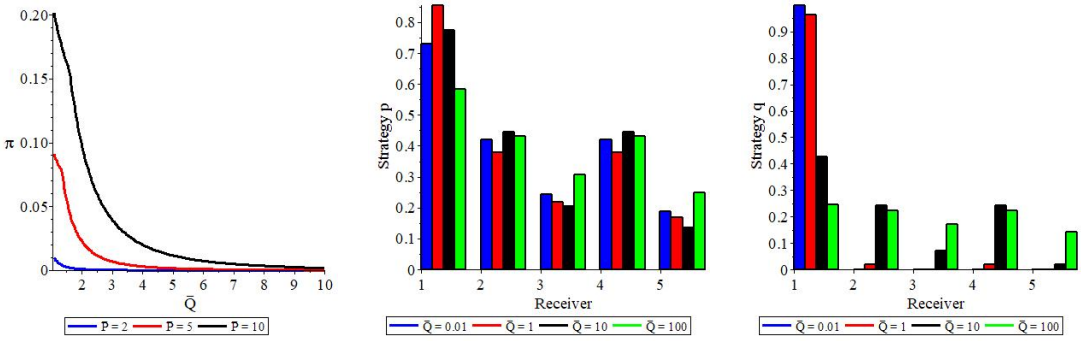
$$Nh/g^2 = (0.3, 1, 3, 1, 5). \quad (45)$$

For this reason, the largest jamming effort is focused on receiver 1 while receiver 2 and receiver 4 face approximately equal-level of interfering signals. Fig. 2(a) illustrates that an increase in the total jamming power leads to a decrease in the PCC, while an increase in the total transmission power reduces such negative effect from jammer's interference. Fig. 2(c) illustrates normalized jammer's strategies, i.e., $q/\overline{Q}$, while Fig. 2(b) illustrates transmitter's strategies for total jamming power budget $\overline{Q} \in \{0.1, 1, 10, 100\}$. By *(44)* and *(45)*, we have that $\mathcal{I} = \{1\}$. That is why, jamming efforts for small total power jamming budget $\overline{Q}$ is focused on receiver 1 (Proposition 8), while jamming efforts for large total jamming power budget $\overline{Q}$ tends to uniform distribution over all the receivers (Proposition 7). Fig. 3(a) illustrates that an increase in the TCC ϵ leads to a decrease in the PCC, while Fig. 3(b) and Fig. 3(c) illustrate that the jammer's strategy focuses jamming efforts on receiver 1 due to a waterfilling form of Equation *(33)* and *(45)* for a small TCC. For a large TCC, jammer's strategy tends to a uniform one (Proposition 9). In all of the cases, the transmitter tries to communicate with each receiver (i.e., $p_i > 0$ for all i). This observation makes the MLCC problem remarkably different from standard OFDM communication scenarios where the transmitter, lacking sufficient transmission resources, must avoid transmission in some of the channels.



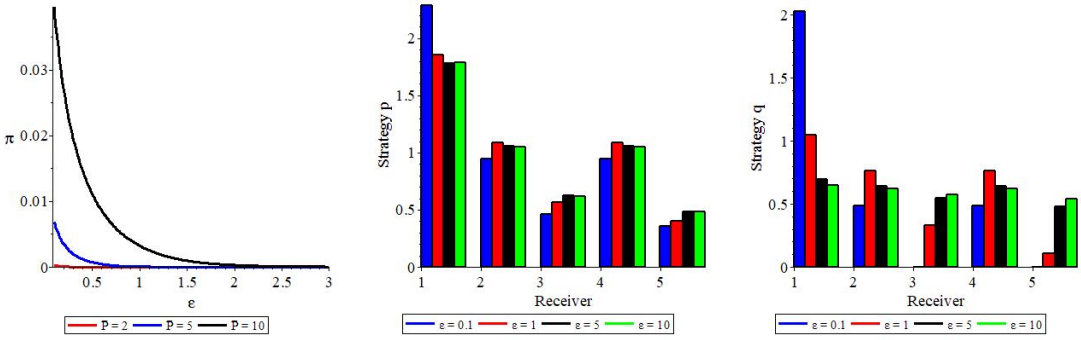
(a) (b) (c)

Fig. 1 – (a) The PCC π, (b) normalized transmitter's strategy p with $\overline{Q} = 3$ and (c) jammer's strategy q with $\overline{Q} = 3$ as functions of $\overline{P}$.



