IEEE Open Journal of the Communications Society

Received 7 August 2023; revised 12 September 2023; accepted 26 September 2023. Date of publication 5 October 2023; date of current version 20 October 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3322383

Enhancing XR Application Performance in Multi-Connectivity Enabled mmWave Networks

MUHAMMAD AFFAN JAVED (Member, IEEE), PEI LIU (Member, IEEE), AND SHIVENDRA S. PANWAR (Fellow, IEEE)

Department of Electrical and Computer Engineering, New York University Tandon School of Engineering, Brooklyn, NY 11201, USA

CORRESPONDING AUTHOR: M. A. JAVED (e-mail: maj407@nyu.edu)

This work was supported in part by NYU Wireless; in part by the NY State Center for Advanced Technology in Telecommunications (CATT); and in part by the NYU IT High-Performance Computing Resources, Services, and Staff Expertise.

ABSTRACT mmWave communications are paving the way for next-generation cellular networks due to their inherent ability to provide high data rates and mitigate interference. Coupled with this are the enormous potential and challenges posed by eXtended Reality (XR) applications which are becoming increasingly ubiquitous. In this paper, we leverage the unique characteristics of mmWave networks to re-think and re-design fundamental network architecture and functions in order to meet the strict requirements of deadline-driven XR applications. We propose a multi-tiered multi-connectivity architecture that allows users (UEs) to connect to multiple base stations (gNBs) simultaneously and switch rapidly between them in case of blockages. By replicating UE data at multiple gNBs close to the UE, we ensure that we satisfy strict Quality of Service (QoS) constraints even with unpredictable, dynamic blockages of the mmWave links. We show through extensive system-level simulations that our network architecture allows us to shield UEs from high handover delays and minimizes data plane interruptions in case of blockages. Moreover, we note that existing algorithms for network functions such as gNB selection and scheduling are not optimized for the multi-connectivity paradigm, nor do they specifically cater to strict deadline constraints or intermittent wireless links. We propose a Deep Reinforcement Learning framework that selects gNBs for data replication by explicitly optimizing to meet strict deadline constraints of XR traffic. Our Deep Learning agent analyzes global state information and predicts the best selection of gNBs to preemptively replicate data for future transmissions. Furthermore, we propose a scheduler based on maximal weight matching, dubbed $\beta-$MWM, which is specifically tailored to exploit multi-connectivity. We show that our Deep Learning based Data Replication Predictor and $\beta-$MWM scheduler perform better than existing, conventional algorithms and result in markedly better performance for XR applications with strict deadlines.

INDEX TERMS Blockages, deadline-driven scheduling, deep learning, DQN, handover, low latency, millimeter wave, mmWave, multi-connectivity, quality of service, reinforcement learning, XR applications.

I. INTRODUCTION

THE PROMISE of eXtended Reality (XR) applications, which include Virtual Reality (VR), Augmented Reality (AR), and Cloud Gaming (CG), has taken the world by storm [1]. These services are the cornerstone of next-generation wireless networks and fundamental changes in network architecture and protocols are needed in order to meet their requirements of high bandwidths, low latencies, and strict deadlines. Currently, it is daunting to support XR applications over 3GPP New Radio (NR) cellular networks because XR does not perfectly fit into the existing classification of fifth-generation (5G) applications and services,

TABLE 1. QoS requirements for XR applications fall outside those for the traditional three categories of services/applications defined for 5G networks [7]. XR is characterized by both a high data rate and strict packet delay budget (PDB) [3], placing it in between 5G eMBB and URLLC.

Application Category	Throughput	Air-link Latency	Reliability	Example Applications
eMBB	0.1 - 10 Gbps	4 ms (single packet)	99.9%	360 degrees video
URLLC	1-10 Mbps	1 ms	99.9% - 99.9999%	robotics, factory automation
mMTC	1-100 Kbps	10 ms - 1hr	90%	smart home
XR	30 - 300 Mbps	10 ms -15 ms (frame)	99% - 99.9%	VR, AR and CG

typically divided into enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), and massive machine-type communications (mMTC). Given its high data rate and strict packet delay budget (PDB) requirements, XR fits into neither of these categories, instead falling somewhere between eMBB and URLLC. The Quality of Service (QoS) requirements of these applications are given in Table 1 [2], [3]. Given massive interest in XR applications, 3GPP has, in recent years, taken some steps towards standardization of requirements and models for XR applications. 3GPP Radio Access Network (RAN) WG1 ("Physical layer," RAN1) performed a Release 17 study on evaluating NR performance for XR applications [3]. This work now continues for Release 18 (the first 5G-Advanced release, tentatively, until 2023), aiming to provide necessary enhancements to better support XR services over NR.

Given the importance of catering to XR applications' QoS requirements in future wireless networks, it is important to take a step back and evaluate the existing capabilities of 5G networks, and identify features that can be exploited to further enhance performance for XR applications. 5G cellular networks have already led the charge into mmWave technology, which operates at frequencies above 24 GHz, thereby utilizing the enormous amount of spectrum available in these frequency bands [4]. At these frequencies, the radio propagation characteristics are starkly different from their microwave counterparts. First, according to the Friis transmission equation [5], the path loss can exhibit 30-40 dB more attenuation. This higher path loss necessitates focusing power into fairly narrow and very directional beams, that can be realized through phased antenna arrays, whose implementation is made possible thanks to the smaller wavelengths that correspond to these frequencies. Furthermore, due to the exacerbated blockage and shadowing effects [6], the wireless links exhibit rapid variations in quality, thereby leading to intermittency in link connectivity between the user (UE) and the base station (gNB).

