KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Information and Communication Technology

Ultra-Reliable and Resilient Communication Service for Cyber-Physical Systems

MILAD GANJALIZADEH

Stockholm, Sweden 2023

Ultra-Reliable and Resilient Communication Service for Cyber-Physical Systems

MILAD GANJALIZADEH

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Thursday the 24th of August 2023, at 2:00 p.m. in F3, Lindstedtsvägen 26, Stockholm.

Doctoral Thesis in Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden 2023

ISBN 978-91-8040-633-8
TRITA-EECS-AVL-2023:54

Printed by: Universitetsservice US-AB, Sweden 2023

Abstract

Cyber-Physical Systems (CPSs) are becoming ubiquitous in modern society, enabling new applications that rely on the seamless interaction between computing, communication, and physical processes. In this context, ultra-reliable low-latency communications (URLLC) emerges as a crucial element, reliably allowing the real-time exchange of critical data.

In wireless networks, reliability is commonly evaluated based on the percentage of packets delivered successfully, with timeliness sometimes considered. Nevertheless, in CPSs, performance is typically assessed by operational metrics such as *availability* (as the ability to provide service at any given time) and *reliability* (as the ability to maintain consistent service over an extended period). To bridge the gap between these two domains, we study the CPSs performance in terms of wireless communications and derive a mapping function between the well-known network metrics (such as packet error ratio) and operational metrics (namely communication service availability and reliability) for deterministic traffic arrivals. This thesis then deals with wireless system orchestration techniques that aim to facilitate URLLC for CPSs, factoring in spectrum and energy efficiency. It investigates two scenarios: i) a single service, where the focus is only on URLLC, and ii) mixed services, where other services simultaneously run on the same network as URLLC.

In the first part, we assume that the impact of other nearby services on URLLC service is negligible. Accordingly, we concentrate on diversity techniques and power control as primary methods to enhance communication service availability and reliability at the cost of redundant transmissions and excessive resource usage. Thus, we devise a deep reinforcement learning (DRL) orchestrator that optimizes the number of hybrid automatic repeat request retransmissions and transmission power to enhance these metrics. We use a deep Q-network framework along with a branching soft actor-critic (BSAC) framework to address scalability issues in per-device orchestration. Our 3GPP-compliant simulations show that our approach achieves significant gains in computational time and memory requirements compared to the state-of-the-art. Besides, our approach requires substantially less energy or spectrum to achieve the target metrics. Additionally, we offer valuable insights into the practical implementation of DRL solutions for URLLC service in real-world wireless communication systems.

In the second part, we examine mixed services with an emphasis on distributed learning as a coexistent service. We consider 5G-NR's quality of service mechanisms to prioritize URLLC traffic and develop models to characterize distributed training workflow in terms of training delay, model size, and convergence. This leads to an optimization problem that uses device selection to minimize distributed learning convergence time, while meeting URLLC availability requirements. We transform this coexistence problem into a DRL problem and tackle it with our adjusted BSAC framework. Our simulations reveal that our approach achieves URLLC service availability performance comparable to the scenario where all communication resources are solely dedicated to URLLC service, and significantly higher than the performance achieved using a static slicing approach with unvarying dedicated resources to slices. Finally, we propose a hierarchical reinforcement learning architecture for dynamic resource slicing on a large timescale, thereby enhancing network flexibility, scalability, and profitability.

Keywords: 5G, availability, cyber-physical systems (CPSs), deep Q-networks (DQN), deep reinforcement learning, distributed learning, machine learning, network slicing, reliability, soft actor-critic (SAC), ultra-reliable low-latency communications (URLLC), wireless communications.

Sammanfattning

Cyberfysiska system (CPS) blir alltmer relevanta och påtagliga i det moderna samhället. I sådana system sker en sömlös interaktion mellan datorberäkningar, kommunikation och fysiska processer. I detta sammanhang spelar system för kommunikation med extremt hög tillförlitlighet och låg fördröjning, s.k. URRLC (Ultra Reliable Low Latency Communication), en nyckelroll vid överföring av viktiga och tidskritiska data. I denna avhandling undersöks existerande tekniska lösningar för att erbjuda URLLC-tjänster i CPS-tillämpningar samt dessas prestanda och begränsningar. Vidare presenteras och utvärderas våra förslag till förbättringar för URRLC-tekniker inom ramen för 5G och inom andra nätarkitekturer.

Avhandlingen inleds med en översikt av de utmaningar som möter konstruktörer av CPS inom trådlösa nätverk, samt med en jämförelse av de vanligaste förekommande trådbundna och trådlösa kommunikationsstandarderna, inklusive 5G. Därefter presenteras en omfattande översikt av olika prestandakriterier för CPS samt för de senaste tekniska lösningar för URLLC-tjänster.

Tillförlitligheten för ett trådlösa nätverk utvärderas vanligen genom att mäta andelen datapaket som levereras i tid. Relevanta prestandamått för CPS är däremot systemets *tillgänglighet* (förmågan att tillhandahålla tjänster vid en godtycklig tidpunkt) och *tillförlitlighet*, som här definieras som förmågan att kunna bibehålla tjänsten över tid. I syfte att överbrygga klyftan mellan dessa två domäner studeras prestanda hos CPS som utnyttjar trådlösa nätverk som kommunikationsmedium. I avhandlingen härleds ett samband mellan väletablerade prestandamått för nätverk (t.ex. paketfel) och prestandamått för CPS (tillgänglighet och tillförlitlighet) för en deterministisk trafikmodell. Denna avhandling analyserar olika implementeringsstrategier för trådlösa system för att dessa skall uppfylla de prestandakrav som ställs på CPS. Kritiska parametrar är här spektrum- och energieffektivitet.

Två scenarier studeras: i) ett där enbart en URLLC-tjänst erbjuds, och ii) i ett scenario där URRLC tjänsten tvingas dela resurser med andra tjänster. Antagandet i det första scenariot är att resurstillgången är så god, att påverkan från övriga tjänster på URLLC är försumbar. I detta scenario koncentrerar vi oss på diversitetsteknologier och effektreglering som primära metoder för att öka kommunikationstjänstens tillförlitlighet till priset av redundanta överföringar och överkonsumtion av resurser. En maskininlärningsteknik, s.k.Deep Reinforcement Learning (DRL)) används för att optimerar omsändningsförfarande (ARQ) och sändareffekt i syfte att förbättra tillgänglighet och tillförlitlighet. . Ett djupt Q-nätverk med en s.k. ”Branching Soft Actor-Critic” (BSAC) teknik, har använts för att lösa skalbarhetsproblem. Vår metod utvärderas med en 3GPP-kompatibel simulator för fabriksautomatiseringsscenarier. Vi kan visa påtagliga vinster vad avser våra prestandamått men även var gäller beräkningstid och minneskrav i jämförelse med den tidigare arbeten. Den skalbara DRL metoden resulterar i betydligt lägre total energiförbrukning och spektrumbehov för att uppnå uppsatta prestandamål. Metoden erbjuder även viktiga insikter i den praktiska implementeringen av DRL lösningar för URLLC i verkliga trådlösa kommunikationssystem.

I det andra scenariot undersöker vi hur tjänster som konkurrerar med URRLC-tjänsten om resurserna, påverkar prestanda. Vi fokuserar på ett fall där ett system för distribuerad maskininlärning är en samexisterande tjänst. Den befintliga tjänsten “5G-NR quality of service” utnyttjas här för att prioritera URLLC-trafik. Vi utvecklar modeller för att karakterisera den trafik som arbetsflödet för maskininlärning (t.ex. träning av fördröjning, modellstorlek, konvergens och noggrannhet) ger upphov till, för att kunna analysera samspelet mellan denna trafik och URLLC-tillgängligheten. Eftersom systemet har begränsade resurser (när det gäller bandbredd och överföringskraft) och tillgänglighetskraven för URLLC är strikta, måste maskinlärningen stå tillbaka och endast låta en delmängd av sina enheter delta i varje iteration av den distribuerade träningen. Vi formulerar ett optimeringsproblem som minimerar den genomsnittliga träningstiden för maskinlärningstjänsten i syfte att uppnå en viss noggrannhet utan att kompromissa med URLLC -tillgängligheten. Vi omvandlar detta samexistensproblem till ett DRL problem och använder åter vår BSAC metod för att angripas problemet. Realistiska simuleringar av scenarier för fabriksautomation har använts för utvärdering av vår metod. Resultatet är en väsentlig minskning av träningstiden inom den distribuerade maskinlärningstjänsten samtidigt som den bibehåller URLLC-tillgängligheten över det krävda tröskelvärdet. Vår metod åstadkommer en högre prestandanivå för URLLC-tillgängligheten jämfört med de konventionella statiska “slicing” metoderna och prestanda snarlika med det första scenariot, där samtliga kommunikationsresurser var dedikerade till URLLC-tjänsten, uppnås. För att övervinna begränsningar i den statiska “slicing” metoden och åstadkomma mera flexibla, skalbara och lönsamma nätverk, föreslår vi en implementering av en hierarkisk RL-metod som dynamisk allokerar kommunikation och beräkningsresurser till tjänster över en större tidskala.

List of Publications

This doctoral dissertation consists of two parts. The first part provides an overview of the research field that I focused on during my Ph.D. studies, along with a brief summary of my contributions to it. The second part comprises the contributions that I have made to this field through published or submitted papers as well as patent applications, as follows:

- **Paper 1:** M. Ganjalizadeh, A. Alabbasi, J. Sachs, and M. Petrova, "Translating Cyber-Physical Control Application Requirements to Network level Parameters," *in IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2020.

- **Paper 2:** M. Ganjalizadeh, P. Di Marco, J. Kronander, J. Sachs, and M. Petrova, "Impact of Correlated Failures in 5G Dual Connectivity Architectures for URLLC Applications," *in IEEE Globecom Workshops (GC Wkshps)*, 2019.

- **Paper 3:** M. Ganjalizadeh, A. Alabbasi, A. Azari, H. S. Ghadikolaei, and M. Petrova, "An RL-based Joint Diversity and Power Control Optimization for Reliable Factory Automation," *in IEEE Global Communications Conference (GLOBECOM)*, 2021.

- **Paper 4:** M. Ganjalizadeh, H. S. Ghadikolaei, A. Azari, A. Alabbasi, and M. Petrova, "Saving Energy and Spectrum in Enabling URLLC Services: A Scalable RL Solution," in *IEEE Transactions on Industrial Informatics*, 2023.

- **Paper 5:** J. R. Baldvinsson, M. Ganjalizadeh, A. Alabbasi, M. Björkman, and A. H. Payberah, "IL-GAN: Rare Sample Generation via Incremental Learning in GANs," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.

- **Paper 6:** M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Interplay between Distributed AI Workflow and URLLC," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.

- **Paper 7:** M. Ganjalizadeh, H. S. Ghadikolaei, D. Gündüz, and M. Petrova, "Device Selection for the Coexistence of URLLC and Distributed Learning Services," *submitted to IEEE Internet of Things Journal*, 2023.

- **Patent 1:** M. Ganjalizadeh, A. Alabbasi, J. A. Ibanez, M. Petrova, and J. Zander, "Methods of Translating Cyber-Physical Control Application Requirements to Radio Parameters," US Patent Application 20220294697A1, Filed Oct. 29, 2020. Available: `https://patents.google.com/patent/US20220294697A1`.

- **Patent 2:** M. Ganjalizadeh, A. Azari, H. S. Ghadikolaei, A. Alabbasi, and M. Petrova, "Methods and Apparatus for Controlling One or More Transmission Parameters Used by a Wireless Communication Network for a Population of Devices Comprising a Cyber-physical System," WO Patent Application WO2022250604A1, Filed May 27, 2021. Available: `https://patents.google.com/patent/WO2022250604A1`.

- **Patent 3:** M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Systems and Methods for Joint Inter- and Intra-slice Orchestration Using Reinforcement Learning," PCT Patent Application PCT/SE2023/050016, Filed Jan. 12, 2023.

In addition to the work included in this thesis, I co-authored the following conference contributions during my Ph.D. studies, which are also relevant to my research:

- **Paper 8:** A. Alabbasi, M. Ganjalizadeh, K. Vandikas, and M. Petrova, "On Cascaded Federated Learning for Multi-tier Predictive Models," *in IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021.

- **Paper 9:** W. Shi, M. Ganjalizadeh, H. S. Ghadikolaei, and M. Petrova, "Communication-Efficient Orchestrations for URLLC Service via Hierarchical Reinforcement Learning," *submitted to IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023.

Acknowledgements

Returning to academia after several years in the industrial sector has been a profoundly enriching journey. For that, I wish to express my sincere gratitude to my main supervisor, Prof. Marina Petrova, whose guidance and support have been instrumental in the success of this endeavor. My co-supervisor from KTH, Prof. Jens Zander, deserves special mention for his priceless insights, helpful comments, and inspiring attitude, for which I am truly grateful. I would also like to voice my deep appreciation to my manager, Dr. Jonas Kronander, who entrusted me with this position and provided persistent support during the challenging times I encountered throughout this industrial Ph.D. project. Besides, I would like to thank Dr. Mikael Prytz, a distinguished member of this project's steering board, for his valuable comments. I would like to express my sincere appreciation to my industrial supervisors during these years, Dr. Abdulrahman Alabbasi, Dr. Joachim Sachs, Asst. Prof. Piergiuseppe Di Marco for their invaluable contributions. And finally, I would like to extend my genuine gratitude to my industrial supervisor and dear friend, Dr. Hossein S. Ghadikolaei, who joined when the confusion was at an all-time high. Since then, his remarkable insights, innovative ideas, and boundless enthusiasm have been a constant source of inspiration and support.

I would like to express my gratitude to Prof. Petar Popovski for accepting the opponent role in my defense. I extend that appreciation to the members of the grading committee, Prof. Liesbet Van der Perre, Prof. Mikael Skoglund, and Dr. Erik Dahlman. Besides, I would like to thank Prof. James Gross for reviewing my Ph.D. proposal, Prof. György Dán for reviewing my doctoral thesis, and Prof. Emil Björnson for serving as the defense chair.

I would like to thank Dr. Amin Azari, Johan Haraldson, Prof. Deniz Gündüz, and Assoc. Prof. Amir H. Payberah for our fruitful discussions and collaborations. I would like to express my appreciation to all of my colleagues at KTH and Ericsson, including my amazing team at Ericsson. Special recognition goes to Morteza (who happens to be the best office neighbor one could ask for), Sara, Meysam, Mustafa, Zinat, Dhruvin, Yasaman, Irshad, Ozan, Afsaneh, Anders, and Yanpeng. I am also indebted to all my friends in Sweden, Hamid, Navid, Forough, Neda, Amir, Mehdi, Elham, Marije, Shekhar, Farshin, Javid, Mazdak, Omid, Kasia, and many others.

I would like to thank my beloved family. Mother, your love and support have given me strength and confidence. I am forever grateful. Father, your presence is dearly missed. To my sister, Morvarid, and those friends who have stood by me, even from a distance, I am immensely grateful. In particular, I would like to extend special appreciation to Ali, Pouya, and Jamal, your enduring friendships have transcended miles and time zones. Finally, I cannot end without expressing my most profound appreciation to Azadeh for her endless support, patience, and love. Your contribution to my life and this journey is truly immeasurable.

Milad Ganjalizadeh
Stockholm, June 2023

The research leading to this doctoral dissertation has received funding from the Swedish Foundation for Strategic Research (SSF).

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	3rd Generation Partnership Project
5G	fifth generation of mobile communication systems
5G-ACIA	5G alliance for connected industries and automation
AI	artificial intelligence
AP	access point
B5G	beyond 5G
BSAC	branching SAC
CAN	controller area network
CPS	cyber-physical system
DDPG	deep deterministic policy gradient
DGD	distributed gradient descent
DL	downlink
DNN	deep neural network
DQN	deep Q-networks
DRL	deep RL
eCDF	empirical cumulative density function
eMBB	enhanced mobile broadband
EtherCAT	Ethernet for control automation technology
FL	federated learning
FSMC	finite state Markov chain
GAN	generative adversarial network
gNB	gNodeB
HARQ	hybrid automatic repeat request

HART	highway addressable remote transducer
HL-RLO	high-level RL orchestrator
HRL	hierarchical RL
HRQ	high-level research question
ICT	information and communication technology
IL-GAN	incremental learning GAN
InF	indoor factory
InF-DH	InF-with dense clutter and high base station height
ISM	industrial, scientific and medical
JSD	Jensen-Shannon divergence
KPI	key performance indicator
LL-RLO	low-level RL orchestrator
LR-WPAN	low rate wireless personal area network
LTE	Long-Term Evolution
MAC	medium access control
MDP	Markov decision process
ML	machine learning
mMTC	massive machine type communication
NGMN	next generation mobile networks
NN	neural network
NR	new radio
NSMF	network slice management function
NSSMF	network slice subnet management function
OFDM	orthogonal frequency-division multiplexing
OT	operational technology
PDCP	packet data convergence protocol
PER	packet error ratio
PHY	physical
PRB	physical resource block
PROFIBUS	process fieldbus
PROFINET	process field network
QoS	quality of service
RAN	radio access network

RL	reinforcement learning
RLC	radio link control
RQ	research question
SAC	soft actor-critic
SIG	special interest group
SINR	signal to noise and interference ratio
TTI	transmission time interval
UL	uplink
URLLC	ultra-reliable low-latency communications

Chapter 1

Introduction

Nowadays, digital devices are everywhere and have become an integral part of our daily lives. As the demand for more advanced, efficient, and affordable technologies continues to grow, new opportunities are emerging to revolutionize the way we interact with the world around us. With the advent of *cyber-physical systems (CPSs)*, we are taking the next step towards integrating technology into our physical world, creating an interconnected network of devices that have the ability to provide real-time interactions and physical awareness, with the potential to transform our daily experiences in ways yet to be imagined.

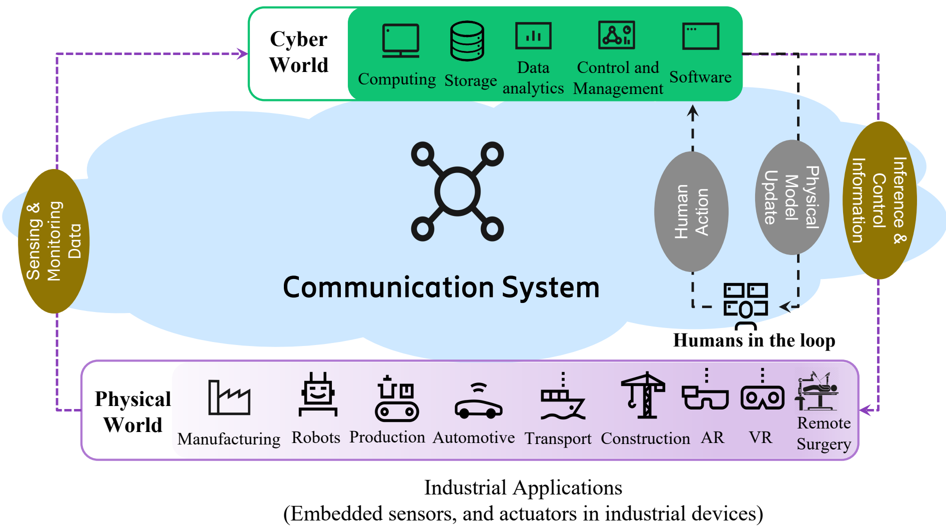


Figure 1.1: Cyber-physical systems.

CPSs, as illustrated in Figure 1.1, are engineered systems[1] in which computation and communication are integrated to control the physical processes, and sometimes, humans are also in the loop. In other words, in CPSs, mechanical and electrical systems (e.g., computers, actuators, and sensors, embedded in products and materials) are networked using software components to collaboratively share information and data from processes to manage logistics and production systems independently. Thus, CPSs surpass the simple control of individual devices to the level of intertwined objects that communicate information about a distinct condition in order to achieve a particular objective with higher accuracy and efficiency. Unlike embedded systems, which are information processing systems integrated into products, the CPSs are a network of interacting devices with physical functions [1]. Therefore, typical CPSs are sensor-based, communication-enabled, autonomous systems [2]. For example, today's cars consist of tens of electronic control units, computer components, and sensors interconnected via hundreds of meters of wiring, forming several networks. Such collection intends to enable several functionalities (e.g., engine control, brake control, and park assist features), thus providing a safer and more efficient driving experience. Another example is the manufacturing industry, which is increasingly caught between the rising demand for customized, high-quality products and the drive to maximize profit margins [3]. Many cyber-physical control applications require the cyber model to interact with humans by which the outcome of the CPS's functions is affected, for instance, in augmented reality, virtual reality, remote surgery, or autonomous driving [4].

1.1 Motivation and Scope

Most of the CPSs fall under the classification of safety-critical systems, implying that faults leading to incorrect service operation may result in significant financial losses or even jeopardize people's lives [5]. Despite this, various types of faults can occur in different parts of a CPS, which may potentially lead to system failures. In the cyber domain, common faults include hardware and software faults, such as random hardware faults or software bugs, which can result in timing errors of real-time applications and operating systems or crash failures. In the physical domain, faults can occur in actuator functions and sensor measurements, and unexpected circumstances in the environment in which CPS operates, such as fire or earthquake, can lead to various errors in different functionalities. Communication faults, which may eventually lead to link failures, lie between these two domains. Since CPSs involve real-time control of physical processes via actuators, message loss, and network-induced delay are typical causes of communication system failures. Network delay affects performance from two perspectives: (i) missing the deadline of hard real-time messages, which has a similar impact as packet loss (e.g., delays in actuator functions), and (ii) delayed control loop, resulting in decision-making based on an

[1] Throughout this thesis, we use the term "system", similar to ISO/IEC/IEEE 15288, as a combination of interconnected elements organized to accomplish one or more stated functions.

outdated physical state, which can cause degradation in control performance and stability [6, 7].

International electrotechnical commission specifies the required failure rate for different safety integrity levels in its guidelines, intending to determine the target reliability for specific safety functions [8]. For example, in control systems, international electrotechnical commission demands the expectation of dangerous failures per hour to be lower than 10^{-5}, 10^{-6}, 10^{-7}, and 10^{-8} for safety integrity levels of 1, 2, 3, and 4, respectively. Moreover, the expected failure rate of industrial devices is often included in their datasheet or catalog (e.g., in [9]). Assuming that failures occur independently across different domains, all cyber, physical, and communication domains should fulfill the overall reliability criteria; otherwise, they become the reliability bottleneck [10].

Although reliability engineering and its associated concepts (such as availability, dependability, and time to failure) have been originally developed for operational technology[2], its available toolset has been little exploited in the information and communication technology (ICT) world. This thesis contributes to filling this gap between OT and ICT by focusing on ultra-reliable wireless communications.

1.1.1 Thesis Scope

In this thesis, our primary objective is to provide ultra-reliable timely communication services for CPSs. To accomplish this objective, the central focus of the thesis is to *study the automation metrics of CPSs with respect to wireless systems and to develop practical methods to optimize wireless systems so they can meet these metrics' requirements efficiently (in terms of spectrum and energy).* Therefore, the scope of this thesis is summarized with the following high-level research questions (HRQs):

- **HRQ1**: What is the connection between OT's reliability key performance indicators (KPIs) with well-known communication system measures, such as packet error ratio (PER)?
- **HRQ2**: How to enable ultra-reliable low-latency communications (URLLC) with a range of strict requirements so that it brings benefits in the form of more reliable and efficient CPSs?

This thesis describes the procedures and methodology for solutions that can address these HRQs and contribute to the answers to these questions, ultimately leading towards the creation of resource-efficient and intelligent CPSs. Throughout this thesis, we assume that the failures in the cyber, physical, and communication domains are statistically independent. Besides, in the majority of this thesis, we assume that the wireless communication system is the bottleneck in the reliability

[2] Operational technology (OT) refers to the hardware and software systems responsible for controlling industrial operations. Examples of OT industry include industrial automation companies, machine builders, and production system manufacturers [11].

chain. Nevertheless, addressing both HRQs, we provide brief investigations on the impact of failures in the cyber domain.

1.2 Communication Service for CPS

The common emphasis on the physical process, communication, and computation in CPS is a crucial enabler for the development of future technologies [12]. From the communication perspective, the interconnection of the cyber and physical worlds puts stringent requirements on the observed quality of service (QoS) in terms of reliability and latency. For the remainder of this section, we discuss the advantages and disadvantages of communication technologies that facilitate cyber-physical control applications.

1.2.1 Wired Communication Limitations

Traditionally, CPS applications have been implemented using wired channels where the communication service is highly predictable. Many technologies have been developed over time, starting with the $4-20\,\mathrm{mA}$ current loop introduced in the 1950s, followed by traditional fieldbus-based protocols [13] (e.g., highway addressable remote transducer (HART), controller area network (CAN), and process fieldbus (PROFIBUS)), and more recently, real-time Ethernet-based solutions [14,15] (e.g., Ethernet for control automation technology (EtherCAT), and process field network (PROFINET)), with each targeting different application domains. While these solutions achieve different performance levels in terms of latency and reliability, they all provide some level of determinism in the communication system. However, wired communication has its inherent limitations and disadvantages, which include:

- *Mobility:* The use of cables in a communication system imposes restrictions on the mobility of devices, making it challenging to support various cyber-physical control applications involving mobile subsystems. For example, wired communications cannot be used for mobile robots, autonomous vehicles, transportation, and many other applications that require dynamic or unrestricted motion [16].

- *Scalability:* Often, control systems go through enlargement and retrofits. Thus, the network is usually obliged to be expanded, and such expansion mandates new wiring and possibly extra switches and access points (APs) in wired networks. This incurs not only additional equipment costs but also labor and maintenance expenses.

- *Small service area:* The maximum length of an Ethernet cable is around 100 meters. Although optimal repeaters and fiber optics can expand the operational distance, there remains a maximum distance beyond which data loss or severe delays in data transfer occur.

- *Long term reliability:* In many cyber-physical control applications, vibration, dust, oil, and high temperatures create a harsh and demanding physical environment in which wires are subject to breaks and aging [17].

