IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 October 2023, accepted 8 November 2023, date of publication 16 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333525

RESEARCH ARTICLE

Radio Propagation Digital Twin Aided Multi-Point Transmission With In-Network Dynamic On-Off Switching

CSABA GYÖRGYI[1], JÓZSEF PETŐ[2], PÉTER VÖRÖS[1], GÉZA SZABÓ[2], (Senior Member, IEEE), AND SÁNDOR LAKI[1], (Member, IEEE)

[1]Information Systems Department, Eötvös Loránd University (ELTE), 1117 Budapest, Hungary
[2]Ericsson Hungary Ltd., 1117 Budapest, Hungary

Corresponding author: Péter Vörös (vopraai@inf.elte.hu)

This work was supported in part by the European Commission through the HORIZON 6G European Smart Networks and Services Joint Undertaking (SNS JU) DESIRE6G under Grant G.A. 101096466; in part by the Project Strengthening the European Institute of Innovation and Technology (EIT) Digital Knowledge and Innovation Communities (KIC) in Hungary under Grant 2021-1.2.1-EIT-KIC-2021-00006; and in part by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, under Grant 2021-1.2.1-EIT-KIC.

ABSTRACT GPUs and programmable data planes have gone through an enormous evolution in the past years. GPUs can be used for modeling the real-world environment accurately, while programmable data planes can monitor the network in real-time and implement novel packet processing and decision logic. In this paper, we investigate how these two distant technologies can be combined to implement efficient coordinated multi-point (CoMP) transmission in an indoor environment. The proposed system implements the concept of dynamic on-off switching (DOOS) among various radio transmitters, relying on two information sources: 1) radio propagation digital twin based on real-time simulations of radio channels in the accurate 3D digital representation of the real industrial environment and 2) traffic load collected by the data plane for each transmitter. The proposed DOOS method implemented as a data plane algorithm dynamically selects a set of radio transmitters for each receiver by mixing information provided by the digital twin and the in-network traffic load measurements. The proposed method has low computational complexity and reduces the number of actively used radio transmitters. The proof-of-concept implementation of the proposed system has been validated in simulations as well as with measurements. The method achieves 69% energy saving for the radio compared to the default CoMP transmission and reception.

INDEX TERMS Digital twin, radio propagation, multi-point transmission, programmable dataplane, P4.

I. INTRODUCTION

There is a bloom of specialized processors to speed certain task-specific computations. There are Graphics Processing Units (GPU) to speed up graphics-related calculations. There are Network processors on the Network Interface Cards (NIC) which have a feature set specifically targeted at the networking application domain. Usually, the design and application life cycle of these processors is narrow, mostly focused on one task. E.g., for the first time, three-dimensional (3D) rendering was only supported by GPUs, the Cyclic Redundancy Check (CRC) calculation of Internet Protocol (IP) header by NICs. After the success of these features in the market, the functionality of the specialized processors has begun to expand.

In many use-cases, such as video encoding and decoding or AI, it has become common to use a General Purpose Graphics Processing Unit (GPGPU) as a modified form of a stream processor (or vector processor) running computational kernels. This concept harnesses the substantial computational capabilities of a contemporary graphics accelerator's shader pipeline, transforming it from a dedicated graphical processing unit into versatile general-purpose computing power, rather than limiting it solely to graphic operations. In some

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

applications that require massive vector operations, this can result in several orders of magnitude higher performance than a traditional Central Processing Unit (CPU). In 2019 Nvidia introduced the first generation of Ray-Tracing Texel eXtreme (RTX) cards, the first in the industry, to implement real-time hardware ray-tracing in a consumer product.

The functionality of NICs has also been extended, leading to smartNICs based on Infrastructure Processing Units (IPS) and a Data Processing Unit (DPU). The born of SmartNICs is the expansion of Software-Defined Networking (SDN). In the realm of networking, SDN involves the division of control functions from forwarding functions. This division facilitates enhanced automation and programmability within the network, essentially enabling a network to function in a manner reminiscent of cloud computing. SDN empowers network engineers and administrators to swiftly adapt to shifts in business needs through a centralized control interface that operates independently of the network's physical hardware. In simpler terms, SDN establishes a central intelligence hub for the network, enabling it to communicate with and direct other network components. The next step in the SDN evolution is happening with the increasing popularity of programmable data planes and the P4 programming language [1] in the network industry. P4 allows a programmer to define the entire packet processing pipeline that can be realized by different targets (e.g., switches, smartNICs, software data planes). GPGPUs and SmartNICs speed up certain calculations and reconfigurations to such an extent that new use cases may appear that were not manageable before due to the computational complexity and the introduced system delay.

In this paper, we focus on a problem of how the energy consumption of an indoor radio system using coordinated multi-point (CoMP) transmission can be reduced by the combination of accurate signal propagation emulation and real-time load measurements in the data plane. Our radio system uses an indoor radio unit generating the radio signals for a set of radio transmitters. All the radio transmitters forward the signal simultaneously to one or more receivers. Our investigation includes how the number of actively used radio transmitters and thus the total energy consumption of the system can be reduced by a Dynamic On-Off Switching method (DOOS) relying on accurate and real-time channel quality and load information.

A. PROBLEM DEFINITION

Our goal is to leverage the accurate modeling capabilities of GPUs and the real-time monitoring capabilities of programmable data planes to achieve energy-efficient CoMP transmission. The specific problem is to develop a system that implements Dynamic On-Off Switching (DOOS) among radio transmitters based on two key information sources: 1) real-time simulations of signal properties using GPUs to create a radio propagation digital twin, and 2) real-time traffic load data collected by the data plane for each transmitter.

The radio propagation digital twin (RPDT) relies on the accurate modeling of the industrial environment and calculates the pathloss values on the rays between the receivers and the transmitters. 3GPP TS 38.213 [2] §7 discusses the User Equipment (UE) uplink power control. It is a complex calculation, and many parameters are missing from the current RPDT implementation to fill in fully. The applied RPDT relies on the log-distance pathloss model [3]. Accordingly, the required UE transmit power (in dBm) from the path loss (in dBm) is calculated by $max(P_{Tx_{dBm}}, 30\ dBm) = P_{Rx_{dBm}} + PathLoss$, where $P_{Rx_{dBm}}$ is the received power at the base station considered with 3 dBm in our model. The max function emulates the effect of the UE power management. The received power is constant to maintain the signal-to-noise ratio (SNR) at the receiver side until the transmitter reaches its enabled maximum transmit power. After that, the pathloss results in reduced received power at the UE and packet drop eventually. The inverse pathloss values are then used as a channel quality indicator in our DOOS method.

We formulate the transmitter selection problem of DOOS as an optimization task. Let the industrial setting have a $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$ set of transmitters and $R = \{r_1, r_2, \ldots, r_m\}$ receivers (e.g., robots). Let $q^t_{\tau,r} \in \mathbb{R}^+_0$ denote the quality of the connection between the $\tau \in T$ transmitter and the $r \in R$ receiver at time t. This value has an inverse relation with the pathloss of the connection, thus the connection can be only used if $q^t_{\tau,r} > h$, where h is a predefined threshold. Similarly, let $l^t_r \in \mathbb{R}^+_0$ non-negative value be the traffic load of the $r \in R$ receiver a time t. The load of a given $\tau \in T$ transmitter at time t can be expressed as $\sum_{r \in R} l^t_r p^t_{\tau,r}$. Our goal is to provide a $p^t_{\tau,r} \in \{0, 1\}$ routing policy for every $\tau \in T, r \in R$ in every time t denoting whether the τ transmitter should transmit ($p^t_{\tau,r} = 1$) to the r receiver or not ($p^t_{\tau,r} = 0$).

To reduce the traffic usage and thus the energy consumption, we require that only half of the available transmitters be used for each r at a given t time:

$$\left\lceil \frac{\sum_{\tau \in T} \chi(q^t_{\tau,r} > h)}{2} \right\rceil = \sum_{\tau \in T} \chi(p^t_{\tau,r} = 1)$$

where $\chi : \mathbb{L} \longrightarrow \mathbb{N}$, and $\chi(true) = 1$ while $\chi(false) = 0$. Besides meeting the previously defined criteria, we want to ensure that the maximum H load of the transmitters is not exceeded:

$$\forall \tau \in T : \sum_{r \in R} l^t_r p^t_{\tau,r} < H$$

This integer-problem can be tackled with SMT-solvers such as Z3 [4] for every t. However, these solutions are too complex to be run on a programmable switch. Moving them to a more resourceful device would mean that they do not have access to the latest l^t_r values — that are promptly available in the switch — due to the latency of the required communication.

To this end, we have designed a proof-of-concept implementation and validated its performance through simulations and measurements. The evaluation criterion is to achieve energy savings for the radio system compared to default CoMP transmission and reception.

B. CONTRIBUTIONS

In this paper, we show how a GPU-assisted radio propagation digital twin accurately modeling the industrial environment and the link propagation can be combined with the real-time load measurements of P4 programmable data planes (running on switches or smartNICs) to efficiently realize multi-point transmission with dynamic on-off switching (DOOS) (see Fig. 1). As a realization of this concept, we present a novel data plane algorithm that considers both the channel quality predictions and the current load of the radio transmitters. The proposed method implements the DOOS mechanism in the pure data plane. It performs in-network data collection and handles quality reporting messages sent by the radio propagation digital twin without the need for any control plane interaction. This design enables ultra-fast reaction time to changes in the network and environmental conditions. The proposed algorithm is simple enough to be implemented in a Tofino ASIC switch.

A demo video on the proposed method is available at [5].

C. STRUCTURE OF THE PAPER

The structure of the paper is the following. In Section II the relevant background is collected in the paper. In Section III the related work is presented. We discuss the DOOS in Section IV, and its relation to digital twins and radio propagation modeling in Section V-B. In Section V-D we discuss how it can be combined with P4 and in Section VI we evaluate the results of our proof-of-concept implementation. Section VII concludes the paper.