(a) (b) (c)

Fig. 2 – (a) The PCC π, (b) transmitter's strategy p and (c) normalized jammer's strategy q as functions of $\overline{Q}$.



(a) (b) (c)

Fig. 3 – (a) The PCC π, (b) normalized transmitter's strategy p with $\overline{Q} = 3$ and (c) jammer's strategy q with $\overline{Q} = 3$ as functions of $\overline{P}$.

Also, in OFDM communication, the transmitter's strategy becomes uniform in a high SINR mode, i.e., large transmission power, [25], while in the MLCC problem the jammer' strategy tends to a uniform one if either the total jamming resource budget is large, or the total transmission resource budget is small.

8. CONCLUSIONS

A problem of multi-link communication connectivity under jamming of a transmitter with a group of receivers when the channels are affected by Rayleigh fading has been formulated as a zero-sum power resource allocation game. Existence and uniqueness of the equilibrium in power allocation strategies have been proven. Thus, in contrast to Colonel Blotto games, if channels are affected by Rayleigh fading, then the stability of communication connectivity in a multi-link system can be maintained without introducing a random factor for a decision maker. Also, the problem of designing the equilibrium power allocation strategies has been reduced to the problem of finding a fixed point of a real-valued function. An algorithm based on the bisection method for finding the fixed point has been developed and its convergence has been proven.

ACKNOWLEDGEMENT

This work was supported in part by the U.S. National Science Foundation under Grant ACI-1541069, Grant CCF-1908308 and Grant CNS-1909186.

9. APPENDIX

9.1 Proof of Theorem 1

Note that $v(p,q)$ is an additively separable function of (p_i, q_i), $i \in \mathcal{N}$ and

$$\frac{\partial v^2(p,q)}{\partial p_i^2} = -\frac{\epsilon q_i g_i(\epsilon g_i q_i + 2h_i p_i)}{(\epsilon g_i q_i + h_i p_i)^2 p_i^2} - \frac{2\epsilon N_i}{h_i p_i^3} < 0 \quad (46)$$

and

$$\frac{\partial v^2(p,q)}{\partial q_i^2} = \frac{\epsilon^2 g_i^2}{(\epsilon g_i q_i + h_i p_i)^2} > 0. \quad (47)$$

Thus, $v(p,q)$ is concave in p and convex in q, and the result follows from the Nash's theorem [2] since sets $\mathcal{P}$ and $\mathcal{Q}$ of feasible strategies of the transmitter and the jammer are compact.

9.2 Proof of Proposition 1

By *(46)*, the NLP problem *(12)* is a concave problem. Thus, to solve the NLP *(12)* we introduce Lagrangian $\mathcal{L}_{T,\omega}(p)$ with ω is a Lagrange multiplier: $\mathcal{L}_{T,\omega}(p) = v(p,q) + \omega\left(\overline{P} - \sum_{i=1}^n p_i\right)$. Then, for a fixed $q \in \mathcal{Q}$, following [21] and the KKT Theorem, $p \in \mathcal{P}$ is the best response if and only if the following condition holds:

$$\frac{\partial \mathcal{L}_{T,\omega}(p)}{\partial p_i} = \frac{\epsilon g_i q_i}{(\epsilon g_i q_i + h_i p_i)p_i} + \frac{\epsilon N_i}{h_i p_i^2} - \omega \begin{cases} = 0, & p_i > 0, \\ \leq 0, & p_i = 0. \end{cases} \quad (48)$$