1.2.2 Wireless Communication Prospects

Enabling CPSs applications using wireless communications offers both unique opportunities and fundamental challenges. While wireless communications provide industrial networks with flexibility and can overcome some of the limitations of wired networks, as discussed in Section 1.2.1, they also present numerous challenges for high-performance control systems. The PER in a wireless system is essentially determined by its signal to noise and interference ratio (SINR), which is a random process due to shadowing, fading, and the presence of interferers. In addition, since a failed transmission is typically retransmitted, SINR can also impact the delay. Therefore, even slight changes in the propagation environment can greatly affect the SINR, and subsequently, the PER and delay. Moreover, industrial environments present additional challenges to the underlying wireless communication system due to electromagnetic interference, mechanical stress, and unfavorable materials [18].

In recent years, the industrial communications community has conducted extensive research to enhance the timeliness, determinism, and reliability of wireless technology. We classify the research on wireless communication technologies for cyber-physical networks into non-cellular and cellular technologies. In the following subsections, we describe the recent advancements in both technologies and explain why we selected the latter as the baseline for our contributions.

1.2.2.1 Non-cellular Technologies

In the following, we describe some well-established and recent non-cellular wireless standards and protocols in order to determine their applicability to support communication service for CPS. We classify these standards and protocols into three main technologies as Bluetooth, low rate wireless personal area network (LR-WPAN), and Wi-Fi.

Bluetooth:

Bluetooth, originally invented by Ericsson, has been standardized and maintained by Bluetooth special interest group (SIG) since 1998[3]. Bluetooth technology was initially designed for short-range, low-power machine-to-machine communications and operates in the unlicensed 2.4 GHz industrial, scientific and medical (ISM) band. The development of Bluetooth Low Energy (BLE) in Bluetooth 4, released in 2010, made this technology applicable to industrial networks where wireless devices have power constraints [20]. Furthermore, the enhancements in Bluetooth 5

[3] IEEE also standardized Bluetooth via IEEE 802.15.1 [19], but the standard is no longer maintained by IEEE.

increased throughput, reduced latency, quadrupled the communication range, and enabled support for mesh networks [21–23]. Although Bluetooth is known for providing low-cost and low-energy solutions, it has been investigated for use in real-time communications [24, 25]. Nevertheless, Bluetooth is inherently a best-effort technology, and although a single-hop communication can achieve latency in the millisecond range, the high-percentile latency over a mesh network can grow to hundreds of milliseconds [26]. Besides, achieving high reliability with Bluetooth can be challenging [27], which makes it unfit for use in industrial control applications.

LR-WPAN:

LR-WPAN is based on IEEE 802.15.4 standards [28] and operates on one of three ISM bands: 868 MHz, 914 MHz, or 2.4 GHz, with up to 1, 2, and 16 channels, respectively [29]. LR-WPAN-based protocols and solutions are widely used in today's industrial networks. For instance, WirelessHART, a wireless version of the wired HART protocol, is a popular choice for industrial control and measurement applications [30]. By implementing a time-synchronized channel hopping protocol on the medium access control (MAC) and networking layers, WirelessHART achieves high reliability through time, frequency, and spatial diversity gains [31]. However, it does not prioritize meeting strict delay requirements [30]. Other LR-WPAN solutions, such as ISA100.11a [32] and 6LoWPAN, enable the use of IP protocols on top of IEEE 802.15.4 [33]. While 6LoWPAN can fulfill latency bounds in the order of a few milliseconds, it may come at the cost of transmission reliability [30].

Wi-Fi:

Wi-Fi is based on IEEE 802.11 standards, with various versions including 802.11 (b/a/g $\cdots$ ax/ay/be), and it operates on one or more of the 2.4 GHz, 5 GHz, and/or 6 GHz ISM frequency bands. Its primary goal was to provide a high-throughput connection to devices located within a short range, leveraging its design for high-bandwidth communication. The availability of such large bandwidth increases the potential for supporting scalable networks. To provide a high data rate over short distances, the network is typically built around a central node, known as the AP [34]. Nevertheless, IEEE 802.11 also allows for the formation of ad-hoc networks without the need for an AP.

In 802.11n, the addition of low-density parity-check codes and multiple-input and multiple-output technology improved the reliability of Wi-Fi networks [35]. The 802.11ax standard introduced orthogonal frequency-division multiple access in both the uplink (UL) and downlink (DL), allowing for simultaneous communication with multiple devices and resulting in a multi-user aggregation gain compared to sequential transmissions in preceding generations. In 802.11ax, the channel is divided into resource units, where each resource unit corresponds to a frequency block with at least 26 orthogonal frequency-division multiplexing (OFDM) subcarriers. This segmentation reduces contention within the service area of one AP.

As the subsequent generation of 802.11ax, 802.11be was expected to include features for extremely high throughput, reduced latency, and improved reliability [36], but only features related to extremely high throughput were included in the standard due to difficulties in meeting the proposed timeline [37]. However, several of these features could benefit CPS applications, such as 320 MHz wide channels, multi-resource unit support [38], and multi-link operations [39]. The ultra-high reliability features are planned for the next study group and are expected to be released in 2027 [40]. The proposed features are envisioned to include improvements to multiple-input and multiple-output technology to support up to 16 spatial streams, hybrid automatic repeat request (HARQ), and multi-AP coordination (for coordinated transmissions) [38, 41]. Additionally, ongoing efforts in the 802.11bf task group are aimed at enabling sensing capabilities via 802.11-compliant waveforms [42]. However, high multipath interference in Wi-Fi networks is a concern, particularly in indoor industrial environments where there may be extreme reflections from walls and other obstacles. Moreover, as the number of devices increases, the SINR in Wi-Fi networks can drop significantly [34].

In addition to standardization activities, there are also tailored solutions built on top of 802.11 standards to address the issues of high multipath interference and low SINR in industrial environments. One example is EchoRing [18], a novel token-passing MAC protocol built on top of the physical layer of 802.11. EchoRing assures medium access to distributed devices and establishes a cooperative relaying scheme to enhance reliability and reduce latency through spatial diversity. Although high levels of determinism may be achieved using this token-passing technique, EchoRing is susceptible to unexpected problems that may arise if the token is lost [17]. Additionally, the delay may increase as the number of industrial devices increases excessively.

1.2.2.2 Cellular Technologies

Nowadays, the communication industry is rapidly advancing towards fifth generation of mobile communication systems (5G) and beyond 5G (B5G) to meet the ever-increasing demands on high data rates, energy efficiency, and uncompromising QoS [43–46]. Compared to earlier mobile communication systems, 5G and B5G are supposed to provide services to a much broader range of application domains and use cases. In addition to traditional human-centric communication for *enhanced mobile broadband (eMBB)* services (e.g., mobile Internet), 5G and B5G are envisioned to enable machine-type communications in a mixture of industrial and societal domains. One service is so-called *massive machine type communication (mMTC)* with an emphasis on low power and low cost wireless connectivity for an enormous number of connected sensor devices. Another service is to support critical machine type communications via *URLLC* [47]. Therefore, URLLC is designed to fulfill strict latency and reliability requirements, and hence, it has the potential to assist cyber-physical control applications [48, 49].

The following key concepts in the new radio (NR) interface of 5G's radio access network (RAN), that has been introduced by 3rd Generation Partnership Project (3GPP) over time, opened doors to supporting URLLC services:

- *Short transmission time interval (TTI):* Unlike previous generations that had a static slot length of 1 ms, 5G allows for various configurations of slot length, referred to as numerology, with a minimum length of 62.5 μs. This was first introduced in Release 15, along with the ability to transmit over a fraction of a slot. Besides, the NR standard enables the subdivision of a slot into smaller units called sub-slots [50].
- *Configured grant:* In Long-Term Evolution (LTE), Release 14 expanded the semi-persistent scheduling framework, enabling devices to access pre-configured grants periodically with a minimum periodicity of one slot. This eliminates the need for devices to request a grant for each UL transmission through the control plane (i.e., physical downlink control channel) [51][4].
- *Faster processing time:* With the advancements in hardware, the processing times of both the devices and the network have become increasingly demanding. For instance, in NR, the device's acknowledgment is required within one slot after receiving a DL transmission. In the MAC and radio link control (RLC) layers, unlike LTE, processing can commence without knowledge of the amount of data to be transmitted. This is especially critical for uplink transmissions, as the device may have only a few OFDM symbols between the reception of the uplink grant and the scheduled transmission time [52].
- *Especial care for QoS handling:* The service data adaptation protocol is a new layer in NR, first introduced in Release 15, and is specified to be on top of the packet data convergence protocol (PDCP) layer. This layer is responsible for mapping between the QoS flows of the 5G core and the data radio bearers [53].

According to Ericsson's measurements in [54], compared to the Release 13 baseline with a scheduling periodicity of 1 ms, utilizing a configured grant and a shorter slot length can result in a 34% and 67% improvement in the 80th percentile round-trip ping delay, respectively. Moreover, by combining both features, the 80th percentile round-trip delay can be reduced by more than 16 ms, leading to an improvement of 77%, from 21 ms to less than 5 ms. Furthermore, the results reported in [55] suggest that the improved processing capabilities and sub-slot functionality available in NR can lead to a significant reduction in DL delay, potentially reducing it by up to 50%.

In the 5G core network, the architecture is service-based, implying that the specification concentrates on the services and functionalities provided by the core network instead of the nodes as such [52]. One of the benefits of such architecture is network *slicing*, where a slice is a logically isolated and independent part of the network, targeting distinct services with different requirements on e.g., latency, reliability, throughput, power consumption [56].

[4] In Release 14, this feature is known as "Instant UL Access" or "Fast UL Access"

Moreover, to facilitate URLLC, various diversity techniques are proposed by 3GPP, including [57, 58]:

- repetitions on the physical (PHY) layer to benefit from the time and frequency diversity [58];
- HARQ retransmissions on the MAC layer [59] and retransmissions of erroneous data on RLC layer [52] to leverage time and frequency diversity; and
- packet duplication on the PDCP layer (using either multi-connectivity to benefit from spatial diversity or carrier aggregation to leverage frequency diversity) [60].

1.2.3 Why 5G and beyond as cyber-physical networks?

A number of characteristics make 5G/B5G especially appealing for the underlying communication systems of CPS, which include the following [61]:

- *High bandwidth:* If managed properly, the high available channel bandwidth in the various 5G frequency bands allows for scalable URLLC solutions.
- *Multi-service support:* 5G provides a unified wireless interface to support heterogeneous services with a variety of requirements. The combination of network slicing in 5G core network, coupled with the introduction of service data adaptation protocol layer in RAN, brings the potential for the support of heterogeneous services in 5G networks.
- *Mobility*: 5G offers built-in functionality for dealing with mobility for various CPS applications.
- *Security:* 5G brings to the industrial sector a tried-and-true security solution that has already been used in cellular networks across the world.
- *Positioning*: It has garnered much attention in earlier generations of cellular technology and continues to receive attention in 5G, particularly in Release 16 and Release 17. With the improvements introduced in Release 17, 5G can attain high location accuracy, reaching decimeter and meter range for indoor and outdoor scenarios, respectively [62–64]. Many emerging CPS applications will benefit from such accurate positioning capabilities.

1.3 Literature Study

1.3.1 Reliability Metrics in Information and Operational Technology

In the early 1930s, Walter Shewhart, Harold F. Dodge, and Harry G. Romig established the foundation for using statistical methods in industrial products' quality control. However, these methods had not been widely adopted until the start of World War II and the development of missile systems. Following World War II, the use of statistical methods in quality control expanded globally. As automation became more prevalent, the demand for advanced control and safety systems increased [65]. Hence, reliability KPIs are well-established concepts in the OT domain. In the following list, we present a summary of the reliability KPIs that we used from OT perspective, considering repairable items[5].

- *Availability* is the ability of an item to fulfill its designated service as expected (i.e., without failure) at a specific instant of time, and is typically represented in the form of probability. Availability can also be expressed as the proportion of time an item is functioning within a defined time period[6] or the percentage of components within a system operating at a given time [65–68].

- *Reliability* is the ability of an item to carry out its designated service, given specific environmental and operational conditions, over a specified time period. It is represented either as a probability that the system operates without failures in a time interval or as the mean duration of time the item operates without failure (also known as mean uptime and mean time to failure) [65–68].

In the ICT community, we recognize that a broad range of KPIs are used to evaluate the reliability of communication systems, and the terms "reliability" and "availability" are frequently used interchangeably [69]. Reliability metrics are traditionally defined on the PHY layer, and indicators, such as outage probability, bit error ratio, or block error ratio, are used to identify the QoS of the channel, as in [70]. In general, the percentage of successful transmissions over the total transmissions seems to be widely used as an indicator of the observed reliability on PHY, MAC, or higher layers. However, in URLLC applications, there is naturally no difference between not receiving a packet and receiving it after its corresponding delay bound [71, 72]. In addition, errors often occur in bursts in wireless communications, and such metrics cannot capture the impact of temporal correlation on reliability performance. For instance, the analysis in [73] shows that the impact of space-time correlation on end-to-end performance is considerable. On top of what has been discussed, and according to 3GPP specifications in [71], sparse packet loss can be tolerated by modern cyber-physical control applications. Such applications typically specify a short period of time, called *survival time*, during which the application can tolerate or even correct the failures in the communication service.

As evident from the definitions of availability and reliability in OT domain, defining a "failure" correctly is crucial to have proper definitions in ICT. In the next section, we present a brief overview of the definitions of failure, error, and fault in both OT and ICT domains.

[5]The term "item" here refers to any component, subsystem, or system that can be considered as a distinct entity [65].

[6]In this case, it can be referred to as steady-state [66] or average availability [65].

1.3.2 Failures, Errors, and Faults

The terms "Failure", "Error", and "Fault" have been well-established in the literature on OT (e.g., [65, 66, 74, 75]). In this context, a system is said to have experienced a *failure* if it is unable to deliver its expected service as per the system's specification. An observed condition or measured value deviating from the theoretically correct or specified value or condition is known as an *error*. Finally, a *fault* is a flaw in the system or its representation that may result in an error if it is executed or triggered [76]. For instance, in software engineering, a software bug is a fault that could lead to an error when the subroutine containing that bug is called. Assuming that this software contributes to the overall system's service, a failure could occur if the service performance deviates from its specification due to this error.

In the realm of ICT, the definition of failure is typically established either at the point-to-point connection level or at the system level, depending on the service specifications. In the case of URLLC service[7], a link failure is defined as the inability to transmit data in the UL and/or DL directions. In studies conducted at the PHY layer, a threshold is often employed, typically based on power level or signal amplitude. References [78, 79] provide further insights into this approach. However, this definition fails to consider the autocorrelation and time dependency of time-varying wireless channels and does not account for the impact of higher-layer mechanisms that can mitigate short outages. To address these issues, recent works such as [77, 80–82] assume a link failure when the received power stays below a given target threshold for a duration exceeding a specified limit. Yet, in multi-user wireless systems, bit error ratio is a function of SINR. Moreover, in URLLC service, a successful reception after a pre-specified delay is deemed as packet loss [71, 72]. As an example, reference [83] defines a link failure as the SINR dropping below a threshold or the failure to meet the delay bound. This definition does not consider higher-layer mechanisms that enhance reliability, such as retransmissions, interface diversity, or repetitions. To overcome this limitation, recent studies, including [84], have considered the impact of interface diversity on reducing failure rates. Nevertheless, in the context of CPS communication, the survival time needs to be considered too [71]. It is important to note that survival time is defined on the reception node and is different from delay bound. The latter is defined per message, whereas the former represents the maximum time that the reception entity remains uninformed. Several recent works, such as [85–87], have taken survival time into account.

1.3.3 Slice Management Architecture

As previously discussed, 5G and B5G aim to support a mixture of heterogeneous use cases, each of which imposes a broad range of performance requirements. Net-

[7]Note that failures may be defined differently for other services, e.g., as a function of throughput for eMBB service [77].

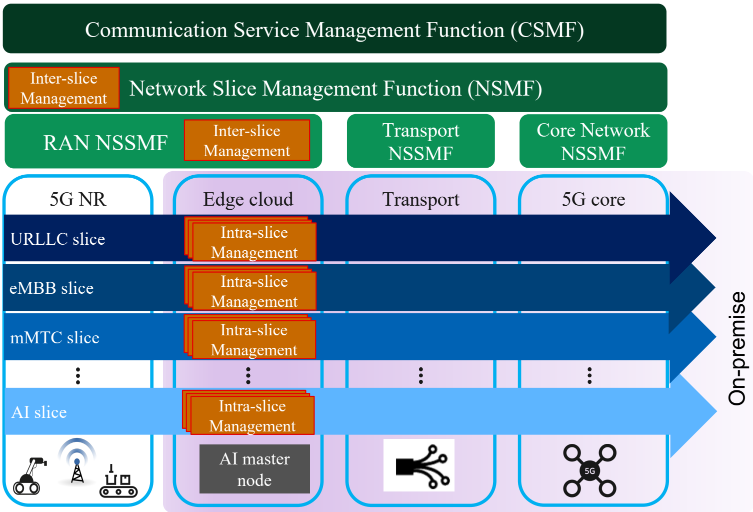


Figure 1.2: The proposed slice management architecture (adapted from figures in [88, 168]).

work slicing is regarded as a key enabler in supporting these vast and sometimes conflicting requirements. Figure 1.2 depicts the proposed slice management architecture to serve as the backbone for this functionality [88, 89]. In this architecture, an intra-domain network slice subnet management function (NSSMF) is considered to provide management services for one or more network slice subnets. This domain-level management function (i.e., NSSMF) is governed by the end-to-end network slice management function (NSMF). In this architecture, orchestrations within a slice (e.g., feature orchestration in [5], or resource allocation in [90])) can be implemented as a network function virtualization orchestrator within its corresponding slice. Besides, as the needs of enterprises evolve, new and unique use cases are expected to emerge. In certain cases, the combination of slice KPIs may be unique, while in others, various services could have common KPIs but differ in their target values. As such, we envision that NSMF maintains a repository of slice templates for common use cases to facilitate the sharing and quick deployment of slices that have similarities [88, 91].

From the network slicing perspective, performance optimization techniques can be addressed from two standpoints: intra-slice and inter-slice management techniques [92]. In intra-slice orchestration techniques, the goal is to achieve the desired service (or services) intents given the allocated resources to the corresponding slice. This includes techniques such as resource allocation for different devices [90], QoS management [93], resource management and feature orchestrations [94] within a single slice. Inter-slice orchestration techniques focus on coordinating and managing different slices within the network. This includes techniques such as resource allocation to different slices [95] and slice selection for different services [96]. Both inter-slice and intra-slice management techniques play a crucial role in ensuring that 5G networks can support the diverse range of use cases, including URLLC, eMBB, and mMTC, and can provide the required level of service for each use case.

Currently, the state-of-the-art focus for enabling URLLC services can be classified into two distinct categories: i) single URLLC slice setting and ii) mixed services setting. For the first category, the focus lies in meeting URLLC's strict requirements through intra-slice orchestrations, given its allocated resources, such as bandwidth, maximum output power, and processing capabilities. This implies that resources are dedicated to URLLC, and the performance is not impacted by other nearby services. In the mixed services setting, URLLC services operate alongside other services within the same network. There are two commonly addressed approaches in the literature for this scenario. The first approach involves sharing resources among heterogeneous services and managing QoS through intra-slice management. The second approach involves having separate slices for URLLC and other coexisting services, leading to full resource separation and inter-slice management to fulfill various service intents. In the rest of this section, focusing on RAN, we conduct a literature review of techniques that address URLLC in both settings.

1.3.4 Enabling URLLC in Single Service Scenarios

To enable URLLC for CPS, one straightforward solution is to allocate excessive resources, in terms of bandwidth and transmission power, to a URLLC slice. However, such an approach is neither spectral nor energy efficient [97]. Another solution is to leverage *reliability enhancement techniques* such as interference management and power control [98] as well as diversity techniques [57], discussed in Section 1.2.2.2. As the wireless channel behaves differently at different points in time, space, and frequency, *diversity techniques* exploit such characteristics to mitigate fading effects by sending the same data at a different time or via distinct paths or frequencies [99]. Nevertheless, these techniques are either (i) reactive (e.g., MAC or RLC retransmissions which happen as a result of negative acknowledgment or time-out), implying that they are associated with an extra delay that the application might not tolerate [52], or (ii) they are always active (e.g., packet duplication), which could be costly in terms of system capacity and energy consumption [100]. Although these techniques have the prospect of providing high reliability, given the limited available resources in wireless communications, blind overuse of these techniques is neither efficient nor effective. For instance, the full power transmission could lead to high interference on the neighboring users, and multiple diversity transmissions could result in a high load in the cell, potentially increasing packet queuing delays to the level that they miss their corresponding delay bounds. Thus, it is necessary to properly utilize/combine such approaches based on various parameters such as load, channel state, and queue size. There have been several research studies on the optimization of diversity techniques and transmission power. References [59, 101] propose a joint resource allocation and HARQ optimization framework to minimize the required bandwidth subject to PER in single-cell scenarios. In [80], the authors propose a multiple selection combined links scheme for multi-connectivity scenarios to fulfill a given reliability requirement. These works make strict assumptions on queue models (e.g., M/GI/∞ with zero queuing delay in [59] and M/G/1 in [101]), or on the channel (e.g., Rayleigh fading channel in [80]) which often are not realistic in cyber-physical production systems.

To overcome the drawbacks of these model-based approaches, there has been increasing interest in using reinforcement learning (RL) to learn from the data and minimize the use of unrealistic assumptions (e.g., [102–104]). Reference [102] proposes a risk-sensitive RL to dynamically orchestrate repetitions in frequency in order to reduce the PER, measured on PHY layer. In this work, like many others, packet error or packet loss is referred to as an event when the transmitted packet is not received or cannot be decoded in the receiving entity. However, in URLLC applications, including cyber-physical applications, a packet is also considered lost if it is received after its corresponding delay bound [71, 72]. Besides, this paper neglects the impact of temporal correlation on reliability. Moreover, the authors in [102–104] assume per-slot orchestration, implying that the entire pipeline (including data transmission, processing, decision making, and applying the decision) should occur within every time slot, which could be as small as $62.5\,\mu$s [105]. Such intensive real-time interactions for service management and orchestration are not envisioned in the near future [106].

1.3.5 Enabling URLLC in Mixed Services Scenarios

The 5G network has brought forth a new era of communication and services by enabling various types of services, such as URLLC, eMBB, and mMTC to coexist within a single network. However, this coexistence of heterogeneous services introduces the challenge of providing QoS that fulfills the specific requirements of each service.

Services Coexisting with URLLC could affect the URLLC performance, particularly in terms of its operational KPIs, for a variety of reasons. One reason is that devices serving other services may increase interference on URLLC traffic, leading to lower SINR and higher PERs and delay. Additionally, they may consume a significant amount of network resources such as bandwidth and processing power, reducing the amount available for URLLC services and potentially increasing the delay in URLLC transmissions. Furthermore, some services may have conflicting QoS requirements, making it difficult to satisfy both URLLC and non-URLLC services at the same time. For example, a video streaming service may require high bandwidth, while URLLC services require low latency. Thus, effective techniques for coexisting and managing these services are essential for maintaining high-quality URLLC performance. Two essential approaches for addressing the coexistence of heterogeneous services are resource sharing and network slicing[8]. In the following, we outline the latest developments in these two areas from the existing literature.

[8] In the RAN literature, the term "slicing" is used ambiguously. However, in our categorization, we assume that slicing should include all gNodeBs (gNBs), and due to its required signaling with NSSMF, it cannot be performed in every slot.

1.3.5.1 Coexistence via Resource Sharing

The resource-sharing approach allows different services to share the same network resources, such as bandwidth, output power, and processing capability. This approach is accomplished through intra-slice orchestrations, implying that the network's resource allocation and management occur within the same slice. There exists a rich literature when it comes to resource sharing for the coexistence of URLLC with other conventional services such as eMBB and mMTC (e.g., [107–110]). To handle such coexistence within 5G networks, existing standardized protocols in 5G-NR for QoS offer the ability to assign each connected device with one or more QoS flows and data radio bearers. These assignments are determined based on the service requirements in the core network [52].

The general practice in this setting is to perform resource allocation either i) to ensure that service-specific KPIs (e.g., throughput, energy efficiency, and reliability for eMBB, mMTC, and URLLC services, respectively) meet their corresponding requirements efficiently (in terms of spectrum or energy consumption), or ii) to maximize/minimize such KPIs. Yet, there are numerous optimization parameters to be concerned with, such as modulation and coding scheme, user-cell association, physical resource block (PRB) allocation, transmit power, number of diversity transmissions, etc. The survey in [45] provides a comprehensive study on resource allocation algorithms for heterogeneous networks. The performance trade-offs, applications, and enabling technologies for eMBB-URLLC coexistence is discussed in [111]. A short overview of radio resource management techniques for mMTC and eMBB traffic is provided in [112].

Leveraging queueing analysis and system-level simulations, reference [113] showed that dynamic multiplexing of URLLC and eMBB traffic in both time and frequency outperforms static partitioning of bandwidth in terms of overall resource utilization. In [114, 115], heuristic algorithms are proposed to dynamically schedule PRBs based on services' requirements, packet size, and control channel overhead. Reference [116] proposed a 2-D PRB allocation and frame structure selection to maximize energy efficiency given URLLC's latency and eMBB's minimum throughput. References [108–110, 117, 118] studied the dynamic scheduling of URLLC traffic using puncturing based schemes in which the URLLC traffic is immediately scheduled over ongoing eMBB transmissions. While puncturing can improve URLLC latency, it may reduce the transmission rates of eMBB traffic due to frequent losses of spectrum resources to URLLC traffic [119]. Alternatively, references [120–123], on top of PRB and power allocation, proposed a hybrid puncturing and superposition approach in which both eMBB and URLLC traffic are superposed at the gNB and decoded using successive interference cancellation at the devices. The performance of URLLC is highly susceptible to inter-cell interference [124]. Such vulnerability is particularly crucial when the coexisting service generates a high load, and if not managed properly, it could potentially lead to compromising energy and spectrum efficiency to meet the stringent URLLC requirements [125, 126]. For instance, reference [127] proposed a coordinated multi-point-based subcarrier and power allocation algorithm to handle the impact of inter-cell interference in URLLC-eMBB coexistence.