To address these challenges, and to maintain an acceptable level of service despite this intermittency, the density of gNBs in mmWave cellular networks is expected to be significantly higher than in sub-6 GHz systems [8]. It will greatly benefit the UEs to harness macro-diversity from the nearby gNBs in sixth-generation (6G) and future cellular networks. Exploiting multi-connectivity in the access network to gain better performance is not a new concept, nor is it unique to mmWave networks. In fact, multi-connectivity was first proposed for sub-6 GHz networks with the introduction of Dual Connectivity (DC) in heterogeneous Long Term Evolution (LTE) networks in 3GPP Release 12 [9]. Although DC contributed to throughput gains, it did not gain much traction in sub-6 GHz networks because the overhead involved in maintaining dual connectivity far outweighed any performance improvements to be had. With the move towards mmWave networks in 5G, multi-connectivity has received renewed interest due to several reasons. First, it is easier for a UE to be within range of multiple gNBs due to the high densification of gNBs required to provide adequate coverage at mmWave frequencies. Second, directional beams in mmWave networks offer an opportunity to provide multi-connectivity without creating excessive interference between neighboring gNBs. Last, meeting the strict QoS constraints of next-generation applications such as XR provides further incentives that make the high overhead cost of multi-connectivity tolerable from a cost-benefit trade-off perspective. In this paper, we leverage mmWave multi-connectivity to propose a multi-tiered network architecture designed to enhance the performance of low latency applications, like XR, which have strict deadlines. We then use the multi-connectivity architecture to propose several enhancements to key network functions, such as gNB selection for data replication and deadline-driven scheduling, in order to extract maximum benefit from our multi-connectivity architecture. The key contributions of this paper are as follows:

- We propose a multi-tiered network architecture for mmWave multi-connectivity in the access network that provides better performance even with conventional scheduling algorithms. We show that our architecture allows us to shield the UEs from high handover latencies in case of blockages, minimizes data plane interruptions and enables fast switching between multiple gNBs.
- We pose the Predictive Data Replication problem as a reinforcement learning problem, and use a Deep Q-Network (DQN) to solve for near-optimal solutions. The DQN agent takes in global state information, including

traffic and channel conditions, and finds the best set of gNBs to replicate data for each UE.
- We also propose a maximal weight matching scheduling algorithm, dubbed β-MWM, which is tailored specifically for deadline-driven traffic in a multi-connectivity enabled network. The β-MWM scheduler strikes a balance between prioritizing traffic with earlier deadlines and prioritizing UEs with better channel conditions.
- We present system-level performance evaluation results for XR applications in a multi-cell mmWave network using the statistical traffic model given in 3GPP standards. Our results show that our DQN Predictor and β-MWM scheduling agent outperform conventional algorithms and lead to better performance for XR traffic with strict deadlines.

The rest of the paper is organized as follows. Section II presents related work. We propose our multi-connectivity architecture in Section III. System models are described in Section IV and problem formulation is presented in Section V. In Section VI we describe our simulation setup and implementation, present results obtained by our simulations, and discuss the key takeaways. Finally, Section VII concludes our paper and highlights possible avenues for future research.

II. RELATED WORK

Multi-connectivity in mmWave networks has been studied in [10], where the impact of gNB discovery time, handover execution times, and degree of multi-connectivity was studied with respect to QoS criteria such as out-of-service probability, outage duration, and radio link failure (RLF) probability. However, the weakness of the proposed architecture was that data would either have to be replicated at all connected base stations, which would be prohibitively expensive in practice, or would have to be redirected from the Master base station to the Secondary base stations, which would incur additional delays. Moreover, the expressions derived in [10] do not provide any explicit performance guarantees for XR applications. In [11], a new transport network architecture was proposed that would enable fast control signaling and leverage multi-connectivity, via a fiber ring, to improve QoS for different applications. Petrov et al. [12] considered different multi-connectivity scenarios to study the impact of the degree of connectivity, and showed that a high degree of multi-connectivity would enhance the reliability of the system at the cost of significant signaling and computation overhead. On a similar note, Gapayenko et al. [13] showed that increasing the degree of multi-connectivity up to 4 could provide benefits in terms of lower outage probability and higher spectral efficiency. In [14], a multi-label classification approach for user association is proposed for multi-connectivity enabled mmWave networks. However, this work has several drawbacks: blockages or mobility are not explicitly addressed in the system model, the optimization problem is framed as a system throughput maximization problem which is not suitable for deadline-driven XR applications and the proposed method is evaluated only for a small test case comprising of 8 gNBs and 8 UEs. Similarly, a joint user association and power control optimization scheme is presented in [15] with the aim of optimizing energy efficiency. However, this work also does not address the unique characteristics of XR applications, nor does it take blockages into account.

With regards to standardization, in Release 12 3GPP introduced the Intra-E-UTRA Dual Connectivity (DC) which is the inter-site DC between two LTE base stations where both base stations are connected to the Evolved Packet Core (EPC). Since then, 3GPP has iteratively expanded on use cases and functionality of dual connectivity, and it is now a key feature of the 5G NR standard. According to the 3GPP NR Release 16 standard [16], Multi-Radio Dual Connectivity (MR-DC) is the term that is generally used for multi-connectivity. With the introduction of 5G NR, 3GPP introduced four configurations for MR-DC, of which only one (NR-NR Dual Connectivity or NR-DC) falls under the standalone architecture and represents the 5G equivalent of the LTE DC. Proposals for multi-connectivity architectures in literature (including 5G NR-DC) assume that multiple connections would be active simultaneously, but don't address limitations in the data plane. As a result, there are two possible ways of dealing with a blockage in the primary link: either replicating data at all connected base stations (which would incur high overhead) or forwarding the data from the primary to the secondary base stations (which would incur additional delays). In order to solve this issue, we replicate data intelligently at a subset of the connected base stations, thus operating at the optimal point between the two extreme solutions listed above.