By *(47)*, the NLP *(13)* is a convex problem. Thus, to solve the NLP *(13)* we introduce Lagrangian $\mathcal{L}_{I,\nu}(q)$ with ν is a Lagrange multiplier as follows: $\mathcal{L}_{J,\nu}(q) = -v(p,q) + \nu\left(\overline{Q} - \sum_{i=1}^n q_i\right)$. Then, similarly, for a fixed $p \in \mathcal{P}$, $q \in \mathcal{Q}$ is the best response if and only if the following condition holds:

$$\frac{\partial \mathcal{L}_{J,\nu}(q)}{\partial q_i} = \frac{\epsilon g_i}{\epsilon g_i q_i + h_i p_i} - \nu \begin{cases} = 0, & q_i > 0, \\ \leq 0, & q_i = 0. \end{cases} \quad (49)$$

By *(48)*, we have that

$$p_i > 0 \text{ for any } i. \quad (50)$$

Then, by *(48)* and *(49)*, $\omega > 0$ and $\nu > 0$ correspondingly. By *(48)*, only two cases arise to consider: (I) $p_i > 0, q_i = 0$ and (II) $p_i > 0, q_i > 0$.

(I) Let $p_i > 0$ and $q_i = 0$. Then, by *(48)*,

$$p_i = \sqrt{\frac{\epsilon N_i}{h_i \omega}}. \quad (51)$$

Substituting *(51)* into *(49)* implies

$$\frac{\epsilon g_i^2}{N_i h_i}\omega \leq \nu^2. \quad (52)$$

(II) Let $p_i > 0$ and $q_i > 0$. Then, by *(48)* and *(49)*, we have that

$$\frac{\epsilon g_i q_i}{(\epsilon g_i q_i + h_i p_i)p_i} + \frac{\epsilon N_i}{h_i p_i^2} = \omega \quad (53)$$

and

$$\epsilon g_i q_i + h_i p_i = \frac{\epsilon g_i}{\nu}. \quad (54)$$

Thus,

$$q_i = \frac{1}{\nu} - \frac{h_i}{\epsilon g_i} p_i. \quad (55)$$

Substituting *(55)* into *(53)* implies

$$\frac{1}{p_i} + \frac{\epsilon N_i}{h_i p_i^2} = \omega + \frac{h_i}{\epsilon g_i}\nu. \quad (56)$$

Solving *(56)* on p_i implies that

$$p_i = \frac{1 + \sqrt{1 + 4\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}}{2\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}. \quad (57)$$

Thus,

$$q_i = \frac{1}{\nu} - \frac{h_i}{\epsilon g_i}\frac{1 + \sqrt{1 + 4\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}}{2\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}. \quad (58)$$

By *(57)*, $p_i > 0$. By *(58)*, $q_i > 0$ if and only if

$$\frac{2}{\nu}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right) > \frac{h_i}{\epsilon g_i}\left(1 + \sqrt{1 + 4\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}\right).$$

The last inequality is equivalent to

$$\frac{1}{\nu}\left(2\omega + \frac{h_i}{\epsilon g_i}\nu\right) > \frac{h_i}{\epsilon g_i}\sqrt{1 + 4\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}. \quad (59)$$

A straightforward calculation shows that *(59)* is equivalent to

$$\frac{\epsilon g_i^2}{N_i h_i}\omega > \nu^2. \quad (60)$$

Thus, q_i, given by *(58)*, is positive if and only if *(60)* holds. Finally, combining (I) and (II) implies the result.

9.3 Proof of Proposition 2

A straightforward substituting $\frac{\epsilon g_i^2}{N_i h_i}\omega = \nu^2$ into *(14)* implies (a).

Let

$$\eta(x) = \frac{1 + \sqrt{1 + ax}}{x} = \frac{1}{x} + \sqrt{\frac{1}{x^2} + \frac{a}{x}}, \quad (61)$$

where a is a positive parameter. It is clear that

$$\eta(x) \text{ is decreasing in } x > 0. \quad (62)$$

Let

$$x = \omega + \frac{h_i}{\epsilon g_i}\nu \text{ and } a = 4\frac{\epsilon N_i}{h_i}. \quad (63)$$

Substituting *(63)* into *(61)*, by *(14)* and *(62)*, yields (b) and (c). (d) follows from (b). (e) follows from (a) and (b), while *(21)* and *(22)* follow from *(18)*, *(19)* and *(20)*.