II. BACKGROUND

This section covers the background grouped according to the topics that are considered in the paper.

A. DIGITAL TWINS AND RELATED MODELING APPROACHES

Digital Twin (DT) is a concept that encompasses more than just raw data. It involves the integration of models, algorithms, and feedback mechanisms to achieve a high degree of synchronization between the physical and virtual realms. Authors of [6] provided a widely recognized definition of Digital Twin, describing it as a probabilistic simulation of a vehicle or system that combines physical models, sensor data, fleet history, and other factors to replicate the behavior of its real-world counterpart. A Digital Twin is not limited to early-stage planning or simulation; it also facilitates real-time monitoring, control, diagnostics, and prognostics during the system's operation. It consists of three essential elements according to [7]: the physical space, the digital space, and bidirectional communication that ensures dynamic mapping between the two. Data flows from the physical space to the virtual space, while information flows from the virtual space back to the physical space. In the manufacturing domain, ISO/DIS 23247-1 [8] defines the Digital Twin as a dynamic model of manufacturing elements, including personnel, products, assets, and process definitions. It updates and adapts in response to changes in the physical counterparts. This definition highlights three types of Digital Twins: those for products, manufacturing assets, and manufacturing processes. In the 5th generation mobile network (5G) domain, [9] argues that the DT has the potential to assess the performance, predict the impact of the environment change, and optimize the 5G network processes and decision-making accordingly. The authors present a concept of cloud DT for 5G networks aiming to perform continuous assessment, monitoring, and proactive maintenance through the closed-loop data from physical entities to the virtual counterparts and vice versa. Within the 5G DT, the digital 5G model will run alongside the physical 5G network to perform operational predictions and enforce optimized decisions into the living network and associated services.

Based on the level of data integration between the physical and digital counterparts, Digital Twin applications can be categorized into three subcategories according to [10]: Digital Model, Digital Shadow, and Digital Twin. A Digital Model represents a comprehensive digital representation of the physical counterpart, such as a simulation or mathematical model. It does not involve automatic data exchange with the physical object. If there is a one-way flow of data from the physical object to its digital representation, it is referred to as a Digital Shadow. Changes in the physical object impact the digital representation. A Digital Twin, on the other hand, involves fully integrated bidirectional data flows, where the digital representation generates feedback to control the physical object. A Digital Twin combines the knowledge derived from modeling activities (Digital Model) and real-world operational data (Digital Shadow). By employing suitable simulation algorithms, an ''Experimentable Digital Twin'' can be created, enabling testing and evaluation of the physical system's performance under different conditions and scenarios.

Considering the specific definitions, this paper presents a digital model of a production cell that is connected to a P4 programmable switch in a Hardware in the Loop (HIL) manner. The P4 switch incorporates various load balancing techniques exclusively within the switch itself (refer to Section V-D2), and its output is interconnected with the digital model of the production cell. This interconnected setup of the P4 switch and the digital model forms a Digital Shadow of the environment. With both the digital model and the digital shadow interconnected within a feedback loop, we argue that a complete Digital Twin exists in the system. Consequently, we will refer to the system as a Digital Twin throughout this paper.

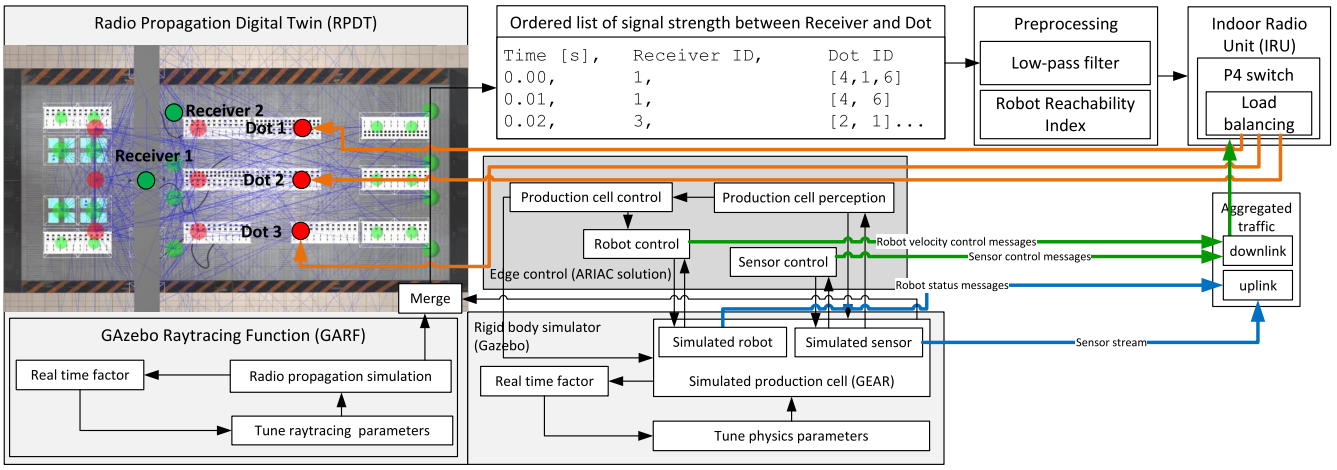

FIGURE 1. The proposed system.

B. RADIO PROPAGATION

A key part of any wireless communications system is the wireless channel where the radio waves are employed to carry the signals or information. Although electromagnetic (EM) waves are governed by the Maxwell equations with appropriate boundary conditions, it is not possible in general to have an analytical solution for the EM field in a realistic propagation environment. The purpose of propagation modeling is to obtain an estimation of field/signal strengths when some of the parameters of the wireless system are given, such as the frequency, terrain characteristics, antenna heights, and so on. There are propagation models, like the Friis equation for free space radio propagation, and there are empirical models which are widely used and are developed based on extensive field measurements like the Hata-Okumura model for urban regions [11].

The handling of MIMO and non-free space radio propagation e.g., indoor environments with dynamic objects is difficult with empirical models. The ray concept is intuitive from our daily experience with the sunlight. When the sunlight goes through an opening (large compared with wavelengths) on a wall and enters the room, we can see the 'ray' which is propagating along a straight line. We can imagine a point source from which many rays are emanating. Consider one of the rays and based on the behavior of this ray we can classify it as one of the following types of rays: a) direct rays with line of sight (LoS) propagation, b) reflected and transmitted rays, c) diffracted rays and d) scattered rays.

A key part of ray-tracing methods is to determine the rays from a source location to a field point. In the simplest case, i.e., in free space, the procedure is trivial: there is only one ray present (the direct ray) which is a straight line from the source to the receive point. In an urban environment which is the most common scenario for using ray-tracing, there may exist many rays from a source location to a field point; each ray may undergo a different number of reflections, diffractions, or their combinations. Determining these rays is not a trivial task but has been a hot research area since the 1990s. It involves two aspects: fast ray-tracing algorithms and accurate field calculations.

There are three main types of ray-tracing algorithms: a) Fermat's principle of least time, b) image method, and c) shooting and bouncing ray (SBR) method and any combination of these. Ray-tracing is a computation heavy and there are several acceleration methods in the literature. There are several algorithmic solutions [11] and there are HW-based solutions like the application of GPUs [12] or computation clusters-based solutions [13].

C. ENERGY SAVING IN RADIO

In order to meet the anticipated 1000x increase in data rates driven by the exponential growth in traffic demand, 5G wireless systems will require substantial deployment of BSs or APs to support high-data-rate services and ensure uninterrupted coverage. Although these BSs are expected to be low-power and small-scale individually, their cumulative energy consumption could be significant, raising environmental and economic concerns. In current cellular networks, one efficient strategy to conserve energy while maintaining the QoS for mobile users is to power down underutilized BSs. However, in 5G systems with new physical layer techniques and highly heterogeneous network architecture, new challenges arise in the design of BS ON-OFF switching strategies. Authors of [14] provide a comprehensive review of recent advances in switching mechanisms in different application scenarios.

D. P4 PROGRAMMABLE DATA PLANES

Although control plane programmability has a long track record, data plane programmability is relatively new. Special languages have evolved to describe the packet processing pipeline in a flexible and possibly portable way. One of them is P4 [1] which gained attention and support both from the academic world and the industry. Thanks to a large number of compilers, it is possible to build P4 code for various target devices ranging from CPUs, NetFPGAs [15], and smartNICs to high-performance ASICs such as Intel Tofino [16].

III. RELATED WORK

This section covers the state-of-the-art grouped according to the topics that are considered in the paper.

A. DIGITAL TWINS

Digital Twins are widely used in the industry for a variety of applications. There are 5G network-related DTs. As the Internet of Things (IoT) is gaining ground and becoming increasingly popular in smart city applications such as smart energy, smart buildings, smart factories, smart transportation, smart farming, and smart healthcare, the digital twin concept is evolving as complementary to its counter physical part.

Reference [17] discusses DTs in relation of 6G also identifying several open research challenges and future directions that need to be addressed before the end-to-end deployment of DT technology in 6G communication systems. Our paper concentrates on the utilization of DT technology in two main areas within the context of 6G communication systems. Firstly, we explore DT technology for achieving RAN moderation and efficient traffic steering in 6G communication systems. Secondly, we explore the potential of DT technology in enhancing radio resource management in 6G communication systems.

There are papers [18], [19], [20] addressing resource allocation challenges in the context of advanced technologies (6G and Industry 4.0) and IoT. They propose novel approaches (CRL and PER-DCW MADDPG) to improve resource allocation efficiency and convergence speed. Additionally, they both highlight the integration of digital twins and related technologies (DTs and DTN) in their respective frameworks. These papers encounter challenges due to the complex modeling of multi-layered services, control loops, and architectures. The proposed algorithms can only be demonstrated using simulated data, which may not efficiently capture the dynamics of real-world network deployments. Our proposed system is applied on real network traffic, implemented on real HWs proving the feasibility of the solution.