The performance of XR applications in different networks and systems has also been an area of keen interest recently. In [17], system-level performance results for XR over a 5G-NR network were presented and several enhancements, such as traffic-aware scheduling, were proposed in order to boost the performance. Petrov et al. [18] also performed a case study that demonstrated that 5G NR can already support XR services, but with a limitation on the number of XR devices per cell at high data rates. A key drawback of these studies is that they fail to explicitly take into account the effect of blockages, which severely affect the performance of any mmWave network. Thus, existing studies have either been done on a) XR application performance in mmWave networks (but without taking multi-connectivity or blockages into account) [17], [18] or b) studying outage probability and duration in multi-connectivity enabled mmWave networks, but without studying the impact on XR application performance [10], [11]. We bridge the gap between these two and argue that it is important to explicitly model blockages and study their impact on XR applications, and then analyze how a multi-connectivity enabled architecture can help in offsetting the impact of those blockages. In [19] deep neural networks and mmWave multicast transmissions are used to

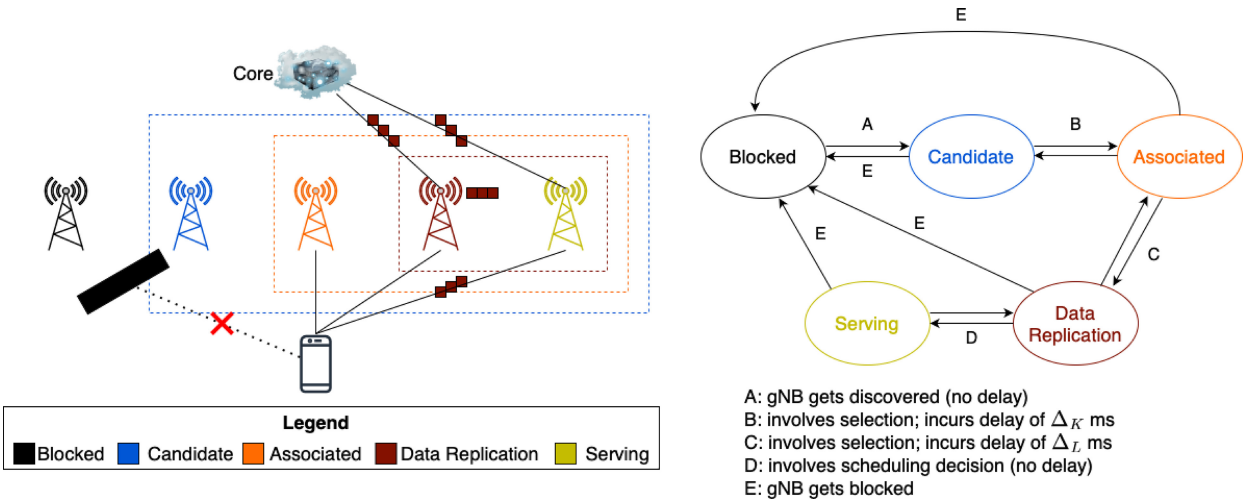


FIGURE 1. Network architecture, illustrating the different tiers of multi-connectivity. Here, $K = 3$ and $L = 2$.

decrease the streaming latency in a cooperative VR environment. The deep learning agent is used to estimate the upcoming viewports of users. The users are grouped based on predicted correlations and proactive multicast resource scheduling is then performed to minimize the latency and traffic volume for VR. While intriguing, this work also does not address the effect of blockages and how data plane interruptions in presence of blockages can be circumvented. Reference [20] investigate the usage of highly dense THz networks for catering to VR traffic. They show that availability of line-of-sight links is critical for performance of THz links, and also take self-blockages and dynamic blockages into account. However, a multi-connectivity THz scenario is not explored.

As XR applications operate under strict deadline constraints, deadline-driven scheduling is of particular interest to us. Existing literature on deadline-driven scheduling can be broadly classified into two categories: a) delay-constrained wireless networks and b) communication networks with hard deadlines. Literature belonging to the first category deal mainly with wireless networks in a scenario where delays are consequential to the network performance and cannot be ignored or ruled out. However, these works often do not associate hard deadlines with traffic and usually use a throughput/utility maximization approach to solving the network problem. Works in the second category, however, associate hard deadlines with traffic in different communication networks subject to different assumptions and constraints. An important point to note here is that these are formulations for communication networks in general, and not for wireless networks in particular. One main aspect of all these works is that they are typically formulated as a network where delivering traffic within the deadline is rewarded and delivering it after the deadline is penalized. The optimization problem typically seeks to maximize the amount of traffic delivered within the deadline. Some common assumptions taken in order to simplify the problem and make it more tractable are:

- Packet arrivals are periodic [21], [22], [23].
- Packets are dropped from the system after their deadline has passed [22], [23], [24].
- The channel is reliable and there are no interference constraints [23], [24], [25].
- Centralized coordination across the network [21].

Reference [26] also models mmWave V2V networks with hard deadline constraints similar to what we propose for XR, and vehicle matching is optimized to minimize delays. However, the vehicular network and scenarios considered are significantly different from our study where we have pedestrian users and XR traffic. To the best of our knowledge, deadline-driven scheduling in multi-connectivity enabled wireless networks has not been addressed in the existing literature.

