INVITED PAPER

Wireless Edge Computing With Latency and Reliability Guarantees

By Mohammed S. Elbamby, Cristina Perfecto, *Student Member IEEE*, Chen-Feng Liu, *Student Member IEEE*, Jihong Park, Sumudu Samarakoon, *Associate Member IEEE*, Xianfu Chen, *Member IEEE*, and Mehdi Bennis, *Senior Member IEEE*

ABSTRACT | Edge computing is an emerging concept based on distributed computing, storage, and control services closer to end network nodes. Edge computing lies at the heart of the fifth-generation (5G) wireless systems and beyond. While the current state-of-the-art networks communicate, compute, and process data in a centralized manner (at the cloud), for latency and compute-centric applications, both radio access and computational resources must be brought closer to the edge, harnessing the availability of computing and storage-enabled small cell base stations in proximity to the end devices. Furthermore, the network infrastructure must enable a distributed edge decision-making service that learns to adapt to the network dynamics with minimal latency and optimize network deployment and operation accordingly. This paper will provide a fresh look to the concept of edge computing by first discussing the applications that the network edge must provide, with a special emphasis on the ensuing challenges in enabling ultrareliable and low-latency edge computing services for mission-critical applications such as virtual reality (VR), vehicle-to-everything (V2X), edge artificial intelligence (AI), and so on. Furthermore, several case studies where the edge is key are explored followed by insights and prospect for future work.

KEYWORDS | Edge computing; edge intelligence; URLLC; vehicle-to-everything; virtual reality.

I. INTRODUCTION

The ever-increasing requirements of wireless services in Media & Entertainment (M&E) as well as in healthcare and well-being demands are transforming the way that the data are communicated and processed. Future networks are anticipated to support a massive number of connected devices requesting a variety of different services such as mobile video streaming, virtual reality (VR), and augmented reality (AR), as well as mission-critical applications. Such services require data, computation, and storage to be performed more often with ultrahigh success rate and minimal latency. Multiaccess edge computing (MEC) has emerged as an infrastructure that enables data processing and storage at the network edge as a means to cut down the latency between the network nodes and the remote servers that typically existed in cloud computing architectures [1]. Instead, edge computing can be provided as a service at the network edge to minimize the service latency and network complexity and save the device nodes' energy and battery consumption.

Edge networking in cellular systems aims to efficiently provide the required connectivity, data access, bandwidth, and computation resources to end devices [2], [3]. Edge

Manuscript received February 14, 2019; revised April 11, 2019; accepted May 5, 2019. This work was supported in part by CWC, Academy of Finland, through the CARMA Project, under Grant 294128, in part by the 6Genesis Flagship under Grant 318927, in part by the Kvantum Institute Strategic Project (SAFARI), in part by the Spanish MINECO through the Project 5RANVIR under Grant TEC2016-80090-C2-2-R, and in part by the (VTT) Academy of Finland thorough the MISSION Project under Grant 319759. The work of C. Perfecto was supported in part by the European Commission through the H2020 5G-PPP Project ESSENCE under Grant Agreement 761592. *(Corresponding author: Mohammed S. Elbamby.)*

M. S. Elbamby, **C.-F. Liu**, **J. Park**, and **S. Samarakoon** are with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland (e-mail: mohammed.elbamby@oulu.fi; chen-feng.liu@oulu.fi; jihong.park@oulu.fi; sumudu.samarakoon@oulu.fi).

C. Perfecto is with the Department of Communications Engineering, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain (e-mail: cristina.perfecto@ehu.eus).

X. Chen is with the VTT Technical Research Centre of Finland, 90571 Oulu, Finland (e-mail: xianfu.chen@vtt.fi).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 17104, South Korea (e-mail: mehdi.bennis@oulu).

Digital Object Identifier 10.1109/JPROC.2019.2917084

base stations (BSs) in the proximity of network users will not only relay content from and to the network core but will also help execute the users' processing tasks, provide customized content and computing services, and control the connectivity and interaction between the coupled network nodes.

In essence, the performance of edge computing is predominantly assessed through two main components, communication between the edge server and the end device, and the processing at the edge server. Furthermore, several optimization aspects are considered to optimize these two components. Optimizing the communication part can be explored through wireless bandwidth and power allocation, edge server selection, computation task distribution, task splitting, and partial task offloading. For the processing part, computation cycle allocation, task queuing and prioritization, joint computing, and predictive computing are critical factors to optimize the computing efficiency.

The focus of the fifth-generation (5G) cellular networks has shifted from merely increasing the data communication rate to providing service-specific performance guarantees in terms of ultrareliability and low latency. This shift is fueled by the emergence of new use cases that require genuine support to critical and latency-sensitive communication services. Nonetheless, ultrareliability and low latency are often seen as contradictory requirements [4], compeling the use of distinctive set of tools to be efficiently realized. Yet, these individually challenging per se requirements are anticipated to be met together for networks of diverse topologies and heterogeneous services.

This paper discusses the feasibility and potential of providing edge computing services with latency and reliability guarantees, supported by the enablers illustrated in Fig. 1. In particular, it first sheds light on the services that can be offered from edge computing networks. It follows by looking into how ultrareliable low-latency communication (URLLC) contributes to and benefits from edge computing. This paper proceeds by presenting the selected use cases that reflect the interplay between edge computing and URLLC. Finally, this paper ends with our concluding remarks and future works.

II. EDGE COMPUTING SERVICES

Legacy network architectures relied on centrally located and centrally controlled servers with high computational and storage powers to provide on-demand computing to network devices [5]. These servers could support a high number of network nodes over a large geographical area. However, the large distance between the cloud computing server and the end-user device results in higher service latency. Moreover, the centralized architecture limited the ability to provide context-aware service and to preserve the user data privacy. Future wireless networks are evolving toward supporting a new set of applications that require minimal latency and high level of service personalization. This motivated the shift toward distributed networking architectures where the network resources are available close to users at the network edge. Edge computing aims to provide computing, content, and connectivity services closer to the data source and consumption points. It is applicable to scenarios with different network environments and use cases. This diversity led to several implementations that did not follow the specific standard or interoperability. The European Telecommunications Standards Institute (ETSI) has been working on solving this issue by providing an efficient standardized MEC that can be integrated across several applications and service providers [6]. MEC also enables providers to deploy edge computing services on top of wireless mobile networks. This will allow cellular operators to integrate computing into the services provided to their users. In this regard, the term edge networking refers to the action and process of serving a user or device at the network edge.

A. Content at the Edge

The idea of leveraging the network edge as a content storage has gained popularity in the last few years [7]. The existing popularity patterns on the contents requested by network users motivated in developing proactive networks. A proactive server can predict popular contents, prefetch them from the core network, and have them stored and readily available at the network edge, hence cutting down delivery times once users request them. Proactive networks require efficient methods to predict the popularity of the content to be cached, as well as high storage capacity to cache this content. Edge caching not only minimizes the service latency but also the load on the backhaul network by prefetching the popular content in the off-peak times [8]–[10]. Furthermore, we envision that the notion of edge content will be extended to include new types of data that can be served from the network edge to support the new use cases. One application to which the future network edge will provide information is the distributed machine learning (ML) application. The tight latency requirements and the need for minimizing the information exchange mandate the development of distributed machine intelligence schemes in which edge servers play a major rule. Edge ML [11], [12] will allow end users to locally develop their own ML models instead of relying on centralized approaches. However, "ML applications" rely on information from other network nodes that affect their state and utility. The network edge role here will be to bring the information necessary for enhancing or complementing the local model close to the user.

B. Computing at the Edge

Processing is becoming an important commodity to cellular applications as content. The use of applications ranging from smart factory, self-driving vehicles, to VR and AR is growing day by day and is becoming more resource greedy and less latency tolerant. While part of the computing load of these applications is served using their local

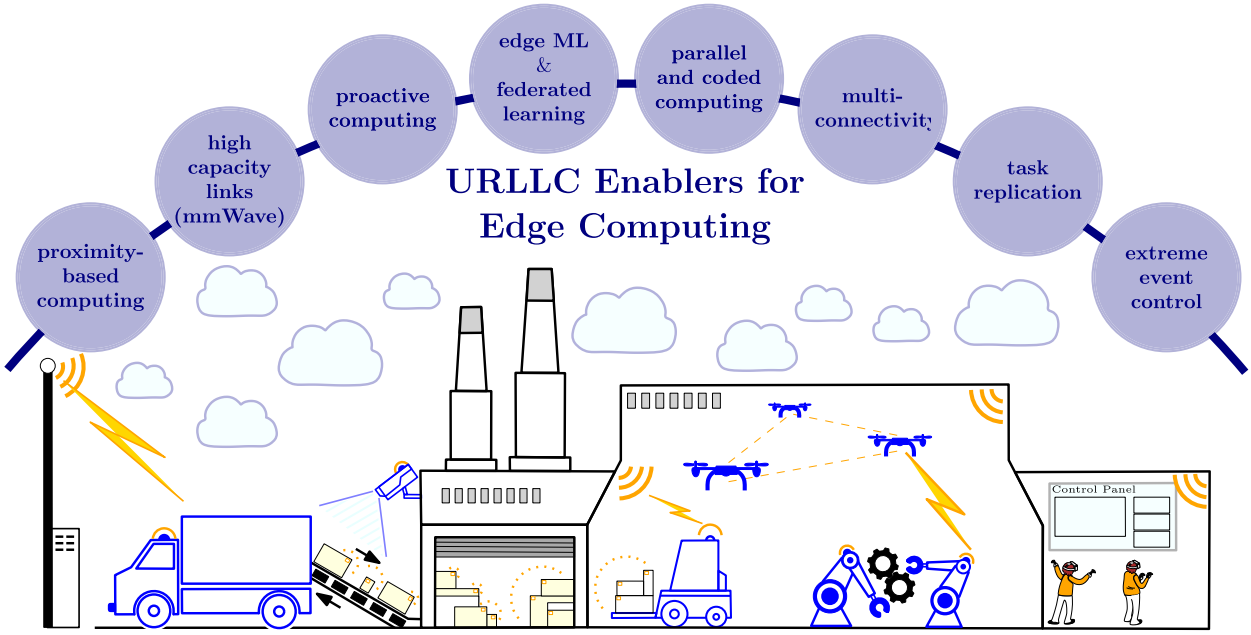


Fig. 1. ***Breakdown of key URLLC enablers for edge computing, exemplified over an Industry 4.0/Smart Factory ecosystem that includes cyber–physical systems, IoT, and MEC.***

processing units, constraints on size, portability, battery lifetime, or lack of full access to task data limit the ability to locally execute computing tasks. Edge computing promises to pool powerful yet proximate computing resources at the network edge, as well as to provide connectivity and seamless information exchange between the neighboring nodes. It is also set to allow for the realization of various 5G verticals that require low-latency and high-reliability computing, such as VR and mission-critical Internet of Things (IoT) applications. Yet, there are several components that need to be addressed to realize low-latency and high-reliable edge computing. Executing computing tasks at the edge often requires the task data to be offloaded to the edge server before execution. This introduces a communication delay that adds to the service latency. In addition, how to queue and schedule the computing tasks at the edge server plays a major role in the queuing and processing latency. Our vision is that the availability of more data and computing power will shape how the edge network performs computing. Similar in vein to proactive content caching, where knowledge of users' preferences and future interests allow for prefetching of their content, data availability and ML will help to speed up the computing tasks of network nodes. Predicting vehicles' future locations and path allows the edge network to proactively render and deliver its high-definition (HD) live map. In VR applications, predicting users' future field of view (FoV) allows rendering the corresponding part of its 360° frame with minimal latency. Several other enablers are vital to achieve ultrareliable and low-latency computing, such as task replications, parallel, and coded computing, which will be addressed in detail in Section II-C.

C. Control at the Edge

Most of the existing cloud and edge computing architectures rely on centralized decision-making schemes that require all the network nodes to send their local states data to a central controller. Instead, distributed decision-making, in which the decision-making process is distributed among the edge servers, will allow for low latency and privacy-preserving operation [13], which is essential for mission-critical applications. Indeed, the control of the network devices' performance requires policies that adapt to their local states. This can be challenging for scenarios where the local state dynamically varies due to the highly dynamic environment or due to the nature of the application, such as in mission-critical applications. Reinforcement learning (RL) solutions can provide efficient control policies that maximize the system rewards by finding policies that map those dynamically changing states into actions. These decision-making policies need to consider the effect of actions on the environment and update the reward accordingly. In centralized architectures, classical RL is often performed offline, not taking into account reliability in decision-making, for example, under noisy feedback. Edge control can provide robust decision-making, where multiagent RL architectures can be used to provide communications' efficient methods that take latency and reliability into account in dynamic and mission-critical environments. Latency stems from the local state exchanges between edge devices, in which the overhead due to the state exchange increases exponentially with the number of devices. This can be addressed using the mean-field game (MFG) theory [14], which can tackle

Table 1 Challenges and Enablers of Realizing Low Latency and High Reliability in Wireless Edge Computing

	Demands/Challenges	Enablers	MEC applications and use cases
Low latency	bandwidth, backhauls	mmWave	extended reality, vehicular edge computing (Sec. 4.1.1 and Sec. 4.1.2)
	low propagation delay	proximity based computing	deep reinforcement learning based task offloading (Sec. 4.2, use case 4)
	computing power, task dependency	parallel and coded computing	[15], [16]
	low propagation delay, energy efficiency	proactive computing	use case 6 in Sec. 4.2
	low prediction delay	edge machine learning	edge computing for federated learning (use case 1 in Sec. 4.2)
High reliability	channel intermittency	multi-connectivity, task replication	use case 6 in Sec. 4.2 and [17], [18]
	low communication cost, data privacy	federated learning	edge computing for federated learning (use case 4 in Sec. 4.2)
	rare event detection	extreme event control	extreme value theoretic edge computing and vehicular federated learning (use cases 2 and 3 in Sec. 4.2)

this by approximating the average state as a collection of agents' instantaneous states.