B. RADIO PROPAGATION SIMULATION

Opal [21], integrated with the Unity 3D game engine in the Veneris framework along with OMNET modules, allows for the rapid and intuitive generation of customized 3D environments using GPU-accelerated ray-tracing. This paper presents the closest existing solution in the state-of-the-art to our applied ray-tracing simulation. The main distinction is that while Opal is implemented on a generic game engine, our tool integrates with Gazebo, which is a highly accurate rigid-body simulator extensively used for robotic simulations. The behavior of the production cell in Gazebo is arguably more representative of a real environment compared to Unity 3D. There are two approaches to simulating physics. In the first approach, game engines prioritize real-time, high-speed execution, where the speed of the game takes precedence over the accuracy of the simulation. This often leads to simplifications in the simulation, primarily affecting visual accuracy but not the overall behavior of the game. Additionally, in multiplayer scenarios, synchronization across various hardware configurations becomes important. The second approach focuses on the accuracy of the simulation, even if it means sacrificing real-time performance. These simulators utilize simulation time rather than wall clock time. In our approach, we leverage an accurate simulator with hardware that supports real-time execution, ensuring both accuracy and real-time performance.

Authors of [22] connect the radio simulation with protocol layer simulation and vehicle movement simulation on real-world maps. They apply a digital twin to propose a city-model-aware deep learning algorithm for dynamic channel estimation in urban vehicular environments. While they suggest the use of geometry-based models such as ray-tracing techniques in the dynamic channel modeling of digital twin simulations, it's acknowledged that these approaches can be computationally intensive. In response, they also recommend an approach based on basis expansion modeling (BEM) to streamline the computational workload within the overall methodology. This allows for achieving a favorable equilibrium between precision and speed. Our approach also supports real-time operation by proper provisioning of the system, but we do not introduce simplification of the radio modeling by the BEM estimations.

C. ENERGY SAVING IN RADIO

Authors of [23] address the challenges posed by the rapid increase in cellular traffic and the need for efficient utilization of resources in 5G heterogeneous networks (HetNets). The authors propose a mechanism that maximizes system energy efficiency through enhanced k-means clustering and a dynamic load-based small cell (SC) switching algorithm. The method considers inter-cluster interference, while our proposed method has multiple dimensions considered during the DOOS operation making more sophisticated decisions.

The coordinated multi-point (CoMP) technique plays a crucial role in 5G mobile networks by supporting high system capacity and enhancing service quality, particularly for users at cell edges. An important aspect of CoMP operation is the decision-making process for cooperative clusters or cluster formation strategies. Authors of [24] propose a load-aware greedy dynamic CoMP clustering mechanism to address this issue. The proposed mechanism builds upon the traffic-based load-aware Dynamic Point Selection CoMP (DPS CoMP) mechanism, aiming to maximize spectral efficiency and balance cell loads within the coordinating area of 5G networks. Our paper analyzes the effects in a production cell compared to the simulated traffic in the previous paper.

D. P4 PROGRAMMABLE DATA PLANES

Numerous works are leveraging the opportunities of P4 to implement custom routing and load-balancing for different domain-specific applications. NetCache [25] implements

a rack-scale key-value store architecture to provide in-network caching and dynamic load balancing across storage servers. Memcached is a popular in-memory caching service, for quick content delivery. In [26] Bremler-Barr et al. introduce MBalancer, an L7 load balancer for Memcached implemented in P4. They duplicate the data on a set of servers and use the SDN controller to update the transmission tables to achieve load balancing without modification in the Memcached clients or servers. Jepsen et al. [27] demonstrate a solution that is able to deliver packets based on predefined subscription criteria, using a novel binary decision diagram-based algorithm to efficiently translate predicates into P4 tables. In [28] Barbette et al. present CHEETAH, a P4 load balancer that supports uniform load distribution and per-connection consistency. It can be deployed either statelessly or statefully on both software and a Tofino-based hardware switch. Tiara [29] enhances the performance of stateful load balancers by distributing tasks among three distinct computing resources: programmable switches, FPGA-based SmartNICs, and server CPUs. Specifically, the switch handles throughput-intensive packet encapsulation/decapsulation, while FPGAs and CPU cores manage memory-intensive real server selection. InFaRR [30] is an algorithm designed for fast rerouting within programmable data planes. Implemented using P4, InFaRR operates without the need for additional headers or network state heartbeats, showing promising results compared to current state-of-the-art solutions.

The high performance of software-defined data planes comes at a price. Every implementation has to deal with various restrictions of both the P4 language and the target architecture.

IV. MULTI-POINT TRANSMISSION WITH DYNAMIC ON-OFF SWITCHING

5G Base Stations (BS) are expected to be small-scale with low power. Even if 5G is designed to be energy efficient, there is a challenge to even further improve as networks get densified with a higher number of base-stations. In existing cellular networks, turning off the under-utilized BSs is an efficient approach to conserve energy while preserving the quality of service (QoS) of mobile users. As [14] collects the practical and implementation concerns two major issues need to be addressed.

A. TIMESCALE OF OPERATION

Since the ON-OFF states directly determine the QoS of users, BS ON-OFF switching is always coupled with other design issues, such as user association and traffic offloading. As discussed, BS ON-OFF switching takes both time and energy, it is thus infeasible to perform ON-OFF switching frequently. However, the system states are usually updated faster, and some technical approaches have to be executed more frequently.

Coordinated multi-point (CoMP) transmission and reception, which allows a user to be served by multiple cooperating BSs, is an effective approach to enhance spectral efficiency and link reliability at the price of increased overhead. The CoMP operation can be implemented with a set of indoor small cells, called Radio Dot System (RDS). The current mode of operation is that all dots assigned to one cell are always actively transmitting and receiving. To achieve this in the downlink, the Indoor Radio Unit (IRU) duplicates the Tx signal from the baseband to the dots. In general, only a few dots within a cell primarily contribute to the desired signal at the receiver, but all anonymously spill interference to adjacent cells. This is especially valid for open areas with few walls, such as industrial environments.

Dynamic on-off switching (DOOS) is the method of dynamic selection of active dot(s). The dot with the lowest pathloss towards the desired user is activated, while all other dots are muted. The muted dots reduce intercell interference for both downlink and uplink and also contribute to energy-efficient operation of the radio.

B. HOW TO ACQUIRE SYSTEM INFORMATION AND WAKE UP WHEN SLEEPING?

When a BS is turned off, the normal transmission between the BS and User Equipment (UEs) is suspended. Hence, the information on nearby UEs, such as CSI and traffic load, cannot be acquired by the BS from the uplink signals. To guarantee the effectiveness of scheduling, the BSs need to be aware of the environment even when they are in sleep mode, so that they can be activated in a timely manner.

As the BS ON-OFF scheduling is combinatorial in nature, it is NP-hard and cannot be solved with standard techniques. It is more challenging when BS ON-OFF is jointly scheduled for trade-offs among multiple performance metrics, as the formulated problem may be mixed integer programming with multiple sets of variables. There are various approaches in literature using AI approaches or game theoretic solutions.

Our approach applied in this paper is a DT-based solution.

V. OUR PROPOSED SYSTEM

In this section, we go through all the components of the proposed system and discuss them in detail. Fig. 1 shows the overall architecture of the system.

A. THE INDUSTRY 4.0 USE CASE

Our main goal is to test our proposed system in a robotic cell and analyze the radio propagation effects on the productivity Key Performance Indicators (KPIs) and the DOOS method in a dynamic environment. To achieve this goal, we take the Agile Robotics for Industrial Automation Competition (ARIAC) 2020 environment [31] as a baseline, and extend it to serve our needs. ARIAC is designed to promote agility in industrial robot systems. The continuously evolving simulated environment called GEAR is implemented in the Robot Operating System (ROS) [32] and is based on the inputs from industrial partners. Thus, it is considered here as a representative instance of a production line requiring agile forward-looking solutions. The ARIAC 2020 edition

implements a work cell with shelves and a conveyor belt, performing pick-and-place actions with two industrial robotic arms deployed on a gantry that is capable of moving on an actuator, while fulfilling incoming orders (see ''Simulated production cell (GEAR)'' in Fig. 1).

1) INPUT

An ''order'' describes the specific product ''parts' required for inclusion in ''shipments'' and dictates their precise placement on Automated Guided Vehicle (AGV) trays. An order is considered complete when all shipments have been assembled and dispatched. Therefore, the primary objective is to retrieve the correct product parts from storage ''bins'' and position them onto AGV trays using robotic arm(s). A ''conveyor belt'' serves as a means to transport both product parts and shipping boxes. This series of pick-and-place tasks becomes notably more challenging due to various ''agility challenges'' that arise, including updates to orders in progress, priority requests, insufficient product availability, and potential equipment failures. To adapt to this dynamic environment, the introduction of different ''sensors'' (such as cameras, break beams, and scanners) into the robotic cell is considered. The performance of the solution is assessed using a ''scoring'' mechanism that combines measures of ''efficiency'' (speed), ''performance'' (precision), and ''cost'' (simplicity).