III. MULTI-CONNECTIVITY ARCHITECTURE

We consider a mmWave wireless network comprising of a set of Base Stations (gNBs), $|\mathcal{M}| = M$, and a set of users (UEs), $|\mathcal{N}| = N$. Thus, there are up to $M \times N$ mmWave links in the system. The critical component of our infrastructure is the UEs' ability to connect to multiple gNBs simultaneously, a feature of emerging 3GPP standards [16]. The cornerstone of this architecture is that it further devolves multi-connectivity into two main tiers, Association and Data Replication, based on the level of connection and data availability [27]. The bifurcation of the multi-connectivity architecture is motivated in part by the overhead costs of replicating UE data at a large number of gNBs. By choosing to associate with a larger number of gNBs, and replicating the data at only a smaller subset of them we can reap the benefits of a higher degree of connectivity while significantly reducing the overhead costs. Moreover, the two-tier architecture allows us to reduce the handover delay experienced by the UEs in the vast majority of blockage scenarios. This allows us to minimize data plane interruptions and boost QoS performance for XR applications. Fig. 1 depicts our multi-connectivity network architecture.

A. MULTI-CONNECTIVITY TIERS

Since the range of mmWave links is quite short, it is possible that some gNBs are out of range of the UEs and, hence, no connection is possible. Even if a gNB is within range, it is possible that it is blocked and hence undiscovered by the UE. We define a set $\mathcal{C}_{n,t} \subset \mathcal{M}$, which comprises all the *candidate* gNBs for user n at time t:

$$\mathcal{C}_{n,t} = \{m : \sigma_{m,n} > \sigma_{th}, |\mathcal{C}_{n,t}| \leq M \forall m \in \mathcal{M}\}, \quad (1)$$

where $\sigma_{m,n}$ is the signal-to-noise ratio (SNR) of the link between the gNB m and UE n, and σ_{th} is the minimum SNR required for a successful connection between a gNB-UE pair. User n is in the range of all gNBs in $\mathcal{C}_{n,t}$ and can associate with any of them.

In the multi-connectivity setting, we assume that a UE can be *associated* with multiple gNBs at the same time. Specifically, the UE maintains a control plane connection with all the gNBs in the Associated set ($\mathcal{K}_{n,t}$). We define K, the degree of association, which determines the maximum number of gNBs a UE will simultaneously associate with, i.e., $|\mathcal{K}_{n,t}| \leq K$.

The set $\mathcal{K}_{n,t} \subset \mathcal{C}_{n,t}$ comprises the gNBs that UE n is associated with at time t. We assume that the best subset of gNBs to associate with is the set of gNBs with the highest channel quality to the UE. The algorithm for selecting $\mathcal{K}_{n,t}$ would start with an ordered set of SNRs and pick the gNBs corresponding to the K highest SNR values. An *associated* gNB-UE pair would have an active control channel open between them and routinely exchange control messages and signaling required to maintain the UE state at the gNB, as well as any signaling required for beam tracking, alignment, and beam switching. However, associated gNBs (except for one) do not have a data plane connection with the UE or up-to-date UE data available for delivery.

A smaller subset of $\mathcal{K}_{n,t}$ is then chosen as the Data Replication set of gNBs ($\mathcal{L}_{n,t}$). gNBs in $\mathcal{L}_{n,t}$ pre-fetch UE data and track UE data delivery status. The set $\mathcal{L}_{n,t} \subset \mathcal{K}_{n,t}$ is the set of all gNBs that are associated with UE n and have copies of UE n's data ready for transmission at time t. It is important to note here that XR applications are highly interactive, and generate data based on the UE's movement and actions. As such, data cannot be buffered in advance as it can be for video-on-demand applications. Instead, our architecture allows all gNBs in $\mathcal{L}_{n,t}$ to receive up-to-date UE data, which can be done via multicast as proposed in [28], and track acknowledgments to keep up-to-date regarding the current delivery status of UEs' data. We also define L, where $L \leq K$, as the degree of replication - another parameter that determines the maximum number of gNBs that will replicate the UE data and have it instantaneously ready for transmission, i.e., $|\mathcal{L}_{n,t}| \leq L$. At any given instance, a UE will have a data plane connection open with only one Serving gNB, which is chosen from $\mathcal{L}_{n,t}$ by the scheduling agent. The scheduling agent's job includes selecting a Serving gNB for the UE from $\mathcal{L}_{n,t}$. Thus, $\mathcal{L}_{n,t}$ consists of one master/serving gNB and several other secondary gNBs. We assume zero delays in the selection of a Serving gNB from $\mathcal{L}_{n,t}$ - hence, there are no data plane interruptions until and unless all gNBs in $\mathcal{L}_{n,t}$ get blocked.

B. HANDOVER PROCESS

The gNB status depends upon whether the link between the gNB and the UE is blocked or unblocked. Until a gNB-UE link becomes unblocked, the gNB cannot be discovered by the UE. Even after a gNB-UE link gets unblocked, it remains undiscovered until the UE discovers the gNB through physical layer procedures, such as cell search and measurement reports. We disregard the gNB discovery time, as the discovery procedure for new gNBs can occur in the background if a UE is still associated with other discovered gNBs. A discovered gNB is a *candidate* for association. The association procedure or the association handover delay (in case one gNB from $\mathcal{K}_{n,t}$ gets blocked, and another gNB from $\mathcal{C}_{n,t}$ is chosen to replace it) takes up to Δ_K ms.

The induction of a gNB from $\mathcal{K}_{n,t}$ to $\mathcal{L}_{n,t}$ incurs an additional handover delay of Δ_L ms, which is the delay incurred in fetching the UE data so that it is available for immediate delivery. This transition also involves selection and is of particular interest to us because it determines the set of gNBs where the UE's data will be replicated. Finally, the scheduling agent picks one gNB from $\mathcal{L}_{n,t}$ to be the Serving gNB. The Serving gNB can change either due to necessity, i.e., if the current Serving gNB gets blocked and the scheduling agent is forced to switch to another gNB, or due to choice, i.e., if the scheduling agent decides that switching to another gNB is the optimal action according to its scheduling policy.