III. URLLC ENABLERS AND CHALLENGES

A. URLLC Overview

The prime focus of the recent groundswell of mission-critical applications, such as autonomous vehicles, immersive VR/AR experiences, industrial automation, and robotics, is to provide services with guaranteed high reliability and low latency. Therein, latency deductions in channel estimations, information exchange among the network elements, decision-making, computation tasks' completion, and memory access within devices have the utmost importance. Along with them, guaranteed low latency in operations, ensuring connectivity, and speed-precision-and-accuracy of computations are essential to assure the reliability of mission-critical applications. Due to the on-device constraints on storage, processing capability, and availability and accessibility of network resources, it is mandatory to utilize the edge servers to maintain the quality of service in mission-critical applications. To support the communication among user devices within mission-critical applications and the edge servers, URLLC, which has been introduced as one of the main services in 5G systems, plays a pivotal role. In this section, we identify the key enablers of reliability and low latency in wireless edge computing networks and the challenges toward realizing each of them. Moreover, in Table 1, we summarize the issues and enablers of providing latency and reliability guarantees in wireless edge computing networks, as well as the applications and use cases that these enablers are targeting.

B. URLLC Enablers for Edge Computing

1) Low-Latency Enablers: There are several components that contribute to latency in edge networking. In this regard, enabling low latency requires several techniques to be implemented and integrated together at different levels of edge networking systems. At the communication level, proximity-based computing and millimeter-wave (mmWave) links play major roles in reducing task offloading latency from edge devices to servers by reducing distance attenuation and providing broad bandwidth with high directionality, respectively. In addition, mmWave also enables wireless backhauling [19], [20] that facilitates edge servers' prefetching popular content with low latency. At the processing level, proactive computing provides significant latency reduction while maximizing resource efficiency by avoiding repetitive and redundant on-demand computing [17], [21], [22]. Next, coded computing is effective in reducing parallel computing latency, which eliminates the dependence of processing tasks, thereby minimizing the worst case latency due to a straggling task. Last but not least, ML is crucial in supporting low-latency mission-critical applications, by empowering edge servers and devices to locally carry out their decision-making.

Low-Latency Enabler 1 (High-Capacity mmWave Links): Driven by the spectrum shortage below 6 GHz, communications in the radio frequencies (RFs) encompassing the electromagnetic spectrum from 30 to 300 GHz, i.e., the mmWave or International Telecommunications Union (ITU)'s extremely high-frequency (EHF) band, have been attracting growing attention [23]–[25], to the point of being currently considered the most important technology to achieve the 10-Gb/s peak data rates foreseen for the upcoming 5G systems [26]. Having abundantly available spectrum, the main appeal of mmWave communications comes from the use of generous bandwidths that ranging from 0.85 GHz in the 28-GHz band to 5 GHz in the 73-GHz band are more than ten times greater than long-term evolution (LTE)'s 20-MHz cellular channel [27] and grant an important channel capacity increase [28].

However, signal propagation at these frequencies is harsh and inherently different from that at the microwave

band [29], experiencing: 1) higher pathloss for equal antenna gains due to a stronger atmospheric attenuation whereby signals are more prone to being absorbed by foliage and rain; 2) higher penetration losses as mmWaves are blocked when trying to pass through walls, buildings, or obstacles; and 3) higher transmit power consumptions than in lower bands to preserve an equal signal-to-noise ratio (SNR) unless directional antennas together with advanced signal processing that includes massive multiple-input–multiple-output (MIMO) [30] and beamforming (BF) techniques are used. Notably, due to the shorter wavelengths in mmWave bands, it is possible to pack more antennas at the transmitter and receiver devices and, thanks to the spatial degrees of freedom afforded, use analog or hybrid BF—fully digital BF implies having one dedicated RF chain per antenna which currently discourages its use in mmWaves due to the unaffordable power consumption and costs—to build a radiation pattern with narrow beams which will be subsequently steered toward the receivers, while the energy radiated through the sidelobes is minimized or negligible.

To administer high-capacity links with mmWaves, transmitters' and receivers' main lobes need to be precisely aligned toward each other if favored with a clear, unobstructed, line-of-sight (LOS) path. In practice, when a mobile user equipment (MUE) is in the connected state, uplink (UL) control channels are used to periodically feedback to the BS of its best transmit beam index; similarly, downlink (DL) control channels are used to report MUE's best transmit beams. Data transmission is then performed through the best beam pair. However, during initial access (IA) and handover, i.e., in random access, such information on the best beams is not available, which hinders taking full benefit from BF. Henceforth, in analog BF, to discover and then maintain the best transmit–receive beam pairs, a series of techniques referred to as beamtraining or beamsearching is applied. Then, beam tracking is performed to adapt the BF, e.g., due to MUE's movement leading to transmitter–receiver beam misalignments. Nevertheless, a full new directional channel discovery process will need to be triggered if the signal-to-interference-plus-noise ratio (SINR) drops below a certain threshold due to, e.g., blockages and/or interference [31]. As analog BF employs a single RF chain, it is challenging to adjust the beam to channel conditions, leading to some performance loss. Moreover, analog BF does not provide multiplexing gains as it can only operate a single data stream. Therefore, to bring all the benefits of mmWave while benefiting from multiplexing gains for MEC, MIMO hybrid BF architectures, which strike a balance between performance, complexity, and power consumption, should be considered. Finally, as adaptive BF requires precise channel state information (CSI), one of the key challenges for mmWave to work as a low-latency enabler for MEC lies on the availability of expedited CSI acquisition schemes together with directionality-aware mobility and beam management procedures [32].

In Section III-B2, a series of reliability enablers will be discussed to reduce the delay incurred to counteract the intermittent blockages and temporal disruptions of the mmWave channel. Largely, these techniques are in line with the idea of overbooking radio resources as protection against channel vulnerability [33] or to consider risk-sensitive approaches [34].

Low-Latency Enabler 2 (Proximity-Based Computing): Reducing the distance between the application and the MEC server is a key latency enabler. This idea is motivated by the concept of bringing the transmitter and the receiver closer to one another yielding capacity improvements [24]. With the low proximity between the application and the MEC server, over-the-air latency that has a significant contribution to the end-to-end (E2E), sometimes dominating the computing latency, can be greatly minimized.

Network densification, the concept of dense deployment of small cells, remote radio units, and relay heads which has been an attractive research interest during recent years [35]–[40], plays a major role in proximity-based computing. While boosting the capacity and coverage, the dense deployment of access points offers the opportunity of introducing additional computing resources at the network edge. Henceforth, the user devices in the network are capable of uploading their computational tasks to access points and download the corresponding outputs after the processing with high data rates yielding lower latencies.

Another proximity-based computing technique is mobility-assisted MEC. Therein, networks of connected vehicles, unmanned autonomous vehicle (UAV), and robots with high processing power can assist the computational tasks of the users [41], [42]. The high processing power of the above-mentioned devices that are dedicated to users provides low computational latencies. Moreover, their flexible connectivity with the users due to the mobility and high data rates therein due to the proximity offer lower communication latencies, yielding reduced E2E latencies.

Computing location swapping is another proximity-based computing method. Therein, groups of users coexist in either physical (located close by) or virtual spaces (interact and/or share computing tasks). In this regard, proximity alone provides low communication latency, yet it could yield poorly utilized computational resources. Combining the user groups in virtual space and their physical locations, some users can swap their associated MEC servers to improve both computing and communication latencies, resulting in better E2E performance [43].

Although the proximity-based computing enables low latency in MEC, the concept itself brings up new challenges to the network design and resource optimization therein. The increased interference is one of the challenges in both network densification and computing location swapping. Due to the limited availability of both communication and computation resource, increased interference may degrade both UL and DL communications, yielding increased E2E

latency [44]. In this regard, interference avoidance, management, and mitigation techniques as well as use of higher frequency channels are viable remedies. Another challenge is the frequent handover due to the dynamics of environment and user mobility [44], [45]. While handover may incur undesirable latencies, the concept of multiconnectivity (MC) can be utilized, in which users receive computing assistance from several MEC servers.

Low-Latency Enabler 3 (Edge Machine Learning): Inference (or prediction) capabilities with low latency are one of the main reasons for ML to be popular in MEC as well as several other communication applications such as coding, BF, resource optimization, caching, scheduling, routing, and security [18], [46]–[48]. While the majority of the ML-based communication system design literature is rooted in the centralized and offline ML techniques, the upturn of mission-critical applications for a massive number of connected devices demands for the intelligence at the network edge [11], [49]. In contrast to conventionally centralized ML designs, the edge ML is capable of generating inference within an instance at the edge devices, presenting the opportunity to greatly reduce the E2E latency in MEC applications. Such intelligence at the edge devices can predict the uncertainties in channel dynamics, communication and computation resource availability, interference, and network congestion at the local devices, explore and learn about the network environment with minimal additional signaling overheads, and characterize and model the network behavior in which the system performance is analyzed. At the MEC servers, such prior knowledge provides the opportunities to smartly schedule their computing resources and share the results with the corresponding user devices. Furthermore, at the events of connectivity losses, edge ML at the user devices allows the decision-making within the devices using the forecast on system behaviors, allowing uninterrupted end-user service experiences. This ability to operate offline/off-grid can reduce the number of latency-critical parallel tasks at the MEC server, in which network-wide end-user experience is improved.

The challenge of enabling low latency in MEC via edge ML relies on the training latency and inference accuracy therein. In the distributed setting, each edge device lacks the access to the large global training data set, in which training over local data can degrade the inference accuracy. To improve the inference accuracy, edge ML devices may need often cooperation among one another or with a centralized helper, which incurs additional overheads and, thus, increased training latency. In this regard, further investigations need to be carried out to optimize the tradeoff between training latency and inference accuracy depending on the design architectures, communication models, and application requirement.

Low-Latency Enabler 4 (Proactive Computing): Although edge computing is capable of minimizing the latency induced due to the high propagation delay of cloud computing, it still experiences the delay due to offloading the task data to the edge server, processing delay, as well as queuing delay for both operations. While these delays are inevitable in some cases, there exist situations in which the task has already been executed before for another user at a different time. Take, for example, an AR case in which visitors of a specific spot in an exhibition or museum request a specific task of augmenting an object to the view of this spot or the task of object identification by multiple vehicles in intelligent transportation systems (ITSs). Executing these tasks redundantly each time it is requested is certainly not resource efficient and is causing higher delays to these tasks as well as other tasks sharing these resources. Here, executing and caching the results of these tasks in advance, such that they are served when requested with minimal latency, can be a major latency minimizer.

The ideas of prefetching tasks [50] and proactive computing [21], [22] aim to develop techniques that learn and predict which tasks are to be requested in the future and precompute them. Indeed, the success of proactive computing lies on a well-aimed choice of which tasks to proactively compute and which are to leave for real-time processing. Essentially, this involves developing efficient prediction methods that study the popularity patterns of the computing tasks to decide on which tasks to prefetch. The idea also relies on the availability of storage capabilities at the edge servers [51].

Low-Latency Enabler 5 (Parallel and Coded Computing): The computing task data can be distributed over multiple servers in different edge computing scenarios, for example, in a smart vehicle scenario where the navigation map data can be partly stored in several edge servers. Parallel execution of computing tasks over multiple servers significantly impacts the efficiency and speed of task execution. Moreover, it eliminates the need to collect the full task data set in a single entity. For example, partial offloading can be performed where only a partition of the task is offloaded to where its required input data are available [5]. The implementation of parallel computing depends on the correlation between the task partitions, i.e., only partitions that are not dependent on each other can be executed in parallel, whereas the dependent tasks have to be executed sequentially. Task dependence graph models and task partitioning [5], [15] are used to tackle the interdependence between the different task partitions.