1.3.5.2 Coexistence via Network Slicing

Network slicing is a technique used in 5G and B5G mobile networks to divide a physical network infrastructure into multiple virtual networks, each of which can be customized to support specific types of services and traffic patterns with guaranteed QoS levels [56]. Reference [56] provides a survey of 5G network slicing, including its driving forces, key enabling technologies, the related 3GPP standardization, and industrial implementation. In [128], the challenges and open issues related to joint slicing and functional splitting are analyzed, with a focus on the added complexity of instantiating multiple slices.

One of the important parts of network slicing is RAN slicing, in which communication resources like bandwidth or power can be dedicated to different slices based on various factors, such as their performance requirements and load. The per-slice observability of network slicing serves two important purposes in coexistence scenarios. First, it enables the monitoring of resource utilization with regard to the accomplished service performance, which can be utilized for the re-configuration of RAN slices. Second, it provides KPIs, traffic load, and wireless channel conditions per slice, which helps to evaluate compliance with the service level agreements for the corresponding end-to-end network slices. The objective of RAN slicing algorithms is to fulfill the performance requirements of different slices while maximizing resource or energy efficiency. In the literature, both model-based and model-free solutions have been investigated to address the RAN slicing problem for the coexistence of URLLC and other conventional services. Model-based methods [129–135] for resource partitioning model the problem mathematically and use various optimization techniques (e.g., convex optimization in [129,130], or mixed-integer linear programming in [131, 132]) to solve it. Reference [133] develops a dynamic optimization problem in which the bandwidth and power of the URLLC and eMBB slices are optimized to minimize the sum of power consumption and the negative of resource utilization. However, such approximated models often fail to fully reflect the complexity of the real network, leading to their inability to achieve optimal outcomes in practical scenarios.

As an alternative approach to model-based solutions, there has been a recent surge in using model-free solutions, particularly deep RL (DRL), which has shown great promise in managing RANs without relying on prior models (e.g., [136–143]). The inter-cell interference is one of the challenges in RAN slicing that have received little attention in the literature. With the increase in the traffic of URLLC's coexisting service and network density, and the resulting intense inter-cell interference among multiple cells, it is necessary to adopt a coordinated approach that can effectively handle complex inter-cell and inter-slice interactions while maintaining a low system and model complexity. The authors in [141] proposed a bandwidth and coordinated multi-point-based beamforming optimization that maximizes the

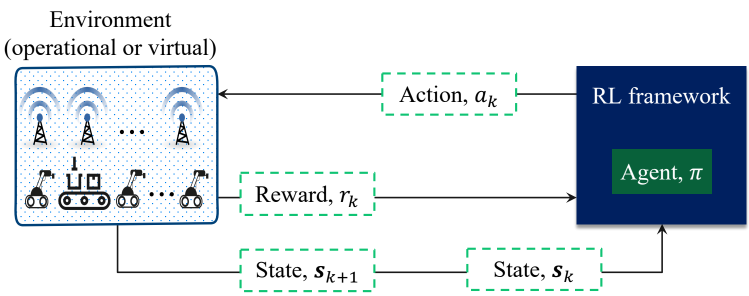


Figure 1.3: The interactions between the agent and the environment in reinforcement learning.

energy efficiency of slices. Reference [139] proposed a multi-agent DRL approach to optimize the resource partitioning for each slice and each cell. Another challenge that has often been neglected in the RAN literature is the trade-off between the signaling overhead of network-wide resource slicing to attain global optimality over the entire RAN and the typically short timescale of resource scheduling to adapt to network dynamics [144]. To address this issue, reference [141] addressed high-frequency beamforming and low-frequency bandwidth orchestration in RAN slicing problem via sample average approximation and alternating direction method of multipliers techniques to support bursty URLLC and multicast eMBB services. In [142], the authors proposed an hierarchical RL (HRL)-based solution to optimize numerology, bandwidth, and transmit power of eMBB and URLLC slices with a lower frequency, along with PRB allocation for each slice with a higher frequency.

1.3.6 RL for Industrial Automation

In the literature, two approaches exist to perform inter-slice and intra-slice orchestrations: i) the model-based and ii) machine learning (ML)-driven. In the former, mathematical models are either developed theoretically or derived through field measurement campaigns. However, deriving these models requires severe assumptions on channel and queue models, which may not realistically reflect the environment of CPSs. Conversely, ML solutions are data-driven and can adjust to network dynamics. Among ML techniques, model-free RL emerges as a powerful tool, and a growing interest in using RL in wireless systems is evident in the current state-of-the-art.

In general, RL algorithms address sequential decision-making by maximizing a reward signal while interacting with an unknown environment. The RL algorithms address problems mathematically described by a Markov decision process (MDP), where state dynamics and reward function are unknown. As depicted in Figure 1.3, at each iteration k, the agent observes a representation of the environment's state $\boldsymbol{s}_k$ from state space $\mathcal{S}$, selects an action a_k from the action space $\mathcal{A}$. Following this action, the agent obtains a numeric reward r_k, and the environment changes to a new state $\boldsymbol{s}_{k+1}$. In RL, a policy determines decision-making, where such policy can be either deterministic (i.e., $\pi(\boldsymbol{s})$ gives the action taken in $\boldsymbol{s}$, where $\pi : \mathcal{S} \mapsto \mathcal{A}$) or stochastic (i.e., $\pi(a|\boldsymbol{s})$ provides the probability of selecting action a in state $\boldsymbol{s}$, where $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$). Hence, the objective is to learn an optimal policy $\pi^\star$ which maximizes the expected accumulated discounted rewards given any starting state $\boldsymbol{s}$ [145].

In this thesis, from all RL algorithms, we use DQN and soft actor-critic (SAC). Even though numerous works have demonstrated RL's effectiveness in addressing emerging issues in wireless communications, there are still challenges and open questions among which we address the following. First, the larger the deployment is, the larger the action space becomes, demanding much memory and computational time for training, possibly to the extent that it becomes difficult or even impossible to apply RL methods on actual deployments. Second, in contrast to other learning scenarios (e.g., image classification and recommendation systems), the vast amount of data required for RL training is not accessible in wireless communication systems. Thus, many RL frameworks rely on simulation data. Typically, these simulators are designed to simplify the system's dynamics and may fail to recognize some hidden patterns. On the other hand, training on the operational network is costly due to RL's random explorations. In the following, we briefly explain both algorithms, and our enhancements and proposals to address these challenges are described in Chapter 3.

1.3.6.1 Deep Q-Networks and Optimization

Q-learning is a model-free reinforcement learning algorithm that works based on the concept of action-value function, $q_\pi(\boldsymbol{s}, a)$, defined as the expected sum of discounted rewards of state $\boldsymbol{s}$ when action a is first taken, and after that, policy π is followed. Hence, the action-value function is defined as [145]

$$\begin{aligned} q_\pi(\boldsymbol{s}, a) :=& \mathbb{E}_\pi \left[\sum_{\kappa=0}^{K-k} \gamma^\kappa r_{k+\kappa} \,\middle|\, \boldsymbol{s}_k = \boldsymbol{s}, a_k = a \right], & (1.1) \\ =& \sum_{\hat{\boldsymbol{s}}, r} \Pr\left(\hat{\boldsymbol{s}}, r | \boldsymbol{s}, a\right) \left[r + \gamma \sum_{\hat{a}} \pi(\hat{a}|\hat{s}) q_\pi(\hat{\boldsymbol{s}}, \hat{a}) \right], & (1.2) \end{aligned}$$

where r_κ is the observed instantaneous reward for iteration κ, K is either a finite number representing the episode length or infinity for continuous RL tasks, and γ is the discount factor. Equation (1.2) presents the Bellman equation for action values. Defining the optimal action-value function as $q^\star(\boldsymbol{s}, a) := \max_\pi q_\pi(\boldsymbol{s}, a)$ for all $\boldsymbol{s} \in \mathcal{S}$ and $a \in \mathcal{A}$, and using (1.2), one can derive the Bellman optimality equations for all states, as [145]

$$q^\star(\boldsymbol{s}, a) = \sum_{\hat{\boldsymbol{s}}, r} \Pr\left(\hat{\boldsymbol{s}}, r | \boldsymbol{s}, a\right) \left[r + \gamma \max_{\hat{a}} q^\star(\hat{\boldsymbol{s}}, \hat{a}) \right], \quad (1.3)$$

and consequently, solve this nonlinear system of equations for $q^\star$. Once $q^\star$ is acquired, one can simply derive the optimal policy $\pi^\star(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{S}$ by selecting any action through which $q^\star(\boldsymbol{s}, a)$ is maximized. Nevertheless, the environment's dynamics (i.e., $\Pr(\hat{\boldsymbol{s}}, r|\boldsymbol{s}, a)$) are typically unknown in real-world industrial applications. Q-learning circumvents the difficulty of obtaining such dynamics by i) sampling the expected values in (1.1), and ii) using existing estimates of the action-value function, $Q(\hat{\boldsymbol{s}}, \hat{a})$, instead of true $q_\pi(\hat{\boldsymbol{s}}, \hat{a})$. In the basic Q-learning, the iterative update at kth iteration is given by [145]

$$Q(\boldsymbol{s}_k, a_k) = Q(\boldsymbol{s}_k, a_k) + \zeta \left[r_k + \gamma \max_{\hat{a} \in \mathcal{A}} Q(\boldsymbol{s}_{k+1}, \hat{a}) - Q(\boldsymbol{s}_k, a_k) \right], \tag{1.4}$$

where $\zeta \in (0, 1)$ is the step-size parameter. In (1.4), the terms inside the bracket represent the error in the action-value function estimate (i.e., the difference between the target and the old estimate). Thus, taking (1.3) into account, (1.4) demonstrates that the learned action-value function, Q, directly approximates the optimal action-value function, $q^\star$, regardless of the behavior policy.

During the training phase, the optimization algorithm trades off between *exploration* and *exploitation*. Exploration describes the process in which the RL agent prioritizes searching for new actions that were not experienced before to improve its action value estimation in different states. In contrast, in the exploitation phase, the RL agent selects actions greedily from what it has learned, i.e., the action with the highest estimated value is taken:

$$\pi(\boldsymbol{s}) = \operatorname*{argmax}_{a \in \mathcal{A}} Q(\boldsymbol{s}, a), \forall \boldsymbol{s} \in \mathcal{S}. \tag{1.5}$$

The exploration phase is often performed via ϵ-greedy action selection method in which an action is either selected randomly with probability ϵ_k, or otherwise using (1.5), where ϵ_k is usually a decaying function in k [145].

In the Q-learning approach, the Q-function is represented by a Q-table, containing states as rows, actions as columns, and Q-values as entries. In practice, the number of states is often too large. This leads to either reducing the number of states (e.g., through quantization) or replacing the Q-table with function approximators. In deep Q-learning [146], the action-value function is represented by a DQN, a neural network (NN) parameterized by $\boldsymbol{\varphi}$. Leveraging the off-policy characteristic of Q-learning, the agent explores the state-action space with some behavior policy (e.g., ϵ-greedy), and each transition $\boldsymbol{b}_k{:=}(\boldsymbol{s}_k, a_k, r_k, \boldsymbol{s}_{k+1}, I_{k+1})$ are stored in a replay buffer, $\mathcal{B}$, MDPs I_{k+1} is a binary parameter that is 0 if the environment terminates at iteration $k{+}1$, and is 1, otherwise (including time-outs) [147]. Then, regardless of sampling technique (e.g., uniform or prioritized experience replay) and the mini-batch size ($|\mathcal{B}_{\mathrm{mb}}|$), we can represent a sampled transition with $\boldsymbol{b}{:=}(\boldsymbol{s}, a, r, \hat{\boldsymbol{s}}, I)$. Besides, reference [146] proposed the use of a target network, parametrized by $\tilde{\boldsymbol{\varphi}}$, which is a duplicate of $\boldsymbol{\varphi}$, except that its parameters are updated after a fixed number of iterations. Then, for each sampled transition, $\boldsymbol{\varphi}$ can be updated in gradient descent direction to minimize the following parameterized mean squared error [146]

$$J(\varphi) = \mathbb{E}_{\boldsymbol{b}} \left[\left(r + I\gamma \max_{\hat{a} \in \mathcal{A}} Q_{\tilde{\varphi}}(\hat{s}, \hat{a}) - Q_{\varphi}(s, a) \right)^2 \right]. \tag{1.6}$$

1.3.6.2 Soft Actor-Critic

SAC is a model-free DRL algorithm. The following characteristics of SAC benefit our industrial automation scenario: i) SAC is an off-policy model-free DRL algorithm in which explorations seek to find an optimal policy maximizing not only the expected accumulated discounted reward but also the expected entropy at each visited state, ii) SAC has an actor-critic architecture where the critic NN estimates state-action pair values, while the actor NN computes the policy, and iii) SAC conquers the massive sampling complications and minimizes the sensitivity of DRL in hyperparameters [148, 149].

The main objective in SAC is to find an optimal stochastic policy that maximizes the sum of discounted expected reward and entropy for upcoming states arising from every state-action tuple $(\boldsymbol{s}_k, \boldsymbol{a}_k)$ weighted by its marginal probability as [149]

$$\pi^\star(\cdot|\cdot) := \underset{\pi(\cdot|\cdot)}{\operatorname{argmax}} \sum_k \mathbb{E}_{(\boldsymbol{s}_k, \boldsymbol{a}_k)} \left[\sum_{l=k}^{\infty} \gamma^{l-k} \mathbb{E}[r_k + \psi \mathbb{H}(\pi(\cdot|\boldsymbol{s})) | \boldsymbol{s}_k = \boldsymbol{s}, \boldsymbol{a}_k = \boldsymbol{a}] \right], \tag{1.7}$$

where $\gamma (\in [0, 1])$ and $\psi (> 0)$ are the discount factor and the temperature parameter specifying the relative importance between the reward and entropy terms, respectively. Furthermore, $\mathbb{H}(\pi(\cdot|\boldsymbol{s}_k)) := \mathbb{E}[-\ln(\pi(\boldsymbol{a}|\boldsymbol{s}_k))]$ is the entropy of policy π at state $\boldsymbol{s}_k$. Introducing entropy in (1.7) guides the policy to explore more broadly while avoiding blatantly unfavorable trajectories.

Here, we describe SAC with prioritized experience replay originally developed for DQN [150]. In this approach, transitions with greater temporal disparities are repeated more frequently, adding a bias toward those actions. To address this issue, we remove this priority during training. Moreover, we describe SAC with target networks and clipped double Q-learning, both originally established for twin delayed deep deterministic policy gradient algorithm. Such additions can mitigate overestimation in value approximation while assuring convergence to a suboptimal policy [151].

The SAC algorithm (and actor-critic methods in general) uses policy iteration, in which the algorithm alternates between policy evaluation (to estimate the action-value function by $Q(\boldsymbol{s}, \boldsymbol{a})$) and policy improvement (to compute π) in the direction of maximizing the first sum in (1.7). In the policy evaluation step, using the soft Bellman backup equations [149], the soft action value function can be computed iteratively as follows:

$$Q(\boldsymbol{s}_k, \boldsymbol{a}_k) = r_{k+1} + \gamma \mathbb{E}[Q(\boldsymbol{s}_{k+1}, \boldsymbol{a}_{k+1}) + \psi \mathbb{H}(\pi(\cdot|\boldsymbol{s}_k))]. \tag{1.8}$$

In large-scale industrial automation problems where the state and action spaces are large, Q and π are approximated in each iteration using deep neural networks (DNNs) (via critics and actors, respectively). As mentioned before, we leverage target networks and clipped double Q-learning. Hence, in the architecture we use in this thesis, there are 4 DNNs for the first critic, its target critic, the second critic, and its target, whose weights are denoted by φ_1, $\tilde{\varphi}_1$, φ_2, and $\tilde{\varphi}_2$, respectively. In addition, there are 2 DNNs for the actor and its target network, whose weights are denoted by $\boldsymbol{\vartheta}$ and $\tilde{\boldsymbol{\vartheta}}$, respectively. Thus, Q, in (1.8), is approximated by 2 DNNs as Q_{φ_i}, $i \in \{1, 2\}$.

Considering the sampled transition $\boldsymbol{b}:=(\boldsymbol{s}, a, r, \hat{\boldsymbol{s}}, I)$, described in the previous Section, φ_1 and φ_2 can be trained by minimizing the mean squared error as

$$J_{\mathrm{Q}}(\varphi_i) \coloneqq \mathbb{E}_{\boldsymbol{b}}\left[\frac{1}{2}\Big(Q_{\varphi_i}(\boldsymbol{s}, \boldsymbol{a}) - \widetilde{Q}(\hat{\boldsymbol{s}}, \tilde{\boldsymbol{a}}, r, I)\Big)^2\right], \tag{1.9}$$

where $\tilde{\boldsymbol{a}}$ is sampled separately from $\pi_{\tilde{\vartheta}}(\cdot|\boldsymbol{s}_{k+1})$, and

$$\widetilde{Q}(\boldsymbol{s}, \boldsymbol{a}, r, I) \coloneqq r + I\gamma\left(\min_{i=1,2} Q_{\tilde{\varphi}_i}(\boldsymbol{s}, \boldsymbol{a}) - \psi \ln(\pi(\boldsymbol{a}|\boldsymbol{s}))\right). \tag{1.10}$$

Note that the minimum represents the smallest value between the two action-value function approximations for clipped double Q-learning [151]. Then, in order to minimize $J_Q(\cdot)$, φ_1 and φ_2 are updated in the direction of gradient descent. To ensure that temporal-difference error remains low, we update target critics' weights gradually by $\tilde{\varphi}_i = \nu\varphi_i + (1-\nu)\tilde{\varphi}_i$ for $i \in \{1, 2\}$ at each DRL iteration.

In the policy improvement step, the actor NN can be updated by minimizing the expected Kullback-Leibler divergence between π_{ϑ} and the exponential of the soft state-action value function, which can be rewritten as

$$J_\pi(\boldsymbol{\vartheta}) \coloneqq \mathbb{E}\left[\psi \ln(\pi_{\boldsymbol{\vartheta}}(\boldsymbol{a}|\boldsymbol{s})) - \min_{i=1,2} Q_{\varphi_i}(\boldsymbol{s}, \boldsymbol{a})\right]. \tag{1.11}$$

To minimize $J_\pi(\cdot)$ based on the latest policy, we employ the re-parameterization technique, from [149], to reformulate the expectation over actions into an expectation over noise, leading to a smaller variance estimator. Therefore, we draw samples from a squashed Gaussian policy such that $\widehat{\boldsymbol{a}} := \tanh\left(\mu_{\mathrm{G},\boldsymbol{\vartheta}}(\boldsymbol{s}) + \sigma_{\mathrm{G},\boldsymbol{\vartheta}}(\boldsymbol{s}) \cdot \boldsymbol{\chi}\right)$, where $\mu_{\mathrm{G},\boldsymbol{\vartheta}}(\cdot)$ and $\sigma_{\mathrm{G},\boldsymbol{\vartheta}}(\cdot)$ are the estimated mean and standard deviation of a Gaussian distribution, respectively, and $\boldsymbol{\chi}$ follows a multivariate Gaussian distribution, whose mean is a vector of 0s, and covariance matrix is the identity matrix. Hence, we can reformulate $J_\pi(\boldsymbol{\vartheta})$ by replacing $\boldsymbol{a}$ with $\widehat{\boldsymbol{a}}$ in (1.11). The policy parameters, $\boldsymbol{\vartheta}$, are then updated in the gradient descent direction as in [149]. Additionally, we update the target actor weights smoothly by $\tilde{\boldsymbol{\vartheta}} = \nu\boldsymbol{\vartheta} + (1-\nu)\tilde{\boldsymbol{\vartheta}}$.

1.4 Research Questions

In this section, we elaborate on our research questions based on our HRQs outlined in Section 1.1.1 and the conducted literature study in Section 1.3.

1.4.1 URLLC Key Performance Indicators

Addressing HRQ1 is influential because understanding the operational performance of practical systems could provide society with a solid foundation for trust in the 5G/B5G to support CPS.

Throughout this thesis, we do not differentiate between packet losses (due to missing the delay bounds) and packet errors (i.e., when the receiver is unable to decode the packet successfully). We consider a communication service failure to occur when there are consecutive packet losses or packet errors for a duration exceeding the survival time. These assumptions are consistent with the 3GPP definitions for CPSs in [62, 71]. Additionally, we use the KPIs defined by 3GPP in [62, 71] as the primary references for communication service availability and reliability, where they are respectively defined as the mean proportion of time and the mean time during which the communication service is delivered to the application layer of the receiving entity according to its required QoS. These definitions are in line with reliability metrics used in OT (e.g., IEC standard in [67]). However, despite their consistency, these definitions remain abstract. Accordingly, to address HRQ1, we focus our efforts on addressing the following in-depth research question (RQ) (see Paper 1 and Patent 1 in Section 1.5.1 for our solution approach):

- **RQ1.1**: How to mathematically model 3GPP definitions of communication service KPIs for cyber-physical applications (i.e., communication service availability and reliability)?

Nevertheless, such reliability KPIs are of no use to a network designer during deployment or a network controller during operation. Therefore, we put our efforts into bridging the gap between traditional ICT performance metrics (e.g., PER) and communication service performance metrics (e.g., availability) by addressing the below RQ (see Paper 1 and Patent 1 in Section 1.5.1 for our solution approach):

- **RQ1.2**: How could a network designer translate such communication service KPIs to meaningful network layer parameters without the need to perform computationally complex simulations to validate years of reliability?

It is worth noting that this RQ was originally raised by 5G alliance for connected industries and automation (5G-ACIA) forum as a result of discussions around the white paper in [152]. Our contributions were partly delivered through an internal report.

1.4.2 Enabling URLLC Service in 5G/B5G

We study URLLC in two settings. Firstly, in the single service setting, we assume an arbitrary amount of resources (such as bandwidth and maximum output power) are allocated to URLLC service, and our focus is to fulfill stringent requirements of URLLC given these allocated resources. Hence, in this setting, we assume that there is no impact from other nearby services on URLLC. Secondly, we assume that mixed services are present. In this setting, we assume URLLC service operates concurrently with another service on a network. We consider distributed learning service as the secondary service, and we aim to minimize its convergence time while meeting the strict operational KPIs of URLLC.

1.4.2.1 Single Service

As discussed in Section 1.3.4, several studies have explored enhancing reliability through diversity techniques (e.g., [80, 84, 86, 100, 153]). As an example of overlooking the benefits of such techniques, the next generation mobile networks (NGMN) alliance[9] conducted a study in [153] to investigate the sensitivity of URLLC performance to two multi-connectivity architecture alternatives proposed by 3GPP. However, the evaluations were based on the assumption of independent spatial paths and, thus, independent packet errors across different links. Nevertheless, the presence of shadowing cross-correlation across different wireless links has a considerable impact on the performance of the wireless system (as observed in simulation results from [100, 154], and the mathematical framework in [83]). Consequently, analyzing the impact of correlation on end-to-end reliability and considering it in the design of networks supporting URLLC applications remain an open problem. The necessity to conduct such analysis to contribute to the ongoing standardization activities in [155] on multi-connectivity scenarios motivated us, at Ericsson, to formulate the below RQ (see Paper 2 in Section 1.5.2 for our solution approach):

- **RQ2.1.1**: What is the impact of correlated spatial paths on end-to-end reliability in multi-connectivity scenarios when employing packet duplication? How does this influence the design of networks supporting cyber-physical applications?

In response to the aforementioned RQ, our evaluations emphasized the importance of dynamic path selection in multi-connectivity scenarios.

In the literature concerning the optimization of wireless systems for efficient support of URLLC service, there has been little discussion regarding near real-time RL solutions. Additionally, most of the existing works have been evaluated on small-scale software modules developed by researchers for specific use cases. Hence, the scalability of such solutions is questionable when it comes to practical scenarios. Finally, towards achieving stringent delay and reliability objectives, the energy efficiency of the designed solutions has been usually neglected in prior arts (i.e., the energy costs associated with different approaches to achieving an objective have not been considered). Thus, such lack of practical and end-to-end analysis that is compliant with standardization activities by 3GPP, particularly in terms of communication service KPIs for CPSs, motivated us to investigate the following RQ (see Papers 3-5 and Patent 2 in Section 1.5.2 for our solution approach):

[9] NGMN is a mobile telecommunications association of mobile operators, manufacturers and research institutes.

- **RQ2.1.2**: How to develop an orchestrator module to manage reliability enhancement techniques in order to fulfill stringent requirements of URLLC such that the orchestrator is

  1. aligned with 3GPP specification for cyber-physical control applications in [71], i.e., it takes into account the operational requirements of URLLC;
  2. efficient, in terms of spectrum utilization and energy consumption;
  3. scalable, i.e., capable of managing a large number of devices; and
  4. practical, i.e., it considers the 5G-NR protocol stack [156], with its complications and impact on delay and reliability, and is aligned with the current vision for edge cloud development (for instance, as in [89,106,157]).

1.4.2.2 Mixed Services

For the service coexisting with URLLC, we have chosen the distributed learning service. As previously discussed in Section 1.3.5, there is a rich literature on the coexistence of URLLC with conventional services, such as eMBB and mMTC. However, to the best of our knowledge and as of the time of writing this thesis, there is no literature available on the coexistence of distributed learning service and URLLC that considers both communication and learning aspects. Nevertheless, numerous recent cyber-physical control applications leverage distributed learning to improve efficiency and system performance, such as industrial automation [5,158], and autonomous driving [159]. The reason to select distributed learning traffic rather than centralized training is that the centralized training of ML models can be impractical in many wireless communication applications because of (i) the distributed nature of the data generated/collected by mobile devices, (ii) privacy concerns on sharing the local data with a central server especially when the computational server is managed by a third party operator, and (iii) limited wireless resources (in terms of bandwidth and power). Therefore, privacy-preserving distributed learning techniques have become the cornerstone of recent advancements in artificial intelligence (AI) applications over wireless networks [160].