Consider the following blockage scenarios, and how they translate to data plane interruptions at the UE:

- *Serving gNB gets blocked:* Instantaneous switching occurs to another gNB in $\mathcal{L}_{n,t}$. No handover delay is incurred nor is there any data plane interruption.
- *non-Serving gNB in $\mathcal{L}_{n,t}$ gets blocked:* The gNB is immediately dropped from $\mathcal{C}_{n,t}$, $\mathcal{K}_{n,t}$ and $\mathcal{L}_{n,t}$. After a handover delay of Δ_L ms, a new gNB from $\mathcal{K}_{n,t}$ is added to $\mathcal{L}_{n,t}$. Similarly, to replace the blocked gNB, a new gNB from $\mathcal{C}_{n,t}$ is added to $\mathcal{K}_{n,t}$ after a handover delay of Δ_K ms. However, these handovers occur in the background and do not interrupt the UE data plane as long as there is still one unblocked gNB available in $\mathcal{L}_{n,t}$.
- *gNB in $\mathcal{K}_{n,t}$ gets blocked:* The gNB is immediately dropped from $\mathcal{C}_{n,t}$ and $\mathcal{K}_{n,t}$. After a handover delay of Δ_K ms, a new gNB from $\mathcal{C}_{n,t}$ is added to $\mathcal{K}_{n,t}$. There is no UE data plane interruption.
- *All gNBs in $\mathcal{L}_{n,t}$ get blocked concurrently:* UE experiences a maximum data plane interruption of Δ_L ms, the time needed for unblocked gNBs from $\mathcal{K}_{n,t}$ to be added to $\mathcal{L}_{n,t}$.
- *All gNBs in $\mathcal{K}_{n,t}$ get blocked concurrently:* UE experiences a maximum data plane interruption of $(\Delta_K + \Delta_L)$ ms, while new gNBs from $\mathcal{C}_{n,t}$ are chosen for $\mathcal{K}_{n,t}$, and $\mathcal{L}_{n,t}$ is chosen from the new $\mathcal{K}_{n,t}$.

Thus, the UE will be out-of-service, and hence experience data plane interruption, in the following scenarios: 1) UE is out of coverage or completely blocked from all of the gNBs in its coverage region, i.e., $\mathcal{C}_{n,t} = \emptyset$, 2) all the gNBs in $\mathcal{L}_{n,t}$ get blocked, and an unblocked gNB from $\mathcal{K}_{n,t}$ is not added promptly enough due to handover execution times to prevent a period of blockage, and 3) all the gNBs in $\mathcal{K}_{n,t}$ get blocked, and an unblocked gNB from $\mathcal{C}_{n,t}$ is not added promptly enough due to handover execution times to prevent a period of blockage.

Of course, the degree of association, K, and the degree of replication, L, are two important parameters that influence the extent to which the UE is shielded from data plane interruptions in case of blockages. Associating with, and replicating the data, at a larger number of gNBs results in significantly larger overhead costs. We explore this trade-off between better performance and larger overhead to determine the optimal choice of K and L.

IV. SYSTEM MODEL

The inherent randomness of the environment is captured by two important parts of the model: the channel state model which models the mmWave links, and the UE traffic model which models the statistics of the arrival processes at the UEs and the parameters of the associated XR traffic.

A. CHANNEL MODEL

The mmWave channel for each gNB-UE link is modeled according to the broadband statistical spatial channel model (SSCM) [29] developed by NYU and used in NYUSIM. A spatial consistency procedure developed by NYU is also implemented to provide spatially correlated line-of-sight/non-line-of-sight probabilities [30]. It has been demonstrated in [31], [32] that mmWave networks tend to be noise-limited rather than interference-limited in dense deployments of mmWave networks due to highly directional beamforming and sensitivity to blockages. Therefore, we also adopt this assumption in this work and the signal-to-noise-ratio (SNR) is considered instead of the signal-to-interference-plus-noise-ratio (SINR) at the receiver.

1) PATH LOSS MODEL

We use the close-in free space reference distance (CI) path loss model with a 1 m reference distance and an extra attenuation term due to various atmospheric conditions [29]. The path loss (in dB) is given by:

$$PL(f, d) = FSPL(f, 1m) + 10n \log_{10}(d) + AT + \chi_\sigma \quad (2)$$

where d is the 3-D transmitter-receiver separation distance in meters, and n is the path loss exponent ($n = 2$ for free space). χ_σ is the shadow fading (SF) modeled as a log-normal random variable with zero mean and σ standard deviation in dB. AT is a total atmospheric absorption term, which depends on the carrier frequency. FSPL (f,1m) is the free space path loss in dB at a transmitter-receiver separation distance of 1m at the carrier frequency f in GHz:

$$FSPL(f, 1m) = 20 \log_{10} \frac{4\pi f \times 10^9}{c} \quad (3)$$

$$= 32.4[dB] + 20 \log_{10} f \quad (4)$$

The path loss exponent (PLE) and shadow fading standard deviation values for Urban Micro-cellular (UMi) scenario are displayed in Table 2 [33].

TABLE 2. PLEs and shadow fading standard deviations for UMi scenario.