2) PROCESSING

The competitors provide algorithms for the ''Production cell control'' including solutions for ''Production cell perception'' and the low-level ''Robot control'' and ''Sensor control'' tasks. For the solution we used the winning algorithm discussed in [33]. This solution is assumed to be running in the edge cloud (''Edge control (ARIAC solution) in Figure 1'').

a: REAL TIMING ASPECT

While hard real-time systems have strict time limits, known as deadlines, and are designed to meet those deadlines, soft real-time systems do not have a mandatory requirement to meet deadlines for every task. Our system falls into the category of soft real-time systems. We offer the following methods to ensure that the ''Simulated production cell'' operates in real-time. The first method involves appropriate provisioning of the system. The ''Real-time factor'' (RTF) indicates the simulation speed of Gazebo [34]. A factor of one signifies that the simulation runs in real-time, while a lower factor indicates slower performance. A well-provisioned system capable of real-time processing for the simulation environment typically achieves an RTF of around one with less than 100% CPU utilization for the Gazebo process. This prerequisite for the simulation environment should be tested before initiating the DOOS operation to fine-tune the environment size and the number of objects accordingly. The ''Rigid body simulator'' section in Figure 1 illustrates a control loop designed to ensure real-time operation. We have developed a Gazebo plugin that monitors the RTF value over a sliding window of 10 seconds. If the RTF drops below 1, the plugin begins to reduce the ''maximum number of contacts'' parameter in Gazebo. This parameter influences the calculation of physics in Gazebo by constraining the maximum number of contacts between two entities, such as face-to-face collisions, potentially sacrificing accuracy. In specific cases where the real production cell status is updated in real-time and there is no need to calculate physics in the rigid body simulation, the complete physics calculation can be switched off to reduce computational load.

3) OUTPUT

The solution provides velocity control packets from the ''Robot control'' and the ''Sensor control'' towards the ''Simulated robot'' and the ''Simulated sensor'' in GEAR via downlink radio channel and receives the robot status messages (joint position information) and the sensor data stream via uplink radio channel. The above-generated traffic is collected in the ''Aggregated traffic'' and serves as input for the ''Indoor Radio Unit''. This is the user plane traffic that needs to be routed towards the dots in a smart way. Note that the solution for the competition is designed in a way that it can query the position of any part at any time, and it assumes that the gathered information is correct. It means that there is no constant uplink camera stream, just when it is triggered by the sensor control solution – usually in case of replanning in the production cell control due to a new or updated order or in case of an agility challenge –, though in that case all the sensor data is queried simultaneously.

B. RADIO PROPAGATION DIGITAL TWIN (RPDT)

The ''Radio Propagation Digital Twin'' (RPDT) provides further input for the IRU. This is the control plane traffic that serves as the strategy for how the user plane traffic is routed from the IRU toward the dots. The other component of the RPDT is the ''GAzebo Ray-tracing Function'' (GARF).

1) INPUT

The whole status of the ''Simulated production cell'' including the position of the shelves, actuators, conveyor belts, and all the objects serves as the input for the ''Radio propagation simulation''

2) PROCESSING

To integrate the ray-tracer with our production cell environment, we used Gazebo's plugin system. We created a plugin called the GAzebo Ray-tracing Function (GARF) ROS plugin. It manages the loaded Gazebo models, which consist of links and their collision shapes (e.g., sphere, plane, mesh, etc.), and the positions of the transmitter and receiver nodes and which model they are connected to, so nodes that are on moving models can be updated. All the collision shapes are converted to meshes. When the plugin notices a new mesh (e.g., it was loaded by another Gazebo plugin), it saves the

mesh, then transforms it to its world position, and uploads the transformed mesh to the RF path tracer. The plugin runs the RF ray-tracer at a set interval time in simulation time. The RF path tracer is notified of every change either in the environment or in the positions of the transmitters and receivers. A change in any of these triggers a ray-trace run. No changes in the environment result in returning the previously provided results.

a: REAL TIMING ASPECT

Similarly to Section V-A2, this section also deals with a soft real-time component. The system is provisioned to ensure that the radio propagation simulation can meet the deadlines set by the robot control loops. As an example, the UR5e's [35] typical velocity control loop operates at 500 Hz, which means that control packets need to be transmitted every 2 milliseconds. Therefore, the performance of the radio propagation calculations must fall within this timeframe. We have employed a GPU-based tool [36] that leverages Nvidia GPUs' ray-tracing acceleration capabilities (RTX). This tool supports the calculation of all types of rays, capturing the intricate physical properties of radio frequency (RF) transmission and delivering predictive accuracy. It enables the calculation of radio propagation on high-resolution and complex indoor geometries, considering detailed surface materials that influence RF propagation, such as metallic coatings and metal features in places like factories. Additionally, it considers dynamic setups, such as the mobility of users and objects, which can cause intermittent connectivity between radio emitters and detectors. A notable feature of the applied RF path tracer is its ability to provide ray-traced paths with millisecond-level granularity on a standard-sized manufacturing cell. The calculation complexity of radio propagation depends primarily on two main parameters: the number of meshes in the environment and the number of rays traced over the production cell. To manage this complexity, we have implemented a control loop in GARF that monitors the duration of radio propagation within a 10-second sliding window. If the duration exceeds the control loop deadline, the number of rays is automatically reduced.

3) OUTPUT

The results contain all the paths between every node and the pathloss. This gets visualized in the Gazebo GUI and published on a ROS topic. This ROS topic is used by our next ROS node, sending the results to the P4 pipeline. The content is an ''ordered list of signal strength between the receivers i.e., user equipments (UEs), and the transmitters i.e., radio dots'' (see the top box in Figure 1).

C. PREPROCESSING

The ''ordered list of signal strength between the receiver and Dot'' serves as an input for the ''Preprocessing'', which can use this list in commanding the IRU into one of the following four operational modes: 1) robust connection-oriented, 2) energy saving oriented, 3) robot movement based (see Section V-C2) and 4) load balanced (see Section V-D2). The ''Preprocessing'' node prepares a list for the P4 programmable IRU that can perform multicasting, i.e., the DOOS accordingly.

1) LOW-PASS FILTER

High-quality ray-tracing involves the casting of numerous rays for each pixel to accurately simulate complex lighting interactions. However, achieving a noise-free image can demand an impractical number of rays, resulting in a rendering process that is excessively slow and computationally intensive. To address this challenge, denoising offers a means to strike a balance between rendering quality and efficiency, rendering real-time or interactive ray-tracing feasible. Various denoising techniques [37] primarily aim to enhance the clarity and visual appeal of ray-traced images. In our case, the issue with the IRU lies in its robustness regarding the received signal strength among the receivers and dots, even in the presence of noise. However, this robustness often leads to frequent reordering of the signal list, causing frequent activation and deactivation of the dots. The extent of reordering directly impacts the sleep duration of the dots. To address this, we introduce a low-pass filter into the system to flatten the noise associated with reordering over time. In Section VI-D, we delve into how different sleep modes can be achieved by configuring the appropriate low-pass filter settings.

2) REACHABILITY INDEX

The pathloss results can be easily organized. In this section, we aim to illustrate how KPIs related to robots can be incorporated into the routing process. The primary prerequisite for the feasibility of this approach is its execution speed, which should generate output at a frequency of approximately 10 Hz within the control loop. One viable method for achieving this is through the integration of the Reachability Index into the path ordering process.

We defined a threshold (5 dB in our experiments) for the pathloss similarity. If the difference of pathloss between two rays is less than the threshold we do a secondary order of the rays. We check the intersections of the two given rays with the Reachability Map [38]. The Reachability map characterizes the accessibility of a particular robot model by dividing its surroundings into discrete segments, generating positions within the environment, and computing valid Inverse Kinematics (IK) solutions for these positions. These accessible positions for the robot are linked to discrete spheres within the environment. Each sphere's accessibility within the environment is quantified using a Reachability index. We aggregate the reachability indexes of the intersecting spheres and make comparisons between them. We select the ray with the lower sum of reachability index values. This choice is based on the understanding that a lower reachability index indicates a challenging-to-access

area, reducing the likelihood of intersection with the robot tool (notably, our setup positions the receivers on the tools).

a: REAL TIMING ASPECT

From a performance point of view, the reachability map is calculated offline, and only the reachability index and the intersection with the related spheres need to be calculated in the magnitude of the control loop frequency.

D. INDOOR RADIO UNIT

The Indoor Radio Unit (IRU) duplicates Tx signals from the baseband unit to the radio dots and implements CoMP transmission. The Input of the system is the ''Aggregated traffic'' as the user plane traffic and the control plane traffic coming from the ''Preprocessing'' node. The output of the system is the routed traffic towards the dots.

In this section, we introduce the proof-of-concept implementation of the proposed RPDT-assisted Multi-point Transmission method in P4 and evaluate it on an Intel Tofino-based hardware switch. Our method handles downlink traffic, dynamically selects the set of active radio dots for each receiver, and multicasts the Tx signal accordingly. We have to note that our solution is purely implemented in the data plane and thus does not require any control plane interaction. Data plane programming is about high performance. Thus they usually require O(1) algorithms. P4 enforces this requirement by excluding loops from the language. P4 has many other restrictions on its own to keep control flows simple and efficient. The Tofino even promises to run every code line rate if it compiles (in exchange for more constraints). Our algorithm also has constant time complexity.

For each receiver, the Radio Propagation Digital Twin generates a status message whenever the active set of the receiver is changed. Each status message is encoded into a UDP packet and carries the new list of active radio dots for a given receiver. Inside the switch, we store the active set elements in registers. Since registers in P4 are array-like structures, we first map the identifier (e.g., MAC address) of the receiver to its register index, using a simple match-action table. Assuming that the set of active radio dots is limited to at most n elements, the data plane program maintains $n + 1$ register arrays: one register array is used for storing the number of active radio dots ($\leq n$), and each element in the active set is stored in a separate register array instance (n register arrays where the receiver's index position in the different instances stores the active elements). In our PoC implementation, the maximum size of an active set is limited to 8.

Upon the arrival of a new status message for receiver R, the data plane program updates the R's active set by filling the used register array instances at R's index position. The data plane pipeline can be finished by either dropping the status message or sending an acknowledgment message back to the RPDT component.

Note that the active set values could also be stored in a match-action table, but in this case, the update would require control plane interaction, resulting in larger delays and CPU load and making the proposed solution less scalable.