9.4 Proof of Proposition 3

(a) follows from *(25)*.

Let

$$f(\nu) = \frac{\nu}{c + a\nu} + \frac{\sqrt{1 + b(c + a\nu)}}{c + a\nu}, \quad (64)$$

where $a = h_i/(\epsilon g_i)$, $b = 4\epsilon N_i/h_i$ and $c = \omega$. Then

$$\frac{df(\nu)}{d\nu} = \frac{c}{(c + a\nu)^2} + \frac{(a^2 b\nu^2 + 3abc\nu + 2bc^2 + 2c)}{2(a\nu + c)^2\sqrt{ab\nu + bc + 1}} > 0. \quad (65)$$

Now we establish that

$$\tilde{Q}(\omega, \nu) = 0 \text{ for } \nu \geq \overline{\nu} \text{ and } \omega \in [\Omega_\infty, \Omega_0] \quad (66)$$

By *(25)*, $\tilde{Q}(\omega, \nu) = 0$ if and only if for all i the following inequality holds:

$$2\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right) \leq \frac{h_i}{\epsilon g_i}\nu\left(1 + \sqrt{1 + 4\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}\right). \quad (67)$$

Since $\omega \geq \Omega_\infty > 0$, substituting $\omega = \Omega_\infty$ into the right-side of *(67)* implies that if the following inequality holds then *(67)* also holds:

$$2\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right) \leq \frac{h_i}{\epsilon g_i}\nu\sqrt{4\frac{\epsilon N_i}{h_i}\frac{h_i}{\epsilon g_i}\nu}. \quad (68)$$

Taking into account notations *(27)* and that $\omega \leq \Omega_0$, *(68)* holds for all i if the following inequality holds:

$$A + B\nu \leq C\nu^{3/2} \quad (69)$$

with A, B and C given by *(27)*. It is clear that Equation *(26)* has the unique positive root $\nu = \overline{\nu}$. Then, *(69)* holds for $\nu \geq \overline{\nu}$, and *(66)* follows.

Thus, *(25)*, *(64)*, *(65)* and *(66)* imply (b). (c) follows from *(25)* and *(61)-(63)*. Also, (b), (c), *(66)* and Proposition 2(e) imply (d).

Since the right-side of Equation *(23)* is increasing from zero for $\nu = 0$ to infinity for $\nu \uparrow \infty$ while, by (d), the right-side of Equation *(23)* is decreasing and reaches zero for $\nu \geq \overline{\nu}$, (e) follows.

9.5 Proof of Theorem 2

By Proposition 1, all the equilibrium strategies have to have the form given by *(14)* and *(15)*, where (ω, ν) is a positive solution of *(16)* and *(17)*. By Proposition 2, *(14)* establishes a bijection relation between ω and ν given by function $\omega = \Omega(\nu)$. Substituting this function into *(15)* yields into Equation *(23)* of one variable ν. By Proposition 3, this equation has the unique root. Thus, the equilibrium also is unique, and the result follows.

9.6 Proof of Proposition 4

By *(30)*, *(52)* cannot hold for any i. Thus, by *(14)* and *(15)*, $p_i > 0$ and $q_i > 0$ for any i. Substituting *(30)* into *(14)* and *(15)* implies

$$p_i(\omega, \nu) = \frac{1}{\omega + \nu h_i/(\epsilon g_i)}, \quad (70)$$

$$q_i(\omega, \nu) = \frac{\omega/\nu}{\omega + \nu h_i/(\epsilon g_i)} \text{ for any } i. \quad (71)$$