Scenario	PLE	Shadow Fading Std Dev (dB)
LOS	2	4.0
NLOS	3.2	7.0

2) SPATIAL CONSISTENCY PROCEDURE

The close-in free space reference distance (CI) path loss model with a 1 m reference distance used in NYUSIM is a drop-based channel model. In a drop, the drop-based channel model generates a static and independent channel impulse response (CIR) at a particular transmitter-receiver separation distance. However, there is no correlation between different drops. The shortcoming of a drop-based channel model is that it generates independent channel coefficients for different distances, even if these points are close to each other. To realize spatial consistency while calculating path loss, spatially-correlated line-of-sight/non-line-of-sight conditions are generated [30]. A 2-dimensional (2-D) grid map is generated to contain values of spatially correlated line-of-sight/non-line-of-sight condition in a simulated area. The granularity of the map is set to be 1 m, which means the distance between two neighboring grid points is 1 m. The map of line-of-sight/non-line-of-sight condition is initialized by assigning an independent and identically distributed Gaussian random variable at each grid point. A 2-D exponential filter is applied to the map, which is given by:

$$h(p, q) = \exp\left(-\frac{\sqrt{p^2 + q^2}}{d_{co}}\right) \quad (5)$$

where p and q are coordinates with respect to the center of the filter and d_{co} is the correlation distance, i.e., the distance over which the large scale parameters are assumed to be spatially correlated. Applying this 2-D filtering, the correlated values in the map are calculated by:

$$M_c(i, j) = \sum_p \sum_q h(p, q) M(i - p + 1, j - q + 1) \quad (6)$$

where M_c is the correlated map and M is the initialized independent map. The correlation distance in a Urban Micro-cellular line-of-sight scenario is set to be 15 m. A transformation from Gaussian distribution to uniform distribution is required to generate spatially correlated uniform

random variables, which is given by:

$$\tilde{u} = g^{-1}(\tilde{v}) = F_{\tilde{u}}^{-1}(F_{\tilde{v}}(\tilde{v})) \quad (7)$$

where $\tilde{u}$ and $\tilde{v}$ are the spatially correlated uniform and Gaussian random variables, respectively, and $F_{\tilde{u}}$ and $F_{\tilde{v}}$ are the cumulative density functions of the uniform distribution and Gaussian distribution, respectively.

The method of deciding the line-of-sight/non-line-of-sight condition at a certain location is to compare a uniformly distributed random variable to line-of-sight probability at that location. There are several line-of-sight probability models available in the literature - we use the NYU squared model [34] for Urban Micro-cellular scenario, which is given by:

$$Pr_{LOS}(d) = \left(\min\left(\frac{d_1}{d}, 1\right)\left(1 - e^{\frac{-d}{d_2}}\right)\right) + \left(e^{-\frac{d}{d_2}}\right)^2, \quad (8)$$

where $d_1 = 22m$ and $d_2 = 100m$. Thus, the line-of-sight or non-line-of-sight condition of a UE at a certain location is determined by comparing correlated value $\tilde{u}$ to the line-of-sight probability $Pr_{LOS}(d)$:

$$\text{Condition} = \begin{cases} \text{line-of-sight} & \text{if } \tilde{u} \leq Pr_{LOS}(d) \\ \text{non-line-of-sight} & \text{if } \tilde{u} > Pr_{LOS}(d) \end{cases} \quad (9)$$

Note that we don't give priority to line-of-sight links over non-line-of-sight links when selecting gNBs for $\mathcal{K}_{n,t}$. Instead, gNBs for $\mathcal{K}_{n,t}$ are chosen based solely on the SNR.

By generating a map of spatially correlated line-of-sight/non-line-of-sight conditions, similar shadow fading values are observed at closely spaced locations, which is a more accurate representation of reality than independent values for close locations used in the drop-based model.

3) CHANNEL CAPACITY EVALUATION

Once the effective path losses are determined between all UE-gNB pairs, we can compute the received power, and hence the average SNR at each gNB:

$$P_R[dB] = P_T[dB] - PL[dB] + G_T[dB] + G_R[dB] \quad (10)$$

$$SNR[dB] = P_R[dB] - P_N[dB] \quad (11)$$

where P_R is the received power, P_T is the transmitted power, G_T and G_R are the transmitter and receiver gains, respectively, and P_N is the noise power.

In an actual cellular system, the achieved rate will depend on the average SNR through a number of factors including the channel code performance, channel quality indicator (CQI) reporting, rate adaptation and Hybrid automatic repeat request (HARQ) protocol. However, we abstract this process and assume a simplified, but widely-used, model [35], where the spectral efficiency is assumed to be given by the Shannon capacity with some loss δ:

$$\rho = \log_2\left(1 + 10^{0.1(SNR-\delta)}\right), \quad (12)$$

where ρ is the spectral efficiency in bps/Hz, and the SNR and loss factor δ are in dB. The spectral efficiency gives us the available capacity for each UE-gNB link, and the scheduling agent uses this information when deciding on scheduling different UEs.

B. DYNAMIC BLOCKAGE MODEL

Dynamic blockages in mmWave cellular networks are extensively studied in [36], [37] assuming a homogeneous Poison Point Process (PPP) with dynamic blocker density λ_B in the disc $B(o, R)$. The blocker arrival rate, or blockage rate, α_i at the i^{th} gNB-UE link is considered Poisson and was derived in [36], [37] as

$$\alpha_i = \Theta r_i, i = 1, 2, \ldots, m, \quad (13)$$

where r_i is the 2D distance, ignoring height, between the i^{th} gNB-UE pair.

Θ is proportional to the blocker density γ_B and is given by

$$\Theta = \frac{2}{\pi}\gamma_B V \frac{h_B - h_R}{h_T - h_R}, \quad (14)$$

where V is the speed of the blocker and h_B, h_T and h_R are the heights of the blocker, the transmitter, and the receiver, respectively.

We model the blocker arrival process as Poisson with parameter α_i blockers/sec. Note that there can be more than one blocker simultaneously blocking the link - if a second blocker arrives while the first blocker is still blocking the link, the blockage duration is extended. Furthermore, we assume the blockage duration of a single blocker is exponentially distributed with parameter μ. The blocking event of a gNB-UE link follows an on-off process with α_i and μ as blocking and unblocking rates, respectively. In the event of a blockage, the Received Signal Strength Indicator (RSSI) of the gNB-UE link is zero, and hence the corresponding channel capacity is also zero. When there is no blockage, the NYUSIM channel model described earlier is used to calculate the path loss and, hence, the channel capacity.