A challenge in realizing parallel computing, however, is the resulting high interserver communication load. Moreover, it suffers from the straggling effect, where a missing result from a single node delays the entire computation process. The concept of coded computing has shown to address both of these challenges [16]. Through exploiting the redundancy in the task partitions' execution at different servers, coded multicast messages, e.g., via maximum distance separable (MDS) codes, can be used to deliver the results of the missing partitions simultaneously to multiple servers. This approach significantly reduced the amount of data that has to be communicated between the

servers, at the expense of more redundant task executions at each server. Coded computing also helps in minimizing the overall computing latency through minimum latency codes. In a conventional parallel computing task, each server executes a partition of the task and returns its result to the client. In this model, one delayed or failed partition will cause a delay or failure to the entire task. Alternatively, by generating redundant task data that are coded combinations of the original task data and executing these coded tasks, the result can be recovered by decoding the data from only a subset of the servers, eliminating the effect of a delayed or failed result. Optimizing the creation of the redundant coded tasks enables an inverse linear tradeoff between the computing latency and the computing load [52].

2) High-Reliability Enablers: For MEC to fulfill its role and run applications on devices behalf, i.e., offloading the computing, it needs to be able to operate below stringent latency values, which are unachievable in traditional mobile cloud computing (MCC) systems or too demanding to be run locally due to excessive computational and communication power

In this regard, to exploit both the high capacity of 5G mobile connections and the extensive computing capabilities located at the edge cloud, the concept of reliability is introduced with a twofold interpretation. In the first place, we find the classical notion of reliability related to error-robustness guarantees. As such, it allows to be tackled at different layers, including the reliability of the wireless link at the physical layer (PHY). Another fundamental notion of reliability, which has been widely adopted for wireless communications and standardization bodies as the Third Partnership Project (3GPP), is that of reliability understood as a probabilistic bound over the latency.

Understood in its most classical form, it is common that a toll in return for ensuring high reliability will have to be paid in the form of additional/increased delays. For instance, at the PHY layer, the use of parity, redundancy, and retransmission will increase the latency. Also, in multiuser environments, allocating multiple sources to a single user while clearly beneficial at an individual level could potentially impact the experienced latency of the remaining users.

Next, we will set forth some of the enablers for both notions of reliability.

High-Reliability Enabler 1 (Multiconnectivity): Compared to wired transmissions, in wireless environments, temporary outages are common due to impairments in the SINR. These originate from, among others, stochasticity of the wireless channels, fluctuating levels of interference, or mobility of the MUEs. The term MC [53] encompasses several techniques developed with the overarching aim of enhancing effective data rates and the mobility robustness, i.e., the reliability, of wireless links. For that purpose, MC exploits different forms of diversity to cut down on the number of failed handovers, dropped connections, and, generally speaking, radio-link failures (RLFs) that might cause service interruptions [54], [55].

MC solutions are classified as intrafrequency or interfrequency, i.e., depending on whether they operate using the same frequency or, otherwise, combine multiple carrier frequencies. Examples of the former include coordinated multipoint (CoMP) [56] transmissions and single-frequency networks (SFNs) [57]. CoMP involves a set of techniques that exploit rather than mitigating intercell interference (ICI) to improve the performance at the cell edge. On performing joint processing, dynamic point selection (JP/DPS) or coordinated scheduling and beamforming (CS/CB) in the UL/DL, BSs operate effectively as if assembled in a distributed multiple antenna system. SFNs embody a form of synchronous multicell transmission whereby various sources use the same time and frequency resource to noncoherently transmit signals to a receiver. The multiple received copies will be then constructively combined if their propagation delays are tightly bounded or, else, will induce intersymbol interference (ISI) [58].

As for interfrequency MC, carrier aggregation (CA) [59] and dual connectivity (DC) are its most noteworthy examples. In CA, contiguous or noncontiguous component carriers, possibly allocated to several different BSs, are combined and the scheduling and interference management orchestrated over these frequency bands aiming to enhance the resulting system's capacity. As for DC, this framework provides the solutions for interfrequency, heterogeneous networks (HetNets) scenarios, and different wireless standards MC so that a user equipment (UE) will be simultaneously connected, respectively, in two different frequencies, to two different types of BSs or two different wireless standards [60]. Recently, the idea of DC for mmWave and microwave bands has been proposed [36], [61] as an effective approach to facilitate cellular mmWave IA [62] as well as mmWave handover [63]. In such a manner, mmWave and sub-6-GHz DC can team together to augment the reliability of the mmWave working as a fallback to compensate eventual mmWave channel vulnerability, e.g., to blocking events. Finally, the benefits of integrating communication interface diversity for reliability purposes are also studied in [64] in the context of machine type communications (MTCs).

SFN operation is proposed in use case 6 detailed in Section IV-B. The goal is to protect against mmWave channel intermittence by increasing the rate of those links between the millimeter-wave access points (mmAPs) and the virtual reality players (VRPs) that, otherwise, would jeopardize the immersive experience.

High-Reliability Enabler 2 (Task Replication): While MC can boost the reliability in the presence of channel fluctuations, it requires coordination between the different servers that are connected to the end user. However, when coordination is not possible, reliability can still be enhanced through the task replication. Similar to packet replication in data communication, a user can offload a computing

task to multiple servers that are not connected to each other and receive the result from whichever has the result ready first. This mechanism provides more guarantees of task execution, at the expense of reduced system capacity, due to the underutilization of computing servers. One realization of this concept is proposed in [65], namely, hedged request is when the user sends one replica of the task to the server that is believed to be most suitable and then follows by sending another replica to an additional server after some delay. Completion pending remaining requests are canceled once a result is received from any server.

While task replication can be efficient in ensuring the reliability in the case of channel dynamics, it incurs significant additional load. To combat this, one can offload the task to an additional server only when the delay from the first server exceeds a certain threshold [65] This approach is investigated in [17]. Therein, it is shown that imposing such a condition can significantly curb the latency variability without inducing much additional load.

High-Reliability Enabler 3 (Federated Machine Learning): While performing ML inference at the network edge yields low latency, distributed training of their ML models across different edge nodes improves the inference reliability. To be specific, each learning agent optimizes its ML model during the training phase so as to maximize the inference accuracy over locally available training data. The measured inference accuracy at the training phase is, however, not always identical to the inference accuracy at the test phase, primarily because of unseen training data samples. This accuracy gap is known as the generalization error that measures the inference reliability under unseen data samples [66]. A straightforward way to reduce the generalization error is exchanging training data samples among edge nodes. Data exchange, however, incurs extra communication and computation cost and may not be available for user-generated private data. To address this problem, federated learning (FL) has recently been proposed [67], [68], in which edge nodes exchange and aggregate their local ML models, thereby preserving data privacy, avoiding extra computation, and reducing communication overhead when ML model sizes are sufficiently smaller than data sizes.

FL is still a nascent field of research, calling for codesigning communication, computation, and ML architectures [11], [49]. For instance, the original FL algorithm has the communication payload size being proportional to the ML model sizes and thus cannot deal with deep neural network (NN) models. Proper model compression and parameter quantization techniques are thus needed while trading the increased communication efficiency off against the reduced accuracy. Furthermore, the server in current FL algorithms simply aggregates uploaded local models, although it has higher computation resources compared to the edge devices. Along with these FL architectures, computing task offloading, task scheduling, and resource allocations should be jointly optimized toward achieving reliability under uncertainties on MEC operations, including unseen data samples, channel fluctuations, and time-varying communication and computation resources.

High-Reliability Enabler 4 (Extreme Event Control): As mentioned previously, one reliability notion is the probability of violation or failure over a latency bound, which can be mathematically expressed as $\Pr(\text{Latency} > L_{\text{bound}})$. This probability ranges from 10^{-3} to 10^{-9} depending on the mission-critical application in 5G networks [69]. To meet the ultrareliability requirements, we should focus on the extreme events with very low occurrence probabilities. However, in classical communication systems, the designed approaches are based on the expected metrics, e.g., average rate and average latency, in which the random event realizations with higher probability distribution function (PDF) values dominate the system performance. In other words, the conventional average-based approaches are inadequate for enhancing reliability performance, and instead, we need to take into account the metrics or statistics, which are related to or affect the extreme events, such as: 1) worst case measurement, e.g., largest latency in the network; 2) tail/decay behavior of the complementary cumulative distribution function (CCDF); 3) very low bound violation probability; and 4) threshold deviation and its higher order statistics, e.g., variance, while designing the URLLC-enabled MEC systems. To analytically analyze these metrics and statistics, extreme value theory (EVT) [70], [71] is a useful methodology for mathematical characterization and, thus, provides a powerful framework for extreme event control. Let us introduce the fundamental theorems in EVT as follows, which characterize the aforementioned metrics and their statistics.

Theorem 1 (Fisher–Tippett–Gnedenko Theorem [70]): We consider n independent and identically distributed (i.i.d.) samples from a random variable X, i.e., $X_1, \cdots, X_n \overset{i.i.d.}{\sim} X$ and define $Z_n := \max\{X_1, \cdots, X_n\}$. If Z_n converges to a nondegenerate distribution as $n \to \infty$, we can approximate the limit as a generalized extreme value (GEV) distribution that is characterized by a location parameter $\mu \in \mathbb{R}$, a scale parameter $\sigma > 0$, and a shape parameter $\xi \in \mathbb{R}$.

Among them, the shape parameter governs the GEV distributions' tail behaviors [71], which are sorted into three types depending on the value of ξ.

1) When $\xi > 0$, the GEV distribution has a heavy-tailed CCDF that is more weighted than an exponential function.
2) When $\xi = 0$, the GEV distribution has a light-tailed CCDF that has a thinner tail than an exponential function.
3) When $\xi < 0$, the GEV distribution has a short-tailed CCDF that has a finite upper endpoint at $z = \mu - \sigma/\xi$.

When $\xi \geq 0$, the upper endpoint of the CCDF approaches infinity.

Theorem 2: (von Mises Conditions [71]): In Theorem 1, the characteristic parameters $(\mu,\ \sigma,\ \xi)$ of the approximated GEV distribution can be asymptotically found as per $\mu = \lim_{n\to\infty} F_X^{-1}(1-1/n)$, $\sigma = \lim_{n\to\infty}(1/nf_X(F_X^{-1}(1-1/n)))$, and $\xi = -1 - \lim_{x\to\infty}([1-F_X(x)]f_X'(x)/[f_X(x)]^2)$.

Theorem 3: (Pickands–Balkema–de Haan Theorem [70]): Consider the random variable X in Theorem 1 and a threshold d. As $d \to F_X^{-1}(1)$, the CCDF of the excess value $Y|_{X>d} = X - d > 0$ can be approximated as a generalized Pareto distribution (GPD) whose mean and variance are $\tilde{\sigma}/(1-\xi)$ and $(\tilde{\sigma}^2/(1-\xi)^2(1-2\xi))$, respectively.

Analogous to the GEV distribution, the GPD is characterized by a scale parameter $\tilde{\sigma} > 0$ and a shape parameter $\xi \in \mathbb{R}$. In Theorems 1 and 3, ξ is identical, while $\sigma = \tilde{\sigma} + \xi(\mu - d)$. Note that Theorems 1 and 2 provide a way to characterize the worst case metric and its tail behavior, whereas Theorem 3 is directly related to the bound violation and its statistics. Since the characteristic parameters of the GEV distribution and GPD are identical or related, the results of these three theorems are complementary to one another.

Nevertheless, some tradeoffs and dilemmas exist when we apply the results of EVT and estimate the characteristic parameters. For example, we need to trade off data availability, which affects the performance, convergence speed, and estimation accuracy. Specifically, given N i.i.d. realizations of X (i.e., N/n realizations of Z_n), larger n theoretically gives the better approximation of the GEV distribution but slows down the convergence of parameter estimations due to the less availability of data samples of Z_n. The similar tradeoff between high threshold d and availability of threshold-exceeding data can be found in Theorem 3. Additionally, if the distribution of X, e.g., delay of a single user, is unknown beforehand, this agnostic makes Theorem 2 difficult to characterize the network-wide largest delay. Fortunately, thanks to the mature development in the ML field, the aforementioned issues can be tackled by using the ML approaches, in which unsupervised learning provides a way to infer a mathematical expression of the unknown distribution, while the lack of available data is addressed in an FL manner by aggregating and averaging the estimated characteristic parameters of all distributed devices.

IV. APPLICATIONS AND USE CASES

In this section, we elaborate on some of the prospective services and applications for which offloading their computing tasks to the edge significantly improves their performance in terms of latency and reliability. In particular, we focus on two scenarios where offloading task computing to the network edge will be beneficial: 1) when end users have limited computing capabilities, e.g., VR head-mounted devices (HMDs) and 2) when end users have sufficient computing and energy resources but are accessible only to a fraction of the entire information for the computation input, e.g., vehicular edge computing scenarios. We follow by presenting different edge computing use cases in which the URLLC enablers are utilized.

A. Edge Computing Applications

1) Extended Reality: Extended reality (XR) is an umbrella term that covers all virtual or combined real-virtual environments, including VR, AR, and mixed reality (MR). These environments differ in the nature of the content a user sees or interacts with. While VR describes environments where users are fully immersed in a virtual world, AR refers to the view of a virtual environment that is merged or supplemented by elements or inputs from the real world. AR can be categorized as a special case of the more general MR, which refers to the environments that mix together real and virtual elements that can interact with each other.