As distinct services, both URLLC and distributed learning performance have been widely investigated in the existing literature. However, the coexistence of URLLC service, with its stringent requirements, and distributed learning workflow, with its unique traffic model and performance characteristics, have not yet been discussed in the literature. Such coexistence introduces new fundamental challenges as well as unique trade-offs between URLLC latency and availability KPIs on the one hand, and convergence time of distributed training on the other hand. Notice that there are fundamental differences between distributed learning services and other traditional communication services. In particular,

1. The performance of these conventional services is characterized by the statistics of a communication metric such as throughput or energy consumption. In contrast, the distributed learning service is an iterative and collaborative

task striving to solve an optimization problem as quickly as possible. Therefore, its performance is not only influenced by the transmission capabilities of a single device but also by the potential statistical correlation of data from multiple devices.

2. In addition to well-known decision parameters for other communication services, distributed learning has unique decision parameters such as model size, algorithm choice, and device selection that differentiate it from other communication services.

These are unique characteristics of distributed learning that distinguish it from other conventional communication services. Hence, **assuming the already existing 5G-NR features for QoS handling**, we formulate the in-depth RQ as (see Paper 6 in Section 1.5.3 for our solution approach):

- **RQ2.2.1**: What are the trade-offs between URLLC's operational KPIs and distributed learning's training delay and accuracy? How does AI setting (e.g., model size or the number of participating devices) impact its own and URLLC performance?

The potential statistical correlation of data allows for a carefully chosen subset of devices to be used for learning. In this context, higher quality updates from a few devices may prove more advantageous than lower quality updates from a larger number of devices [161]. Besides, missing parameters/gradients of some devices can be approximated by the ones of other devices or by their own parameters/gradients sent in previous iterations. Hence, another RQ is defined as (for details on our solution approach, please see Paper 7 in Section 1.5.3)

- **RQ2.2.2**: How can we enhance the distributed training workflow using device selection to minimize the convergence time of distributed learning service while still fulfilling URLLC's stringent operational requirements?

Although 5G QoS handling can ensure the prioritization of URLLC traffic, the interference of AI traffic can still lead to deterioration in URLLC service performance. Another approach to reducing such impact even further is to leverage separate resources (in terms of bandwidth and power). For instance, if each of these services uses different frequency ranges, there is no inter-interference among the devices from these distinct services. However, it is not clear how much resources should be allocated to each of these services. The more resources that are allocated to one service, the better performance this service can reach (in terms of convergence time for distributed learning and operational KPIs for URLLC). Therefore, we formulate the inter-slice orchestration RQ as (refer to Patent 3 in Section 1.5.3 for our solution approach):

- **RQ2.2.3**: What is the potential gain of performing joint user-selection and slicing (i.e., dedicating a portion of total bandwidth and power to each service) to minimize the convergence time of the distributed learning service, while still fulfilling the URLLC's stringent operational requirements?

1.5 Thesis Contributions

This section summarizes the main contributions of this thesis towards understanding CPSs' operational requirements in terms of wireless communications and enabling URLLC service in 5G and B5G systems.

1.5.1 Key Performance Indicators

To address RQ1.1 and RQ1.2, we combine basic definitions of reliability theory [65] with the communication service requirements (defined by 3GPP [71]) to develop concrete mathematical models for the URLLC service reliability and availability. Moreover, we propose a finite state Markov chain (FSMC) to keep track of burst errors and approximate the error length distributions. The proposed framework aims to provide a mapping function that takes the application's survival time into account and maps the requirements on the application layer to the expected network performance or vice versa. A thorough elaboration of this contribution can be found in the following paper and patent [162, 163]:

- **Paper 1:** M. Ganjalizadeh, A. Alabbasi, J. Sachs, and M. Petrova, "Translating Cyber-Physical Control Application Requirements to Network level Parameters," *in IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2020.

- **Patent 1:** M. Ganjalizadeh, A. Alabbasi, J. A. Ibanez, M. Petrova, and J. Zander, "Methods of Translating Cyber-Physical Control Application Requirements to Radio Parameters," US Patent Application 20220294697A1, Filed Oct. 29, 2020. Available: `https://patents.google.com/patent/US20220294697A1`.

Contribution: As mentioned in Section 1.4.1 for RQ1.1, the high-level problem for these works was raised by a 5G-ACIA work item. Nevertheless, the author of this thesis conducted the literature review, narrowed down the high-level problem, formulated the problem statement, developed the analytical model, performed the performance evaluation, and authored the original draft of the publication and patent application.

1.5.2 Enabling URLLC in Single Service Scenarios

In single service scenarios and on the benefits of spatial diversity (as an example of diversity techniques), we first perform an investigation on the impact of shadowing correlation on the end-to-end reliability to address RQ2.1.1. In this study, we analyze the significance of such correlation on the design of a system supporting

URLLC applications. The results of this contribution are presented in the below paper [10]:

- **Paper 2:** M. Ganjalizadeh, P. Di Marco, J. Kronander, J. Sachs, and M. Petrova, "Impact of Correlated Failures in 5G Dual Connectivity Architectures for URLLC Applications," *in IEEE Globecom Workshops (GC Wkshps)*, 2019.

Contribution: The idea for this paper was originally inspired by a study conducted by NGMN alliance, as described in their report [153]. The author of this thesis formulated the problem, developed the analytical results, carried out the simulations, analyzed the numerical results, and authored the original draft of this paper.

Then, we take a step further and develop a model-free framework to address RQ2.1.2. In this study, we develop a DRL framework to dynamically orchestrate power control and diversity techniques to maximize our reliability KPIs. Our orchestrator performs near real-time control, which is in line with current developments of edge computing for cellular networks [106]. This contribution is reported in the following publications [5, 164]:

- **Paper 3:** M. Ganjalizadeh, A. Alabbasi, A. Azari, H. S. Ghadikolaei, and M. Petrova, "An RL-based Joint Diversity and Power Control Optimization for Reliable Factory Automation," *in IEEE Global Communications Conference (GLOBECOM)*, 2021.
- **Patent 2:** M. Ganjalizadeh, A. Azari, H. S. Ghadikolaei, A. Alabbasi, and M. Petrova, "Methods and Apparatus for Controlling One or More Transmission Parameters Used by a Wireless Communication Network for a Population of Devices Comprising a Cyber-physical System," WO Patent Application WO2022250604A1, Filed May 27, 2021. Available: `https://patents.google.com/patent/WO2022250604A1`.

Contribution: The author of this thesis formulated the problem and system model, developed the reliability KPIs and the proposed DQN algorithm, implemented the simulator on top of Ericsson's proprietary simulator and performed the performance evaluation. While the original draft of the paper was written by the author of this thesis, the initial patent application draft was written in collaboration with the second and third inventors.

We further extend our solution to address RQ2.1.2 by proposing a scalable architecture called branching SAC (BSAC). In BSAC, the action representation of each user is distributed across separate neurons. Such decomposition of users' actions enables the linear growth of the action space with an increasing number of users. This significantly reduces the computational and memory complexities of the solver, and consequently, our BSAC solution can orchestrate within an environment where many industrial devices exist. Moreover, we take spectral and energy efficiency into consideration. This contribution is thoroughly discussed in the following publication [94]:

- **Paper 4:** M. Ganjalizadeh, H. S. Ghadikolaei, A. Azari, A. Alabbasi, and M. Petrova, "Saving Energy and Spectrum in Enabling URLLC Services: A Scalable RL Solution," in *IEEE Transactions on Industrial Informatics*, 2023.

Contribution: The author of this thesis developed the MDP formulation and the BSAC algorithm, implemented the simulator on top of Ericsson's proprietary simulator, evaluated the performance, and authored the original draft of the manuscript.

Nevertheless, model-free orchestration for URLLC applications is challenging because these applications require extreme system performance, and thus, the tail distribution of metrics resulting in failure events is often small. In other words, infrequent encounters of rare events (i.e., communication failures in this case) during training may lead to suboptimal decision-making in deployment. To address the practicality challenge posed by RQ2.1.2, we propose incremental learning GAN (IL-GAN), an incremental learning framework built on top of conventional generative adversarial networks (GANs). IL-GAN uses balanced replay and layer freezing techniques in its training and is able to generate samples uniformly, regardless of the frequency of their occurrence in the dataset. We published our contribution in the following paper [165]:

- **Paper 5:** J. R. Baldvinsson, M. Ganjalizadeh, A. Alabbasi, M. Björkman, and A. H. Payberah, "IL-GAN: Rare Sample Generation via Incremental Learning in GANs," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.

Contribution: This paper is based on the M.Sc. thesis work, which was co-supervised by the author of this thesis. The idea for the M.Sc. thesis was developed by the author of this thesis and the third author, with the author of this thesis contributing through weekly discussions. The initial draft of the paper was jointly written by the author of this thesis and the first author.

1.5.3 Enabling URLLC in Mixed Services Scenarios

Towards addressing RQ2.2.1, we leverage our models for the operational metrics of the URLLC service (i.e., communication service availability) and develop models for essential parameters that characterize distributed training workflow (i.e., training delay, model size, and convergence time) and investigate the interplay between them. We leverage system level and link level 3GPP-compliant 5G-NR simulations. Besides, we use RLC flow prioritization between URLLC and distributed learning services. Within our simulations, we investigate the impact of the number of participating AI devices on URLLC performance. Moreover, we study the influence of soft synchronous protocols on the training delay of distributed learning. A detailed elaboration regarding this contribution can be found in the following paper [166]:

- **Paper 6:** M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Interplay between Distributed AI Workflow and URLLC," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.

Contribution: The author of this thesis developed the problem formulation and system model, designed the KPIs for distributed learning service in collaboration with the second author, implemented a simulator based on Ericsson's proprietary platform, and conducted the performance evaluation. Additionally, the author was responsible for writing the initial draft of the paper.

To address RQ2.2.2, we formulate an optimization problem that minimizes the average training latency of the distributed training to reach ϵ optimality while sustaining URLLC's communication service availability requirements in our resource-limited system (in terms of bandwidth and transmission power). Then, we transform the formulated coexistence optimization problem into an MDP and design an action masking technique to put a lower bound on the minimum number of AI devices required to participate in each iteration of distributed training. However, the solution may select a higher number of devices than this minimum to address the so-called *straggling effect*. To deal with the unknown dynamics of our complex cellular system, we propose a data-driven approach that optimizes the device selection policy via a BSAC-based framework. An in-depth description of this contribution can be found in the following article [167]:

- **Paper 7:** M. Ganjalizadeh, H. S. Ghadikolaei, D. Gündüz, and M. Petrova, "Device Selection for the Coexistence of URLLC and Distributed Learning Services," *submitted to IEEE Internet of Things Journal*, 2023.

Contribution: The author of this thesis developed the problem formulation and system model, and proposed the MDP formulation and SAC framework. The author also implemented distributed learning models on Ericsson's proprietary simulator and conducted the performance evaluation. The paper was jointly written by the author of this thesis and the co-authors.

Unlike previous work, and to address RQ2.2.3, we develop a hierarchical DRL framework which takes network-wide large timescale slicing into consideration. Further information regarding this contribution can be found in the following patent application [168]:

- **Patent 3:** M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Systems and Methods for Joint Inter- and Intra-slice Orchestration Using Reinforcement Learning," PCT Patent Application PCT/SE2023/050016, Filed Jan. 12, 2023.

Contribution: The author of this thesis developed the problem formulation and system model, worked with the second inventor to develop the HRL framework, and authored the patent application in collaboration with co-inventors.

1.6 Thesis Organization

The remainder of this thesis is structured into five chapters, which are organized as follows:

- Chapter 2 studies the relationship between network metrics and the operational metrics (i.e., communication service availability and reliability as observed by the application layer).
- Chapter 3 investigates the dynamic orchestration techniques in wireless systems to enable URLLC within a single slice (i.e., given the allocated bandwidth and power to URLLC slice).
- Chapter 4 investigates the trade-offs and management techniques to enable the coexistence of URLLC and distributed learning services.
- Chapter 5 concludes the research studies presented in this thesis and highlights the limitations and potential future extensions of the conducted research.

Chapter 2

KPIs: A Journey From OT to ICT World

In [71], 3GPP specifies the communication service performance requirements (in terms of availability and reliability) for various cyber-physical control applications and use cases. These metrics reflect the performance from the application layer perspective and consider survival time. In this chapter, we leverage the fundamental principles of reliability theory to bridge the gap between the KPIs used in ICT (e.g., PER) and those used in OT (e.g., availability). Specifically, we develop concrete mathematical models for the reliability and availability of URLLC service and provide an approximation between ICT metrics and OT metrics.

2.1 System Model

CPSs consist of end devices executing diverse functions, all contributing to the control of physical processes. Examples of such end devices are sensors and actuators. In such a system, the communication system is responsible for the timely delivery of i) sensor data to gNBs and then to a nearby controller, and ii) computed or emergency tasks to the actuators. Once the packets are delivered, the application layer performs the requested function. However, packet losses might occur in wireless networks for reasons such as interference, fading, and delay. Here, packet loss reflects all packets that do not reach their intended recipient within their delay bound due to factors such as incorrect decoding at the receiving end, delay (caused by excessive retransmissions or congestion), or loss (for instance, resulting from restrictions on retransmission limits in MAC and RLC or synchronization issues). Recall that sporadic packet loss in the underlying network does not necessarily affect operational performance. Survival time, denoted by T_{sv}, defines the maximum time period that the application layer can tolerate consecutive losses in the underlying network [71]. For instance, consider a scenario where new positions are periodically transmitted to mobile robots. In this case, a limited number of consecutive packet losses does not significantly affect the overall performance of the robots. When a new packet is successfully received, the application layer can use techniques such as interpolation to estimate the missing positions. Hence, we discuss the performance of a CPS on two levels, as observed by: (i) the application and (ii) the network. The former refers to the communication service performance, whereas the latter is the performance that the network can provide up to the PDCP layer.

2.2 Understanding the Network and Operational Performance

In the network layer, we define the channel state, which is described by a Bernoulli state variable $X_i^\Gamma(n) \sim \mathrm{Ber}(1 - e_i^\Gamma(n))$, where $e_i^{(n)}$ represents the packet decoding error probability for nth transmission to ith URLLC device in $\Gamma \in \{\mathrm{DL}, \mathrm{UL}\}$ direction. This probability is a function of SINR, the code rate of the transmitted packet, and various settings in wireless communication (e.g., the number of retransmissions, multi-connectivity). Considering the delay, we can define the network state variable as

$$Y_i^\Gamma(n) := \mathbb{1}\left\{D_i^\Gamma(n) \leq D_{i,\Gamma}^{\mathrm{req}},\ \mathrm{AND}\ X_i^\Gamma(n) = 1\right\}, \tag{2.1}$$

where $D_i^\Gamma(n)$ and $D_{i,\Gamma}^{\mathrm{req}}$ represent the end-to-end delay of nth packet of ith device and the delay requirement for the ith device, in Γ direction, respectively. Assuming that packet n is received at time t_n, we define the continuous network state variable as

$$Y_i^\Gamma(t) := \sum_{n=1}^{\infty} Y_i^\Gamma(n)\left[h(t - t_n) - h(t - t_{n+1})\right], \tag{2.2}$$

where $h(\cdot)$ is 1 for $t \geq 0$ and 0 otherwise.

In the application layer, consecutive packet losses shorter than the survival time, T_{sv}, can be tolerated or even, in some cases, corrected [71, 162]. We can describe the communication service state by a binary function

$$Z_i^\Gamma(t) = \begin{cases} 0, & \text{if communication service for device } i \text{ in } \Gamma \text{ is failed,} \\ 1, & \text{if communication service for device } i \text{ in } \Gamma \text{ is operational.} \end{cases} \tag{2.3}$$

The communication service state variable can be derived as:

$$Z_i^\Gamma(t) := \max_{t - T_{\mathrm{sv}} \leq \tau \leq t} Y_i^\Gamma(\tau). \tag{2.4}$$

Figure 2.1 illustrates the relationship between $Z_i^\Gamma(t)$ and $Y_i^\Gamma(t)$.

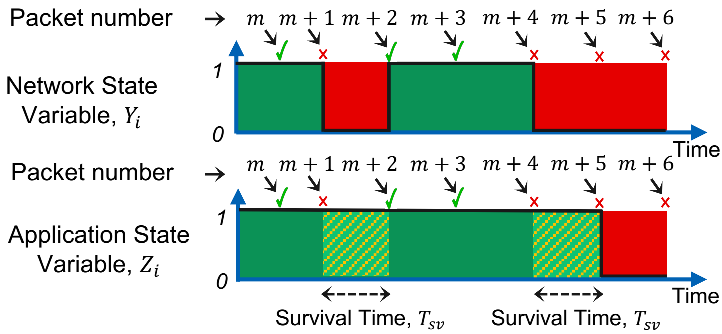


Figure 2.1: An example showing the relationship between network state variable, $Y_i\,(t)$, and communication service state variable, $Z_i\,(t)$ [5].

2.2.1 Operational Performance

2.2.1.1 Communication Service Availability

Availability is an item's ability to perform its functions given the specified requirements at an arbitrary time [65]. Therefore, in wireless systems, availability (or average availability) at the application layer is defined as the mean proportion of time during which the communication service is delivered according to the required QoS with interruptions that are shorter than T_{sv}. That is,

$$\alpha_i^{\Gamma} := \lim_{t\to\infty} \Pr\left(Z_i^{\Gamma}(t) = 1\right) = \lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{T} Z_i^{\Gamma}(t)dt. \tag{2.5}$$

In URLLC, the requirement is often defined in the form of [72]

$$\Pr\left\{\alpha_i^{\Gamma} \le \alpha_i^{\mathrm{req}}\right\} \le \lambda,\ \ \forall i \in \mathcal{U}, \tag{2.6}$$

where α_i^{req} is the communication service availability requirement of the use case that URLLC device i serves, λ is the sensitivity of this use case to α_i^{req}, and $\mathcal{U}$ is the set of industrial devices. We follow [71] in assuming that the requirement for UL and DL availability is the same for a given use case.

2.2.1.2 Communication Service Reliability

Communication service reliability is defined as the mean duration of time that the communication service is operational (i.e., $Z_i^{\Gamma}(t) = 1$). Let $N_i^{\Gamma}(t)$ denote the number of times the crossing from $Z_i^{\Gamma}(t) = 1$ to $Z_i^{\Gamma}(t) = 0$ occurs. Then, we can define the reliability of ith device as:

$$\tau_i^{\Gamma} := \lim_{T\to\infty} \frac{1}{N_i^{\Gamma}(T)} \int_{t=0}^{T} Z_i^{\Gamma}(t)dt. \tag{2.7}$$

Note that the reliability requirement can be similarly defined as (2.6).

2.3 Operational Performance Approximation

The 3GPP specification in [71] outlines not only the required communication service reliability and availability but also provides the typical survival time for various use cases in cyber-physical control applications. In this section, we aim to approximate the operational metrics (i.e., communications service availability and reliability) by utilizing network layer metrics such as the PER. For simplicity, we consider single-user performance in any transmission direction, and as a result, the corresponding notations are omitted in the equations presented here.

To derive the mapping between operational and network performance, we propose using an FSMC to keep track of the length of burst errors in the communication system. We assume that the traffic is periodic with a period of T_{c}, allowing us to calculate the survival time, T_{sv}, as the number of consecutive packet failures that the application can tolerate, N_{sv}, where

$$N_{\mathrm{sv}} := \left\lfloor \frac{T_{\mathrm{sv}}}{T_{\mathrm{c}}} \right\rfloor. \tag{2.8}$$

The state space for a URLLC device can be partitioned as

$$\mathcal{S} := \{\mathcal{U}^{\mathrm{S}}, \mathcal{D}^{\mathrm{S}}\} = \left\{\mathcal{U}^{\mathrm{N}}, \mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(N_{\mathrm{sv}})}, \mathcal{D}^{\mathrm{S}}\right\}, \tag{2.9}$$

where $\mathcal{U}^{\mathrm{S}}$ and $\mathcal{D}^{\mathrm{S}}$ denote the up state (i.e., where the communication service state variable in (2.3) is 1) and down state (i.e., where the communication service state variable in (2.3) is 0) for communication service, respectively. Besides, $\mathcal{S}^{(n)}$ represents a state with n consecutive packet losses, and $n \in [1, N_{\mathrm{sv}}]$. Hence, the state space is divided into a set of up states (i.e., $\{\mathcal{U}^{N}, \mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(N_{\mathrm{sv}})}\}$), and a down state, $\mathcal{D}^{\mathrm{S}}$. The state space is broken down to capture burst length up to the size of survival time and consequently to have a proper characterization between operational and network performance. Therefore, the subset $\{\mathcal{S}^{(1)}, \ldots, \mathcal{D}^{\mathrm{S}}\}$ represents network outages, and the network is available only in state $\mathcal{U}^{\mathrm{N}}$. Figure 2.2 illustrates the proposed FSMC and the transition probabilities for different states. In this figure, $\bar{\tau}_{\mathcal{U}_{\mathrm{N}}}$ and $\bar{\tau}_{\mathcal{D}_{\mathrm{N}}}$ denote the network's mean uptime (where network state variable Y in (2.1) is 1), and mean downtime or mean outage duration (where network state variable Y in (2.1) is 0), respectively.

An irreducible Markov chain is one where all states are interconnected. The period of a state is the greatest common divisor of all integers m where there is a non-zero probability of returning to the original state after m transitions. Thus, a state with a period of 1 is termed aperiodic. Given that the proposed FSMC is irreducible and its states are aperiodic, it possesses a unique stationary distribution (proof in [169]). Using Markov chain properties, we can derive communication service availability as the sum of steady-state probabilities of states $\{\mathcal{U}^{\mathrm{N}}, \mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(N_{\mathrm{sv}})}\}$ (i.e., the ones representing the communication service up state, $\mathcal{U}^{\mathrm{S}}$, in Figure 2.2). Hence, communication service availability can be approximated as follows:

$$\alpha = 1 - p(1 - \bar{\tau}_{\mathcal{D}_{\mathrm{N}}}^{-1})^{N_{\mathrm{sv}}}, \tag{2.10}$$

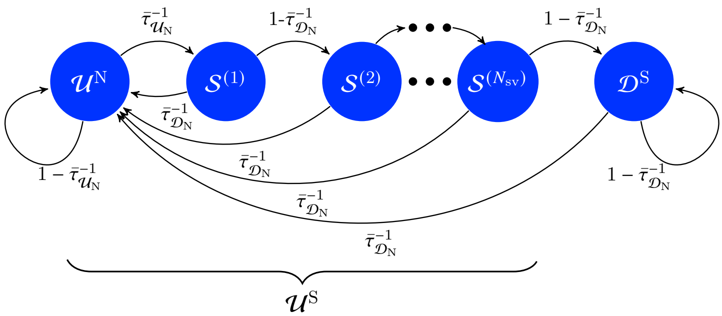


Figure 2.2: Proposed Markov chain to keep track of burst length (adapted from [162]).

where p is the PER. Then, the communication service reliability can be approximated as follows:

$$\tau = \frac{1}{p\bar{\tau}_{\mathcal{D}_\mathrm{N}}^{-1}(1-\bar{\tau}_{\mathcal{D}_\mathrm{N}}^{-1})^{N_\mathrm{sv}}} - \bar{\tau}_{\mathcal{D}_\mathrm{N}}. \tag{2.11}$$

2.3.1 Selected Results

To evaluate our approximation model for reliability and availability, we performed both link level and network level simulations for our factory automation scenario. We established the simulation environment through link level simulations, calculating path gains and 3D channel data. Network level simulations were conducted to simulate MAC and upper layers in a multi-cell multi-user scenario. The factory was represented as a rectangular plant with dimensions of $120 \times 50 \times 11\,\mathrm{m}^3$. We adopted the 3GPP Indoor Hotspot (InH) model from [170], initially intended for office scenarios. Nevertheless, we used the blockage model in [170] to adjust it for our large factory, with concrete walls, ceilings, and floors, filled with numerous metallic objects. In the network level simulations, radio units are positioned in two rows at the height of 10 m, each 20 m apart from its neighbors. We configured the system with 12 radio units, the furthest left and right ones being 15 m and 10 m from the factory's long and short walls, respectively. Device traffic is based on the motion control use case from [71], involving cyclic traffic with a 2 ms period, a delay bound of 2 ms, and a packet size of 20 Bytes.

Figure 2.3 demonstrates the impact of survival time on communication service availability and reliability under different load conditions. The solid bars on the left represent the results of our simulations, the black-hatched bars in the middle represent values derived from (2.10) and (2.11) for availability and reliability, respectively, while the white-dotted bars on the right represent values derived under the assumption that packet losses occur independently (see [171] for derivations).

As shown in Figure 2.3a, in a scenario with 50 users, simulations show that communication service availability can meet the "eight 9s" (0.99999999) requirement when the survival time is triple the cycle time (i.e., $T_\mathrm{sv} = 6\,\mathrm{ms}$). This result

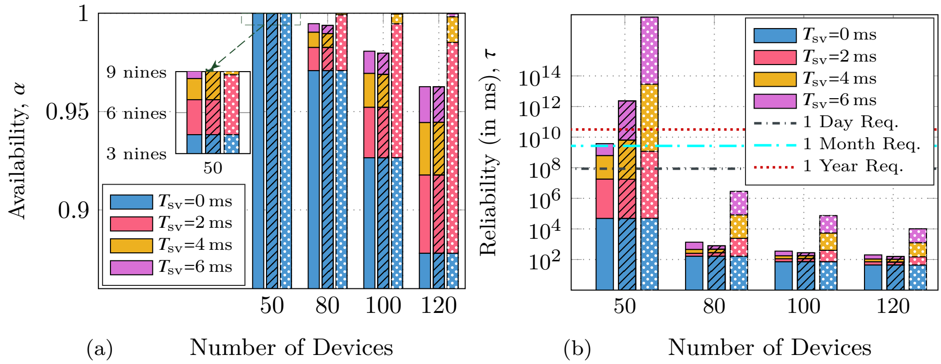


Figure 2.3: Impact of survival time, T_{sv}, on application layer's (a) availability, and (b) reliability, based on i) simulation results (plain colored bars on the left side), ii) our theoretical results (black hatched in the middle), and iii) approximations based on results from [171] which assumes independent packet loss (white dotted on the right side) [162].

highlights the substantial impact of survival time on availability. As another example, in the 80-user scenario, the 97% availability can increase to as high as 99.4% when $N_{sv} = 3$, which may be sufficient for applications with more flexible requirements, such as plant asset management. Figure 2.3 indicates that the assumption of independent packet losses may result in severe optimism on the application's performance approximation. For instance, in Figure 2.3a, the approximation based on independent failures for the 80-user scenario with a survival time (T_{sv}) of 2 ms yields an availability of 0.999, sufficient for positioning traffic. However, actual availability, as determined by simulations, falls short of 0.99. Figure 2.3a and Figure 2.3b demonstrate that our model provides a reasonable approximation of communication service performance, except for the 50-user scenario when survival time exceeds 4 ms. Our thorough analysis revealed that the simulation duration contributed to the discrepancy between simulated and theoretical outcomes in this case. In the 50-user scenario, we ran 600-second simulations 250 times, totaling less than 87 days of simulation time. However, validating reliability over ten years is not feasible with this type of discrete event-based simulation, as excessive detail leads to extended run times.