Dividing *(71)* by *(70)* implies that

$$q_i(\omega, \nu) = (\omega/\nu)p_i(\omega, \nu). \quad (72)$$

Summing up *(72)* by i and taking into account that $p \in \mathcal{P}$ and $q \in Q$ we have that

$$\overline{Q} = (\omega/\nu)\overline{P}. \quad (73)$$

Thus, $\nu = \overline{P}\omega/\overline{Q}$. Substituting this ν into *(70)* implies that

$$p_i(\omega, \nu) = 1/(\omega(1 + h_i\overline{P}/(\epsilon g_i \overline{Q}))). \quad (74)$$

Since $p \in \mathcal{P}$, summing up *(74)* yields that $\omega = \sum_{i\in\mathcal{N}} 1/(\overline{P}(1 + h_i\overline{P}/(\epsilon g_i\overline{Q})))$. This and *(74)* yield *(31)*, while *(32)* follows from *(31)* and *(72)*.

9.7 Proof of Proposition 5

Since $\overline{P} \uparrow \infty$ there is at least one i such $p_i \uparrow \infty$. Then, by *(54)*, $\nu \downarrow 0$ and for large $\overline{P}$ we have that

$$p_i \approx \epsilon g_i/(h_i\nu). \quad (75)$$

By *(56)*, for large p_i we have that

$$1/p_i \approx \omega + (h_i/\epsilon g_i)\nu. \quad (76)$$

Thus, $\nu \downarrow 0$ while $\overline{P} \uparrow \infty$. Moreover, by *(75)* and *(76)*,

$$\omega/\nu \downarrow 0 \text{ for } \overline{P} \uparrow \infty. \quad (77)$$

Then, since $\sqrt{1+ax} \approx 1 + ax/2$ for small x, *(15)* implies that

$$q_i \approx \left[\frac{1}{\nu} - \frac{h_i}{\epsilon g_i}\frac{2 + 2\frac{\epsilon N_i}{h_i}\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}{2\left(\omega + \frac{h_i}{\epsilon g_i}\nu\right)}\right]_+ = \left[\frac{\omega}{\nu^2(\omega/\nu + h_i/(\epsilon g_i))} - \frac{N_i}{g_i}\right]_+. \quad (78)$$

By *(77)* and *(78)*, we have that

$$q_i \approx \frac{g_i}{h_i}\left[\frac{\epsilon\omega}{\nu^2} - \frac{N_i h_i}{g_i^2}\right]_+. \quad (79)$$

Let $\tau = \epsilon\omega/\nu^2$. Substituting this τ into *(79)* and taking into account that $q \in \mathcal{Q}$ imply *(33)* and *(35)*.
By *(14)* and *(75)*, we have that

$$p_i \approx \begin{cases} \sqrt{\frac{\epsilon N_i}{h_i\omega}}, & \tau \le \frac{N_i h_i}{g_i^2}, \\ \frac{\epsilon g_i}{\nu h_i}, & \tau > \frac{N_i h_i}{g_i^2}. \end{cases} \quad (80)$$

Summing up *(80)* and taking into account that $p \in \mathcal{P}$ yields the following relation: $\overline{P} = \sqrt{\frac{\epsilon}{\omega}}\left(\sum_{\tau\le N_i h_i/g_i^2}\sqrt{\frac{N_i}{h_i}} + \sqrt{\tau}\sum_{\tau>N_i h_i/g_i^2}\frac{g_i}{h_i}\right)$. This allows to define $\sqrt{\epsilon/\omega}$. Substituting this $\sqrt{\epsilon/\omega}$ into *(80)* implies *(34)* and *(36)*, and the result follows.