XR is anticipated to be one of the leading applications to leverage edge computing. Providing high-quality XR experience comes with high computation resource demand. At the same time, XR applications are highly sensitive to delay. Typically, a maximum E2E delay, also known as motion-to-photon (MTP) delay, of 15–20 ms can be tolerated in VR. Higher delay values trigger what is known as motion sickness, resulting from a visual-motor sensory conflict. This makes it unrealistic to rely on remote cloud servers for processing. On the other hand, processing XR locally on the user device has several complications. First, XR devices, such as HMDs and smartphones, are often equipped with limited computing capabilities. This limitation is due to the device size, manufacturing cost, as well as the heat generated from powering the device. Second, running applications on different types of devices, with different hardware, operating systems, and platforms is a challenging task. For these reasons, existing standalone XR devices often provide limited content quality. Standalone VR headsets operate with reduced frame resolution and frame rate [72], whereas AR headsets, such as Microsoft HoloLens, restrict the amount of renderable polygons [73].

For these reasons, the success of XR requires providing high computation and storage resources close to the end users. In this regard, edge computing is an intuitive solution to provide such services [74]. Today's most powerful VR headsets rely on edge computers to perform sophisticated rendering. However, wired connections are still used between the headsets and the edge servers due to the high rate requirement of VR applications. This limits the mobility and convenience of VR users and hence decreases the Quality of Experience (QoE).

The need for a better XR QoE and the advancement in wireless communication capabilities motivate the development of wireless XR systems that incorporate powerful edge computers and high-capacity wireless links [18], [74]–[77]. The mmWave communication can provide large spectrum and high data rates, making it a solid candidate

for wireless XR. Moreover, the directionality of mmWave links allows for leveraging multiuser transmission techniques, such as multicasting and broadcasting to deliver common and correlated content to multiple users in a way that minimizes the communication delay. However, directional mmWave links suffer outages due to signal blockage. This affects the link signal quality and increases the channel variability and, hence, decreases the link reliability. MC can be a viable solution to provide robust mmWave communication. Using MC, an XR user maintains multiple simultaneous communication links with multiple servers.

2) Vehicular Edge Computing and V2X/V2V for ADAS: Future autonomous driving vehicles comprised as nodes of the Internet of Vehicles (IoV), a larger mobility network which can be considered as an extended application of the IoT to ITSs [78], will operate as hubs, integrating multiple technologies and consuming and producing massive volumes of data [79]. The advanced driver-assistance systems (ADASs) to be equipped in these vehicles, especially those pertaining to the area of traffic safety, heavily depend on reliable and instantaneous decision-making processes that hinge on inputs from multiple sensory data sources, including laser imaging detection and ranging (LIDAR), automotive radar, image processing, and computer vision [80]. As an example, we can think of successful object identification from LIDAR point clouds or speed and trajectory prediction for dynamic objects moving within a vehicle's vicinity. Hereof, it is essential that these vehicles are equipped with powerful computing and processing capabilities to swiftly handle high data volumes rather than solely relying on cloud services that, in the above-mentioned example, may classify the objects or predict trajectories from raw data with higher accuracy but, possibly, incurring to do so in unacceptable delays. Moreover, for next-generation ADAS, it is envisaged that vehicles will communicate with each other as well as with an increasingly intelligent roadway infrastructure through the use of vehicle-to-everything (V2X) and vehicle-to-vehicle (V2V) communications, ultimately exploiting high-capacity mmWave links [81], [82]. Consequently, the cumbersome volume of locally generated data could be exacerbated by the acquisition of data from the environment and the surrounding vehicles.

Indeed, vehicular edge computing will play a pivotal role to support delay-sensitive as well as future emerging multimedia-rich applications in vehicular networks, which is buttressed by the growing body of literature devoted to the area of content-centric applications of vehicular MEC [83]–[86] which are frequently combined with ML to provide reliability as edge analytics [87] to leverage huge volume of information [86] or to provide an integrated framework for dynamic orchestration of networking, caching, and computing resources in next-generation vehicular networks [88].

Being not nearly as tightly constrained by size or by the access to a power supply as their counterpart IoT devices or smartphones, the computational and storage capabilities in vehicular terminals could allow them to run locally or collaboratively, using vehicles as the infrastructures for communication and computation as proposed in [89], resource-hungry applications.[1] In this regard, provided that computing and processing capabilities may not be the limiting factor, a second advantage of running these applications in the network edge is substantiated by the availability of data collected from multiple vehicles in edge servers. Access to this information raw or preprocessed can augment individual vehicles' situational awareness by extending their own sensing range. Resorting to edge contents can thus provide a bigger picture at acceptable delays.

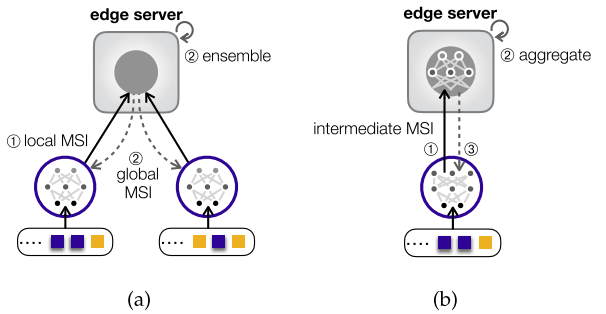


Fig. 2. ***Edge ML architectural splits. (a) Data split. (b) Model split.***

The latter idea is exemplified in the third use case in Section IV-B where the information from different vehicles is combined in the network edge following FL principles and used to refine a global model for transmission queue length distribution for the purpose of providing ultrareliable low-latency V2V communications.

B. Use Cases

Next, we present different case studies in which the URLLC enablers are utilized in edge computing settings.

Use Case 1 (Edge Computing for Federated Machine Learning): As addressed in Sections III-B1 and III-B2, edge ML is envisaged to be a key enabler for URLLC, in which both inference and training processes of ML models, e.g., NNs, are pushed down to the network edge [11]. This direction of edge ML has been fueled by FL [67], [68], [91]–[94] under a data split architecture [see Fig. 2(a)], where edge devices collectively train local models with their own user-generated data via a coordinating edge server that aggregates locally computed model updates, referred to as model state information (MSI). The MEC framework can further improve FL by codesigning with training architectures and algorithms. In view of this, on the one hand, each

[1]However, the longer product's life span in the automotive industry (according to the U.S. Department of Transportation since 2018, the average age of on-the-road vehicles is over 11 years [90]) could quickly turn onboard central processing unit (CPU)/graphical processing unit (GPU) processing capabilities obsolete.

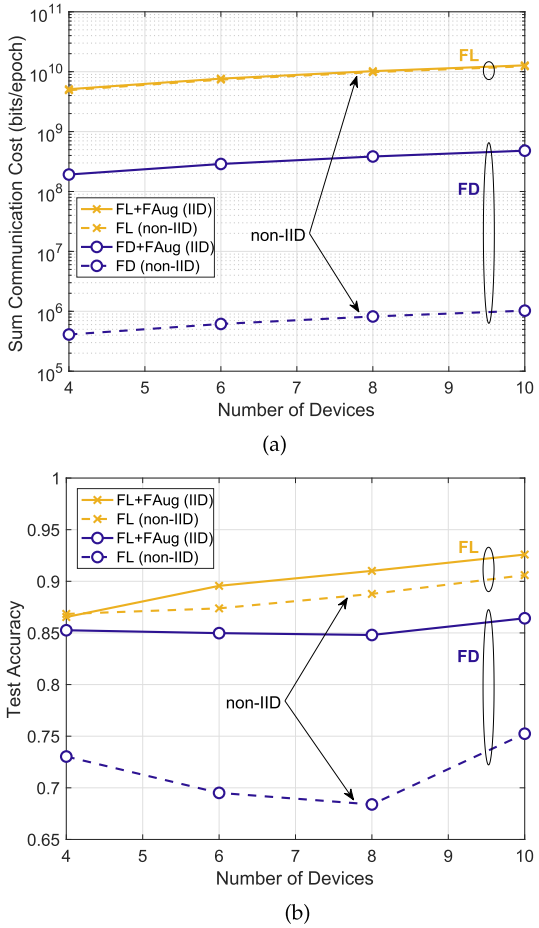


Fig. 3. ***Communication cost and inference accuracy of FL and FD with or without FAug in the MNIST classification problem, where each device stores a five-layer convolutional NN (CNN). For FAug, the conditional GAN consists of a four-layer generator NN and another four-layer discriminator NN. (a) Communication cost. (b) Test accuracy.***

edge device is able to optimize the MSI type depending on the NN model size and channel quality. As done in FL, one can exchange the model parameter MSI whose payload size is proportional to the model size, which is not feasible for deep NNs under poor channel conditions. Alternatively, one can exchange model output MSI whose payload size is independent of the model size, referred to as federated distillation (FD) [95]. As shown in Fig. 3(a), this fundamentally results in FDs incomparably smaller communication payload per MSI exchange than FL and can thereby better cope with poor channel conditions.

On the other hand, the edge server can assist in the training process by exploiting its extra computation and communication resources. A compeling example is to rectify the non-IID training data set incurred by the user-generated nature of data, in which entirely uncorrelated (nonidentical) and/or too similar (nonindependent) data samples across devices negate the benefit of distributed training [96]. To this end, in federated augmentation (FAug) [95], the edge server first collects a few seed samples from edge devices and oversamples them (e.g., via Google's image search for visual data) through its fast connection to the Internet. Then, the edge server can utilize its high computing power for training a generative model (e.g., conditional generative adversarial network (GAN) [97]). Downloading the trained generator empowers each device to locally augment deficient data samples until reaching an IID training data set. With FAug, both FL and FD yield higher test accuracy as shown in Fig. 3(b), at the cost of slight increase in communication cost as shown in Fig. 3(a).

Finally, a very deep NN (e.g., Inception V4 NN model consuming 44.3 GB [98]) cannot fit into a single device's memory and has to be partitioned into multiple segments stored across edge devices and server, i.e., model split [see Fig. 2(b)]. Here, the model's local and offloaded computations should be orchestrated over wireless links by optimizing the partitioning strategy based on the NN's topology and constituent layers. This calls for a novel MEC framework that takes into account not only communication and computation resources but also NN forward and backward propagation dynamics intertwined with channel dynamics.

Use Case 2 (Extreme Event-Controlled MEC): For the extreme event-controlling computation and communication codesign in [99] and [100], we studied a multiuser MEC scenario as shown in Fig. 4, in which multiple MEC servers with different computation capabilities are deployed. In this setting, the UE manages its local resource (i.e., total power budget) for computation and communication, i.e., task offloading, while the MEC server schedules its computational resources for the UEs' offloaded tasks. Herein, we consider the length of the task queue as a latency measurement since queuing latency can be reflected by the queue length. For the reliability concerns, we are concerned about the bound violation probability and higher order statistics of threshold deviation as highlighted in high-reliability enabler 4. In this regard, we first impose a constraint on the queue length[2] bound violation probability as

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \Pr\left(Q(t) > d\right) \leq \epsilon \ll 1 \tag{1}$$

where d and ϵ are the given bound and tolerable violation probability, respectively. Let us further focus on the excess value over the bound d, which is denoted by $X(t)|_{Q(t)>d} = Q(t) - d > 0$. By applying Theorem 3, we approximate the exceedances as a GPD with the characteristic parameters $(\tilde{\sigma}, \xi)$. The mean and variance are $\mathbb{E}\big[X(t)|Q(t) > d\big] \approx (\tilde{\sigma}/1-\xi)$ and $\text{Var}\big(X(t)|Q(t) > d\big) \approx (\tilde{\sigma}^2/(1-\xi)^2(1-2\xi))$, respectively. We can find that the smaller $\tilde{\sigma}$ and ξ, the smaller the mean value and

[2]The notation Q generalizes the lengths of all task queues at the UEs and MEC servers.