1) DUPLICATING TX SIGNAL

Downlink Tx signals encapsulated in Ethernet frames need to be forwarded to multiple transmitters listed in the receiver's active set. Assuming that the Tx signal's destination is receiver R, R's index can be used to determine the current size of its active set. This also means the number of copies to be created.

The new copies are created by the multicast engine that appears on the egress side. Since rerouting is not possible on the egress pipeline of Tofino ASIC, we might need to recirculate each copy in order to send them out on the correct port.

After recirculation, each copy can be distinguished and thus can be forwarded to one of the radio dots listed in the active set, including sending it out on the proper port, tagging it, or both. To this end, the ith replicated packet reads only the ith register instance and uses the read transmitter identifier to set routing (outgoing port, tagging, etc.).

2) PAIRWISE LOAD-BALANCING

The RPDT-assisted method described previously only takes into account the list of active transmitters reported by the RPDT component. RPDT models the environment and the signal propagation between transmitters and receivers, but does not deal with the network traffic. However, the traffic load on the various transmitters can be easily measured in P4 programmable data planes. We have extended our RPDT-assisted method with two pairwise load-balancing schemes that can distribute the traffic of multi-point transmissions among the radio dots listed in the active set. For each receiver, our approach selects a fixed number (K) of transmitters from the active set where K is generally less than the number of active transmitters. Since P4 and programmable switching ASICs result in different constraints on the computational complexity and on the way how registers and other extern objects can be used, we have designed a relatively simple algorithm. To facilitate our work, we assume that the RPDT component sends the active-set elements in decreasing order of the channel quality. The main idea of our load balancing algorithm is that we pair the elements of the active set according to ordering: the first two elements with the best channel quality form the first pair, the third and fourth elements are in the second pair, etc. Note that it may happen that the last pair is incomplete which is also handled by our algorithm (the pair only contains one radio dot). Instead of forwarding Tx signal frames towards all the active transmitters, the proposed load balancing method chooses only one transmitter from each pair according to various criteria. It assumes that each pair groups radio dots with similar channel quality. Note that the list is reported by the RPDT component and thus other grouping is also possible and can be implemented easily. We investigated two possible selection criteria: 1) random choice, and 2) load

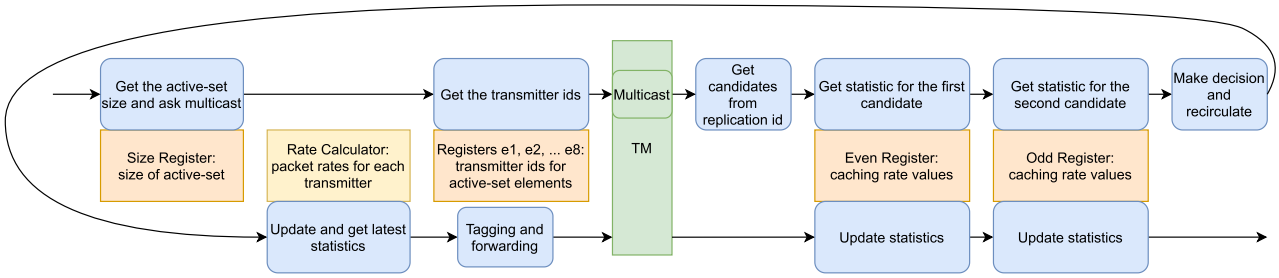

FIGURE 2. The proposed system.

comparison-based selection. 1) Random choice: We randomly select one of the elements from each pair. This method is stateless and easy to implement in P4 with a small overhead. 2) Comparison-based selection: Using some available externs (e.g., registers, low pass filters), the method continuously monitors the transmitters' load and chooses the less stressed element from each pair.

Fig. 2 depicts the data plane pipeline of the Comparison-based selection method. The implementation is challenging since hardware targets pose particular constraints when stateful objects like registers or low-pass filters are used. These extern objects can only be touched once during the packet processing pipeline. Our pipeline computes exponentially weighted moving averages of packet rates to keep track of the usage statistics of each transmitter. The load statistics are needed at two different points of the pipeline: 1) we need to read the load values for the comparison of load values in each pair, and 2) after the decision is made we have to update their value since the outgoing packets needs to be taken into account in the load statistics. We solved this issue by introducing two additional register arrays (as cache registers) that contain the last load statistics calculated for the elements of each pair. These stored values are used in the pairwise load comparison of our method. After one recirculation, the statistics can be updated and the new values can be stored in the appropriate cache registers. One can note that the caching mechanism might fail in certain situations. If we never choose a transmitter after a certain point of time, the cache will never be updated for this transmitter, causing potentially sub-optimal decisions. We can keep the cache up to date in many ways with negligible overhead, e.g., periodically generating extra packets to force the update and then drop these packets, or the control plane can reset the unused cache registers after a predefined timeout.

a: REAL TIMING ASPECT

The P4 language is a domain-specific language that does not support loops and recursion. Thanks to this design decision, every code that contains a constant number of recirculations/resubmissions has an $\mathcal{O}(1)$ runtime. Since our solution contains a maximum of one recirculation, we can claim this runtime. Furthermore, the Intel Tofino compiler guarantees that every compiling code is executed at line-rate. The only hindering factor could be a congestion in the traffic managers queue. However, our use case can safely assume relatively low-bandwidth usage.

VI. EVALUATION

In this section, we evaluate the proposed system from various aspects.

A. EFFECT ON THE PRODUCTIVITY KPIs

For evaluation purposes, we can use the ARIAC [31] scoring as KPIs as we did e.g., in [39]. These are not typical robotic-related KPIs like accuracy or repeatability, but they provide a normalized indicator of the performance of the robot cell during a complex scenario. However, it is important to note that the baseline setup is a grid deployment (see Fig. 3) of the dots with manual fine-tuning of the positions. This ensures that there is always connectivity between the transmitters and receivers with at least two rays all the time. In this way, there is no connection outage during the execution of the order fulfillment with the CoMP operation. Thus, we assume a well-set-up working production cell with certain DOT deployment. Enabling the DOOS operation does not alter the always-on connectivity of the robot cell, thus the *productivity KPIs* remain unaffected.

B. PERFORMANCE OF RPDT

The simulated robot cell contains 9 transmitters, and 30 receivers (27 are static, 3 are on the moving robot arm). The rays' simulated frequency can be set up, which is used for path loss computation. In our experiments, it was 2 GHz. We trigger the RF path tracer component of RPDT every 100 simulated milliseconds and only recompute the ray-trace in case of an environment change. A computation round takes approx. 50 ms on a desktop PC with an Nvidia 2060 RTX GPU. The computation time depends mainly on how many transmitters are considered, how many rays are projected, how many voxels are in the environment, etc. The main message is that the real-time calculation capability of the setup i.e., the robots' control-loop frequency can be maintained with proper parameter settings.

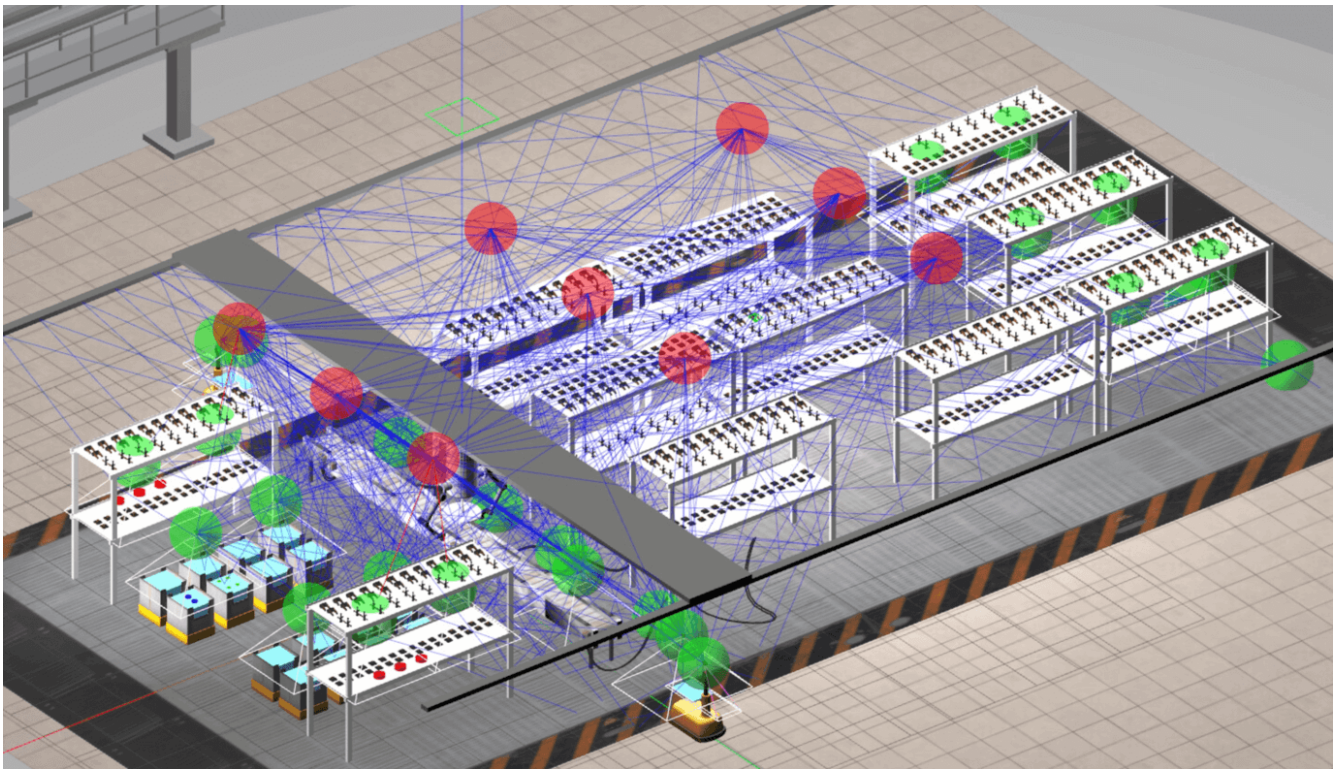

FIGURE 3. View on the production cell /red spheres – dots, green spheres – UEs, blue lines – rays/.