9.8 Proof of Proposition 6

Note that $p_i \downarrow 0$ since $\overline{P} \downarrow 0$. Then, by *(53)*,

$$\omega \uparrow \infty. \quad (81)$$

Since $q \in \mathcal{Q}$, by *(55)*, ν is upper-bounded on ω. This, *(14)* and *(15)* imply that $q_i > 0$ for any i. Then, by *(55)*,

$$q_i \approx 1/\nu \text{ for all } i \text{ and small } \overline{P}. \quad (82)$$

Since $q \in \mathcal{Q}$, summing up *(82)* by $i \in \mathcal{N}$ implies that

$$\nu \approx n/\overline{Q}. \quad (83)$$

Then, *(82)* and *(83)* imply *(38)*.
Substituting *(83)* into *(53)*, by *(81)*, implies that for small $\overline{P}$ the following approximation holds:

$$p_i \approx \sqrt{\epsilon N_i/(h_i\omega)}. \quad (84)$$

This and the fact that $p \in \mathcal{P}$ implies *(37)*.

9.9 Proof of Proposition 7

Since $q \in \mathcal{Q}$, $\overline{Q} \uparrow \infty$ implies that $q_i \uparrow \infty$ for at least one i. Then, by *(55)*, $\nu \downarrow 0$. Thus, by *(55)*, for large $\overline{Q}$ we have that

$$q_i \approx \nu \text{ for all } i. \quad (85)$$

Taking into account that $q \in \mathcal{Q}$ summing up *(85)* yields that $\nu = \overline{Q}/n$. Substituting this ν into *(85)* implies *(40)*. While substituting $\nu = \overline{Q}/n$ into *(14)* and taking into account that $\overline{Q} \uparrow \infty$ imply *(39)*, and the result follows.

9.10 Proof of Proposition 8

First prove that ν cannot tend to infinity while $\overline{Q}$ tends to zero. Assume that $\nu \uparrow \infty$. Then, by *(55)*, $p_i \downarrow 0$ where $q_i > 0$. Thus, by *(53)*, $\omega \uparrow \infty$. So, by *(14)*, $p_i \downarrow 0$ also for $q_i = 0$. This contradicts the fact that $p \in \mathcal{P}$. Thus, ν cannot tend to infinity while $\overline{Q} \downarrow 0$. Then, since $q_i \downarrow 0$ while $\overline{Q} \downarrow 0$, by *(55)*, we have that

$$\nu \approx \epsilon g_i/(h_i p_i) \text{ for } q_i > 0. \quad (86)$$

While, by *(53)*, since $q_i \downarrow 0$, we have that,

$$p_i \approx \sqrt{\epsilon N_i/(h_i\omega)} \text{ for } q_i > 0. \quad (87)$$

By *(14)*, $p_i = \sqrt{\epsilon N_i/(h_i\omega)}$ for $q_i = 0$. This, *(87)* and the fact that $p \in \mathcal{P}$ implies *(41)*.
Substituting *(87)* into *(86)* implies *(42)*, and the result follows.

9.11 Proof of Proposition 9

By *(8)*, since $p \in \mathcal{P}$, the equilibrium strategies are defined by ratio $\overline{P}/\epsilon$, and the result follows.

9.12 Proof of Proposition 10

First note that procedure $\Omega(\nu)$ reflects the bisection method to solve equation $P(\omega, \nu) = \overline{P}$ for a fixed ν. By Proposition 2, this procedure converges. Procedure *Strategies()* reflects the bisection method to solve equation $\tilde{Q}(\Omega(\nu), \nu) - \overline{Q}\nu = 0$. By Proposition 3, this procedure also converges. Thus, Algorithm 1 converges as a superposition of two converging procedures.

REFERENCES

[1] S. Vadlamania, B. Eksioglub, H. Medala, and A. Nandia. "Jamming attacks on wireless networks: A taxonomic survey". In: *International Journal of Production Economics* 172 (2016), pp. 76–94.

[2] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjrungnes. *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge University Press, 2012.

[3] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang. "Coping with a Smart Jammer in Wireless Networks: A Stackelberg Game Approach". In: *IEEE Transactions on Wireless Communications* 12 (2013), pp. 4038–4047.

[4] A. Garnaev, A. Petropulu, W. Trappe, and H.V Poor. "A Jamming Game with Rival-Type Uncertainty". In: *IEEE Transactions on Wireless Communication* 19 (2020), pp. 5359–5372.