2.4 Summary

Modern cyber-physical control applications can endure occasional packet loss. In this chapter, we presented the application of fundamental reliability theory metrics to wireless communication scenarios and derived a function that maps communication service performance to conventional network metrics and vice versa. Our model incorporates PER, packet loss burst length, and the application's specified survival time. Our approximation's advantages are twofold. Firstly, it simplifies the network design process by estimating the operational performance of the communication service, reducing the need for extensive and time-consuming simulations. Secondly, it equips the network controller to balance the tradeoff between communication service performance and utilization during operation dynamically, leveraging the real-time performance approximations it delivers.

Chapter 3

Enabling URLLC in Single Service Scenarios

Wireless communication systems supporting cyber-physical production applications must meet strict communication service requirements. Diversity techniques and power control are prominent and widely employed approaches to reduce latency and increase reliability. However, they may result in redundant transmissions and overuse of resources. In this chapter, we investigate DRL algorithms to improve operational KPIs for URLLC service. In Section 3.2, we design a DRL orchestrator for power control and HARQ retransmissions to optimize these reliability KPIs. To tackle scalability issues in per-device orchestrations, we develop a new BSAC framework in which a separate branch represents the action space of each industrial device. Our orchestrator allows for near real-time control and can be deployed in the edge cloud. In Section 3.3, we validate our solution with a realistic, and 3GPP-compliant simulator for factory automation scenarios. Section 3.4 delves into the practical considerations of implementing a DRL algorithm in real-world wireless communication systems. Finally, Section 3.5 summarizes this chapter.

3.1 Problem Formulation

We reuse the system model introduced in Section 2.1. However, we focus on DL transmissions (i.e., Γ=DL) within a 5G deployment with several gNBs, where each gNB corresponds to a number of cells. We consider a set of cells $\mathcal{C} := \{\mathrm{CL}_1, \mathrm{CL}_2, \ldots, \mathrm{CL}_{|\mathcal{C}|}\}$ in the network. Each cell CL_i serves a set of devices $\mathcal{U}_i := [U_i]$, with U_i as the total number of devices per CL_i, and $U := \sum_i U_i$ as the total number of URLLC devices. We use diversity techniques and power control to enhance CPS reliability. Let N_f be the number of diversity techniques we can control. Examples of a technique include HARQ retransmissions and packet duplication at the PDCP layer. We can compactly show the diversity setting for jth device in cell i as

$$\boldsymbol{f}_{ij} := \left(f_{ij}^{(1)}, f_{ij}^{(2)}, \ldots, f_{ij}^{(N_{\mathrm{f}})}\right), \tag{3.1}$$

where $f_{ij}^{(\iota)}$ is the setting for a diversity technique such that $f_{ij}^{(\iota)} \in \{1, 2, \ldots, N_\iota\}$, where $\iota \in \{1, 2, \ldots, N_{\mathrm{f}}\}$, and N_ι denotes the maximum diversity transmissions that the diversity technique ι can have.

The goal for the system is to find the configuration of diversity setting and transmission power level (i.e., $\boldsymbol{f}_{ij}$ and p_{ij} for $\forall i \in \mathcal{C}, \forall j \in \mathcal{U}_i$) in order to maximize the communication service reliability and availability of a CPS. We define the orchestration problem for a given set of reliability enhancement techniques as follows:

$$\max_{\boldsymbol{f}_{ij}, p_{ij}} \eta_\alpha \sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{U_i} \alpha_{ij} + \eta_\tau \sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{U_i} \tau_{ij}, \tag{3.2a}$$

$$\text{s.t.} \quad p_{ij} \in (0, p_{\max}], \ \forall i \in \mathcal{C}, \forall j \in \mathcal{U}_i, \tag{3.2b}$$

$$f_{ij}^{(k)} \in \{1, 2, \cdots, N_\iota\}, \ \forall \iota \in \{1, 2, \cdots, N_{\mathrm{f}}\}, \forall i \in \mathcal{C}, \forall j \in \mathcal{U}_i, \tag{3.2c}$$

where $\eta_\alpha > 0$ and $\eta_\tau > 0$ in (3.2a) denote the scaling coefficient of availability and reliability, respectively. Furthermore, (3.2b) represents the maximum allowable transmission power to jth device in cell i, while (3.2c) ensures that diversity transmissions are non-negative integers within a certain limit. It is worth noting that both (3.2b) and (3.2c) should be respected at all decision time epochs. In (3.2) and the following equations in this section, we omit Γ notation from Section 2.2 as our focus is only on DL here.

The communication service availability and reliability of a system are functions of the channel state variable, $X_{ij}(n)$, as well as the end-to-end delay of packets at receiving devices (see equations (2.1) to (2.7)). Channel state variable and end-to-end delay depend on many variables such as instantaneous path gain, SINR, code rate, and transmission buffer status where each of these variables can be impacted by $\boldsymbol{f}_{ij}$ and p_{ij}. Hence, joint modeling of the channel state variable and end-to-end delay is a highly-complex task and requires significant assumptions on the channel, traffic, and queue models.

3.2 An RL-Powered Orchestrator

In this section, we concentrate on using DRL to orchestrate the DL transmission powers and HARQ retransmissions in a flexible manner.

3.2.1 Transformation to Deep RL Problem

In (3.2), determining the impact of variables $\boldsymbol{f}_{ij}$ and p_{ij} on our reliability KPIs requires in-depth models of the channel and queue, which may contain approximations that are not necessarily accurate in real-world scenarios. Given the complexity of the optimization problem outlined in (3.2) and the difficulty in obtaining precise models for its components, the use of RL becomes a compelling option.

This framework can be seen as a closed-loop control problem in which the agent takes actions and learns their impacts based on the feedback received from the environment. Therefore, we propose to formulate the orchestration problem defined in (3.2) as an infinite horizon MDP. The state space, $\mathcal{S}$, action space, $\mathcal{A}$, and set of all possible rewards, $\mathcal{R}$, are determined by the dynamic interactions between the orchestrator and the RAN environment. However, it is not possible to calculate the transition probability function ($p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$) in our complex and dynamic environment. To address this challenge, in Section 3.2.1.1, we present a solution that employs BSAC to solve the infinite horizon fully observable discounted MDP problem.

State Space, $\mathcal{S}$: The state at iteration k, $\boldsymbol{s}_k$, comprises both explicit and implicit QoS variables. On the former, and as described in Section 2.2, communication service availability and reliability are a function of PER, network outage duration (i.e., the mean time that Y_{ij} is zero), and survival time for the specific use case. Accordingly, we propose including the PER and the mean packet loss burst length to each device in the state space, while it is not necessary to incorporate the fixed survival time parameter into the model. These metrics can be estimated from empirical measurements within the RL step period, Δt_k (i.e., $t_{k+1} - t_k$). For the implicit QoS variables, we determine the following metrics as having the most impact on the network's QoS, either directly or indirectly: i) path gain (g_{ij}), ii) SINR (β_{ij}), iii) end-to-end delay (D_{ij}), iv) RLC buffer status (q_{ij}), v) the number of used resource blocks, and vi) the number of diversity transmissions. For the first three metrics (i-iii), the empirical distribution (i.e., $\widehat{F}_x(\Delta t_k)$ for $x \in \{\beta_{ij}, g_{ij}, D_{ij}\}$) justly characterizes the environment. Hence, we use certain statistics of these measures, specifically the 5th and 95th percentile, median, and mean. The remaining measures (iv-vi) are count metrics, for which a simple mean suffices in the state representation.

Action Space, $\mathcal{A}$: The action space consists of the decision parameters the RL agent uses to interact with the environment. In practice, many RL frameworks quantize the continuous transmit power into several discrete levels [172]. We define the set of power levels as $\mathcal{P} := \{p_{\min}, p_1, p_2, \cdots, p_M, p_{\max}\}$. Subsequently, the action space for each device, which includes diversity transmissions, becomes $\mathcal{A} := [N_1] \times [N_2] \times \cdots \times \left[N_{N_\mathrm{f}}\right] \times \mathcal{P}$, where $[N] := \{n | n \in \mathbb{N}, n \leq N\}$, and $\times$ represents Cartesian product.

The Reward Function: From (2.5) and (2.7), we understand that communication service availability and reliability are measured in infinite time, whereas the temporal granularity of the RL control loop is defined by a step period, Δt_k. Consequently, we propose incorporating estimators of these two measures into the reward function. Even though estimating these long-term measures over a short step period may lead to inaccuracies, such an estimation effectively reflects the impact each action has on the communication service performance in the short-term future. The availability estimator can be defined as

$$\bar{\alpha}_{ij}(\Delta t_k) := \frac{1}{\Delta t_k}\int_{t_k}^{t_k+\Delta t_k} Z_{ij}(t)dt. \tag{3.3}$$

Nevertheless, as reliability is measured in time units, it can not be reliably estimated over brief time intervals. Thus, we define crossing rate from $Z_{ij}(t) = 1$ to $Z_{ij}(t) = 0$ as $l_{ij} := \lim_{T\to\infty} N(T)/T$. Leveraging (2.5) and (2.7), we have

$$\tau_{ij}=\frac{\alpha_{ij}}{l_{ij}}, \tag{3.4}$$

where l_{ij} can be estimated at each step period by $\bar{l}_{ij}(\Delta t_k) := N(\Delta t_k)/\Delta t_k$. We propose to use $\bar{a}_{ij}(\Delta t_k)$ and $\bar{l}_{ij}(\Delta t_k)$ in the reward function instead of α_{ij} and τ_{ij}. Therefore, we transform (3.2a) to maximize the availability estimator and minimize the crossing rate. We define the reward function for kth RL iteration as

$$r_k := \frac{1}{\nu U}\left(\nu\sum_{i=1}^{|C|}\sum_{j=1}^{U_i}\bar{\alpha}_{ij}(\Delta t_k) - (1-\nu)\sum_{i=1}^{|C|}\sum_{j=1}^{U_i}\bar{l}_{ij}(\Delta t_k)\right). \tag{3.5}$$

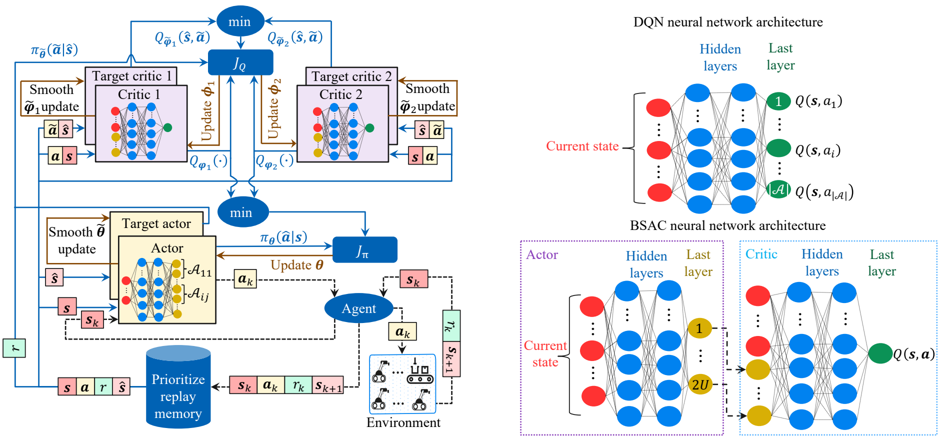
In this expression, ν is the weight specifying the relative importance of availability and crossing rate. The term $1/\nu U$ scales the reward function to have an upper bound of 1. Moreover, the step period of RL is denoted by Δt_k at the end of which the reward is calculated at time t_{k+1}. For example, in our evaluations, we set Δt_k to one second, gather availability and crossing rate statistics in this one-second interval, and compute the reward at time $t + 1$ using these statistics.

3.2.1.1 Branching Soft Actor-Critic

One of the main challenges of DQN and SAC is their poor scalability. In the following, we present our NN architectural enhancement of SAC, *BSAC*, to tackle this issue.

Our initial step involves utilizing continuous action in actor-critic algorithms, which we then quantize into discrete values representing the settings for all devices in the factory. In this setting, we implemented our orchestrator using both deep deterministic policy gradient (DDPG) [173] and SAC, which is a stochastic policy gradient method. Our evaluations suggest that SAC outperforms DDPG in terms of convergence rate and the average reward for our orchestration problem. However, both algorithms' performance dramatically declines as the number of devices increases.

To address the scalability issue concerning the size of action space (which grows exponentially with the number of industrial devices in DQN), we introduce BSAC in which the action representation of each device is distributed across distinct neurons. Consequently, our BSAC solution can orchestrate a significantly higher number of



(a) (b)

Figure 3.1: The detailed architecture of BSAC algorithm in (a), and the simplified neural network architecture comparison between DQN and BSAC, in (b). Almost all other discrete action space DRL methods have similar architecture to DQN for their actors [94].

devices within a CPS environment. Figure 3.1a illustrates the detailed BSAC architecture in which double Q-learning, target critics, and target actor are used, while Figure 3.1b depicts the simplified NN architecture of both DQN and BSAC. As Figure 3.1 shows, in the actor's last layer, we dedicate two neurons to the action space of each device, $\mathcal{A}_{ij}$, representing the mean and standard deviation of a Gaussian distribution of an element in $\widehat{\boldsymbol{a}}$. Therefore, we can reformulate $J_\pi(\boldsymbol{\vartheta})$ by replacing $\boldsymbol{a}$ with $\widehat{\boldsymbol{a}}$ in (1.11). Finally, we quantize each continuous element in $\widehat{\boldsymbol{a}}$ to get the selected action for each industrial device. For example, if the action set is identical on all industrial devices (i.e., $\mathcal{A}_{\mathrm{u}} := \mathcal{A}_{ij}, \forall i \in \mathcal{C}, \forall j \in \mathcal{U}_i$), the action space in our problem becomes $|\mathcal{A}| = |\mathcal{A}_{\mathrm{u}}|^U$. Thus, the final layer of our BSAC actor only comprises of $2U$ neurons, in comparison to the NN in DQN which would require $|\mathcal{A}_{\mathrm{u}}|^U$ neurons (refer to Figure 3.1b). This decomposition of device actions allows for the linear growth of the NN's last layer size as the number of devices increases, significantly reducing the solver's computational and memory complexities.

3.3 Selected Results

We modeled the environment as a small factory with dimensions of $15 \times 15 \times 11\,\mathrm{m}^3$. Similar to simulations in Section 2.3.1, we employed a modified indoor Hotspot model incorporating the blockage model B from [170]. Moreover, we assumed that a gNB is located at the factory's center with a 10 m height and is configured with a 3-sector cell setting. We assumed the bandwidth allocated to URLLC traffic, W_{u}, is 10 MHz, unless otherwise stated. The devices are arranged to emulate high-interference scenarios within the factory. The control application's traffic ingress to devices is modeled based on [71] and assumed to be periodic with a 2 ms cycle, and delay tolerance of 2.5 ms (i.e., $D_{ij}^{\text{req}} = 2.5\,\text{ms}, \forall i \in \mathcal{C}, \forall j \in \mathcal{U}_i$).

To evaluate the performance of our proposed orchestrators, BSAC and DQN (represented as `bsacOrch` and `dqnOrch`, respectively), we compared them against the following baselines.

- `vanillaScheduler`: All resource blocks were configured with p_{max}, while setting the maximum number of transmissions to 2.
- `doubleBandwidth`: We increased the bandwidth from 10 MHz (default value) to 20 MHz and configured all resource blocks with p_{max}, while setting the maximum number of transmissions to 2.
- `doubleTransPower`: We increased the total output power from 0.5 W (default value) to 1 W, while setting the maximum number of transmissions to 2.

We trained both of the orchestrators (i.e., `dRlOrch` and `bsacOrch`) on a server with Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz, 8 cores and 64 GB of random access memory. To ensure a fair comparison, we trained both orchestrators for one million iterations, and as our step period is set to 1 simulation second, we ran 10000 simulations, each for 100 simulation seconds. Besides, we utilized the same random seed for both the exploration and exploitation phases in both orchestrators.

Figure 3.2a presents the empirical cumulative density function (eCDF) of communication service availability. Each sample in the distribution represents the availability of each device over a 100-second simulation duration. This figure indicates that our `bsacOrch` successfully enhances the 5th percentile device availability to 0.99982, i.e., achieving a three nines availability. This availability level indicates that a factory operating under the `bsacOrch` framework can fulfill the three nines availability with a 0.95 probability (i.e., $\Pr(\alpha_{ij} \leq 0.999) \leq 0.05$, as described in (2.6)). Such availability represents an 18.3% increase over the 5th percentile availability offered by `vanillaScheduler`. Interestingly, the `doubleBandwidth` benchmark, which employs twice the bandwidth, only achieves an 18.1% increase in the 5th percentile of availability compared to the `vanillaScheduler` (i.e., `bsacOrch` achieves this gain with half the bandwidth, significantly saving operational network costs). Our `bsacOrch` even manages to outperform `doubleTransPower`, which employs twice the transmission power, by a margin of 0.5% in terms of 5th percentile availability. This gain can be attributed to the efficient management of the trade-offs between delay, power, and packet loss, as also highlighted in delay and resource block distributions (see [5]).

Figure 3.2b depicts the mean and 5th percentile device availability of both `vanillaScheduler` and `bsacOrch` for different numbers of devices. This figure reveals that the performance gain of `bsacOrch` over `vanilla-Scheduler` in terms of mean and 5th percentile availability grows with the increase in the number of devices. Such growth is likely due to its increased freedom to perform interference

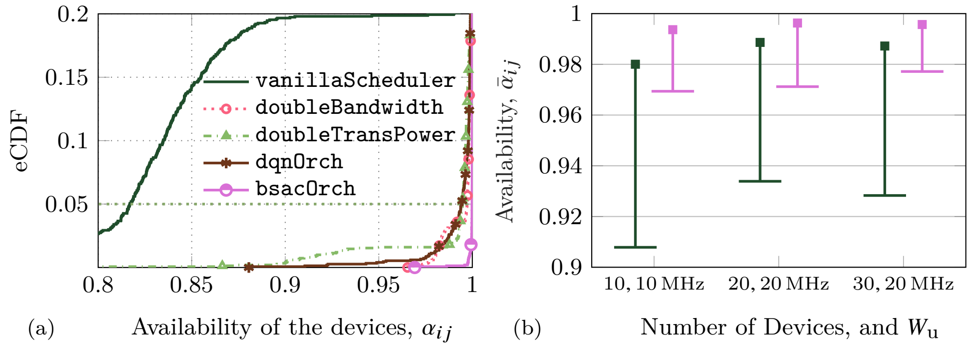


Figure 3.2: Comparison of `bsacOrch` in terms of communication service availability of the devices. (a) depicts the empirical CDF of `bsacOrch` and baselines with 5 devices. The horizontal dotted line in (a) shows the 5th percentile availability. (b) depicts the mean (square) and 5th percentile (line) availability for `vanillaScheduler` (in green) and `bsacOrch` (in purple) (adapted from [94]).

management. Our solution's scalability allows it to tackle larger optimization problems that challenge baseline approaches. Notably, we could not run `dqnOrch` for more than 5 devices on our high-performance server due to its hefty computational and memory demands (refer to [94] for more detailed descriptions).

Figure 3.3 demonstrates the transmission energy consumption of our `bsacOrch` compared to `vanillaScheduler`, `doubleBandwidth`, and `doubleTransPower`. Recall that the last two baselines showed a similar level of performance to `bsacOrch` regarding communication service availability and reliability. The results in Figure 3.3 indicate that `bsacOrch` consumes 1.9x energy compared to `doubleBandwidth` but utilizes only half the bandwidth. Conversely, it consumes 52% less energy than `doubleTransPower` while offering superior availability and reliability. These outcomes suggest that by orchestrating diversity techniques and power control, we can achieve the level of availability and reliability that state-of-the-art solutions reach with either double bandwidth or energy consumption. Notice that we did not account for energy consumption during training as `bsacOrch` is trained once until a significant change happens. In many factory automation use cases, such drastic changes seldom occur; thus, we barely need to retrain the agent.

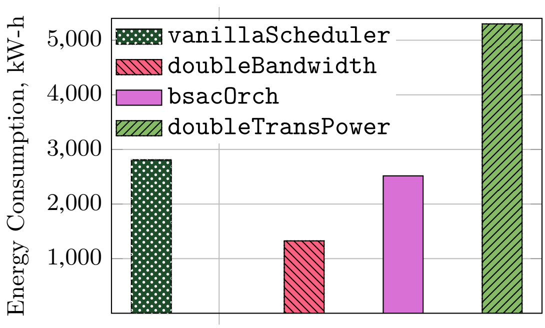


Figure 3.3: Energy consumption of `bsacOrch` compared to baselines (adapted from [94]).

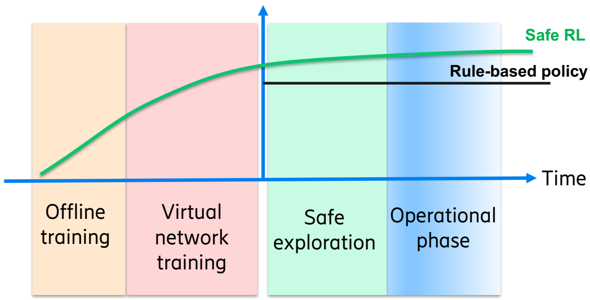


Figure 3.4: Efficient first-time model deployment using offline training and training via a virtual network, outside ordinary operational phase.

3.4 Implementation in Real Deployments

There are two main challenges in implementing a DRL algorithm in real-world wireless communication systems. First, unlike other learning scenarios, such as image classification and recommendation systems, obtaining vast amounts of data with sufficient random explorations is often challenging. Hence, many RL frameworks, including ours, rely on simulations, which may not accurately mirror the dynamics of actual systems. Training on the operational network is also costly due to the random explorations in DRL algorithms. Second, in the case of URLLC applications that require extreme system performance, the infrequent occurrence of rare events, such as communication service failures, during training can result in suboptimal decision making when deployed. In the following sections, we present our perspective on tackling these challenges.

3.4.1 From Simulation to Real Deployment

MLOps refers to a set of practices focused on deploying and managing ML models in an operational environment in a reliable and efficient manner [174]. It extends the well-known DevOps concept commonly used in software engineering. ML models, including RL agents, are developed and tested in isolated experimental systems before being deployed in the operational environment. In existing telecom use cases, RL models are only retrained upon detecting model or data drift, which can lead to a decrease in KPIs. This approach may not be feasible for use cases with strict requirements, such as URLLC. For instance, applying our orchestration solution in RANs may prove challenging due to the non-stationary environment, leading to frequent model and data drift, which in turn necessitates frequent retraining of potentially heavy DRL algorithms. On the other hand, if retraining is disabled, suboptimal decision making may follow, which is not a desirable outcome.

To tackle the challenges associated with implementing RL algorithms in real-world deployments, one potential solution involves a pre-operational three-step process. Such a process involves initial training during the development phase,

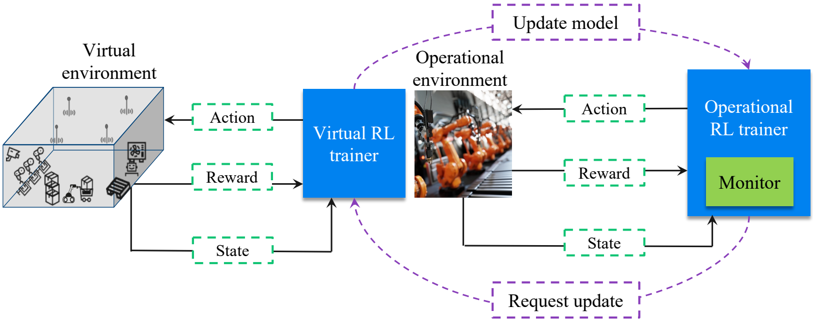


Figure 3.5: The general architecture of the continuous update of operational RL model. The internal monitor block of the operational RL trainer tracks the KPIs and initiates model update requests as needed.

fine-tuning in a pre-launch initialization phase, and minor adjustments throughout a controlled operational phase [175], somewhat similar to the concept displayed Figure 3.4. However, this approach might not be viable for scenarios requiring continuous deployment.

To address this issue, we suggest combining offline, virtual, and safe exploration phases continuously, as demonstrated in Figure 3.5. The learning process outlined in (1.9)-(1.11) supports both off-policy and offline training. In off-policy learning, where interactions with the environment are possible, new experienced transitions get added to the buffer, with several transitions from the buffer updating the new policy. However, the behavior policy may differ from the target policy. On the other hand, offline training employs a dataset collected via an unspecified behavior policy and does not involve any interactions with the environment. Yet, the actors and critics can still undergo training via on-policy learning. In such a case, the algorithm only accesses the most recent transition (i.e., there is no buffer), and consequently, the actions are sampled and executed based on the latest policy estimate (i.e., the behavior policy and target policy are the same). This characteristic is particularly useful when dealing with URLLC services, where stringent requirements cannot be compromised. It facilitates the training of DNNs in a number of ways: i) using an existing dataset in offline mode, ii) employing a virtual network (e.g., digital twin or realistic simulations) in off-policy mode, while the episodes can run in parallel to speed up the learning procedure, and iii) utilizing the operational network (e.g., in safe exploration mode) for further fine-tuning of the DNNs' weights. Figure 3.4 graphically represents the potential effectiveness of this procedure when being applied to the operational environment.