C. TRAFFIC MODEL

The traffic model we assume for this study is based on the 3GPP XR (Extended Reality) traffic models proposed in [3]. Specifically, we use a generic single-stream down-link model that can be used for VR, AR, and CG applications.

The downlink traffic is modeled as a sequence of video frames arriving periodically at the gNB according to a specified video frame rate. Random jitter, which follows a truncated Gaussian distribution, is super-imposed on the periodic arrivals to get the actual arrival time of the frames at the gNB. The size of each frame is also random according to a truncated Gaussian distribution.

Each traffic flow of a UE is assigned a specific traffic type: VR, AR, or CG. The traffic type of the flow determines the underlying parameters for the distributions governing the frame size, jitter, and packet delay budget of the flows. Each flow consists of a sequence of frames, and each frame is further broken up into IP packets of 1500 bytes for delivery.

TABLE 3. Statistical parameters for frame size.

Parameter	Unit	Baseline Values
Mean: M	byte	$(R \times 10^6)/F/8$
STD	byte	10.5% of M
Max	byte	150% of M
Min	byte	50% of M

IP packets belonging to the same frame have the same delay budget and arrive at the gNB simultaneously.

Each UE has a separate buffer at the gNB, so traffic from different UEs do not share a buffer. This means a UE flow cannot experience head-of-line (HOL) blocking from another UE's flow.

1) FRAME SIZE

Given R, the data rate of the flow in Mbps, and F, the frame generation rate of the flow in frames per second (fps), the frame size is modeled as a random variable following a truncated Gaussian distribution with the statistical parameters given in Table 3 [3].

2) FRAME ARRIVAL

The frame arrival rate is determined by the frame rate, F, which is given in frames per second. Hence, the inter-arrival time for the frames is given by the inverse of the frame rate. The periodic frame arrivals implicitly assume fixed delay contributed by the network. However, in a real system, the varying processing and transit delays introduce jitter in frame arrival times at the gNB. In this model, the jitter is modeled as a random variable that is added on top of the periodic arrivals. Thus, the jitter follows a truncated Gaussian distribution with zero mean, 2 ms standard deviation, and a truncation range of $[-4, 4]$ ms [3].

The given parameter values and frame generation rates ensure that the frame arrivals are always in order, i.e., the arrival time of the next frame is always later than that of the previous frame. The periodic arrival with jitter, therefore, gives the arrival time for the frame with index $k(= 1, 2, 3, \ldots,)$ as:

$$T\big[k|\text{with jitter}\big] = \frac{k \times 1000}{F} + J \text{ ms}, \tag{15}$$

where J is a truncated Gaussian random variable capturing the jitter. Note that the actual arrival times of traffic for each UE could be shifted by a UE-specific arbitrary offset.

3) FRAME DELAY BUDGET (FDB)

The latency requirement of XR traffic in the air interface is modeled as a limited time budget for a frame to be transmitted over the air from a gNB to a UE. The delay a frame incurs in the air interface is measured from the time that the frame arrives at the gNB to the time that it is successfully, *fully* transferred to the UE.

TABLE 4. Traffic parameters for VR, AR, and CG traffic.

Parameter	VR	AR	CG
Data Rate (Mbps)	45	45	30
Frame Rate (fps)	60	60	60
Frame Delay Budget (ms)	10	10	15

If a frame exceeds its FDB, it is considered to have *expired* and is no longer useful owing to the time-sensitive nature of XR applications. Hence, expired frames are immediately dropped and counted as a failed delivery. A partially delivered frame that expires is also considered a failure. If a frame is fully delivered within its FDB, it is said to be successfully delivered. The value of the FDB varies for different applications (see Table 4)

4) TRAFFIC TYPE PARAMETERS

XR traffic can be broadly classified into three main categories, each with its own set of parameters governing the data rate, frame rate, and FDB: VR, AR, and CG. The parameters for these various XR applications, according to 3GPP specifications [3], are specified in Table 4.

D. MOBILITY MODEL

The user mobility is modeled by a Random Waypoint model [38]. The UEs are initially dropped uniformly into an area around the gNBs. Each UE then randomly selects a destination within the grid and moves towards it with a constant velocity uniformly distributed between 0 and 3 kmph [39]. Upon reaching its destination, a UE selects a new destination.

V. PROBLEM FORMULATION

The first problem we explore is predictive data replication, i.e., selecting $\mathcal{L}_t = \{\mathcal{L}_{n,t}, \forall n\}$, where data would be preemptively replicated [40]. The problem of selecting $\mathcal{L}_t$ from $\mathcal{K}_t$ is interesting and non-trivial. It is not simply sufficient to pick the best links with respect to channel quality because other factors such as existing load at the gNBs and the UE traffic delay budgets and statistics need to be taken into account. The selection of $\mathcal{L}_t$ will directly impact our scheduler's performance because it will determine which gNBs are considered as possible servers in the scheduling decision-making process. Thus, the selection of $\mathcal{L}_t$ directly influences the scheduling decision, and hence the on-time delivery of the XR traffic.

A. PREDICTIVE DATA REPLICATION

We formulate our problem as a Markov Decision Process (MDP), which we can then use for our deep reinforcement learning algorithm. Since the selection of $\mathcal{L}_{n,t}$ is independent for each UE, we can decouple the MDP for each UE. Hence the state and action spaces are defined for a single UE n, while the reward function ensures that we optimize globally over the entire system. Let $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ be the

predictive data replication policy, where $\mathcal{S}$ is the set of states, and $\mathcal{A}$ is the set of actions. We now describe the state and action sets, and the reward function.