C. PERFORMANCE OF LOAD BALANCING

The performance of the pairwise load balancing schemes introduced in Sec. V-D2 was evaluated both in simulations and on a real hardware switch.

Our simulation-based analysis was carried out in SimPy [40], a discrete event simulator, simulating random walks of 10 robots in a square-like environment with 5×5 transmitters organized in a grid. The robots always approach a random location (set-point) with a random speed, and Tx signal frames are generated every 8 to 16 ms chosen uniformly at random. The robots do not have size, and they can not collide. A transmitter is included in the active set of a robot if it is inside a given radius. The random walks were always the same for each tested algorithm. Furthermore, a hypothetical 500 pps load limit (H) as defined in Section I-A is introduced, but not enforced by the simulation. This limit is marked by the red line in the figures. Note that the simulations were repeated many times with different random seeds and led to similar results.

Fig. 4 depicts the load on the most used transmitter for a 200s-long simulation period. The figure compares the results of our three proposed RPDT-assisted Multi-point transmission schemes: 1) 'no LB' used as baseline when all the dots in the active set are used for forwarding Tx signal frames, 2) 'random LB' shows when one radio dot from each pair is selected uniformly at random, while 3) 'compared LB' when the radio dot with less load is chosen from each pair.

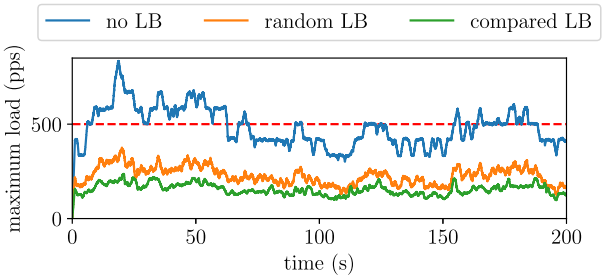

FIGURE 4. The maximum load experienced in a simulation scenario with 1 second aggregation.

It is not surprising that the maximum load is significantly smaller when only a subset of the active transmitters are used (both 'random LB' and 'compared LB'). It is more meaningful to compare the load comparison-based approach to the random selection. One can observe that 'compared LB' generally results in a lower maximum load than the random approach (mean: $\sim$ 68.54% of the random, median: $\sim$ 67.54% of the random).

Fig. 5 depicts the empirical cumulative distribution function of the observed load values on all the transmitters. A dotted line indicates highlights the point where the curve reaches 1.0 corresponding to the maximum observed load. One can see that even the simple random choice-based approach can result in a satisfying traffic reduction. Up to the 80th percentile, the two distributions basically run together.

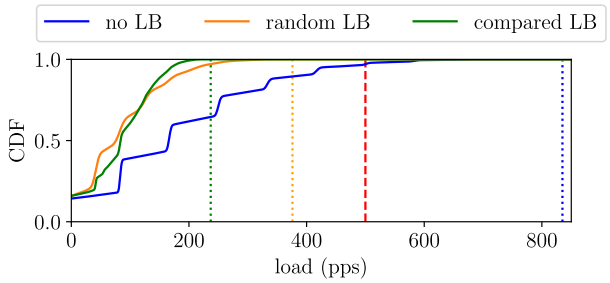

FIGURE 5. Distribution of the observed load values on the transmitters (simulation).

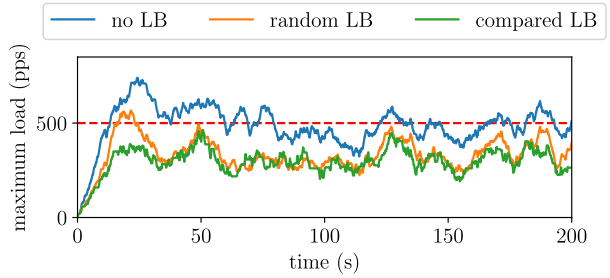

FIGURE 6. The maximum load experienced in a hardware measurement with 1 second aggregation.

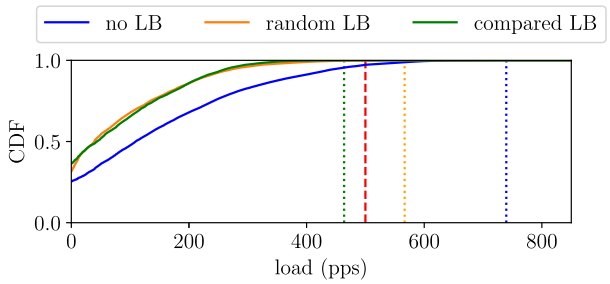

FIGURE 7. Distribution of the observed load values on the transmitters (hardware measurements).

In addition to simulations, we also measured the performance of the proposed solutions using a Tofino-based hardware switch. Note that the Tofino ASIC was chosen to demonstrate that the proposed pipeline is lightweight and can even be implemented in real programmable hardware. The test traffic was the same as in the simulated cases. Fig. 6 depicts the load on the most used transmitter in the measurement period. One can observe that though the load comparison-based approach results in a lower maximum load than the random choice in most cases, the difference between the two strategies is not significant (mean: $\sim$ 90.37% of the random, median: $\sim$ 90.67% of the random). The difference between the simulation and hardware measurements can be explained by the traffic bursts that did not happen in the simulator. Fig. 7 shows that the two load distributions are almost identical in this case.

The results of a random walk might suggest that random choice can be almost as good as the load comparison-based strategy. However, in some realistic scenarios, this is not the case. Fig. 8 depicts the case when we have 3 non-moving robots (R1, R2, R3) and 3 transmitters (T1, T2, T3). R1 sees T1 and T2; similarly, R3 sees T2 and T3; but R2 only sees T2. The random choice approach would only locally balance the traffic of R1 and R3; thus, half of their traffic would use T2, which is the only option for R2. However, if we always choose the option with the lower load, R1 uses T1 primarily, and R3 uses T3, thus leaving enough bandwidth for R2 who can only use T2.

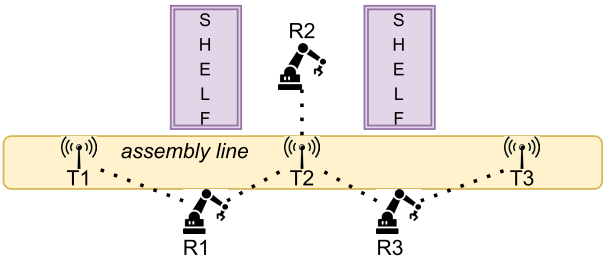

FIGURE 8. Example setup where random choices are not beneficial.

D. ENERGY SAVING

The DOOS operation provides benefits in terms of energy saving. Reference [41] provides a summary of base station sleep mode levels and their corresponding power consumption. Depending on the idle time duration, the base station may then enter a specific sleep mode while a proper wake-up time is ensured to continue normal operation. These different sleep modes can be created at the hardware sub-component level. By grouping the sub-components with similar transition times, we used four sleep modes (SM) in our power saving calculation: active, SM2 needed 1 ms, SM3 with 10 ms, and SM4 with 1s necessary sleep time to enter into this mode.

Fig. 9 shows the time spent in various sleep modes in the function of a given operation strategy. The leftmost column shows the 'all active' column, which means that during the CoMP operation, all 9 dots are transmitting all the time. There are three actuators in the production cell the gantry on the linear actuator and the two UR5e robotic arms. We assumed traffic between the 27 sensors and the transmitters every 10 ms. This provides input for the load balancing mechanism to work with. As the UR5e robotic arm has a 500 Hz control-loop for remote controlling, this induces traffic every 2 ms. We assume that the actuators send traffic whenever they are controlled to move. The 'raytrace' column refers to the case when those dots are sending data that have rays between the transmitter and receivers. It can be seen as the dots cover well the production cell, there is little chance to switch down a dot that has no active ray for any of the receivers. The 'X best signal' columns refer to the strategy when the list of rays is grouped by the receiver (see Fig. 1) and ordered according to pathloss. The best-performing X number of dots are selected from each receiver. This strategy provides the most robust communication for the actuators. It ensures that the receivers

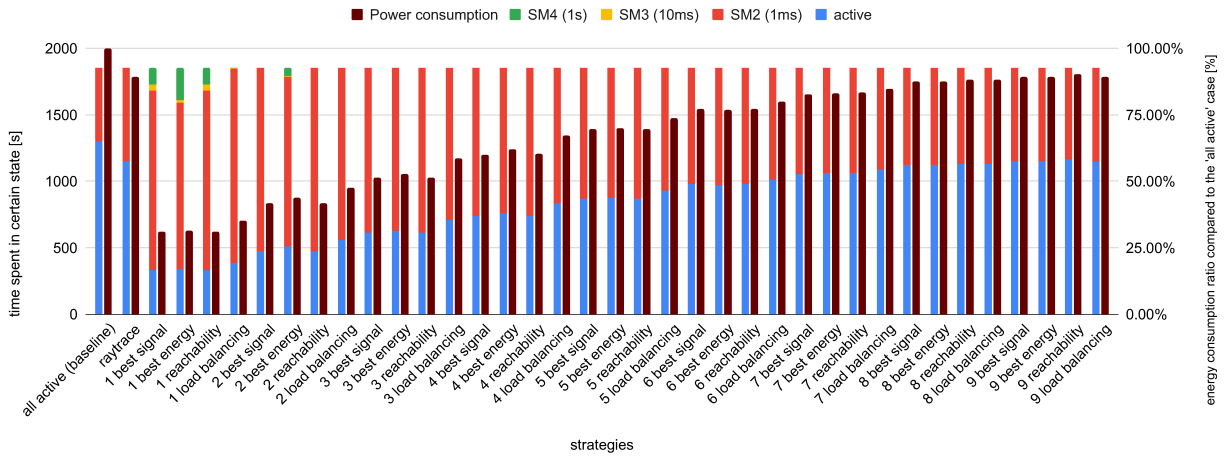

FIGURE 9. The time spent in various sleep modes in the function of applied strategy.