3.4.2 Generating the Tail

Training on datasets with a dominant head distribution and a long tail distribution can be problematic. The model parameters are frequently overwritten due to

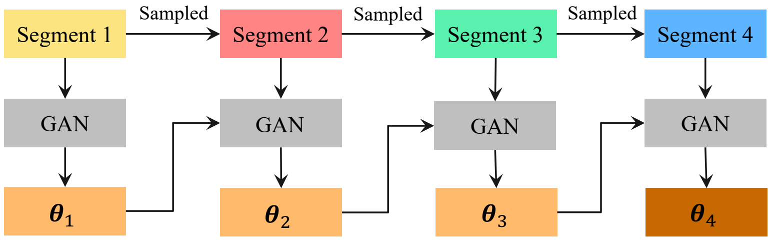


Figure 3.6: The training procedure of IL-GAN [165].

updates from the head distribution, resulting in a model that aligns well with the head distribution but overlooks the tail. Therefore, the conventional GANs that undergo training on the entire dataset typically tend to mimic the head distribution accurately while neglecting the tail of the distribution. To address this issue, we adapt the incremental learning approach introduced in [176] for image classification and segmentation tasks to enhance generative models in learning and generating data from the tail of the distribution.

The fundamental concept of incremental learning involves learning in segments, where the acquisition of new segments does not result in forgetting previous ones. The basic algorithm for IL-GAN works as follows: the dataset is partitioned into segments, with the first segment representing the head distribution (i.e., the segment with the highest number of samples). The model is then trained on the data from the first segment until a specific convergence criterion is met or for a predetermined number of epochs. Once training on the first segment is completed, the process continues with the next segment, and so on, until the model is trained on the data from all segments. Figure 3.6 illustrates the architecture of IL-GAN. The focus of incremental learning is on mastering new data segments. We propose *balanced replay* and *freezing* techniques during the training of GANs to mitigate the issue of catastrophic forgetting.

Balanced replay is a sampling strategy that ensures a balance between data from past segments and the current segment in training. This approach enables the model to learn and represent all previously observed segments, including the current one, effectively modifying its weights to retain information from old segments while learning new ones. As a result, the model can generate data across all segments, even those with few samples. Because the old segments are continuously represented, there should be no catastrophic forgetting.

Freezing of the nodes in NNs is a well-known technique for preserving a feature representation of a data batch [177]. Freezing prevents certain parts of the model from being updated during backpropagation [178]. The idea is to maintain some of the representation learned in each segment, helping the model remember old classes even with less representation during training on new segments. Various freezing methods have been proposed in the literature, including layer freezing, vertical freezing, or gradually changing vertical freezing [179]. Nevertheless, we

Table 3.1: Jensen-Shannon Divergence For Channel Estimate Dataset

Segment	Percentage of data	cWGAN JSD	IL-GAN JSD
1	54%	0.0211	0.0193
2	23%	0.0524	0.0107
3	22%	0.0636	0.0184
4	0.3%	0.27	0.0548
5	0.0002%	1.0	0.264

adopt the approach of incremental freezing of vertical layers. Initially, all weights are unfrozen, but after training the first segment, some weights and biases between the input and first layers are frozen. After training on the second segment, some weights between the first and second layers are frozen. This process continues until the training on the data from the last segment is completed. The motivation behind incremental freezing is to keep an equal representation of each segment in the model. This method enables uniform data generation from all segments, including the tail.

3.4.2.1 Selected Results

To evaluate the performance of IL-GAN, we employed the Jensen-Shannon divergence (JSD), a metric for comparing probability distributions. JSD is symmetric and bounded to be in $[0, 1]$, where 0 indicates that the distributions are the same, and 1 implies that there are no common elements in them. Additionally, we chose the conditional Wasserstein GAN from [180] as the baseline for our training and generation process. We then applied the procedure described in Section 3.4.2 to enhance it for IL-GAN. Besides, we normalized the data to the range $[0, 1]$ and utilized the Sigmoid activation function in the generator's output nodes to generate continuous variables.

Table 3.1 presents the data distribution among different segments and the JSD for a simulated channel estimate dataset. We used the same setting as in Section 3.3 for our simulations. This table shows a clear head distribution in the channel estimate dataset. The first segment has more than half of the data points in the dataset, and the first three segments include over 99% of the data. As this table confirms, although conditional Wasserstein GAN performs well in the first three segments, its performance dramatically declines in the last two segments. Nevertheless, IL-GAN performs better on all segments, and most notably, compared to conditional Wasserstein GAN, it reduces the JSD considerably on the last two segments and overall, decreases the average JSD of all segments by 73%.

3.5 Summary

In this chapter, we designed a DRL orchestrator for power control and retransmissions, and proposed BSAC framework for scalable device orchestration. We evaluated our solution using a realistic, 3GPP-compliant simulator. Our results validate that our proposed solution facilitates energy and spectrum savings compared to legacy methods, while still achieving the same level of performance for URLLC service. On the big picture, our feature orchestrator can be implemented as an inner loop to inter-slice management techniques, aiding the outer loop orchestrator by requiring fewer resources to satisfy the stringent demands of URLLC service. Additionally, this chapter explores the practical considerations for implementing an DRL algorithm for URLLC service in real-world wireless communication systems.

Chapter 4

Enabling URLLC in Mixed Services Scenarios

Recently, many new 6G distributed ML applications have been developed to boost efficiency and improve system performance, such as fault prediction in industrial automation [158], and autonomous driving [181]. The remarkable performance of AI models largely stems from the exhaustive training they undergo, which relies heavily on the vast amount of data collected by edge devices. This chapter focuses on understanding and optimizing the coexistence of distributed learning and URLLC services. In Section 4.1, we introduce a soft synchronous distributed learning protocol in which the central node distributes the global model updates upon receiving local updates from a subset of the available devices. Additionally, we present the system model and distributed learning metric, namely convergence time. In Section 4.2, we formulate the coexistence problem considering the demands of both the distributed learning and URLLC services. We then propose a DRL-based framework for efficient device selection to address the formulated problem and evaluate its performance via a realistic 5G simulator. Besides, as a complementary approach, we propose an HRL strategy to address the problem by combining device selection with network-wide slicing. Finally, this chapter is summarized in Section 4.3.

4.1 Coexistence With Distributed Learning Service

In this section, we delve into key issues with executing distributed learning over wireless systems, describe the setup for our coexistence scenario, and introduce the metrics for evaluating distributed learning service.

4.1.1 Distributed Learning over Wireless Communications

The primary hurdles in implementing distributed learning over wireless networks stem from its two main characteristics: i) the dynamics of wireless propagation, which can be affected by various factors such as noise, interference, and fading, and ii) the scarcity of resources [182]. The latter becomes even more pressing as distributed training often involves exchanging large models or model parameters multiple times [182]. From a communication service standpoint, several studies have focused on resource management and, more explicitly, device selection techniques to improve the performance of distributed learning in terms of training loss or convergence time (e.g., [161, 183–186]). For instance, [185] examines the impact of resource management, device selection, and transmit power on the convergence of federated learning (FL) and optimizes these wireless parameters to minimize the FL training loss. Reference [186] introduces a FL algorithm tailored to manage data from heterogeneous devices. Assuming a strongly convex and smooth loss function, this algorithm optimizes the resource allocation to improve its convergence. Reference [161] indicates that, when selecting devices, considering both the updates from devices and their channel conditions can enhance the performance of distributed learning. However, none of these works account for the scenario where the performance of distributed learning is impacted not just by the limitations of the wireless network but also by the demands of the higher priority service.

4.1.2 System Model

In addition to the described system model in Section 2.1 where we consider a set of $\mathcal{U}:=[U]$ industrial devices in the factory hall, we assume that the AI devices are distinct from the industrial devices, and there exist a set of $\mathcal{N}:=[N]$ AI devices. To update the global model at each iteration, the AI central node needs to receive relevant information from n out of the N AI devices. To counteract the straggler effect, the central node may request updates from $\mathcal{N}_{m,k} \subseteq \mathcal{N}$ at iteration k, where $|\mathcal{N}_{m,k}|=m_k(\geq n)$ (i.e., the central node may request m_k-n additional backup devices, mitigating the straggler issue in a synchronous distributed learning scenario).

To model a distributed learning workflow, we consider a network of N AI devices that cooperatively solve a distributed learning problem. Assuming that $\mathcal{N}_{n,k} \subseteq \mathcal{N}_{m,k}$ is the subset of size n whose updates the central node receives first at iteration k, then iteration k of an abstract distributed algorithm reads:

$$\boldsymbol{w}_{k+1}=A\big(\boldsymbol{c}_{i,k}, \boldsymbol{w}_k\big), \quad \text{for} \quad \forall i \in \mathcal{N}_{n,k} \tag{4.1a}$$

$$\boldsymbol{c}_{i,k}=C_i(\boldsymbol{w}_k), \quad \text{for} \quad \forall i \in \mathcal{N}_{m,k} \tag{4.1b}$$

where function A represents an algorithm update of the decision variable $\boldsymbol{w}_k$, function C_i picks out the relevant information, $\boldsymbol{c}_{i,k}$, that node i uploads to the server to run the algorithm. This general algorithmic framework covers many ML algorithms, including FL and distributed gradient descent (DGD), with or without data compression. For example, when C_i returns a stochastic gradient, say $\widehat{\nabla} f_i(\boldsymbol{w}_k)$, and $A = \boldsymbol{w}_k - \eta \sum_i \widehat{\nabla} f_i(\boldsymbol{w}_k)/n$ for some positive step size η, we recover n-sync and synchronous DGD for $n(<N)$ and $n(=N)$ [187], respectively. When C_i returns updated local model parameters of AI device i and A takes an averaging step over a subset of $n(\leq N)$ AI devices, we recover FL (n-sync or synchronous). Without loss

of generality, and for the sake of simplicity, we assume that the gradients' noise is independent and identicaly distributed [188].

4.1.3 Distributed Learning KPI: Convergence Time

The convergence time of the distributed learning can be described by two metrics: training delay and the number of iterations to reach certain optimality.

We denote the selected devices at iteration k by an indicator vector of $\mathcal{N}_{m,k}$ as $\boldsymbol{\pi}_k^{\mathrm{u}}=\left[\left[\boldsymbol{\pi}_k^{\mathrm{u}}\right]_1, \left[\boldsymbol{\pi}_k^{\mathrm{u}}\right]_2, \ldots, \left[\boldsymbol{\pi}_k^{\mathrm{u}}\right]_N\right]$, where $\left[\boldsymbol{\pi}_k^{\mathrm{u}}\right]_i \in\{0,1\}, \forall i \in \mathcal{N}$. Although the central node requests a subset $\mathcal{N}_{m,k}$ (i.e., $\mathbf{1}^T \boldsymbol{\pi}_k^{\mathrm{u}}=m_k$) to participate in the training, it waits only for $n(\leq m_k, \forall k \in [K])$ local gradients/models, then the training delay at the central node for the kth iteration, d_k^{AI}, can be derived as

$$d_k^{\mathrm{AI}}\left(\boldsymbol{\Pi}_k^{\mathrm{u}}\right) := \min\left\{\min_{\substack{\mathcal{N}_{n,k} \subseteq \mathcal{N}_{m,k}, \\ |\mathcal{N}_{n,k}|=n}} \left(\max_{i \in \mathcal{N}_{n,k}}\left(d_{i,k}^{\mathrm{D}}+d_{i,k}^{\mathrm{pr}}+d_{i,k}^{\mathrm{U}}\right)\right)+d_k^{\mathrm{pr}}, T^{\max}\right\}, \tag{4.2}$$

where $\boldsymbol{\Pi}_k^{\mathrm{u}} := \left[\boldsymbol{\pi}_1^{\mathrm{u}}, \ldots, \boldsymbol{\pi}_k^{\mathrm{u}}\right]$ is the device selection matrix, $d_{i,k}^{\mathrm{D}}$, $d_{i,k}^{\mathrm{pr}}$, and $d_{i,k}^{\mathrm{U}}$, are the delay of global model's DL transmission, local training (represented in (4.1b)), and UL transmission of local gradients/models for kth iteration of the ith device, respectively. It is worth noting that $d_{i,k}^{\mathrm{D}}$ and $d_{i,k}^{\mathrm{U}}$ contain the transmission processing, payload transmission, occurred retransmissions, and queuing delay, and thus, are a function of $\boldsymbol{\Pi}_k^{\mathrm{u}}$. Moreover, d_k^{pr} is the kth iteration processing delay required to perform the global model update (4.1a) on the central node. Hence, (4.2) computes the maximum aggregated communication and processing delay for each subset of $\mathcal{N}_{n,k}$ with n elements. The subset with the lowest delay is then selected to determine $d_k^{\mathrm{AI}}(\cdot)$. Nevertheless, to prevent an indefinite waiting time at the central node, we set a limit on the maximum allowable delay per iteration, denoted as $T^{\max}$. Figure 4.1 shows the training delay in a n-sync distributed training scheme.

The distributed learning service proceeds until the global training loss function reaches a specified level of optimality. By $K_{\min}$, we represent the minimum number of iterations required for the distributed learning algorithm to converge. To determine $K_{\min}$, we can find "critical points" as the set of points where the norm of their derivative is 0. The local minima and maxima are a subset of these points. In general, under some regularity assumptions, we can often converge to an approximate critical point (a point wherein the norm of the gradients gets smaller than some positive ϵ)[1]. We use the first-order necessary condition to determine these critical points by searching for points where gradients disappear. In reality, for the sake of computational efficiency, we assume the global model converges when the approximate first-order necessary condition is satisfied, and we terminate the

[1]Note that a critical point may not be an optimal point, e.g., a saddle point. However, in the case of a convex function, a critical point is the global minima.

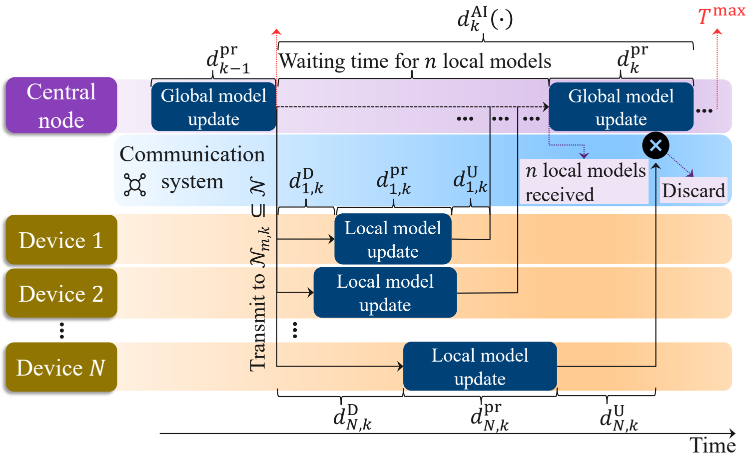


Figure 4.1: The illustration of training delay in distributed training workflow [167].

algorithm accordingly. Hence, $K_{\min}$ can be formally defined as

$$K_{\min} := \min \left\{ k \middle| \; \mathbb{E}\left[\left\| \widehat{\nabla} f(\boldsymbol{w}_k) \right\|_2 \right] \leq \epsilon, k \in \mathbb{N} \right\}, \tag{4.3}$$

where ϵ is a small positive constant. In our experiments, we estimate the expectation term numerically utilizing the Monte Carlo method. The literature studies on convergence analysis reveal that, in n-sync schemes, an increase in the required number of devices participating in the global update (i.e., n) leads to a decrease in $K_{\min}$ (e.g., convex and non-convex objectives for DGD algorithm in [189], or non-convex objective functions for FL in [190]). In [167], we provide several examples in which it is possible to theoretically approximate $K_{\min}$ for DGD and FL algorithms considering factors such as the smoothness, strong/non-convexity of the objective functions, and other relevant technical assumptions.

4.2 Enhancing Distributed Learning Procedure

Based on the findings presented in Section 4.1, it is evident that device selection algorithms that jointly consider the performance of distributed learning and the coexisting URLLC service are of utmost importance. Additionally, introducing dynamic slicing approaches that optimize bandwidth allocation for both the AI slice and URLLC slice can potentially enhance the coexistence performance.

4.2.1 Device Selection

The primary objective is to enable the coexistence of URLLC and distributed learning services over a wireless network by employing a device selection scheme. To achieve this, we consider the definition of $d_k^{\mathrm{AI}}(\mathbf{\Pi}_k^{\mathrm{u}})$ in (4.2), the availability requirement in (2.6), and assume an n-sync scheme for distributed training procedure. The joint optimization problem for distributed learning over a wireless network can thus be formulated as follows:

$$\min_{\mathbf{\Pi}_{K_{\min}}^{\mathrm{u}}} \sum_{k=1}^{K_{\min}} d_k^{\mathrm{AI}}(\mathbf{\Pi}_k^{\mathrm{u}}), \tag{4.4}$$

$$\text{s.t.} \quad \Pr\left\{\alpha_i^{\Gamma} \leq \alpha_i^{\mathrm{req}}\right\} \leq \gamma, \ \ \forall i \in \mathcal{U}, \forall \Gamma \in \{\mathrm{UL}, \mathrm{DL}\}, \tag{4.4a}$$

$$\mathbf{1}^{\intercal}\boldsymbol{\pi}_k^{\mathrm{u}} \geq n, \ \ \forall k \in [K_{\min}], \tag{4.4b}$$

$$\boldsymbol{\pi}_k^{\mathrm{u}} \in \{0,1\}^N, \ \ \forall k \in [K_{\min}]. \tag{4.4c}$$

To fulfill the required number of local updates in the global update of (4.1) in the n-sync scheme, (4.4b) ensures that the central node selects at least n AI devices. Yet, it can select any number of backup devices to tackle the straggling issue (i.e., $m_k = \mathbf{1}^{\intercal}\boldsymbol{\pi}_k^{\mathrm{u}}$, and $m_k \in [n, N], \forall k \in [K]$). Moreover, (4.4c) indicates that the device selection policy is a binary vector. Note that (4.4b) and (4.4c) must be respected at all decision time epochs.

The communication service availability and training delay are a function of the channel state variable, $X_i^{\Gamma}(t)$, and the end-to-end delay of each packet (see Section 2.2). Besides, due to URLLC service's high priority, the amount of URLLC load impacts the distributed training delay considerably.

4.2.2 Transformation to MDP Problem

In optimization problem (4.4), characterizing the effect of $\boldsymbol{\pi}_k^{\mathrm{u}}$ on our KPIs necessitates explicit modeling of the channel and queues, involving approximations that may not be valid in practice. Hence, we propose to model the device selection problem (4.4) as a finite horizon MDP. Nevertheless, as in Section 3.2, it is not viable to derive the transition probabilities in our complex environment. Accordingly, towards solving our device selection problem, we leverage BSAC (described in Section 3.2.1.1) to address the finite horizon fully observable discounted MDP problem.

While transforming optimization problem (4.4) into an MDP problem, it is important to consider that the two services operate on different timescales. Specifically, the performance of URLLC is measured based on the actual time, whilst each time step in the control loop corresponds to the duration of one iteration of the distributed training algorithm (i.e., $d_k^{\mathrm{AI}}(\cdot)$ for kth iteration). Consequently, the control loop is not periodic in actual time and is triggered by the central node. To highlight the time instants when iteration k begins and ends, we use Δt_k (i.e., $t_{k+1} - t_k$) as necessary. In the following, we describe the key components of the MDP formulation.

4.2.2.1 State Space, $\mathcal{S}$

We categorize the environment's state at kth iteration (i.e., $\boldsymbol{s}_k \in \mathcal{S}$) into three classes i) URLLC related observations, ii) AI related observations, and iii) gNBs level observations. In the following, we describe these three classes.

URLLC QoS variables: As described in Section 2.3, communication service availability of each URLLC device, as the main URLLC KPI, is a function of PER, network outages' duration, and survival time. Therefore, the state should include both (UL/DL) PER and (UL/DL) mean packet loss burst length, estimated via empirical measurements within Δt_{k-1}. Along with these metrics that explicitly impact the communication service availability, cell association, (UL/DL) buffer size (at t_k), (UL/DL) SINR, and (UL/DL) delay are other measures that implicitly affect the availability. However, SINR and delay statistics might vary significantly during, possibly long, Δt_{k-1}. Hence, we represent their distribution using specific statistics of these measures, i.e., 1st percentile, 5th percentile and median of the SINR distribution, and 95th percentile, 99th percentile, and median of the delay distribution. Utilizing such percentiles is well motivated by URLLC extreme availability performance, which, under proper system design, is affected by the tail of delay and SINR distributions [57].

Distributed learning delay variables: The training delay of each iteration, $d_k^{\mathrm{AI}}(\boldsymbol{\Pi}_k^{\mathrm{u}})$, is a function of the device selection; accordingly, we include a binary variable in the state indicating if this device has been selected in the last iteration. As indicated by (4.2), the delay of the DL transmission of the global model, $d_{i,k}^{\mathrm{D}}$, and the delay of the UL transmission of local gradients/models, $d_{i,k}^{\mathrm{U}}$, directly impacts the training delay, and thus, should be included in the state. Besides, (UL/DL) buffer size (at t_k), and (UL/DL) SINR of the underlying transmissions have an implicit impact on delay, so we also include them in the state. To focus on the overall statistics, we describe the SINR distribution of the underlying transmissions for each AI device using its 5th percentile, median, and 95th percentile. It is important to note that there is no available empirical measurement for $d_{i,k}^{\mathrm{D}}$, $d_{i,k}^{\mathrm{U}}$, and SINR of AI devices whose central node does not receive their local models in iteration k (i.e., if $i \notin \mathcal{N}_{n,k}$). However, if an AI device was chosen at t_{k-1} but its local update is not among the first n received updates, its buffer size at t_k will not be empty.

gNB level observations: On the gNB level, the number of resource blocks each service has consumed in both UL and DL directions has a substantial impact on both training delay and availability. Consequently, we incorporate the mean number of allocated resource blocks (per slot), within Δt_{k-1}, for each service in the state.

4.2.2.2 Action Space, $\mathcal{A}$

In accordance with (4.4), the action vector at iteration k should be the device selection vector $\boldsymbol{\pi}_k^{\mathrm{u}}$. Nevertheless, to eliminate selections that do not follow condition (4.4b), we define $\boldsymbol{a}_k$ as a continuous vector where each element describes the action for an AI device (i.e., $\boldsymbol{a}_k \in [-1, 1]^N$). Then, the mapping from $\boldsymbol{a}_k$ to $\boldsymbol{\pi}_k^{\mathrm{u}}$ can be performed through

$$\boldsymbol{\pi}_k^{\mathrm{u}} := \begin{cases} \mathbb{1}\{\boldsymbol{a}_k \geq \mathbf{0}\}, & \text{if } \mathbf{1}^\intercal \mathbb{1}\{\boldsymbol{a}_k \geq \mathbf{0}\} \geq n, \\ \mathbb{1}\{\boldsymbol{a}_k \geq a_k^{(n)} \mathbf{1}\}, & \text{otherwise}, \end{cases} \tag{4.5}$$

where $a_k^{(n)}$ is the nth largest element in vector $\boldsymbol{a}_k$.

4.2.2.3 The Reward Function

The communication service availability of URLLC devices, as defined in (2.5), is evaluated over an infinite time, while the granularity of the DRL algorithm is determined by the duration of each distributed training round, Δt_k. As a result, we incorporate the availability estimator from Equation (3.3) as part of the reward function. Although estimating long-term availability based on a short period may not be entirely accurate, it effectively represents the device selection outcome from the application's perspective. Based on the optimization objective in (4.4) and the constraint in (4.4a), the kth iteration reward is defined as

$$r_k := \upsilon \exp\left(\zeta \min\left\{ \min_{\substack{i \in \mathcal{U}, \\ \Gamma \in \{\mathrm{UL},\mathrm{DL}\}}} \left(\hat{\alpha}_i^{\Gamma}(\Delta t_k) - \alpha_i^{\mathrm{req}}\right), 0 \right\}\right) + (1-\upsilon)\, \frac{T^{\max} - d_k^{\mathrm{AI}}(\boldsymbol{\Pi}_k^{\mathrm{u}})}{T^{\max}}, \tag{4.6}$$

where υ ($\in [0, 1]$) is the weight distinguishing the relative importance between URLLC service's reward and distributed learning service's reward. In the URLLC reward, $\left(\hat{\alpha}_i^{\Gamma}(\Delta t_k) - \alpha_i^{\mathrm{req}}\right)$ is negative for those URLLC devices that did not meet their corresponding availability requirement within Δt_k, regardless of the transmission direction, Γ. Accordingly, our reward function design enforces the DRL agent to maximize the availability of the worst URLLC device among those that do not meet their availability requirements. Moreover, ζ is a design parameter which is a function of the sensitivity (λ), and the precision that the maximum availability requirement needs (i.e., $\zeta \propto \max_i \left(\alpha_i^{\mathrm{req}}\right)/\lambda$). The device selection policy gets the full reward on the URLLC part (i.e., υ) when all devices fulfill their availability requirements. For the distributed learning reward, the shorter $d_k^{\mathrm{AI}}(\cdot)$ is, the larger the $\boldsymbol{\pi}_k^{\mathrm{u}}$'s reward becomes. Moreover, to minimize the tail of the per-device availability distribution and the average training delay, URLLC reward decreases exponentially while the reduction in distributed learning reward is linear.

The learning procedure for our device selection solution in the coexistence scenario using BSAC is outlined in Algorithm 1. In our design, the device selection policy for each device is determined by the actions represented by two neurons in the actors' last layer (refer to Figure 3.1), and then transformed via (4.5).