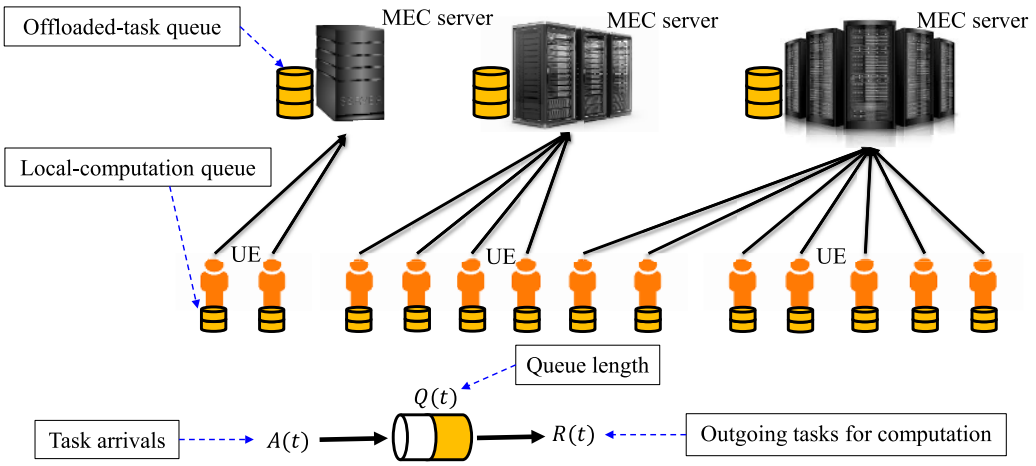


Fig. 4. ***Extreme Event-Controlled MEC architecture.***

variance. Since the approximated GPD is just characterized by the scale and shape parameters, we impose thresholds on these two parameters, i.e., $\tilde{\sigma} \leq \tilde{\sigma}^{\text{th}}$ and $\xi \leq \xi^{\text{th}}$. Subsequently, applying the two parameter thresholds and $\text{Var}(X) = \mathbb{E}[(X)^2] - \mathbb{E}[X]^2$, we consider the conditional constraints on the mean and second moment of the excess queue length

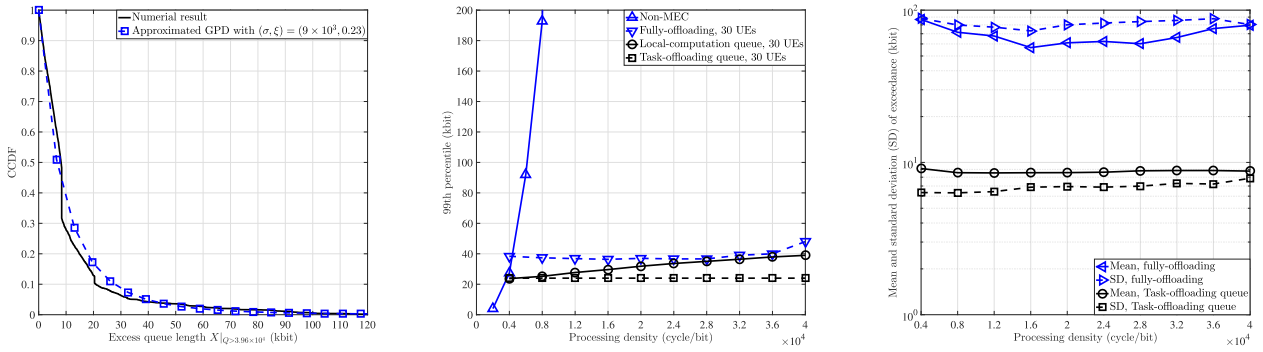
$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\big[X(t)|Q(t)>d\big] \leq \frac{\tilde{\sigma}^{\text{th}}}{1-\xi^{\text{th}}} \tag{2}$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\big[[X(t)]^2|Q(t)>d\big] \leq \frac{2\big(\tilde{\sigma}^{\text{th}}\big)^2}{\big(1-\xi^{\text{th}}\big)\big(1-2\xi^{\text{th}}\big)}. \tag{3}$$

Considering the three requirements mentioned earlier for the extreme events, we trade off the UE's computation power and communication power in the extreme event-controlling computation and communication codesign.

The effectiveness of characterizing threshold deviation by the Pickands–Balkema–de Haan theorem, i.e., Theorem 3, is verified in Fig. 5(a). Therein, $\Pr(Q > d) = 3.4 \times 10^{-3}$ with $d = 3.96 \times 10^4$. Additionally, in contrast with the schemes without edge computing and without local computation capability, the extreme event-controlling approach achieves the better performance in terms of the extreme event-related metrics shown in Fig. 5(b) and (c), in the considered MEC system.

Use Case 3 (EVT/FL Ultrareliable Low-Latency V2V Communication): The idea of how to combine EVT and FL to enable URLLC in vehicular communication networks, referred as `extFL`, is discussed in our preliminary study [101] and illustrated in Fig. 6. Here, vehicles observe their queue length samples and utilize the tail distribution of queue lengths at the vehicular transmitters over the whole edge network to optimize their transmission decisions such that the worst case queue lengths are minimized while ensuring reliability in terms of queuing latency. The analytical parametric model of the aforementioned tail



(a) (b) (c)

Fig. 5. ***(a) Tail distributions of the excess queue length and the approximated GPD of exceedances, (b) 99th percentile of the queue length, and (c) mean and standard deviation of exceedances over the 99th percentile queue length, versus processing density.***

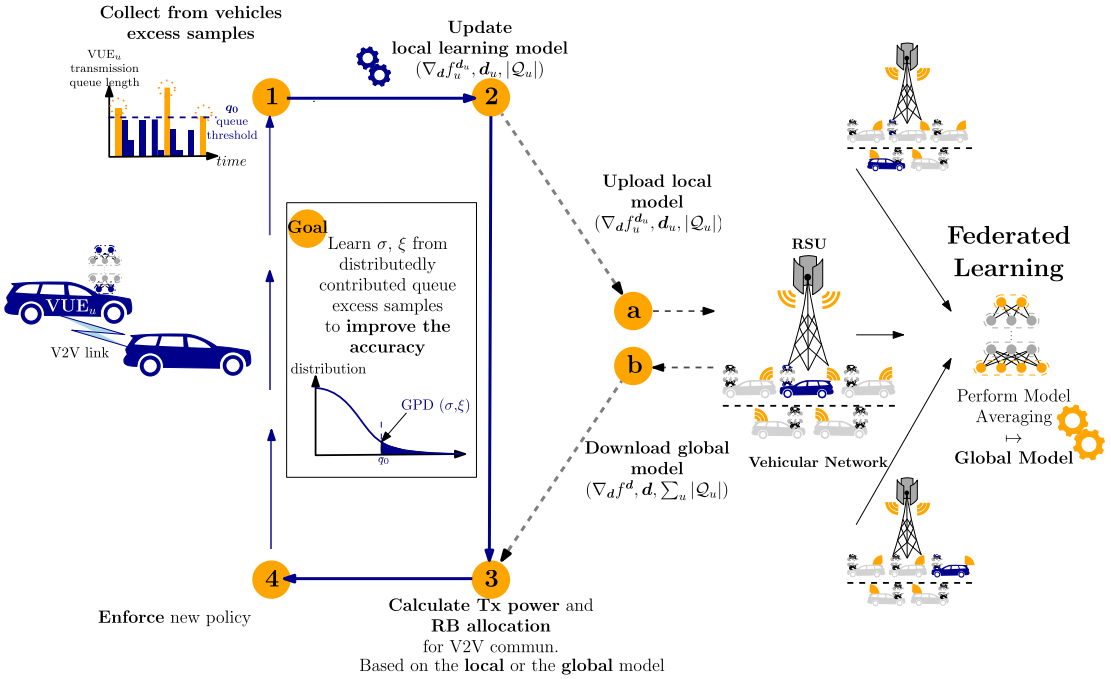


Fig. 6. ***Operational structure of EVT parametric FL (extFL).***

distribution is obtained via EVT. Naturally, the evaluation of the above-mentioned parameters is carried out by gathering all queue length samples at a central controller, the MEC server, with the additional costs of communication and computation overheads. In contrast to the centralized approach, here, FL is used to reduce the communication payload by allowing individual vehicles to learn the tail distribution by exchanging a simplified model (two gradient values) instead of their raw local queue length samples, i.e., enabling URLLC with the aid of ML at the edge devices.

The goal is thus to minimize the network-wide power consumption of a set of vehicular user equipments (vUEs) while ensuring low queuing latencies with high reliability. However, there still exist worst case vUEs experiencing high latencies with a low probability whose performance losses are captured by extreme events pertaining to vehicles' queue lengths exceeding a predefined threshold with nonnegligible probability. The principles of EVT characterize the tail distribution of the queue lengths exceeding a predefined threshold by a GPD with two parameters scale and shape. The concepts in maximum likelihood estimate (MLE) are used along FL to estimate the scale and shape parameters of the queue tail distribution locally at each vUE over the queue length samples. Therein, occasionally, local estimations and the gradients of MLE known as local model at each vUE are shared with the MEC server. The MEC server does model averaging and shares the global model with the vUEs to update their local estimations. Using the knowledge of the tail distribution over the network, the transmit power of each vUE is optimized to reduce the worst case queuing delays.

Fig. 7(a) compares the amount of data exchanged and the achieved V2V communication reliability of `extFL` with a centralized tail distribution estimation model, denoted as `CEN`. Note that the `CEN` method requires all vUEs to upload all their queue length samples to the RSU and to receive the estimated GPD parameters. In contrast, in `extFL`, vUEs upload their locally estimated learning models and receive the global estimation of the model. As a result, `extFL` yields equivalent or better end-user reliability compared with `CEN` for denser networks while reducing the amount of data exchange among vUEs and the RSU. The worst case vUEs' queue lengths, i.e., queue lengths exceeding q_0, are compared in Fig. 7(b). Here, the mean indicates the average queuing latency of the worst case vUEs, while the variance highlights the uncertainty of the latency. As the number of vUEs increases, it can be noted that both the mean and the variance in `extFL` are lower than that in `CEN`. The reason for the above-mentioned improvement is the reduced training latency in `extFL` over `CEN`.

Use Case 4 (Deep Reinforcement Learning for Optimized Edge Computing Task Offloading): The task offloading decision-making in edge computing networks is a challenging task in the presence of environmental dynamics. This situation is aggravated in ultradense networks, where solutions to break the curse of dimensionality is desperately needed. In [102] and [103], a discrete-time Markov decision process was adopted to model the problem of expected long-term MEC performance optimization in an ultradense radio access network, where a number of BSs are available for computation task offloading. For a representative wireless charging-enabled MUE, whether to execute an arriving computation task at the local mobile

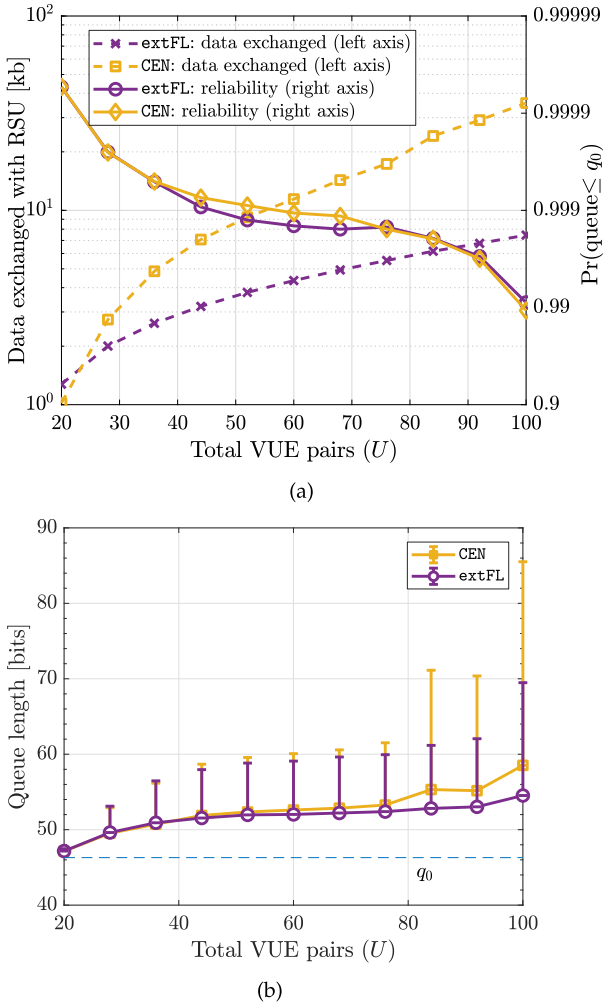


Fig. 7. ***Comparison between CEN and extFL. (a) Amount of data exchanged between RSU and VUEs (left axis) and the achieved reliability (right axis). (b) Mean and variance of the worst case VUE queue lengths.***

device or to offload the task for edge server execution via one of the BSs should adapt to the environmental dynamics in an intelligent manner. These environment dynamics may consist of random computation task arrivals, time-varying communication qualities between the MU and the BSs, and the sporadic energy availability at the mobile device. The challenges for the problem-solving lie in the lack of any *a priori* knowledge of any environment dynamic statistics along with the high-dimensional state space. A deep RL technique shows the power of achieving an optimal solution.

More specifically, the objective of the MUE is to minimize an expected infinite-horizon discounted cost given by

$$Q(s,a) = \mathsf{E}\left[\sum_{t=1}^{\infty} (\gamma)^{t-1} \cdot c\left(s^t, a^t\right) | s^1 = s, a^1 = a\right] \quad (4)$$

where $\gamma \in [0,1)$ is the discount factor, while the immediate cost $c\left(s^t, a^t\right)$ after performing an action a^t under a state s^t at each time slot t takes into account the incurred task execution delay and the penalty of failing to process an arriving computation task. Once we obtain the optimal Q-function, the optimal action a^* can be made by the MUE following $a^* = \arg\min_a Q(s,a)$ under a state s. Instead of using a conventional Q-learning to find the optimal Q-function, we resort to a deep-Q network (DQN) [104] $Q(s,a;\boldsymbol{\theta})$ to approximate $Q(s,a)$ with $\boldsymbol{\theta}$ being the set of parameters of the NN. The procedure of the deep RL for MEC performance optimization is briefly depicted as in Fig. 8.