[5] F.M Aziz, J.S Shamma, and G.L Stuber. "Jammer-type estimation in LTE with a smart jammer repeated game". In: *IEEE Transactions on Vehicular Technology* 66 (2017), pp. 7422–7431.

[6] T. Song, W.E Stark, T. Li, and J.K Tugnait. "Optimal Multiband Transmission Under Hostile Jamming". In: *IEEE Transactions on Communications* 64 (2016), pp. 4013–4027.

[7] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H.V Poor. "Two-Dimensional Antijamming Mobile Communication Based on Reinforcement Learning". In: *IEEE Transactions on Vehicular Technology* 67 (2018), pp. 9499–9512.

[8] N. Abuzainab and W. Saad. "Dynamic Connectivity Game for Adversarial Internet of Battlefield Things Systems". In: *IEEE Internet of Things Journal* 5 (2018), pp. 378–390.

[9] Y. Gao, Y. Xiao, M. Wu, M. Xiao, and J. Shao. "Game Theory-Based Anti-Jamming Strategies for Frequency Hopping Wireless Communications". In: *IEEE Transactions on Wireless Communications* 17 (2018), pp. 5314–5326.

[10] A. Garnaev, A. Petropulu, W. Trappe, and H.V Poor. "A Muli-Jammer Game with Latency as the User's Communication Utility". In: *IEEE Communications Letters* 24 (2020), pp. 1899–1903.

[11] A. Garnaev, A. Petropulu, W. Trappe, and H.V Poor. "A Power Control Game with Uncertainty On the Type of the Jammer". In: *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2019.

[12] N. Namvar, W. Saad, N. Bahadori, and B. Kelleys. "Jamming in the Internet of Things: A Game-Theoretic Perspective". In: *Proc. IEEE Global Communications Conference (GLOBECOM)*. 2016.

[13] Y. Wu, B. Wang, and K.J.R Liu. "Optimal power allocation strategy against jamming attacks using the Colonel Blotto game". In: *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*. 2009, pp. 1–5.

[14] G.D Nguyen, S. Kompella, C. Kam, J.E Wieselthier, and A. Ephremides. "Wireless link connectivity under hostile interference: Nash and Stackelberg equilibria". In: *Proc. 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 2016.

[15] G.D Nguyen, S. Kompella, C. Kam, J.E Wieselthier, and A. Ephremides. "Impact of Hostile Interference on Wireless Link Connectivity". In: *IEEE Transactions on Control of Network Systems* 5 (2018), pp. 1445–1456.

[16] R. El-Bardan, S. Brahma, and P.K Varshney. "Power control with jammer location uncertainty: A game theoretic perspective". In: *Proc. 48th Annual Conference on Information Sciences and Systems (CISS)*. 2014, pp. 1–6.

[17] A. Garnaev and W. Trappe. "Stability of communication link connectivity against hostile interference". In: *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2017.

[18] M. Hajimirsaadeghi and N.B Mandayam. "A dynamic Colonel Blotto game model for spectrum sharing in wireless networks". In: *Proc. 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2017, pp. 287–294.

[19] M. Labib, S. Ha, W. Saad, and J.H Reed. "A Colonel Blotto Game for Anti-Jamming in the Internet of Things". In: *Proc. IEEE Global Communications Conference (GLOBECOM)*. 2015, pp. 1–6.

[20] C. Guo, Y. Zhang, M. Sheng, Wang X, and Y. Li. "α-Fair Power Allocation in Spectrum-Sharing Networks". In: *IEEE Transactions on Vehicular Technology* 65 (2016), pp. 3771–3777.

[21] E. Altman, K. Avrachenkov, and A. Garnaev. "Fair Resource Allocation in Wireless Networks in the Presence of a Jammer". In: *Performance Evaluation* 67 (2010), pp. 338–349.

[22] E. Altman, K. Avrachenkov, and A. Garnaev. "Closed Form Solutions for Water-Filling Problem in Optimization and Game Frameworks". In: *Telecommunication Systems Journal* 47 (2011), pp. 153–164.