4.2.3 Selected Results

As in previous chapters, we considered a factory automation scenario where URLLC and distributed learning services coexist. To elaborate, we designed a 3D model

Algorithm 1: SAC-Based Algorithm for Device Selection in the Coexistence of URLLC and Distributed Learning

Input: Set of AI devices $\mathcal{N}$, Required number of local models in global update n;
Output: Device selection policy as a function of RAN state;
Initialize: φ_1, φ_2, and $\boldsymbol{\vartheta}$ and set $\tilde{\varphi}_1 \leftarrow \varphi_1$, $\tilde{\varphi}_2 \leftarrow \varphi_2$, and $\tilde{\boldsymbol{\vartheta}} \leftarrow \boldsymbol{\vartheta}$, $k \leftarrow 1$;
▹ `episode: numbers of iterations during which distributed training converges`
foreach *episode* **do**
  Set initial device selection from previous virtual training or random;
  while *true* **do**
    Receive $\boldsymbol{c}_{i,k}$ from n AI devices, or $T^{\max}$;
    Observe $\boldsymbol{s}_k$ (measured within Δt_{k-1});
    if $d_k^{\mathrm{AI}} < T^{\max}$ **then** Compute $\boldsymbol{w}_{k+1}$ as in (4.1a);
    else $\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k$;
    Sample an action, $\boldsymbol{a}_k \sim \boldsymbol{\pi}_{\boldsymbol{\vartheta}}(\cdot|\boldsymbol{s}_k)$;
    Transmit $\boldsymbol{w}_{k+1}$, via RAN, to selected AI devices ($\boldsymbol{\pi}_k^{\mathrm{u}}$, derived from (4.5));
    Observe $\boldsymbol{s}_{k+1}$, and calculate r_{k+1} via (4.6);
    if *distributed learning converges* **then** $I_{k+1} \leftarrow 0$;
    else $I_{k+1} \leftarrow 1$;
    Store $(\boldsymbol{s}_k, \boldsymbol{a}_k, r_k, \boldsymbol{s}_{k+1}, I_{k+1})$ in $\mathcal{B}$;
    if $I_{k+1} = 0$ **then** break;
    $k \leftarrow k + 1$;
  end
  ▹ `Training DNNs`
  if $|\mathcal{B}| \geq |\mathcal{B}|_{\min}$ ***and*** $k = l|\mathcal{B}_{\mathrm{mb}}|, \forall l \in \{1, 2, \ldots\}$ **then**
    Randomly sample a mini-batch $\mathcal{B}_{\mathrm{mb}}$ from $\mathcal{B}$;
    forall $(\boldsymbol{s}, \boldsymbol{a}, r, \hat{\boldsymbol{s}}, I) \in \mathcal{B}_{\mathrm{mb}}$ **do**
      Derive $\widehat{Q}$ using (1.10), where $\tilde{\boldsymbol{a}} \sim \boldsymbol{\pi}_{\tilde{\boldsymbol{\vartheta}}}(\cdot|\hat{\boldsymbol{s}})$;
    end
    $\varphi_i \leftarrow \varphi_i - \frac{1}{|\mathcal{B}_{\mathrm{mb}}|} \sum_{\forall b \in \mathcal{B}_{\mathrm{mb}}} \widehat{\nabla} J_{\mathrm{Q}}(\varphi_i)$, for $i \in \{1, 2\}$;
    $\boldsymbol{\vartheta} \leftarrow \boldsymbol{\vartheta} - \frac{1}{|\mathcal{B}_{\mathrm{mb}}|} \sum_{\forall b \in \mathcal{B}_{\mathrm{mb}}} \widehat{\nabla} J_{\boldsymbol{\pi}}(\boldsymbol{\vartheta})$, where (1.11) uses $\widehat{\boldsymbol{a}}$;
    $\tilde{\varphi}_i \leftarrow \nu \varphi_i + (1-\nu) \tilde{\varphi}_i$ for $i \in \{1, 2\}$;
    $\tilde{\boldsymbol{\vartheta}} \leftarrow \nu \boldsymbol{\vartheta} + (1 - \nu) \tilde{\boldsymbol{\vartheta}}$;
  end
end

of a small factory of size $40 \times 40 \times 10\,\mathrm{m}^3$ with 4 gNBs at the height of 8 m, and with an inter-site distance of 20 m. Unlike previous chapters, we applied the recent 3GPP channel model for the indoor factory with dense clutter and high base station height (InF-DH) use case, where the gNB and devices are respectively positioned higher and lower than the average clutter height [170]. We considered RLC in acknowledged mode for distributed learning traffic, while RLC was configured in unacknowledged mode for URLLC flow (due to slow RLC retransmissions). Besides, we employed strict priority scheduling, where URLLC flow had higher priority than AI flow, implying that AI packets are only scheduled when no URLLC packets are in the queues.

For the performance evaluation of our algorithm (shown as `dRlAgent` in the figures), we set up 20 URLLC devices. At each simulation, the URLLC devices appear at random positions and move in 1D in a random direction at a speed of 30 km/h within a short distance of that position. We compared our proposed solution with three types of baselines:

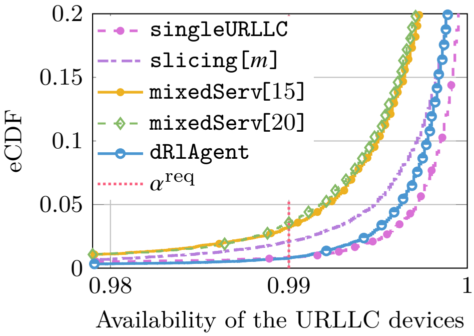
- `singleURLLC`: We did not set any distributed training traffic in our industrial automation scenario, and the URLLC devices leveraged the entire 40 MHz bandwidth.

- `mixedServ[`m`]`: In addition to URLLC devices, we had a total of 50 AI devices. We kept the required number of devices that the central node waits for constant (i.e., n=15), and randomly picked a set of m participating devices (i.e., $m_k = m, \forall k \in \mathbb{N}$), where $m \in \{15, 20, 30, 40, 50\}$.

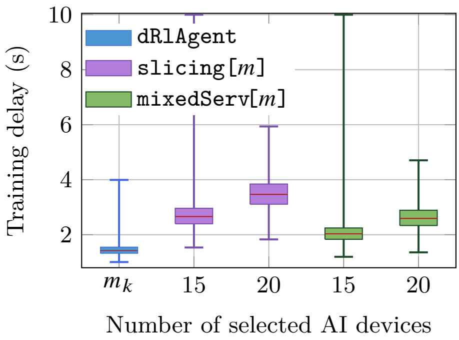
- `slicing[`m`]`: We assigned 25% of resources to the URLLC service (i.e., 10 MHz bandwidth and 0.125 W for total DL transmit power) and the rest to the distributed learning service. We kept U, N, n the same as `mixedServ[`m`]` baseline. Besides, we randomly picked a set of m participating devices (i.e., $m_k = m, \forall k \in \mathbb{N}$), where $m \in \{15, 20\}$.

In our scenario, the `singleURLLC` baseline represents the best possible performance on the URLLC availability. In `dRlAgent`, and to calculate the reward in (4.6), we set υ and ζ to 0.5, and 100, respectively. Moreover, we assumed that all URLLC devices serve a single use case, and thus, set the availability requirement to 0.99 (i.e., $\alpha^{\text{req}} = \alpha_i^{\text{req}} = 0.99, \forall i \in \mathcal{U}$). Note that distributed training does not progress if less than n local models are collected by the central node within a time duration of T^{max}, and thus, RL iterations could differ from distributed learning iterations. However, if there is no strict latency limitation from distributed learning tasks, we can set T^{max} adequately large to ensure that time-outs happen seldom. Although our RAN simulator can handle such situations, for simplicity, we set T^{max}=10 s.

Figure 4.2 shows the distribution of our main KPIs. Figure 4.2a demonstrates the eCDF of URLLC devices' availability, where each sample is the UL or DL availability of one URLLC device in one simulation during its entire simulation time. In `slicing[`m`]`, due to the slicing of the bandwidth and gNB transmission power, the availability distribution is identical for any arbitrary m. As this figure suggests, compared to `singleURLLC` and `slicing[`m`]`, the availability of the URLLC devices decreases in `mixedServ[15]` and `mixedServ[20]`, likely due to the inter-cell interference introduced by AI devices. Although the scheduler adjusts the modu-



(a)



(b)

Figure 4.2: The empirical CDF of URLLC devices' availability, $\bar{\alpha}_i^{\Gamma}$, in (a), and the distributed learning's training delay, d_k^{AI} in (b), both for the benchmark with random URLLC devices. Each box plot represents the minimum, 25th percentile, median, 75th percentile, and maximum of the observed training delay samples [167].

lation and coding scheme to deal with this additional interference[2], it still affects the availability, such that the availability requirement of 0.99 can be met with a sensitivity of around 0.03 in `mixedServ[15]` and `mixedServ[20]`, rather than 0.008 in `singleURLLC`. As Figure 4.2a suggests, the `dRlAgent` manages to maintain the availability of AI devices close to the level of `singleURLLC` up to α^{req}, even though the URLLC traffic appears at random locations. Surprisingly, `slicing[`m`]` can meet a 0.99 availability requirement with λ=0.02, which is more than twice the sensitivity `singleURLLC` and `dRlAgent` can support. It is highly likely that such lower availability performance for `slicing[`m`]` arises from scenarios where a large number of devices are associated with the same gNB.

Figure 4.2b displays the distributed training delay, with each box representing the minimum, 25th percentile, median, 75th percentile, and maximum of the observed training delay samples. As our findings in [166] suggest, the training delay increases as the number of selected devices grows. Such an increase can also be observed for both `slicing[`m`]` and `mixedServ[`m`]` boxes. Furthermore, the lower training delay statistics in `mixedServ[15]` and `mixedServ[20]` compared to `slicing[15]` and `slicing[20]` respectively, indicates that the distributed learning service generally demands more resources than what is dedicated to it in `slicing[`m`]`. As this figure confirms, our `dRlAgent` improves the median training delay at least by 30% (compared to `mixedServ[15]`), and decreases the maximum observed training delay at least by 15% (compared to `mixedServ[20]`). Hence, our `dRlAgent` successfully orchestrates the distributed training traffic in two ways: i)

[2]Note that this decrease in availability happens regardless of the scheduler's configuration. For instance, a higher target block error rate cannot overcome the additional interference, and conversely, a lower target block error rate induces additional delay. Both scenarios contribute to decreased availability.

it minimizes the impact on the availability of the URLLC devices, ensuring their requirements are negligibly affected, and ii) it achieves lower training delay statistics.

4.2.4 An Example of Inter and Intra-Slice Orchestration

Now let us denote the set of resources by $\mathcal{L} := [L]$ (e.g., bandwidth, power, and processing capability), the set of slices by $\mathcal{V} := [V]$, and the ratio of resources that are allocated to slice v by $\boldsymbol{\rho}^v \in [0,1]^L, \forall v \in \mathcal{V}$. Then, network slice subnet management should always keep the following condition:

$$[\boldsymbol{\rho}^v]_l \leq 1, \;\; \forall l \in \mathcal{L} \tag{4.7}$$

where $[\boldsymbol{\rho}^v]_l$ denotes the element l of vector $\boldsymbol{\rho}^v$, and (4.7) implies that the ratio of an arbitrary resource allocated to all slices should be less than (e.g., reserved for emergency service) or equal to one. Thus, we can define the slicing matrix as $\boldsymbol{P} := [\boldsymbol{\rho}^{\mathrm{U}}, \boldsymbol{\rho}^{\mathrm{AI}}, \ldots]$. However, for simplicity, we only consider the coexistence of URLLC and AI services here, where each service is assigned a single slice. Hence, we have $\boldsymbol{\rho}^{\mathrm{U}} = 1 - \boldsymbol{\rho}^{\mathrm{AI}}$. The objective of the network-wide slicing problem aligns with Equation (4.4). However, apart from the device selection policy, the large timescale slicing decisions, denoted by $\boldsymbol{P}$, should also be optimized. Consequently, we have a two-tier orchestration problem, as described below.

- On the high-level orchestration, both services are taken into consideration. Hence, the objective is to minimize the sum of distributed learning service's training delays and fulfill the stringent requirements of URLLC via efficient slicing of the network resources as a high-level decision variable.
- On the low-level orchestration, we propose an optimization method that uses device selection to minimize (4.4). Note that $K_{\min}$ is only a function of the low-level decision variable (i.e., device selection), while the training delay depends on both high-level and low-level decision variables.

Figure 4.3 presents the overall architecture of our HRL framework and its interconnections. As this figure shows, there are five main components in this architecture:

- data processing unit, which collects the data from the environment;
- translation entity, which maps the QoS related data to URLLC service's operational KPIs (e.g., estimations via (3.3), or approximations via (2.10) for communication service availability);
- planner, which applies the device selection policy to the operational devices;
- high-level RL orchestrator (HL-RLO), which is an RL agent that optimizes the slice resources on a large timescale, $\Delta t_{\kappa}^{\mathrm{h}}$; and

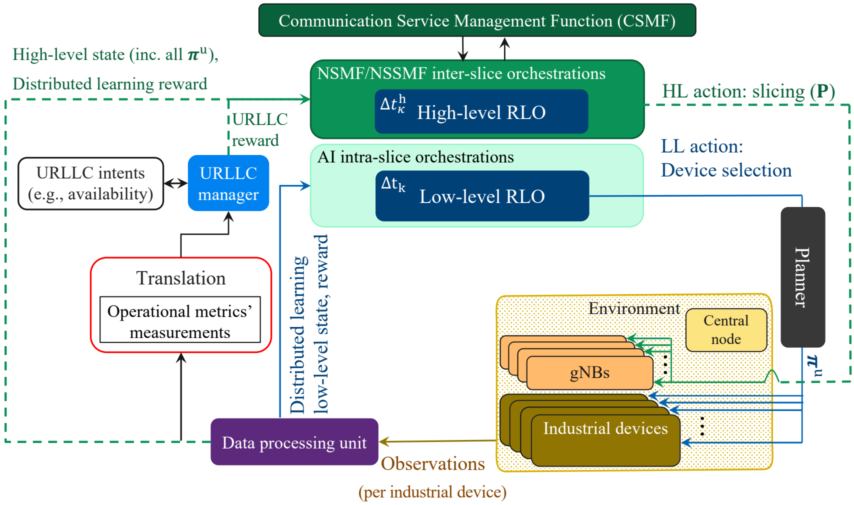


Figure 4.3: The architecture of our HRL solution for joint inter-slice and intra-slice orchestration on the coexistence of distributed learning and URLLC services [168].

- low-level RL orchestrator (LL-RLO), which is an RL agent that optimizes device selection policy relatively on a small timescale, Δt_k.

In addition to URLLC, distributed learning, and cell-level measurements described in Section 4.2.2.1, the HL-RLO state should include the device selection policies taken within its timescale, $\Delta t_\kappa^{\mathrm{h}}$. Regarding its long-term reward, the distributed learning term in (4.6) should be averaged over all iterations within $\Delta t_\kappa^{\mathrm{h}}$. In contrast, the URLLC term can be kept the same but with long-term availability estimations. Nevertheless, the LL-RLO state at each iteration includes only distributed learning delay variables in Section 4.2.2.1 and takes the distributed learning reward in (4.6).

4.3 Summary

In this chapter, we investigated the performance optimization of distributed learning service when coexisting with the URLLC service from two perspectives: i) resource sharing among services, and ii) network-wide slicing of the resources. On the former, we proposed a DRL-based framework that selects a subset of devices to perform distributed training, aiming to minimize training delay while preserving URLLC service availability. Our thorough 3GPP-compliant 5G simulations indicate that our scheme substantially reduces the median, 75th percentile, and maximum observed training delays, all while maintaining URLLC availability near levels achieved when all network resources are exclusively dedicated to URLLC. Concerning network-wide resource slicing, we proposed a HRL framework to dynamically allocate resources to these services over a larger timescale, thus realizing the benefits of increased observability and business profitability through slicing. This chapter provides valuable insights into adaptive distributed learning traffic management, contributing towards a sustainable coexistence between URLLC and distributed learning services.

Chapter 5

Conclusions and Future Work

CPSs are complex systems that integrate physical and computational components to enable advanced automation, control, and monitoring capabilities. URLLC service is a crucial enabler for CPSs, as it ensures the timely and reliable delivery of data flows among CPS's components. Together, CPSs and URLLC have the potential to create new opportunities in areas such as Industry 4.0, autonomous vehicles, and the healthcare industry. In this thesis, we investigated the operational metrics of CPSs in the context of wireless communications and developed ML methods to steer wireless systems toward fulfilling these operational requirements efficiently, especially in terms of spectrum and energy. This chapter responds to the main RQs posed in Section 1.4 by offering insightful concluding remarks. Subsequently, potential future research directions and open problems in the field are presented.

5.1 Concluding Remarks

The first contribution of this thesis is to understand the operational metrics of CPSs in terms of wireless communications by addressing the following RQs.

- **RQ1.1**: How to mathematically model 3GPP definitions of communication service KPIs for cyber-physical applications (i.e., communication service availability and reliability)?
- **RQ1.2**: How could a network designer translate such communication service KPIs to meaningful network layer parameters without the need to perform computationally complex simulations to validate years of reliability?

To provide an effective communication service for a CPS, it is essential to consider its application's specific needs. In order to address these RQs, we incorporated basic principles of reliability theory in wireless communications to develop mathematical models for communication service reliability and availability. Leveraging these models, we developed a FSMC to model that tracks burst packet losses and gets an approximation of the error length distributions. The proposed framework provides a mapping function that considers the application's survival time and maps the communication service requirements to the expected network performance. Our findings reveal that assuming independent packet losses can lead to an inflated evaluation of the communication service performance, potentially resulting in unsatisfied application requirements. However, our mapping allows network designers to delicately approximate communication service performance, thus eliminating the need for complex and time-consuming simulations to validate years of reliability.

Next, we shifted our attention toward techniques to enable URLLC service for CPSs, explicitly focusing on use cases in industrial automation. Diversity techniques and power control emerged as primary strategies to reduce latency and improve the reliability of wireless communications, albeit at the cost of redundant transmissions and resource overuse. The subsequent contributions of this thesis are to answer the following RQs.

- **RQ2.1.1**: What is the impact of correlated spatial paths on end-to-end reliability in multi-connectivity scenarios when employing packet duplication? How does this influence the design of networks supporting cyber-physical applications?

- **RQ2.1.2**: How to develop an orchestrator module to manage the reliability enhancement techniques in order to fulfill stringent requirements of URLLC such that the orchestrator is energy and spectrum efficient, scalable, and aligned with current standards?

In another attempt to understand the impact of wireless systems' dynamics, our analysis for packet duplication on dual connectivity architectures showed that a small correlation among spatial paths leads to several orders of magnitude growth in end-to-end loss rate, potentially resulting in a completely different design choice. Therefore, towards addressing RQ2.1.2, we leveraged 3GPP-compliant symbol-level simulations for our evaluations. Accordingly, we proposed a DRL-powered orchestrator to dynamically control transmission power and the maximum number of diversity transmissions to maximize the communication service reliability and availability based on certain statistics, such as SINR and delay, from industrial devices. To ensure scalability, we implemented the orchestrator using BSAC, a model-free DRL algorithm that represents the action space of each device through separate branches in the actor network. The results of our extensive simulations confirmed the superior performance of the proposed solution compared to the state-of-the-art benchmarks, including the DQN based solutions. In contrast, a similar level of performance could be reached by the legacy scheduler at the cost of a double bandwidth or 1.9x transmit energy. Thus, our orchestration approach provides a promising avenue for saving energy and spectrum in enabling URLLC service for 5G and beyond 5G networks.

Towards enabling mixed services, where URLLC service coexists with distributed learning service, this thesis aims to address the following RQs through its contributions.

- **RQ2.2.1**: What are the trade-offs between URLLC's operational KPIs and distributed learning's training delay and accuracy? How does AI setting (e.g., model size or the number of participating devices) impact its own and URLLC performance?
- **RQ2.2.2**: How can we enhance the distributed training workflow using device selection to minimize the convergence time of distributed learning service while still fulfilling URLLC's stringent operational requirements?
- **RQ2.2.3**: What is the potential gain of performing joint user-selection and slicing (i.e., dedicating a portion of total bandwidth and power to each service) to minimize the convergence time of the distributed learning service, while still fulfilling the URLLC's stringent operational requirements?

To address RQ2.2.1-3, we developed a communication model for synchronous distributed training on top of our system-level simulator and studied the trade-offs between URLLC service availability and the convergence time of distributed training. The already existing 5G-NR QoS handling was employed to separate the traffic of the two services. Our results indicate that the introduction of distributed training traffic has a significant unfavorable impact on URLLC availability, which reduces with the number of participating AI devices. While an increase in the number of AI devices can improve the training accuracy of the distributed learning, we demonstrated that it could also result in an unacceptable increase in training delay. Therefore, we proposed a DRL-based solution to run distributed learning using a carefully selected subset of devices. In our framework, we incorporated an action masking technique to ensure that each iteration of distributed training involves a minimum number of AI devices while also allowing for a higher number of devices to address the straggling effect. Additionally, we employed the BSAC framework from our previous contribution to this mixed service framework and distributed the device selection policy of each AI device across distinct neurons for improved scalability. Our results indicated that our scheme can significantly reduce the total training delay while maintaining the availability of URLLC devices close to their required level without dedicating all network resources to URLLC. Finally, we proposed a HRL architecture to overcome the limitations of static slicing by dynamically allocating communication and computation resources to these services on a larger timescale, contributing towards flexible, scalable, and profitable networks.

5.2 Future Research Directions

With the integration of URLLC into CPS, we can create systems that can quickly and accurately respond to changing conditions in real-time, delivering new levels of efficiency, reliability, and convenience to our lives. In recent years, access to mobile network data has presented fresh opportunities for mobile operators and vendors. By harnessing this data, they can design, optimize, and maintain their networks more effectively than ever before. The latest advancements in ML and processing technologies enable the application of ML techniques to improve network management and performance metrics. In this thesis, we have investigated several key challenges in providing URLLC service for industrial applications. In the following, we outline three possible extensions to our work that have not been substantially explored in the existing literature.

Safe RL Deployment

Collecting vast amounts of data with sufficient random exploration is challenging in cyber-physical networks. As a result, many RL frameworks, including ours, rely on virtual environments such as simulators or digital twins. However, this approach has two potential issues. Firstly, there may be a discrepancy between the virtual and operational environment that always exists. Secondly, changes in the operational environment can cause deviations between the two environments. In Section 3.4.1, we proposed a simple pull method where the operational agent requests a model update by monitoring the URLLC KPIs. For such an approach, we need to assume that the virtual environment is always up-to-date, including the latest changes in the operational environment. However, training within a virtual environment can also be energy-intensive and necessitates retraining in the event of alterations in the operational environment. Furthermore, each update carries a cost of safe explorations for tuning the new model, necessitating a halt in industrial production. A potential research direction to tackle these challenges involves optimizing the frequency of virtual model training and developing methods to minimize the discrepancy between the two environments, thereby reducing production downtime.

Distributed learning as a Service

In this thesis, we considered distributed learning as a distinct service in a coexistence scenario and optimized the underlying trade-offs between the URLLC performance and the distributed training workflow in a wireless communication system (discussed in Chapter 4). However, our optimization was limited to device selection. An alternative approach to control the distributed learning load is quantizing the exchanged messages. This approach can reduce the communication overhead per iteration and the interference footprint, though it may necessitate additional iterations to achieve convergence. One potential future work is to develop novel approaches that can adjust the quantization level based on the distributed learning algorithm, the URLLC load, and the network's interference footprint.

Efficient Network management

In RAN, the timescale within which a decision variable should be optimized varies. In Chapter 3, we showed that large timescale control could still improve the URLLC service performance significantly. However, in the literature, most RL solutions train all the decision variables on the same timescale. On the one hand, if the decision variables are optimized at a frequency lower than the optimal frequency, it may result in poor performance in the target KPIs (e.g., availability for URLLC). On the other hand, optimizing the decision variables at a higher frequency than optimal may result in additional redundant signaling, leading to higher energy and communication resource consumption. For instance, adaptive modulation and coding scheme selection or beamforming can be designed for every time slot, while routing decisions or base station association variables may remain the same for many time slots. Forcing the variables to be optimized in every time slot unnecessarily complicates the decision space and significantly increases energy consumption, processing, and communications. Moreover, as we move further from the gNBs to the central cloud (i.e., to the right-hand side of Figure 1.2), the lower bound on the control timescale becomes tighter. However, there is more processing capability available at a lower cost. Therefore, in future work, it is crucial to consider the impact of the control loop frequency and the optimal placement of decision variables in terms of service characteristics and requirements.

Bibliography

[1] P. Marwedel, *Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things*, 4th ed. Cham, Switzerland: Springer Nature, 2021.

[2] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (IoT)," *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015.

[3] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, and S. Trimpe, "Wireless control for smart manufacturing: Recent approaches and open challenges," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 441–467, 2021.

[4] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for Industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467–3501, 2019.

[5] M. Ganjalizadeh, A. Alabbasi, A. Azari, H. S. Ghadikolaei, and M. Petrova, "An RL-based joint diversity and power control optimization for reliable factory automation," in *IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[6] T. S. Alemayehu and J.-H. Kim, "Dependability analysis of cyber physical systems," *IET Computers & Digital Techniques*, vol. 11, no. 6, pp. 231–236, 2017.

[7] Y. Du, W. Cao, J. She, M. Wu, M. Fang, and S. Kawata, "Disturbance rejection for input-delay system using observer-predictor-based output feedback control," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4489–4497, 2020.

[8] *Functional safety of electrical/electronic/programmable electronic safety-related systems*, IEC 61508, 2017.

[9] Advantech Co., Taipei, Taiwan, *2020-2021 IIoT Master Catalog*, 2021. [Online]. Available: https://advcloudfiles.advantech.com/campaign/2020-2021_IIoT_Master_Catalog/index.html

[10] M. Ganjalizadeh, P. Di Marco, J. Kronander, J. Sachs, and M. Petrova, "Impact of correlated failures in 5G dual connectivity architectures for URLLC applications," in *IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[11] "Integration of 5G with time-sensitive networking for industrial communications," 5G-ACIA, White Paper, Frankfurt, Germany, Feb. 2021. [Online]. Available: https://5g-acia.org/whitepapers/integration-of-5g-with-time-sensitive-networking-for-industrial-communications

[12] R. Baheti, H. Gill, T. Samad, and A. M. Annaswamy, "Cyber-physical systems," *The Impact of Control Technology*, pp. 161–166, 2011. [Online]. Available: www.ieeecss.org

[13] *Industrial communication networks - Profiles Part 1: Fieldbus profiles*, IEC 61784-1, 2019.

[14] *Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3*, IEC 61784-2, 2019.

[15] M. Ganjalizadeh, M. Jonsson, and K. Kunert, "Reliability enhancement for hard real-time communication in industrial packet-switched networks," in *Multiple Access Communications (MACOM)*, M. Jonsson, A. Vinel, B. Bellalta, and E. Belyaev, Eds. Cham: Springer, 2014, pp. 59–74.