In Fig. 9, we compare the average cost performance from the `Proposed` deep RL algorithm with three baselines as follows.

1) *Local:* Whenever a computation task arrives, the MUE executes it at the local mobile device using the queued energy units.
2) *Server:* All arriving computation tasks are offloaded to the edge server for computing via the BSs with the best communication qualities.
3) *Greedy:* When the computation task queue as well as the energy queue are not empty at a time slot, the MUE decides to execute the task locally or at the cloud to achieve the minimum immediate cost.

We configure a DQN of one hidden layer with 512 neurons. The replay memory is assumed to have a capacity of 5000 and we select the size of the mini-batch as 100. From Fig. 9, we can clearly see that compared to the baselines, the deep RL algorithm realizes the best performance at average cost. A higher task arriving probability ρ indicates a longer average task execution delay and, hence, a larger average cost. As the average energy arrival rate increases, the average cost improves due to the decreased failure of processing an arriving computation task.

Use Case 5 (Edge ML-Enabled 360° VR Multicast Transmission): Our previous work in [18] considered merging ML and mmWave multicasting to optimize the proactive wireless streaming of FoV-based HD 360° videos in a multiuser VR environment with low-latency guarantees. Hereof, the use of edge ML to predict users' FoV in advance is pivotal to leverage interuser correlations and curb the latency. These predicted correlations will ultimately drive both how contents are transmitted and the BF decisions at the mmWave BSs.

A VR theater scenario consisting of a network of VR users watching different HD 360° VR videos streamed in the mmWave band over a set of distributed small cell base stations (SBSs) is studied. The SBSs will report users' six-degree-of-freedom (6DoF) pose as well as CSI and produce multiple spatially orthogonal beams to serve shared FoV video content to a group of users (multicast) or a single beam (unicast) following the scheduling decisions adopted at the edge controller. By optimizing video frame admission and user scheduling, the goal is to provide a highly reliable broadband service for VR users that deliver

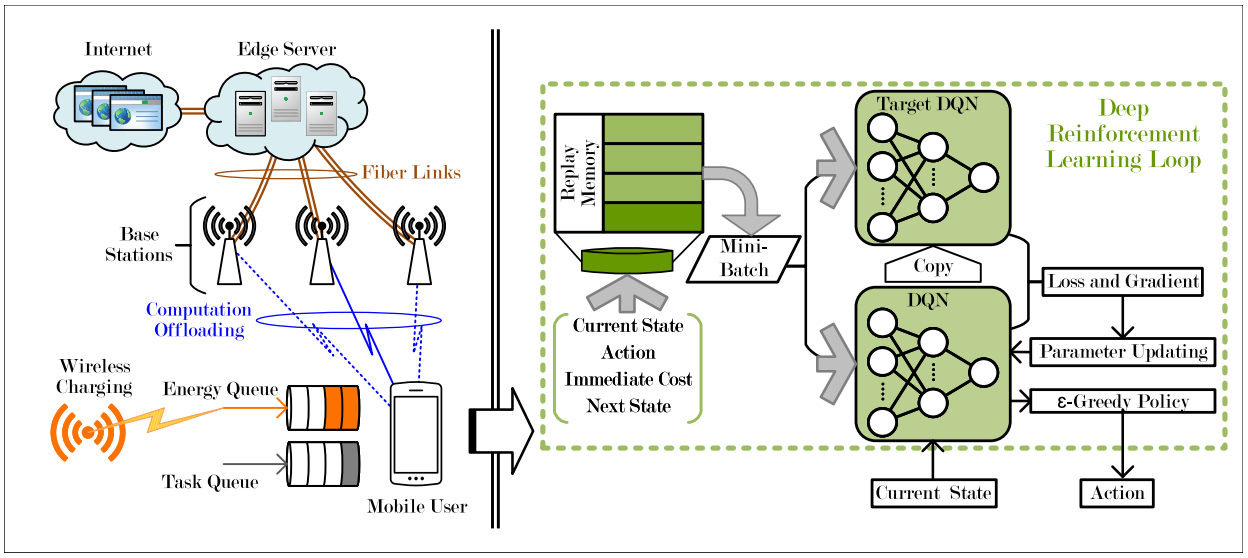


Fig. 8. *Illustration of deep RL for mobile edge computing performance optimization.*

HD videos with a latency that is below the MTP latency limits with very high probability.

To achieve this proactive content transmission and perform a head movement pattern recognition predicting users' upcoming tiled-FoV, a sequential learning model based on gated recurrent units (GRUs) [105], [106] is selected. Specifically, GRUs are a form of recurrent neural networks (RNNs) that include a double gating mechanism to govern the impact of past hidden states over the new output states and effectively tackle long-term dependences. To that purpose, an architecture based on two layers of GRU cells with a hidden state size equal to 512 separated by a rectified linear unit (ReLU) activation is stacked. The output is then fed to a serial-to-parallel (S/P) layer and to a dense neural layer. Given the multilabel nature of the learning model, a sigmoid activation layer maps the N-sized dense output to the N logits, one for each tile in the equirectangular (EQR) projection of the 360° VR video frame, which are binarized with a cutoff layer such that

$$\widehat{y}_{u,n}^{f_p} = \begin{cases} 1, & \sigma(\boldsymbol{W}_d \boldsymbol{h}_f^{(2)} + \boldsymbol{b}_d)_n \geq \gamma_{th} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\boldsymbol{W}_d$ and $\boldsymbol{b}_d$ are the weights and biases of the dense fully connected layer, respectively, and γ_{th} is the threshold value for the cutoff layer. The predicted FoV for a user u and frame index $f_p = f + T_H$ is retrieved as $\widehat{\mathcal{N}}_u^{f_p} = \{n \in [1, ..., N]: \widehat{y}_{u,n}^{f_p} = 1\}$.

Fig. 10 shows an overview of the building. The output of the deep recurrent neural network (DRNN) is fed to a user clustering module and the former constitutes one of the inputs for a scheduler, the Lyapunov drift-plus-penalty approach. In addition to our proposed scheme MPROAC+, the performance of three reference baselines with reactive unicast and multicast and proactive multicast transmission capabilities, correspondingly, UREAC, MREAC, and MPROAC is evaluated. Our proposed approach incorporates a penalty whereby quality is traded in exchange for not violating a maximum latency bound. For simulation purposes, a small size theater with the capacity of 50 users with SBSs that are located at ceiling level in its upper four corners is selected. Fig. 11 evaluates the impact of the requested HD video quality by representing the average and 99th percentile delay, the HD delivery rate and Jaccard index measured while 30 users watch one out of the 3 available VR videos for an increasing requested video chunk size.

Fig. 11 clearly shows the tradeoff between frame delay and HD streaming rate. As the chunk size increases, the average and 99th percentile delays increase

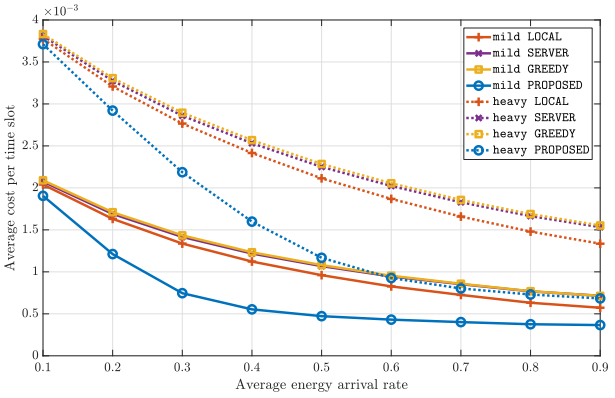


Fig. 9. *Average cost per time slot versus average energy arrival rate under* MILD *(ρ=0.3) and* HEAVY *(ρ=0.5) task arrival probabilities, respectively, represented with solid and dashed lines.*

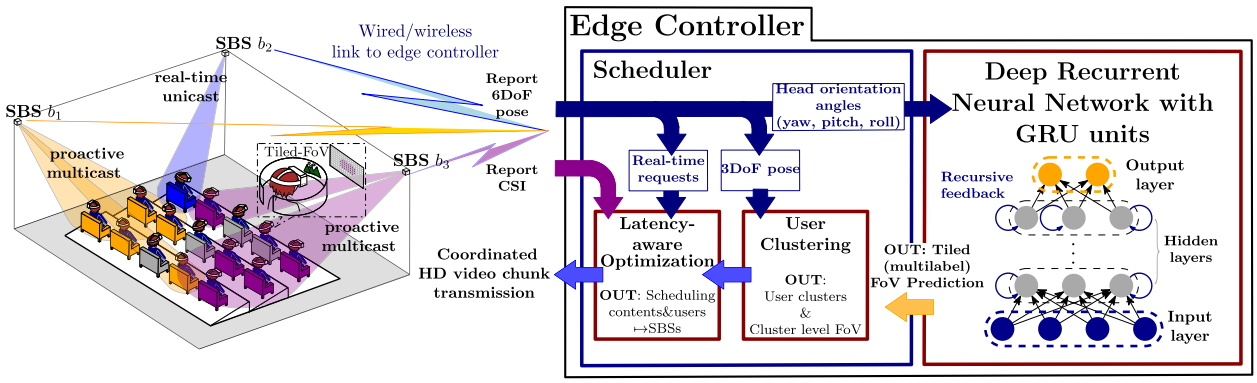


Fig. 10. ***Operational structure and building blocks of the edge controller that coordinates the DRNN FoV prediction-aided proactive content quality adaptation for the mmWave 360° VR video streaming.***

for the different schemes. Moreover, comparing UREAC with the other schemes, it is shown that multicasting brings 40%–50% increase in the HD rate and 33%–70% latency reduction through the utilization of shared FoVs of different users. By delivering the predicted frames in advance, both MPROAC and MPROAC+ minimize the average delay without sacrificing the HD quality rate. Moreover, our proposed MPROAC+ scheme is shown to also keep the worst delay values bounded due to imposing the constraint over the latency.

The tradeoff between frame delay and quality is further illustrated and the results for different values of the Lyapunov parameter V_δ are compared; as V_δ increases, the scheduling algorithm prioritizes maximizing users' HD delivery rate, whereas at lower values, the scheduling algorithm prioritizes keeping the delay bounded with high probability. This comes at the expense of having lower HD delivery rate.

Finally, the Jaccard similarity in Fig. 11(d) illustrates the tradeoffs between effective versus transmitted contents. At low traffic loads, the Jaccard index is low, which is due to the large amount of excess data delivered due to transmitting an estimated user-/cluster-level FoV. As the traffic load increases, the proactive schemes transmit more real-time frames, which increases the Jaccard index. The Jaccard index decreases again at higher traffic loads as the effect of missed frames increases [once the average delay is close to reaching the deadline, as can be seen in Fig. 11(a)].

Use Case 6 (MEC-Enabled Multiuser VR Gaming Arcade): We consider a practical use case of wireless VR to deliver a low-latency service to the multiuser scenario of users playing VR video games in a gaming arcade, as illustrated in Fig. 12. This scenario, which is fully detailed in our previous work [74], is highly demanding due to the tight latency tolerance in VR as well as the state dynamics of the user due to the game-specific actions taken by themselves or by other players that affect what content should be shown to them. The users are served wirelessly through multiple mmAPs wired to edge computing and storage servers. These servers receive the users' 3-D location coordinates and their 3-D pose that consists of roll, pitch, and yaw angles, and their game-related actions. The servers will render the corresponding frames in HD resolution and deliver it wirelessly to users. Hence, the latency consists of the processing latency at the server and the communication latency to deliver the HD frames expressed as

$$D_{uf}(t) = \xi_{fu}(D^{\text{cp}}_{uf}(t) + D^{\text{cm}}_{uf}(t) + \tau_{\text{EP}}) \tag{6}$$

where ξ_{fu} represents a binary indicator that equals 1 when the HD video frame is delivered to VRP u and equals 0 if the low-quality (LQ) frame is delivered, D^{cp}_{uf} and

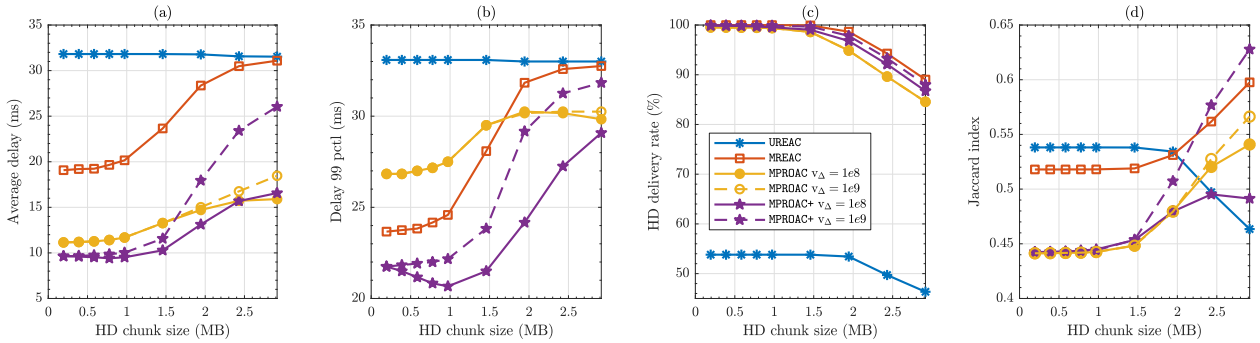


Fig. 11. ***(a) Average delay, (b) 99th percentile delay, (c) HD delivery rate, and (d) Jaccard index performance in $_{ST\text{-}3V}$, respectively, as a function of the HD chunk size, for $V = 3$ videos, $K = 2 \times V$ clusters, $T_H = 5$ frames, and Lyapunov tradeoff $V_\delta = 110^8$ and $V_\delta = 1 \cdot 10^9$.***