are all covered with the necessary number of dots. We do not switch off a dot if it can provide the best connectivity to a receiver. Note that this strategy does not mean that X dot is active in the whole production cell, but X best dots are selected for each receiver. If we simply select X active dots for the whole production cell, it would result in connection loss for certain areas. The 'X best energy' columns refer to the strategy when a certain dot is selected to be active, it does remain active till all the connections are lost without considering the pathloss. This results in rare on-off switching of the dots and ensures that the inactive dots can remain inactive for the longest period of time. This strategy achieves the least energy consumption and the most energy save. The 'X reachability' columns refer to the robot movement-related strategy introduced in Section V-C2, in which those rays are preferred that are blocked the least by the gantry and the two robot arms. The 'X load balancing' columns refer to the strategy introduced in Section V-D2. The number of active dots is increased from 1 to 9 till we reach the all-active set.

The sleep modes need to be weighted with the power consumption. This can be seen in the 'power' column compared to the 'all active' column which is considered the baseline with 100%. It can be seen that from the conservative energy focused strategy that achieves 31% energy consumption compared to the 'all active' case, choosing between various strategies can provide energy saving and robustness of connectivity as well.

1) DISCUSSION ON ENERGY EFFICIENCY

To assess the impact of energy efficiency mechanisms on energy savings, it is crucial to consider both the power consumption models and the energy efficiency metrics employed. These metrics should be comprehensive, reliable, and widely accepted to enable meaningful comparisons. Moreover, they must encompass the energy consumed by the system under investigation and the network-level performance measurements, including coverage, capacity, and delay. To fulfill these objectives, various standard development organizations (SDOs) such as the European Telecommunications Standards Institute (ETSI) Environmental Engineering Technical Committee, ITU-T Study Group 5, and 3GPP Technical Specification Group RAN have defined metrics for evaluating mobile network energy efficiency across different operating conditions. The contributions of these SDOs primarily focus on overall system performance, accounting for demographic areas, load scenarios, and radio access technologies. In contrast, the academic research community has proposed energy efficiency metrics that are specific to individual links, enabling more tailored network optimization for energy efficiency.

Given that enhancing energy efficiency is a key goal of 5G networks, where system capacity and associated power consumption are adjusted according to network load, load-aware metrics play a vital role in the development of next-generation green communication networks. In this regard, ETSI has introduced the mobile network data energy efficiency metric, $EE_{DV}[bit/J]$ [42], which quantifies the ratio between the data volume, DV, delivered in the network and the energy consumption, EC, observed during the corresponding time period required for data delivery.

Further calculations exist that involve fine-grained considerations such as weighing deployment scenarios (e.g., urban, suburban) and load levels. Our current focus is on end-to-end (E2E) network slicing-based metrics. Network slicing is a promising 5G technology that enables the simultaneous support of multiple services with diverse characteristics and requirements within a single network. The 3GPP is actively working on evaluating energy efficiencies for three main service families in Release 17: enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC), and massive machine-type communication (mMTC). Notably, a network slice is typically defined end-to-end, covering the radio access network (RAN), 5G core (5GC), and transport network.

For eMBB slices, the 3GPP specifies the usage of the EEDV metric (defined in Eq. 14 in [43]). Additionally, for URLLC and mMTC slices, where data volume is not the primary KPI, the 3GPP has introduced more appropriate metrics to characterize energy efficiency. In the case of URLLC slices, the 3GPP defines the $EE_{URLLC,Lat}$ metric [43], which represents the inverse of the average end-to-end latency divided by the energy consumption of the network slice $EE_{URLLC,Lat} = (T_{e2e}EC)^{-1}$, where T_{e2e} refers to the overall system end-to-end latency.

We argue that our proposed method does not alter the end-to-end latency. On one hand, the selection strategy consistently chooses the optimal path for transmission, and additional strategies are applied solely to ensure link robustness. On the other hand, the UE power control defined in 3GPP TS 38.213 [2] guarantees unaffected link quality and implicitly preserves end-to-end latency. In our case, robot control and sensor traffic operate at a constant bitrate. The signal-to-noise ratio does not significantly impact throughput, as the distances in the indoor scenario are limited, and the access points are set up with similar transmission power to provide uniform CoMP quality across the entire production area. There are further cases where the calculations become more fine-grained, incorporating throughput (as in Eq. 22 in [44]) or considering a large number of devices by including the ratio of the number of UEs in the slice (as in Eq. 23 in [44]).

Our method primarily affects the EC term in the formulas. The ratio of the EC term is influenced by the strategies illustrated in Figure 9. Consequently, the energy efficiency can be directly calculated based on the provided power saving, with the reciprocal weighted by a constant specific to the energy efficiency definition. While link-aware metrics exist for describing the energy efficiency of specific radio resource management methods like beamforming or scheduling, our approach considers the energy efficiency of the entire system without delving into such detailed calculations.

2) ABSOLUTE POWER CONSUMPTION

The power consumption of a certain radio cell is deployment-specific. The power consumption ranges can start from the 10W power consumption of the antennas, consisting of apptox. 100-500 W power consumption of a whole cell including the cooling as well [44]. This magnitude can be compared to the power consumption of the Tofino and the GPU that is required for the DT. When considering the energy consumption, it is important to factor in the GPU and Tofino's power usage. This can be adjusted by a fixed constant value, which modifies the energy-saving metrics discussed in this section. Regarding the Tofino, it should not be regarded as an additional cost, especially in terms of energy utilization. Its full power consumption is 1200W (with 300W attributed to the four fans), while the fixed ASIC requires 4.9 W/port and the Tofino itself consumes 4.2 W/port [45]. Additional cost comes at capex not in opex more likely due to lower production volumes, but it should be considered as a replacement for the existing switch rather than an extra cost. Regarding the GPU, its power consumption typically falls in the range of 200-500W, depending on factors such as whether it is an entry-level or high-performance card and whether it operates at base or peak performance levels. This energy consumption can vary significantly based on the specific requirements of the digital twin it needs to compute. Moreover, the environmental parameters can be adjusted in various ways, including mesh resolution, ray calculations, and overall setup, providing flexibility in energy usage optimization.

E. LIMITATIONS

When considering the DT, the primary challenge lies in maintaining the real-time status of both active and passive elements within the DT compared to the actual production cell. In cases where active elements are controlled from the edge, both control and status information are readily available and can be calculated. However, passive elements, such as a fallen bolt, require manual updates or the implementation of techniques that actively scan the environment and regularly update the DT (as demonstrated in [46]). In addition to the aforementioned conceptual limitation, the computational power of both the GPU and CPU places constraints on the complexity of the environment that can be processed in a soft real-time manner.

Decreasing the number of used transmitters inherently reduces reliability. If a robot receives information through a single transmitter, losing the connection with that transmitter would also lose control over that robot. Besides having more transmitters, our solution can be easily extended in a way that at least 2 transmitters are selected from the active set.

In-network computing and programmable data planes usually introduce constraints in exchange for high performance. We used a relatively strict Intel Tofino to evaluate our idea. However, investigating the potential and feasibility of a P4 programmable IRU looks promising. We expect that it could be more flexible than the current Tofino ASIC since the use case allows for less than 6.4 Tbps line-rate throughput.

One might be concerned whether we introduce additional packet reordering to the system since it might affect the performance of the actuators. First, it is important to note that CoMP operation resolves packet reordering issues on the radio level. Moreover, if the active set contains every possible transmitter with access to the destination and we transmit through all of them, we can state that we do not introduce additional reordering. Since we included every accessible transmitter in the active set, the ones excluded can not reach the destination, thus not affecting the reordering whether we use them or not.

VII. CONCLUSION

In this paper, we investigated how Dynamic On-off switching can be achieved by recent advances in technology. We considered two emerging applications. One is the radio

propagation simulation with GPUs, while the second is the application of P4 programmable switches. Both technologies can operate in a production cell's actuators control-loop frequency time frames. With the combination of the two techniques, we could achieve 69% energy saving for a DOOS use case in terms of radio equipment compared to the default Coordinated Multi-point (CoMP) transmission and reception case. We introduced various strategies that can be selected to choose between robust communication and energy-saving focused operation, also considering the robot kinematics induced and the P4's load balancing capability induced ray selection strategies.

REFERENCES

[1] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014.

[2] (Jan. 2023). *3GPP TS 38.213, 5G; NR; Physical Layer Procedures for Control*. [Online]. Available: https://www.3gpp.org/ftp/specs/archive/38_series/38.213/38213-h40.zip

[3] T. S. Rappaport, *Wireless Communications—Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[4] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Proc. Theory Pract. Softw., 14th Int. Conf. Tools Algorithms Construct. Anal. Syst.* Berlin, Heidelberg: Springer-Verlag, 2008, pp. 337–340.

[5] (2021). *Demo Video*. [Online]. Available: https://youtu.be/FlqxND4xGms

[6] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf. 14th AIAA*, 2012, p. 1818.

[7] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101837.

[8] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles*, Standard ISO 23247-1:2021, 2021. [Online]. Available: https://www.iso.org/standard/75066.html

[9] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5G and beyond," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 10–15, Feb. 2021.