[23] V. Baston and A. Garnaev. "A Search Game with a Protector". In: *Naval Research Logistics* 47 (2000), pp. 85–96.

[24] E. Altman, K. Avrachenkov, and A. Garnaev. "Jamming Game in Wireless Networks with Transmission Cost". In: *Network Control and Optimization*. Ed. by T. Chahed and B. Tuffin. Vol. 4465. LNCS. Springer, 2007, pp. 1–12.

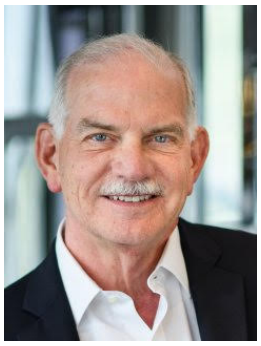
[25] A. Garnaev, W. Trappe, and A.Petropulu. "Equilibrium Strategies for an OFDM Network That Might be Under a Jamming Attack". In: *Proc. 51st Annual Conference on Information Systems and Sciences (CISS)*. 2017, pp. 1–6.

AUTHORS

Andrey Garnaev received a Ph.D. degree in applied mathematics and D.Sc. degree in computer science and applied mathematics from Saint Petersburg State University, St. Petersburg, Russia, in 1987 and 1997, respectively. He is currently with WINLAB, Rutgers University, USA. His current research interests are in applications of game theory and optimization theory in network security, wireless communication and related fields.

Wade Trappe is a Professor in the Electrical and Computer Engineering Department at Rutgers University, and Associate Director of the Wireless Information Network Laboratory (WINLAB), where he directs WINLAB's research in wireless security. He has led several federally funded projects in the area of cybersecurity and communication systems, projects involving security and privacy for sensor networks, physical layer security for wireless systems, a security framework for cognitive radios, the development of wireless testbed resources (the ORBIT testbed, www.orbit-lab.org), and new RFID technologies. His experience in network security and wireless spans over 15 years, and he has co-authored a popular textbook in security, *Introduction to Cryptography with Coding Theory*, as well as several monographs on wireless security, including *Securing Wireless Communications at the Physical Layer* and *Securing Emerging Wireless Systems: Lower-layer Approaches*. Professor Trappe served as an editor for *IEEE Transactions on Information Forensics and Security (TIFS)*, *IEEE Signal Processing Magazine (SPM)*, and *IEEE Transactions on Mobile Computing (TMC)*. He served as the IEEE Signal Processing Society representative to the governing board of IEEE TMC. Professor Trappe is an IEEE Fellow.

Narayan B. Mandayam is a Distinguished Professor and Chair of Electrical and Computer Engineering at Rutgers University, where he also serves as Associate Director of WINLAB. Using constructs from game theory, communications, and networking, his work has focused on system modeling, information processing as well as resource management. His recent interests include enabling privacy in IoT, building resilience in smart city infrastructures and trustworthy knowledge creation on the Internet. He received the 2015 IEEE COMSOC Advances in Communications Award, the 2014 IEEE Donald G. Fink Award, and the 2009 Fred W. Ellersick Prize from the IEEE Communications Society. He is also a recipient of the Peter D. Cherasia Faculty Scholar Award from Rutgers University (2010), the National Science Foundation CAREER Award (1998), the Institute Silver Medal from the Indian Institute of Technology (1989), Kharagpur and its Distinguished Alumnus Award (2018). He is a Fellow and Distinguished Lecturer of the IEEE.



H. Vincent Poor (S'72, M'77, SM'82, F'87) received a Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. During 2006 to 2016, he served as the dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications* (Cambridge University Press, 2021). Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.

INDEX OF AUTHORS

International
Telecommunication
Union

Telecommunication
Standardization Bureau (TSB)

Place des Nations
CH-1211 Geneva 20
Switzerland

ISSN: 2616-8375
Published in Switzerland
Geneva, April 2021