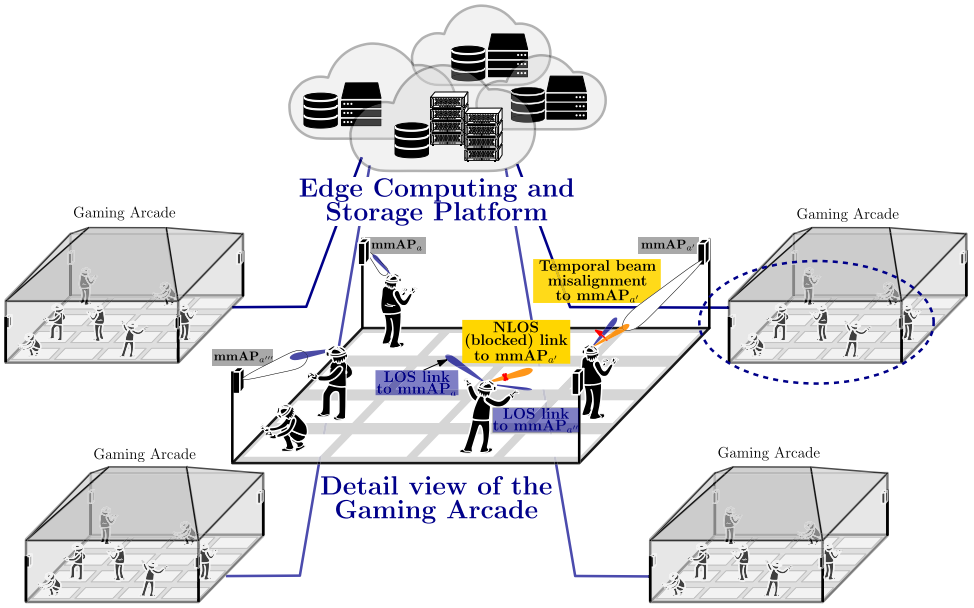


Fig. 12. ***Representation of a group of VR gaming arcades where HD frame computation is offloaded to an MEC platform such that input actions of the VRPs might impact the virtual environment shown to a subset of the remaining VRPs. The detailed view of the bottom arcade also illustrates several LOS and nonline-of-sight mmWave link states, e.g., link blockage and VRP and mmAP beam misalignment.***

D_{uf}^{cm} are the computing and communication delays of HD frame f initiated from user u, respectively, and τ_{EP} is the processing latency that accounts for the edge server processing, storage processing, and the UL transmission of user pose and action data. Let the computing delay D_{uf}^{cp} be expressed as follows:

$$D_{uf}^{\text{cp}}(t) = \left(\frac{\kappa L_{fu}^{\text{HD}}}{c_e} + W_{uf}(t)\right) z_{fu}(t)(1 - y_{fu}(t)) \qquad (7)$$

where c_e is the computation capability of edge server e, $z_{fu}(t)$ and $y_{fu}(t)$ indicate that the video frame f of user u is scheduled for computing and is cached in the fog network at time instant t, respectively, and W_{uf} is the computation waiting time of HD frame f of user u in the service queue, defined as $Q(t)$. Furthermore, let the communications delay D_{uf}^{cm} be given as

$$D_{uf}^{\text{cm}}(t) = \arg\min_{d_u} \sum_{t'=D_{uf}^{\text{cp}}(t)+1}^{D_{uf}^{\text{cp}}(t)+d_u} \left(T_t r_u(t') \geq L_{fu}^{\text{HD}}\right) \qquad (8)$$

where the arg min function is to find the minimum number of time slots needed for the video frame f to be delivered.

Here, we study two enablers to minimize the latency and boost the reliability of the VR gaming experience. For the computing latency, we investigate how prior knowledge of users' future pose using prediction methods affects the computing the latency. We leverage the results from the previous works as in [107], which states that the users' future pose in the next hundreds of milliseconds can be predicted with high accuracy to proactively predict, render, and cache the users' upcoming frames, subject to computation and storage resource availability. For the communication parts, the use of MC is considered to associate a user with more than one mmAPs if the SINR with its serving mmAP falls below a given threshold. Specifically, SFN operation is considered where multiple mmAP use the same frequency and time resource to transmit to the intended user.

Fig. 13 compares the communications and computing latency of our PROPOSED scheme that considers both

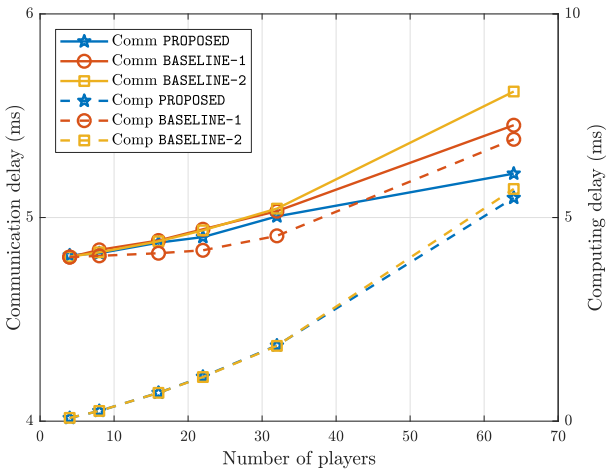


Fig. 13. ***Communication delay (solid lines) and computing delay (dashed lines) for different schemes as the number of players varies for an arcade of 16 mmAPs, each equipped with an edge computing unit.***

enablers of proactive computing and MC, with BASELINE-1 that does not have either of the two enablers and BASELINE-2 that considers only proactive computing. By looking into the computing latency in Fig. 13, we can see that the schemes with proactive computing significantly minimizes the computing latency, whereas a look at the communication latency shows the gain achieved using MC. Comparing the communication latency of BASELINE-1 and BASELINE-2 also shows that the proactive computing, which improved the computing performance, also slightly increases the communication latency. This is due to having to send additional data due to the errors in prediction, in which the correct data have to be retransmitted in real time.

V. CONCLUSION AND FUTURE OUTLOOK

Edge computing is an essential component of future wireless networks, in which several challenges need to be overcome to realize the vision of ultrareliable and low-latency edge computing. Chief to this vision is leveraging multiple high-reliability and low-latency enablers applied for different types of services and use cases. In this paper, we have discussed edge networking services and examined key enablers to achieve low-latency and high-reliability networking. Moreover, we showcased how the network resources can be optimized for a selection of use cases characterized by their shared need for edge networking. As the vision of 5G starts to materialize beyond its initial inception toward imminent first commercial deployments, we envision a realization of edge computing hand in hand with the development of URLLC and distributed artificial intelligence (AI) able to deal with dynamic and heterogeneous environments and provide seamless computing, content, and control services while preserving data privacy and security. ■

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[2] G. Klas, "Edge computing and the role of cellular networks," *Computer*, vol. 50, no. 10, pp. 40–49, 2017.

[3] A. Ndikumana *et al.* (2018). "Joint communication, computation, caching, and control in big data multi-access edge computing." [Online]. Available: https://arxiv.org/abs/1803.11512

[4] B. Soret, P. Mogensen, K. I. Pedersen, and M. C. Aguayo-Torres, "Fundamental tradeoffs among reliability, latency and throughput in cellular networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 1391–1396.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[6] ETSI. (2018). *Multi-Access Edge Computing (MEC)*. Accessed: Apr. 2, 2019. [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[7] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[8] E. Bastug, M. Bennis, and M. Debbah, "Social and spatial proactive caching for mobile data offloading," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 581–586.

[9] S. Tamoor-ul-Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-Aho, "Caching in wireless small cell networks: A storage-bandwidth tradeoff," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1175–1178, Jun. 2016.

[10] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 945–949.

[11] J. Park, S. Samarakoon, M. Bennis, and M. Debbah. (Dec. 2018). "Wireless network intelligence at the edge." [Online]. Available: https://arxiv.org/abs/1812.02858

[12] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proc. ACM Workshop Mobile Edge Commun. (MECOMM)*, 2018, pp. 31–36.

[13] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed intelligence in Internet of Things," *IEEE Access*, vol. 5, pp. 16441–16458, 2017.

[14] J.-M. Lasry and P.-L. Lions, "Mean field games," *Jpn. J. Math.*, vol. 2, no. 1, pp. 229–260, 2007.

[15] X. Chen, Q. Shi, L. Yang, and J. Xu, "ThriftyEdge: Resource-efficient edge computing for intelligent iot applications," *IEEE Netw.*, vol. 32, no. 1, pp. 61–65, Jan./Feb. 2018.

[16] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded MapReduce," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep./Oct. 2015, pp. 964–971.

[17] M. S. Elbamby, M. Bennis, W. Saad, M. Latva-Aho, and C. S. Hong, "Proactive edge computing in fog networks with latency and reliability guarantees," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 209, Aug. 2018.

[18] C. Perfecto, M. S. ElBamby, J. Del Ser, and M. Bennis. (2018). "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework." [Online]. Available: https://arxiv.org/abs/1811.07388

[19] C. Dehos, J. L. González, A. De Domenico, D. Kténas, and L. Dussopt, "Millimeter-wave access and backhauling: The solution to the exponential data traffic increase in 5G mobile communications systems?" *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 88–95, Sep. 2014.

[20] T. K. Vu, M. Bennis, S. Samarakoon, M. Debbah, and M. Latva-Aho, "Joint in-band backhauling and interference mitigation in 5G heterogeneous networks," in *Proc. 22nd Eur. Wireless Conf. Eur. Wireless*, 2016, pp. 1–6.

[21] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–6.

[22] J. Oueis, "Joint communication and computation resources allocation for cloud-empowered future wireless networks," Ph.D. dissertation, Univ. Grenoble Alpes, Grenoble, France, Feb. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01366449

[23] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, May 2013.

[24] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[25] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[26] M. Xiao *et al.*, "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sep. 2017.

[27] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks—With a focus on propagation models," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 6213–6230, Dec. 2017.

[28] M. R. Akdeniz *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014.

[29] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-wave communications: Physical channel models, design considerations, antenna constructions, and link-budget," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 870–913, 2nd Quart., 2018.

[30] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, "Millimeter-wave massive MIMO: The next wireless revolution?" *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 56–62, Sep. 2014.

[31] M. Rebato, J. Park, P. Popovski, E. De Carvalho, and M. Zorzi. (Jun. 2018). "Stochastic geometric coverage analysis in mmWave cellular networks with realistic channel and antenna radiation models." [Online]. Available: https://arxiv.org/abs/1806.04193

[32] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi. (2019). *A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies*. [Online]. Available: https://ieeexplore.ieee.org/document/8458146

[33] S. Barbarossa, E. Ceci, and M. Merluzzi, "Overbooking radio and computation resources in mmW-mobile edge computing to reduce vulnerability to channel intermittency," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2017, pp. 1–5.

[34] T. K. Vu, M. Bennis, M. Debbah, M. Latva-Aho, and C. S. Hong, "Ultra-reliable communication in 5G mmWave networks: A risk-sensitive approach," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 708–711, Apr. 2018.

[35] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 4th Quart., 2016.

[36] J. Park, S.-L. Kim, and J. Zander, "Tractable resource management with uplink decoupled millimeter-wave overlay in ultra-dense cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4362–4379, Jun. 2016.

[37] J. Park, S. Y. Jung, S.-L. Kim, M. Bennis, and M. Debbah, "User-centric mobility management in ultra-dense cellular networks under spatio-temporal dynamics," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[38] J. Park, S.-L. Kim, M. Bennis, and M. Debbah, "Spatio-temporal network dynamics framework for energy-efficient ultra-dense cellular networks," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–7.

[39] H. Kim, J. Park, M. Bennis, S.-L. Kim, and M. Debbah, "Ultra-dense edge caching under spatio-temporal demand and network dynamics," in *Proc. IEEE ICC*, Paris, France, May 2017, pp. 1–7.

[40] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-Aho, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1267–1280, May 2016.

[41] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[42] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular micro clouds as virtual edge servers for efficient data collection," in *Proc. 2nd ACM Int. Workshop Smart, Auto., Connected Veh. Syst. Services (CarSys)*, New York, NY, USA, 2017, pp. 31–35.

[43] J. Park, P. Popovski, and O. Simeone, "Minimizing latency to support VR social interactions over wireless cellular systems via bandwidth allocation," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 776–779, Oct. 2018.