[10] A. Ladj, Z. Wang, O. Meski, F. Belkadi, M. Ritou, and C. Da Cunha, "A knowledge-based digital shadow for machining industry in a digital twin perspective," *J. Manuf. Syst.*, vol. 58, pp. 168–179, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861252030128X

[11] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[12] R. Felbecker, L. Raschkowski, W. Keusgen, and M. Peter, "Electromagnetic wave propagation in the millimeter wave band using the NVIDIA OptiX GPU ray tracing engine," in *Proc. 6th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2012, pp. 488–492.

[13] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5G and beyond wireless communications: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 10–27, 1st Quart., 2019.

[14] M. Feng, S. Mao, and T. Jiang, "Base station ON-OFF switching in 5G wireless networks: Approaches and challenges," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 46–54, Aug. 2017.

[15] J. W. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo, "Netfpga—An open platform for gigabit-rate network switching and routing," in *2007 IEEE Int. Conf. Microelectronic Syst. Educ. (MSE7)*, 2007, pp. 160–161.

[16] *Explore the Power of Intel® Programmable Ethernet Switch Products*. Accessed: Oct. 4, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch.html

[17] N. P. Kuruvatti, M. A. Habibi, S. Partani, B. Han, A. Fellan, and H. D. Schotten, "Empowering 6G communication systems with digital twin technology: A comprehensive survey," *IEEE Access*, vol. 10, pp. 112158–112186, 2022.

[18] Z. Huang, D. Li, J. Cai, and H. Lu, "Collective reinforcement learning based resource allocation for digital twin service in 6G networks," *J. Netw. Comput. Appl.*, vol. 217, Aug. 2023, Art. no. 103697. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804523001169

[19] L. Tang, Y. Du, Q. Liu, J. Li, S. Li, and Q. Chen, "Digital-twin-assisted resource allocation for network slicing in industry 4.0 and beyond using distributed deep reinforcement learning," *IEEE Internet Things J.*, vol. 10, no. 19, pp. 16989–17006, Oct. 2023.

[20] A. Mozo, A. Karamchandani, M. Sanz, J. I. Moreno, and A. Pastor, "B5GEMINI: Digital twin network for 5G and beyond," in *Proc. NOMS IEEE/IFIP Netw. Operations Manage. Symp.*, Apr. 2022, pp. 1–6.

[21] E. Egea-Lopez, J. M. Molina-Garcia-Pardo, M. Lienard, and P. Degauque, "Opal: An open source ray-tracing propagation simulator for electromagnetic characterization," *PLoS ONE*, vol. 16, no. 11, Nov. 2021, Art. no. e0260060, doi: 10.1371/journal.pone.0260060.

[22] C. Ding and I. W. Ho, "Digital-twin-enabled city-model-aware deep learning for dynamic channel estimation in urban vehicular environments," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1604–1612, Sep. 2022.

[23] J. Natarajan and B. Rebekka, "An energy efficient dynamic small cell on/off switching with enhanced K-means clustering algorithm for 5G HetNets," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 29, no. 2, p. 209, 2023.

[24] S. Chantaraskul, P. Tarbut, and K. Nuanyai, "Load-aware greedy dynamic CoMP clustering mechanism for DPS CoMP in 5G networks," *Int. J. Networked Distrib. Comput.*, vol. 11, no. 1, pp. 31–48, Jun. 2023.

[25] X. Jin, X. Li, H. Zhang, R. Soulé, J. Lee, N. Foster, C. Kim, and I. Stoica, "Netcache: Balancing key-value stores with fast in-network caching," in *Proc. 26th Symp. Operating Syst. Princ.*, 2017, pp. 121–136.

[26] A. Bremler-Barr, D. Hay, I. Moyal, and L. Schiff, "Load balancing memcached traffic using software defined networking," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, Jun. 2017, pp. 1–9.

[27] T. Jepsen, A. Fattaholmanan, M. Moshref, N. Foster, A. Carzaniga, and R. Soulé, "Forwarding and routing with packet subscriptions," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2020, pp. 282–294.

[28] T. Barbette, C. Tang, H. Yao, D. Kostić, G. Q. Maguire Jr., P. Papadimitratos, and M. Chiesa, "A high-speed load-balancer design with guaranteed per-connection-consistency," in *Proc. 17th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2020, pp. 667–683.

[29] C. Zeng, L. Luo, T. Zhang, Z. Wang, L. Li, W. Han, N. Chen, L. Wan, L. Liu, and Z. Ding, "Tiara: A scalable and efficient hardware acceleration architecture for stateful layer-4 load balancing," in *Proc. 19th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2022, pp. 1345–1358.

[30] F. L. Verdi and G. V. Luz, "InFaRR: In-network fast ReRouting," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 3, pp. 2319–2330, Sep. 2023.

[31] A. Downs, Z. Kootbally, W. Harrison, P. Pilliptchak, B. Antonishek, M. Aksu, C. Schlenoff, and S. K. Gupta, "Assessing industrial robot agility through international competitions," *Robot. Comput.-Integr. Manuf.*, vol. 70, Aug. 2021, Art. no. 102113.

[32] (Sep. 2023). *Robot Operating System (ROS)*. [Online]. Available: https://www.ros.org/

[33] A. Vidács and G. Szabó, "Winning ARIAC 2020 by KISSing The BEAR: Keeping things simple in Best Effort Agile Robotics," *Robot. Comput.-Integr. Manuf.*, vol. 71, Oct. 2021, Art. no. 102166. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584521000508

[34] (2021). *Gazebo*. [Online]. Available: http://gazebosim.org/

[35] (2023). *Universal Robot UR5e*. [Online]. Available: https://www.universal-robots.com/products/ur5-robot/

[36] (2021). *Next-Generation Simulation Technology to Accelerate the 5G Journey*. [Online]. Available: https://www.ericsson.com/en/blog/2021/4/5g-simulation-omniverse-platform

[37] T. Viitanen, M. Koskela, K. Immonen, M. J. Mäkitalo, P. Jääskeläinen, and J. Takala, "Sparse Sampling for real-time ray tracing," in *Proc. VISIGRAPP (GRAPP)*, 2018, pp. 295–302.

[38] A. Makhal and A. K. Goins, "Reuleaux: Robot base placement by reachability analysis," 2017, *arXiv:1710.01328*.

[39] G. Szabó, S. Rácz, N. Reider, H. A. Munz, and J. Peto, "Digital twin: Network provisioning of mission critical communication in cyber physical production systems," in *Proc. IEEE Int. Conf. Ind. 4.0, Artif. Intell., Commun. Technol. (IAICT)*, Jul. 2019, pp. 37–43.

[40] T SimPy. (2019). *Simpy*. [Online]. Available: https://simpy.readthedocs.io/en/latest/
[41] B. Debaillie, C. Desset, and F. Louagie, "A flexible and future-proof power model for cellular base stations," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–7.
[42] (Oct. 2020). *ETSI ES 203 228 V1.3.1 (2020-10)*. [Online]. Available: https://www.etsi.org/deliver/etsi_es/203200_203299/203228/01.03.01_60/es_203228v010301p.pdf
[43] (Jun. 2023). *3GPP TS 28.554, Management and orchestration; 5G end to end Key Performance Indicators (KPI)*. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/28_series/28.554/28554-ha0.zip
[44] D. Lopez-Perez, A. De Domenico, N. Piovesan, G. Xinli, H. Bao, S. Qitao, and M. Debbah, "A survey on 5G radio access network energy efficiency: Massive MIMO, lean carrier design, sleep modes, and machine learning," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 653–697, 1st Quart., 2022.
[45] A. Agrawal and C. Kim. (2020). *Intel Tofino2 - A 12.9Tbps P4-Programmable Ethernet Switch*, Intel HotChips. [Online]. Available: https://hc32.hotchips.org/assets/program/conference/day2/HotChips2020_Networking_Tofino.pdf
[46] H. Zhao, M. Tomko, and K. Khoshelham, "Interior structural change detection using a 3D model and LiDAR segmentation," *J. Building Eng.*, vol. 72, Aug. 2023, Art. no. 106628. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710223008070

CSABA GYÖRGYI received the M.Sc. degree in computer science from Eötvös Loránd University, Hungary, where he is currently pursuing the Ph.D. degree. He is participating in EIT Digital's Doctoral Program, working with Ericsson, Hungary. He is also a Visiting Researcher with the University of Vienna.

JÓZSEF PETŐ received the M.Sc. degree in computer engineering from the Budapest University of Technology and Economics, in 2018, where he is currently pursuing the Ph.D. degree. His current areas of research interests include cloud robotics, digital twin, robot simulation, and machine learning applications in robotics and networking.

PÉTER VÖRÖS received the M.Sc. and Ph.D. degrees from Eötvös Loránd University (ELTE), Budapest, in 2014 and 2019, respectively. He is currently an assistant professor. Recently, he has been working on projects on network security, traffic analysis, and programmable data planes.

GÉZA SZABÓ (Senior Member, IEEE) received the Ph.D. degree in informatics, in 2011. He wrote the M.Sc. thesis about comparing various application traffic classification methods, in 2006. He joined Ericsson Research, as an Undergraduate Student, in 2005. Since 2017, he has been working in the field of Industrial 4.0 and evolves robot cells into cyber physical production systems via the help of 5G and AI. He is a delegate in 3GPP SA6 and the Vice-Chair of IEEE P2940 Standardization Group.

SÁNDOR LAKI (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from Eötvös Loránd University, in 2007 and 2015, respectively. He is currently an Assistant Professor with the Department of Information Systems, Eötvös Loránd University. He has authored over 40 peer-reviewed articles and demo papers, including publications at *JSAC*, INFOCOM, ICC, and SIGCOMM. His research interests include active and passive network measurement, traffic analytics, programmable data planes, and their application for new networking solutions.