[16] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008.

[17] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2016.

[18] C. Dombrowski and J. Gross, "Echoring: A low-latency, reliable token-passing mac protocol for wireless industrial networks," in *Proceedings of European Wireless 2015; 21th European Wireless Conference*, 2015.

[19] *IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN)*, IEEE 802.15.1, 2005.

[20] K.-H. Chang, "Bluetooth: a viable solution for IoT? [industry perspectives]," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, 2014.

[21] *Bluetooth Core Specification*, Bluetooth SIG, Bluetooth Specification v5.3, 2019. [Online]. Available: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=521059

[22] *Mesh Profile*, Bluetooth SIG, Bluetooth Specification v1.0.1, 2019. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457092

[23] P. Di Marco, P. Skillermark, A. Larmo, P. Arvidson, and R. Chirikov, "Performance evaluation of the data transfer modes in Bluetooth 5," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 92–97, 2017.

[24] G. Patti, L. Leonardi, and L. Lo Bello, "A Bluetooth low energy real-time protocol for industrial wireless mesh networks," in *42nd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2016, pp. 4627–4632.

[25] L. Leonardi, G. Patti, and L. Lo Bello, "Multi-hop real-time communications over Bluetooth low energy industrial wireless mesh networks," *IEEE Access*, vol. 6, pp. 26 505–26 519, 2018.

[26] P. Di Marco, P. Skillermark, A. Larmo, and P. Arvidson, "Bluetooth mesh networking," Ericsson, White Paper, July 2017. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/bluetooth-mesh-networking

[27] A. Aijaz and M. Sooriyabandara, "The tactile internet for industries: A review," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 414–435, 2019.

[28] *IEEE Standard for Low-Rate Wireless Networks*, IEEE 802.15.4, 2020.

[29] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[30] H. B. Celebi, A. Pitarokoilis, and M. Skoglund, *Wireless Communication for the Industrial IoT*. Cham, Switzerland: Springer Nature, 2020, pp. 57–94.

[31] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 377–386.

[32] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.

[33] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, 2008.

[34] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, "A critical analysis of research potential, challenges, and future directives in industrial wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 39–95, 2018.

[35] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the use of IEEE 802.11n for industrial communications," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1877–1886, 2015.

[36] L. Cariou *et al.*, "IEEE 802.11-18/1231r0: EHT draft proposed PAR," contribution to the IEEE WNG SC, Mar. 2019. [Online]. Available: https://mentor.ieee.org/802.11/documents?is_dcn=1231&is_group=0eht

[37] R. d. Vegt, B. Tian, G. Cherian, and Y. Kim, "IEEE 802.11-22/0708r3: Beyond 'be' - proposed next step," contribution to the IEEE WNG SC, Jul. 2022. [Online]. Available: https://mentor.ieee.org/802.11/dcn/22/11-22-0708-03-0wng-beyond-be-next-step.pptx

[38] C. Deng *et al.*, "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2136–2166, 2020.

[39] G. Lacalle, I. Val, O. Seijo, M. Mendicute, D. Cavalcanti, and J. Perez-Ramirez, "Analysis of latency and reliability improvement with multi-link operation over 802.11," in *IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021.

[40] E. Jeon, W. B. Lee, M. Kim, C. Chung, and J. Han, "IEEE 802.11-22/1392r0: Beamforming improvement for UHR," contribution to the IEEE UHR SG, Sep. 2022. [Online]. Available: https://mentor.ieee.org/802.11/dcn/22/11-22-1392-00-0uhr-beamforming-improvement-for-uhr.pptx

[41] G. Naik, J.-M. Park, J. Ashdown, and W. Lehr, "Next generation Wi-Fi and 5G NR-U in the 6 GHz bands: Opportunities and challenges," *IEEE Access*, vol. 8, pp. 153 027–153 056, 2020.

[42] C. Chen, H. Song, Q. Li, F. Meneghello, F. Restuccia, and C. Cordeiro, "Wi-Fi sensing based on IEEE 802.11bf," *IEEE Communications Magazine*, vol. 61, no. 1, pp. 121–127, 2023.

[43] O. Liberg, M. Sundberg, E. Wang, J. Bergman, and J. Sachs, *Cellular Internet of things: technologies, standards, and performance*, 2nd ed. Cambridge, MA, USA: Academic Press, 2019.

[44] M. Masoudi *et al.*, "Green mobile networks for 5G and beyond," *IEEE Access*, vol. 7, pp. 107 270–107 299, 2019.

[45] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, "A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 668–695, 2021.

[46] M. Tavana, E. Björnson, and J. Zander, “Range limits of energy harvesting from a base station for battery-less internet-of-things devices,” in *IEEE International Conference on Communications (ICC)*, 2022, pp. 153–158.

[47] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai, “5G radio network design for ultra-reliable low-latency communication,” *IEEE Network*, vol. 32, no. 2, pp. 24–31, 2018.

[48] H. Xu, W. Yu, D. Griffith, and N. Golmie, “A survey on industrial internet of things: A cyber-physical systems perspective,” *IEEE Access*, vol. 6, pp. 78 238–78 259, 2018.

[49] F. Alriksson, L. Boström, J. Sachs, Y.-P. E. Wang, and A. Zaidi, “Critical IoT connectivity ideal for time-critical communications,” *Ericsson Technology Review*, vol. 2020, no. 6, pp. 2–13, 2020.

[50] *NR; Physical channels and modulation*, 3GPP, TS 38.211 v17.1.0, 2022.

[51] *NR; Medium Access Control (MAC) protocol specification*, 3GPP, TS 38.321 v17.0.0, 2022.

[52] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*, 2nd ed. Cambridge, MA, USA: Academic Press, 2020.

[53] *Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) specification*, 3GPP, TS 37.324 v17.0.0, 2022.

[54] O. Teyeb, G. Wikström, M. Stattin, T. Cheng, S. Faxér, and H. Do, “Evolving LTE to fit the 5G future,” *Ericsson Technology Review*, vol. 2017, pp. 1–16, 2017.

[55] T. Dudda and A. Shapin. (2021, Feb.) A technical overview of time-critical communication with 5G NR. Ericsson Blog. [Online]. Available: https://www.ericsson.com/en/blog/2021/2/time-critical-communication--5g-nr

[56] S. Zhang, “An overview of network slicing for 5G,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[57] M. Bennis, M. Debbah, and H. V. Poor, “Ultrareliable and low-latency wireless communication: Tail, risk, and scale,” *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[58] T.-K. Le, U. Salim, and F. Kaltenberger, “An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17,” *IEEE Access*, vol. 9, pp. 433–444, 2021.

[59] A. Anand and G. De Veciana, “Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2411–2421, Nov. 2018.

[60] A. Aijaz, "Packet duplication in dual connectivity enabled 5G wireless networks: Overview and challenges," *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 20–28, Sept. 2019.

[61] ——, "Private 5G: The future of industrial wireless," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 136–145, 2020.

[62] *Service requirements for the 5G system*, 3GPP, TS 22.261 v19.2.0, 2023.

[63] *Study on NR positioning enhancements*, 3GPP, TR 38.857 v17.0.0, 2021.

[64] I. Rahman *et al.*, "5G evolution toward 5G advanced: An overview of 3GPP releases 17 and 18," *Ericsson Technology Review*, vol. 2021, no. 14, pp. 2–12, 2021.

[65] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods, and Applications.* Hoboken, NJ, USA: Wiley, 2004.

[66] E. Dubrova, *Fault-tolerant design.* New York, NY, USA: Springer, 2013.

[67] *Dependability management - Part 3-15: Application guide - Engineering of system dependability*, IEC 60300-3-15, 2009.

[68] C. E. Ebeling, *An introduction to reliability and maintainability engineering*, 3rd ed. Long Grove, IL, USA: Waveland Press, 2019.

[69] T. Hoesler, M. Simsek, and G. P. Fettweis, "Mission reliability for URLLC in wireless networks," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2350–2353, nov 2018.

[70] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1688–1692, 1999.

[71] *Service requirements for cyber-physical control applications in vertical domains*, 3GPP, TS 22.104 v18.3.0, 2021.

[72] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (URLLC)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, 2019.

[73] J. J. Nielsen, I. Leyva-Mayorga, and P. Popovski, "Reliability and error burst length analysis of wireless multi-connectivity," in *16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 107–111.

[74] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[75] L. Miclea and T. Sanislav, “About dependability in cyber-physical systems,” in *9th East-West Design & Test Symposium (EWDTS)*, 2011, pp. 17–21.

[76] *ISO/IEC/IEEE International Standard - Systems and software engineering–Systems and software assurance –Part 1:Concepts and vocabulary*, ISO/IEC/IEEE 15026-1:2019(E), 2019.

[77] D. Ohmann and G. P. Fettweis, “Minimum duration outage of wireless Rayleigh-fading links using selection combining,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2015, pp. 681–686.

[78] T. HoBler, M. Simsek, and G. P. Fettweis, “Joint analysis of channel availability and time-based reliability metrics for wireless URLLC,” in *IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206–212.

[79] T. Hößler, L. Scheuvens, N. Franchi, M. Simsek, and G. P. Fettweis, “Applying reliability theory for future wireless communication networks,” in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–7.

[80] T. Hoesler, M. Simsek, and G. P. Fettweis, “Mission reliability for URLLC in wireless networks,” *IEEE Communications Letters*, vol. 22, no. 11, pp. 2350–2353, nov 2018.

[81] T. Hossler, P. Schulz, M. Simsek, and G. P. Fettweis, “Mission availability for wireless URLLC,” in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[82] T. Hößler, “On dependable wireless communications through multi-connectivity,” Ph.D. dissertation, Technical University of Dresden, 2020.

[83] C. She, Z. Chen, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, “Improving network availability of ultra-reliable and low-latency communications with multi-connectivity,” *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5482–5496, 2018.

[84] J. J. Nielsen, R. Liu, and P. Popovski, “Ultra-reliable low latency communication using interface diversity,” *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, mar 2018.

[85] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, “Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1117–1174, 2022.

[86] I. Donevski, I. L. Mayorga, J. J. Nielsen, and P. Popovski, “Performance trade-offs in cyber–physical control applications with multi-connectivity,” *Frontiers in Communications and Networks*, vol. 2, 2021.

[87] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, “Predictive ultra-reliable communication: A survival analysis perspective,” *IEEE Communications Letters*, vol. 25, no. 4, pp. 1221–1225, 2021.

[88] A. Rao, “5G network slicing: cross domain orchestration and management will drive commercialization,” Cisco, White Paper, Sept. 2020. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/network-services-orchestrator/white-paper-sp-5g-network-slicing.pdf

[89] *Management and orchestration; Architecture framework*, 3GPP, TS 28.533 v16.6.0, 2020.

[90] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, “Experienced deep reinforcement learning with generative adversarial networks (gans) for model-free ultra reliable low latency communication,” *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 884–899, 2021.

[91] H. Basilier, J. Lemark, A. Centonza, and T. Åsberg, “Applied network slicing scenarios in 5G,” *Ericsson Technology Review*, vol. 2021, no. 2, pp. 2–11, 2021.

[92] F. Debbabi, R. Jmal, L. Chaari, and R. L. Aguiar, “An overview of inter-slice & intra-slice resource allocation in B5G telecommunication networks,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5120–5132, 2022.

[93] M. Amine, A. Kobbane, and J. Ben-Othman, “New network slicing scheme for UE association solution in 5G ultra dense HetNets,” in *IEEE International Conference on Communications (ICC)*, 2020.

[94] M. Ganjalizadeh, H. S. Ghadikolaei, A. Azari, A. Alabbasi, and M. Petrova, “Saving energy and spectrum in enabling URLLC services: A scalable RL solution,” *IEEE Transactions on Industrial Informatics*, early access, 2023.

[95] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, “Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing,” *IEEE Communications Letters*, vol. 23, no. 8, pp. 1337–1341, 2019.

[96] S. Jošilo and G. Dán, “Joint wireless and edge computing resource management with dynamic network slice selection,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1865–1878, 2022.

[97] A. Azari, M. Ozger, and C. Cavdar, “Risk-aware resource allocation for URLLC: Challenges and strategies with machine learning,” *IEEE Communications Magazine*, vol. 57, no. 3, pp. 42–48, 2019.

[98] R. Abreu, T. Jacobsen, G. Berardinelli, K. Pedersen, I. Z. Kovács, and P. Mogensen, "Power control optimization for uplink grant-free URLLC," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[99] A. Goldsmith, *Wireless Communications.* Cambridge, MA, USA: Cambridge University Press, 2005.

[100] J. Rao and S. Vrzic, "Packet duplication for URLLC in 5G: Architectural enhancements and performance analysis," *IEEE Network*, vol. 32, no. 2, pp. 32–40, 2018.

[101] H. Jang, J. Kim, W. Yoo, and J.-M. Chung, "URLLC mode optimal resource allocation to support HARQ in 5G wireless networks," *IEEE Access*, vol. 8, pp. 126 797–126 804, 2020.

[102] N. Ben Khalifa, V. Angilella, M. Assaad, and M. Debbah, "Low-complexity channel allocation scheme for URLLC traffic," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 194–206, 2021.

[103] S. Khosravi, H. S. Ghadikolaei, and M. Petrova, "Learning-based handover in mobile millimeter-wave networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 663–674, 2021.

[104] ——, "Learning-based load balancing handover in mobile millimeter wave networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–7.

[105] X. Jiang, M. Luvisotto, Z. Pang, and C. Fischione, "Reliable minimum cycle time of 5G nr based on data-driven channel characterization," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7401–7411, 2021.

[106] "O-RAN use cases and deployment scenarios," O-RAN Alliance, White Paper, Feb. 2020. [Online]. Available: https://www.o-ran.org/resources

[107] S. R. Sabuj, A. Ahmed, Y. Cho, K.-J. Lee, and H.-S. Jo, "Cognitive UAV-aided URLLC and mMTC services: Analyzing energy efficiency and latency," *IEEE Access*, vol. 9, pp. 5011–5027, 2021.

[108] A. K. Bairagi *et al.*, "Coexistence mechanism between eMBB and uRLLC in 5G wireless networks," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1736–1749, 2021.

[109] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "eMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 740–743, 2019.

[110] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 477–490, 2020.

[111] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, "URLLC and eMBB in 5G industrial IoT: A survey," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1134–1163, 2022.

[112] N. H. Mahmood, M. Lauridsen, G. Berardinelli, D. Catania, and P. Mogensen, "Radio resource management techniques for eMBB and mMTC services in 5G dense small cell scenarios," in *IEEE 84th Vehicular Technology Conference (VTC)*, 2016.

[113] C.-P. Li, J. Jiang, W. Chen, T. Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," in *European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–5.

[114] G. Pocovi, K. I. Pedersen, and P. Mogensen, "Joint link adaptation and scheduling for 5G ultra-reliable low-latency communications," *IEEE Access*, vol. 6, pp. 28 912–28 922, 2018.

[115] A. Karimi, K. I. Pedersen, N. H. Mahmood, G. Pocovi, and P. Mogensen, "Efficient low complexity packet scheduling algorithm for mixed URLLC and eMBB traffic in 5G," in *IEEE 89th Vehicular Technology Conference (VTC)*, 2019.

[116] W. Sui, X. Chen, S. Zhang, Z. Jiang, and S. Xu, "Energy-efficient resource allocation with flexible frame structure for hybrid eMBB and URLLC services," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 1, pp. 72–83, 2020.

[117] K. I. Pedersen, G. Pocovi, J. Steiner, and S. R. Khosravirad, "Punctured scheduling for critical low latency data on a shared channel with mobile broadband," in *IEEE 86th Vehicular Technology Conference (VTC)*, 2017.

[118] Y. Zhao, X. Chi, L. Qian, Y. Zhu, and F. Hou, "Resource allocation and slicing puncture in cellular networks with eMBB and URLLC terminals coexistence," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 431–18 444, 2022.

[119] M. Almekhlafi, M. A. Arfaoui, M. Elhattab, C. Assi, and A. Ghrayeb, "Joint resource allocation and phase shift optimization for RIS-aided eMBB/URLLC traffic multiplexing," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1304–1319, 2022.

[120] A. Manzoor, S. M. A. Kazmi, S. R. Pandey, and C. S. Hong, "Contract-based scheduling of URLLC packets in incumbent EMBB traffic," *IEEE Access*, vol. 8, pp. 167 516–167 526, 2020.

[121] M. Almekhlafi, M. A. Arfaoui, C. Assi, and A. Ghrayeb, "Joint resource and power allocation for URLLC-eMBB traffics multiplexing in 6G wireless networks," in *IEEE International Conference on Communications (ICC)*, 2021.

[122] M. Darabi, V. Jamali, L. Lampe, and R. Schober, "Hybrid puncturing and superposition scheme for joint scheduling of eMBB and URLLC traffic," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1081–1085, 2022.

[123] Y. Prathyusha and T.-L. Sheu, "Coordinated resource allocations for eMBB and URLLC in 5G communication networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8717–8728, 2022.

[124] K. Hiltunen, Y. Yang, and F. Chernogorov, "Impact of network densification on the performance of a non-public URLLC factory network," in *IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 1570–1575.

[125] Y. Li, M. Sheng, Y. Sun, and Y. Shi, "Joint optimization of BS operation, user association, subcarrier assignment, and power allocation for energy-efficient HetNets," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3339–3353, 2016.

[126] L.-H. Shen, S.-F. Hung, and K.-T. Feng, "Hybrid energy resource allocation for simultaneous wireless information and power transfer for green communications," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2121–2138, 2021.

[127] L.-H. Shen, C.-Y. Su, and K.-T. Feng, "CoMP enhanced subcarrier and power allocation for multi-numerology based 5G-NR networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5460–5476, 2022.

[128] B. Ojaghi, F. Adelantado, A. Antonopoulos, and C. Verikoukis, "Impact of network densification on joint slicing and functional splitting in 5G," *IEEE Communications Magazine*, vol. 60, no. 7, pp. 30–35, 2022.

[129] F. Fossati, S. Moretti, P. Perny, and S. Secci, "Multi-resource allocation for network slicing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1311–1324, 2020.

[130] M. Masoudi, Ö. T. Demir, J. Zander, and C. Cavdar, "Energy-optimal end-to-end network slicing in cloud-based architecture," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 574–592, 2022.

[131] R. A. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, and H. Flinck, "Optimization model for cross-domain network slices in 5G networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1156–1169, 2019.

[132] M. K. Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. A. López, "Resource allocation in an open ran system using network slicing," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 471–485, 2023.

[133] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34 538–34 551, 2020.

[134] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.

[135] Y. Loong Lee, J. Loo, and T. Chee Chuah, "A new network slicing framework for multi-tenant heterogeneous cloud radio access networks," in *International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, 2016, pp. 414–420.

[136] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2416–2428, 2020.

[137] H. Song, L. Liu, J. Ashdown, and Y. Yi, "A deep reinforcement learning framework for spectrum management in dynamic spectrum access," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 208–11 218, 2021.

[138] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[139] I. AlQerm and B. Shihada, "A cooperative online learning scheme for resource allocation in 5G systems," in *IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[140] T. Hu, Q. Liao, Q. Liu, D. Wellington, and G. Carle, "Inter-cell slicing resource partitioning via coordinated multi-agent deep reinforcement learning," in *IEEE International Conference on Communications (ICC)*, 2022, pp. 3202–3207.

[141] P. Yang, X. Xi, T. Q. S. Quek, J. Chen, X. Cao, and D. Wu, "How should i orchestrate resources of my slices for bursty URLLC service provision?" *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1134–1146, 2021.

[142] M. Setayesh, S. Bahrami, and V. W. Wong, "Resource slicing for eMBB and URLLC services in radio access network using hierarchical deep learning,"

IEEE Transactions on Wireless Communications, vol. 21, no. 11, pp. 8950–8966, 2022.

[143] D. Carrillo *et al.*, "Boosting 5G on smart grid communication: A smart RAN slicing approach," *IEEE Wireless Communications*, early access, 2022.

[144] J. Li, W. Shi, P. Yang, Q. Ye, X. S. Shen, X. Li, and J. Rao, "A hierarchical soft RAN slicing framework for differentiated service provisioning," *IEEE Wireless Communications*, vol. 27, no. 6, pp. 90–97, 2020.

[145] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[146] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[147] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev, "Time limits in reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 4045–4054.

[148] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1861–1870.

[149] T. Haarnoja *et al.*, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905* [cs.LG].

[150] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2016.

[151] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1587–1596.

[152] "5G for automation in industry," 5G-ACIA, White Paper, Frankfurt, Germany, Jul. 2019. [Online]. Available: https://5g-acia.org/wp-content/uploads/5G-ACIA_WP_5G-for-Automation-in-Industry_SinglePages.pdf

[153] "5G extreme requirements: End-to-end considerations," NGMN Alliance, White Paper, Wales, U.K., Aug. 2018. [Online]. Available: https://ngmn.org/wp-content/uploads/Publications/2018/180819_NGMN_5G_Ext_Req_TF_D2_2_v2.0.pdf

[154] D. Ohmann, A. Awada, I. Viering, M. Simsek, and G. P. Fettweis, "Achieving high availability in wireless networks by inter-frequency multi-connectivity," in *IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[155] *NR; Multi-connectivity; Overall description; Stage-2*, 3GPP, TS 37.340 v17.0.0, 2022.

[156] *NR; NR and NG-RAN Overall description; Stage-2*, 3GPP, TS 38.300 v17.0.0, 2022.

[157] M. Masoudi and C. Cavdar, "Device vs edge computing for mobile services: Delay-aware decision making to minimize power consumption," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3324–3337, Dec. 2021.

[158] H. Huang, S. Ding, L. Zhao, H. Huang, L. Chen, H. Gao, and S. H. Ahmed, "Real-time fault detection for IIoT facilities using GBRBM-based DNN," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5713–5722, 2020.

[159] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2020.

[160] A. Alabbasi, M. Ganjalizadeh, K. Vandikas, and M. Petrova, "On cascaded federated learning for multi-tier predictive models," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–7.

[161] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3643–3658, 2021.

[162] M. Ganjalizadeh, A. Alabbasi, J. Sachs, and M. Petrova, "Translating cyber-physical control application requirements to network level parameters," in *IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2020, pp. 1–7.

[163] M. Ganjalizadeh, A. Alabbasi, J. A. Ibanez, M. Petrova, and J. Zander, "Methods of translating cyber-physical control application requirements to radio parameters," US Patent Application 20 220 294 697A1, Filed Oct. 29, 2020. [Online]. Available: https://patents.google.com/patent/US20220294697A1

[164] M. Ganjalizadeh, A. Azari, H. S. Ghadikolaei, A. Alabbasi, and M. Petrova, "Methods and apparatus for controlling one or more transmission parameters used by a wireless communication network for a population of devices comprising a cyber-physical system," WO Patent Application WO2 022 250 604A1, Filed May 27, 2021. [Online]. Available: https://patents.google.com/patent/WO2022250604A1

[165] J. R. Baldvinsson, M. Ganjalizadeh, A. AlAbbasi, M. Björkman, and A. H. Payberah, "IL-GAN: Rare sample generation via incremental learning in

GANs," in *IEEE Global Communications Conference (GLOBECOM)*, 2022, pp. 621–626.

[166] M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Interplay between distributed AI workflow and URLLC," in *IEEE Global Communications Conference (GLOBECOM)*, 2022, pp. 4208–4213.

[167] M. Ganjalizadeh, H. S. Ghadikolaei, D. Gündüz, and M. Petrova, "Device selection for the coexistence of URLLC and distributed learning services," 2022*, arXiv:2212.11805* [cs.NI].

[168] M. Ganjalizadeh, H. S. Ghadikolaei, J. Haraldson, and M. Petrova, "Systems and methods for joint inter- and intra-slice orchestration using reinforcement learning," PCT Patent Application PCT/SE2023/050 016, Filed Jan. 12, 2023.

[169] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002.

[170] *Study on channel model for frequencies from 0.5 to 100 GHz*, 3GPP, TR 38.901 v15.0.0, 2017.

[171] Huawei and Siemens, "3GPP, S1-183134: network performance: Communication service availability," 2018. [Online]. Available: https://www.3gpp.org/ftp/tsg_sa/WG1_Serv/TSGS1_84_Spokane/Docs/S1-183134.zip

[172] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, Oct. 2020.

[173] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2016.

[174] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," 2022*, arXiv:2205.02302* [cs.LG].

[175] V. Berggren, K. Dey, J. Jeong, and B. Guldogan. (2022, May) Bringing reinforcement learning solutions to action in telecom networks. Ericsson Blog. [Online]. Available: https://www.ericsson.com/en/blog/2022/3/reinforcement-learning-solutions

[176] X. Hu, Y. Jiang, K. Tang, J. Chen, C. Miao, and H. Zhang, "Learning to segment the tail," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 042–14 051.

[177] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[178] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[179] L. F. Isikdogan, B. V. Nayak, C.-T. Wu, J. P. Moreira, S. Rao, and G. Michael, “Semifreddonets: Partially frozen neural networks for efficient computer vision systems,” in *16th European Conference on Computer Vision (ECCV)*, 2020, pp. 193–208.

[180] Y. Luo and B.-L. Lu, “EEG data augmentation for emotion recognition using a conditional Wasserstein GAN,” in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 2535–2538.

[181] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[182] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, “Distributed learning in wireless networks: Recent progress and future challenges,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, 2021.

[183] K. Wei *et al.*, “Low-latency federated learning over wireless channels with differential privacy,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 290–307, 2022.

[184] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, “Optimizing federated learning in distributed industrial IoT: A multi-agent approach,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, 2021.

[185] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021.

[186] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, “Federated learning over wireless networks: Convergence analysis and resource allocation,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 398–409, 2021.

[187] S. Dutta, G. Joshi, S. Ghosh, P. Dube, and P. Nagpurkar, “Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018, pp. 803–812.

[188] M. Cohen, J. Diakonikolas, and L. Orecchia, "On acceleration with noise-corrupted gradients," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1019–1028.

[189] L. Bottou, F. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[190] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *International Conference on Learning Representations (ICLR)*, 2020.