[44] B. Romanous, N. Bitar, A. Imran, and H. Refai, "Network densification: Challenges and opportunities in enabling 5G," in *Proc. IEEE 20th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2015, pp. 129–134.

[45] R. Arshad, H. ElSawy, S. Sorour, T. Y. Al-Naffouri, and M. Alouini, "Handover management in dense cellular networks: A stochastic geometry approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[46] Y. Wang, M. Narasimha, and R. W. Heath, Jr. (May 2018). "MmWave beam prediction with situational awareness: A machine learning approach." [Online]. Available: https://arxiv.org/abs/1805.08912

[47] N. Kato *et al.*, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 146–153, Jun. 2017.

[48] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 1st Quart., 2018.

[49] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang. (2018). "Towards an intelligent edge: Wireless communication meets machine learning." [Online]. Available: https://arxiv.org/abs/1809.00343

[50] S.-W. Ko, K. Huang, S.-L. Kim, and H. Chae, "Live prefetching for mobile computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3057–3071, May 2017.

[51] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[52] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coding for distributed fog computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 34–40, Apr. 2017.

[53] A. Ravanshid *et al.*, "Multi-connectivity functional architectures in 5G," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2016, pp. 187–192.

[54] B. Soret, P. Mogensen, K. I. Pedersen, and M. C. Aguayo-Torres, "Fundamental tradeoffs among reliability, latency and throughput in cellular networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM) Workshops*, Dec. 2014, pp. 1391–1396.

[55] A. Wolf, P. Schulz, M. Dörpinghaus, J. C. S. S. Filho, and G. Fettweis, "How reliable and capable is multi-connectivity?" *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1506–1520, Feb. 2018.

[56] *Coordinated Multi-Point Operation for LTE Physical Layer Aspects*, document 3GPP TR36.819, 36.819 Rel-11, Sep. 2013.

[57] M. Eriksson, "Dynamic single frequency networks," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 1905–1914, Oct. 2001.

[58] F. B. Tesema, A. Awada, I. Viering, M. Simsek, and G. P. Fettweis, "Multiconnectivity for mobility robustness in standalone 5G ultra dense networks with intrafrequency cloud radio access," *Wireless Commun. Mobile Comput.*, vol. 2017, Jan. 2017, Art. no. 2038078.

[59] *Evolved Universal Terrestrial Radio Access (E-UTRA); Carrier Aggregation Enhancements; UE and BS Radio Transmission and Reception*, document 3GPP TR 36.823, 36.823 Rel-11, Nov. 2013.

[60] F. B. Tesema, A. Awada, I. Viering, M. Simsek, and G. P. Fettweis, "Mobility modeling and performance evaluation of multi-connectivity in 5G intra-frequency networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[61] M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Multi-connectivity in 5G mmWave cellular networks," in *Proc. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2016, pp. 1–7.

[62] M. Giordani, M. Mezzavilla, and M. Zorzi, "Initial access in 5G mmWave cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 40–47, Nov. 2016.

[63] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmWave mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2069–2084, Sep. 2017.

[64] J. J. Nielsen and P. Popovski, "Latency analysis of systems with multiple interfaces for ultra-reliable M2M communication," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2016, pp. 1–6.

[65] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013.

[66] O. Bousquet, U. von Luxburg, and G. Rätsch, "Introduction to statistical learning theory," in *Advanced Lectures on Machine Learning*. Berlin, Germany: Springer, 2004, pp. 169–207.

[67] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. NIPS Wksp. PMPML*, Barcelona, Spain, Dec. 2016.

[68] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, Fort Lauderdale, FL, USA, Apr. 2017.

[69] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[70] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*. London, U.K.: Springer, 2001.

[71] L. de Haan and A. F. Ferreira, *Extreme Value Theory: An Introduction*. New York, NY, USA: Springer, 2006.

[72] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

[73] M. Schneider, J. Rambach, and D. Stricker, "Augmented reality based on edge computing using the example of remote live support," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 1277–1282.

[74] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[75] "Augmented and virtual reality: The first wave of 5g killer apps," ABI Res. Qualcomm, White Paper, Feb. 2017.

[76] J. Park and M. Bennis. (May 2018). "URLLC-eMBB slicing to support VR multimodal perceptions over wireless cellular systems." [Online]. Available: https://arxiv.org/abs/1805.00142

[77] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[78] L. Nanjie, "Internet of vehicles: Your next connection," in *WinWin Magazine*. Huawei Communicate, Dec. 2011.

[79] P. Gao, R. Hensley, and A. Zielke. (Oct. 2014). A Road Map to the Future for the Auto Industry. McKinsey Quarterly. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/a-road-map-to-the-future-for-the-auto-industry

[80] C. Perfecto, J. Del Ser, M. Bennis, and M. N. Bilbao, "Beyond WYSIWYG: Sharing contextual sensing data through mmWave V2V communications," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, vol. 6, 2017, pp. 1–6.

[81] C. Perfecto, J. Del Ser, and M. Bennis, "Millimeter-wave V2V communications: Distributed association and beam alignment," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2148–2162, Jun. 2017.

[82] C. Perfecto, J. Del Ser, and M. Bennis, "On the interplay between scheduling interval and beamwidth selection for low-latency and reliable V2V mmWave communications," in *Proc. IEEE 20th Conf. Innov. Clouds, Internet Netw. (ICIN)*, Mar. 2017, pp. 1–8.

[83] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 98–104, Feb. 2016.

[84] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[85] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan./Feb. 2018.

[86] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, "BEGIN: Big data enabled energy-efficient vehicular edge computing," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 82–89, Dec. 2018.

[87] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 62–70, Mar. 2019.

[88] T. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[89] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[90] *Consumer Reports—Make Your Car Last 200, 000 Miles*. Accessed: Jan. 15, 2019. [Online]. Available: https://www.consumerreports.org/car-repair-maintenance/make-your-car-last-200-000-miles/

[91] H. Kim, J. Park, M. Bennis, and S.-L. Kim. (Aug. 2018). "On-device federated learning via blockchain and its latency analysis." [Online]. Available: https://arxiv.org/abs/1808.03949

[92] M. M. Amiri and D. Gunduz. (Jan. 2019). "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air." [Online]. Available: https://arxiv.org/abs/1901.00844

[93] S. Ha, J. Zhang, O. Simeone, and J. Kang. (2019). "Coded federated computing in wireless networks with straggling devices and imperfect CSI." [Online]. Available: https://arxiv.org/abs/1901.05239

[94] S. Wang *et al.* (Aug. 2018). "Adaptive federated learning in resource constrained edge computing systems." [Online]. Available: https://arxiv.org/abs/1804.05271

[95] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Federated distillation and augmentation under non-IID private data," presented at the NIPS Wksp. MLPCD, Dec. 2018.

[96] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[97] M. Mirza and S. Osindero. (Nov. 2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784

[98] L. Wang *et al.*, "Superneurons: Dynamic GPU memory management for training deep neural networks," *ACM Special Interest Group Program. Lang.*, vol. 53, no. 1, pp. 41–53, Feb. 2018.

[99] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Global Commun. Conf. Workshops*, Dec. 2017, pp. 1–7.

[100] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor. (2019). "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing." [Online]. Available: https://arxiv.org/abs/1812.08076

[101] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah. (Jul. 2018). "Distributed federated learning for ultra-reliable low-latency vehicular communications." [Online]. Available: https://arxiv.org/abs/1807.08127

[102] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[103] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, to be published.

[104] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[105] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Emp. Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[106] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Wksh. Deep Learn.*, 2014.

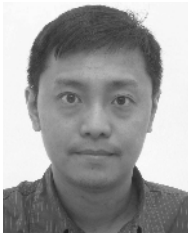
[107] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360° video delivery over cellular networks," in *Proc. 5th Workshop Things Cellular, Oper., Appl. Challenges (ATC)*, New York, NY, USA, 2016, pp. 1–6.

ABOUT THE AUTHORS

Mohammed S. Elbamby received the B.Sc. degree (Hons.) in electronics and communications engineering from the Institute of Aviation Engineering and Technology, Giza, Egypt, in 2010, and the M.Sc. degree in communications engineering from Cairo University, Giza, in 2013. He is currently working toward the Dr.Tech. degree at the University of Oulu, Oulu, Finland.

After completing the M.Sc. degree, he joined the Centre for Wireless Communications, University of Oulu. His current research interests include resource optimization, uplink and downlink configuration, fog networking, and caching in wireless cellular networks.

Mr. Elbamby received the Best Student Paper Award from the European Conference on Networks and Communications in 2017.



Chen-Feng Liu (Student Member, IEEE) received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, in 2009, and the M.S. degree in communications engineering from National Chiao Tung University, Hsinchu, in 2011. He is currently working toward the Ph.D. degree at the University of Oulu, Oulu, Finland.

In 2012, he joined Academia Sinica, Taipei, Taiwan, as a Research Assistant. In 2014, he was a Visiting Researcher with the Singapore University of Technology and Design, Singapore. He was a Visiting Ph.D. Student with the University of Houston, Houston, TX, USA, in 2016, and New York University, New York, NY, USA, in 2018. His current research interests include 5G communications, mobile edge computing, ultrareliable low-latency communications, and wireless artificial intelligence.

Cristina Perfecto (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunication engineering from the University of the Basque Country (UPV/EHU), Bilbao, Spain, in 2000. She is currently working toward the Ph.D. degree with a focus on the application of multidisciplinary computational intelligence techniques in radio resource management for 5G.

From 2016 to 2018, she was a Visiting Researcher with the Centre for Wireless Communications, University of Oulu, Oulu, Finland. She is currently a College Associate Professor with the Department of Communications Engineering, UPV/EHU. Her current research interests include millimeter-wave communications and the application of machine learning in 5G networks.

Jihong Park received the B.S. and Ph.D. degrees from Yonsei University, Seoul, South Korea, in 2009 and 2016, respectively.

From 2016 to 2017, he was a Postdoctoral Researcher with Aalborg University, Aalborg, Denmark. He was a Visiting Researcher with The Hong Kong Polytechnic University, Hong Kong, KTH, Stockholm, Sweden, Aalborg University, and the New Jersey Institute of Technology, Newark, NJ, USA, in 2013, 2015, 2016, and 2017, respectively. He is currently a Postdoctoral Researcher with the University of Oulu, Oulu, Finland. His current research interests include ultradense/ultrareliable/massive multiple-input multiple-output system designs using stochastic geometry and network economics.

Dr. Park's papers on tractable ultradense network analysis received the IEEE GLOBECOM Student Travel Grant in 2014, the IEEE Seoul Section Student Paper Contest Bronze Prize in 2014, and the 6th IDIS-ETNEWS (The Electronic Times) Paper Contest Award sponsored by the Ministry of Science, ICT, and Future Planning of Korea.

Sumudu Samarakoon (Associate Member, IEEE) received the B.Sc. degree (Hons.) in electronic and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2009, the M.Eng. degree from the Asian Institute of Technology, Pathumthani, Thailand, in 2011, and the Ph.D. degree in communication engineering from the University of Oulu, Oulu, Finland, in 2017.

He is currently a Postdoctoral Researcher with the Centre for Wireless Communications (CWC), University of Oulu. His current research interests include heterogeneous networks, small cells, radio resource management, reinforcement learning, and game theory.

Dr. Samarakoon received the Best Paper Award from the European Wireless Conference and Excellence Awards for innovators and the Outstanding Doctoral Student at the Radio Technology Unit, CWC, University of Oulu, in 2016.

Xianfu Chen (Member, IEEE) received the Ph.D. degree in signal and information processing from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, in 2012.

He is currently a Senior Scientist with the VTT Technical Research Centre of Finland, Oulu, Finland. His current research interests include various aspects of wireless communications and networking, with an emphasis on network virtualization, software-defined radio access networks, green communications, centralized and decentralized resource allocation, dynamic spectrum access, and the application of machine learning to wireless communications.

Mehdi Bennis (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from the École polytechnique fédérale de Lausanne, Lausanne, Switzerland, and the Eurecom Institute, Biot, France, in 2002, and the Ph.D. degree in spectrum sharing for future mobile cellular systems from the University of Oulu, Oulu, Finland, in 2009.

From 2002 to 2004, he was a Research Engineer with Imra Europe, Valbonne, France, investigating adaptive equalization algorithms for mobile digital TV. In 2004, he joined the Centre for Wireless Communications (CWC), University of Oulu, as a Research Scientist. In 2008, he was a Visiting Researcher with the Alcatel-Lucent Chair on Flexible Radio, Supélec, Gif-sur-Yvette, France. He is currently an Associate Professor with the University of Oulu and a Research Fellow with the Academy of Finland, Finland. He has coauthored one book and published more than 100 research papers in international conferences, journals, and book chapters. His current research interests include radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond.

Dr. Bennis was a recipient of the prestigious 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, and the 2017 EURASIP Best Paper Award for the *Journal on Wireless Communications and Networking*. He serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.