1) STATES

Let $X(f,n)$ be the size, in number of packets, of frame f of UE n, and $Y(f,n,t)$ be the number of packets of frame f of UE n that are successfully delivered during timeslot t. Let $D(f,n)$ denote the packet delay budget for frame f of UE n. Then, the expiry time for frame f of UE n is given by:

$$t_{expiry}(f,n) = t_{arrival}(f,n) + D(f,n), \tag{16}$$

and the time-till-expiry at time t is given by:

$$t_{time-till-expiry}(f,n,t) = t_{expiry}(f,n) - t \tag{17}$$

. Then, we can see that a frame is successfully delivered iff:

$$\sum_{t=t_{arrival}(f,n)}^{t=t_{expiry}(f,n)} Y(f,n,t) = X(f,n) \tag{18}$$

At time t, the fraction of frame f that has been successfully delivered is given by:

$$Y_{frac}(f,n) = \frac{\sum_{t'=0}^{t'=t-1} Y\big(f,n,t'\big)}{X(f,n)}. \tag{19}$$

The remaining packets of an expired frame are immediately dropped from the buffer, and the frame is not re-transmitted.

The state $s_{n,t}$ of UE n at time-slot t consists of:

- *UE index:* n
- *UE Connectivity Set:* $\mathcal{K}_{n,t}$
- *Global Traffic Information:* The fractions served of all the frames in the buffer and their time-till-expiry:

$$Y_{frac}(f,n)\ \forall f,n.$$
$$t_{time-till-expiry}(f,n,t)\ \forall f,n.$$

- *Global Channel State Information:* $\sigma_t = \{\sigma_{m,n,t}\colon \forall m \in \mathcal{K}_t, \forall m,n\}$.

2) ACTIONS

The action space spans over $\binom{\mathcal{K}_{n,t}}{L}\ \forall L \le K$ where each action is denoted by $\mathcal{L}_{n,t} \in \binom{\mathcal{K}_{n,t}}{L}$

3) REWARD

Since XR applications are primarily deadline driven, our explicit goal is to maximize the number of frames that are delivered within their delay budget or, equivalently, minimize the number of frames that expire.

The total reward accrued during time-slot t is given by:

$$r_t = \sum_{f,n}\left[\frac{Y(f,n,t)}{X(f,n)} - \omega \mathbb{1}(f,n,t)\big(1+Y_{frac}\big)\right], \tag{20}$$

where,

$$\mathbb{1}(f,n,t) = \begin{cases} 1, & \text{if frame } f \text{ of UE } n \text{ expires and has not fully finished transmission by time } t \\ 0, & \text{otherwise} \end{cases}$$

and ω is a constant weight.

The first term in (20) allocates a fractional reward for delivering packets of a frame successfully and sums it over all frames served during time-slot t. The second term in (20) exerts a cumulative penalty in case a frame expires. The cumulative penalty offsets the reward accrued for partial delivery of the expired frame and imposes a higher penalty in proportion to the fraction of the expired frame that had already been delivered. So, for example, a frame that had been 90% delivered before it expired will exert a much higher penalty, a penalty of 1.9, than a frame that had been 10% delivered before it expired, which would only incur a penalty of 1.1. Moreover, we assign a constant weight ω to the penalty to promote faster convergence to a policy that is averse to expiring frames.

Our optimization problem can then be represented as an infinite-horizon decision problem:

$$\arg\max_{\pi} \mathbb{E}\left(r_0 + \sum_{t=1}^{\infty} \gamma^t r_t\right), \tag{21}$$

where future rewards are discounted by a discount factor γ, $0 < \gamma < 1$, and r_t is the reward function given by Eq. (20).

We use Deep Q-Learning, an elegant solution for solving complex MDPs, which uses a Deep Q-Network (DQN) to estimate the Q-function for the MDP [41]. Since a DQN outputs the Q-value corresponding to each action, we can simplify our action space by training our DQN for $L = 1$, so that the DQN just learns to give a *goodness* metric for selecting a gNB to be in $\mathcal{L}_t$. Then, when the DQN is deployed after training, we can use it to scale up to larger values of L by selecting the L gNBs corresponding to the L highest Q-values. Moreover, we make use of techniques such as experience replay and target networks to promote faster convergence. Algorithm 1 describes the Deep Q-learning algorithm for the predictor.

State space explosion is a real issue in a problem of this size and magnitude, making it difficult to train any deep learning network. With 11 gNBs and 35 UEs, we get ~ 600 state variables for the Predictor described earlier. Assuming a lower bound of just 2 unique values for each state variable, it still leads to a state space of size 2^{600}. To reduce the state space to a more manageable size we use an auto-encoder. The architectures of the DQN and Autoencoder, as well as hyper-parameters and training processes, are explained in detail in Section VI. For each UE n, the current state is observed and input to the Autoencoder which encodes it to a smaller code size. The encoded state is passed on to the Data Replication Predictor DQN which outputs $\mathcal{L}_{n,t}$ for UE n. The scheduler uses this information ($\mathcal{L}_{n,t}\forall n$), along with the network state, to take a scheduling action for the

Algorithm 1 Deep Q-Learning for Predictive Data Replication

Initialize Experience Replay Memory D
Initialize DQN with random weights
Initialize encoded state $s_{n,t}, \forall n$
for $t = 1 : T$ **do**
  With probability ϵ, select random $\mathcal{L}_{n,t}, \forall n$
  Otherwise select $\mathcal{L}_{n,t} = \max_{a_{n,t}} Q(s_{n,t}, a_{n,t}), \forall n$
  Execute action $a_{n,t} = \mathcal{L}_{n,t}, \forall n$, observe global reward r_t and state $s_{n,t+1} \forall n$
  Store experience $(s_{n,t}, a_{n,t}, r_t, s_{n,t+1})$ in D for a random UE n
  Set $s_{n,t+1} = s_{n,t} \forall n$
  Sample random minibatch of experiences from D
  Perform a gradient descent step
end for

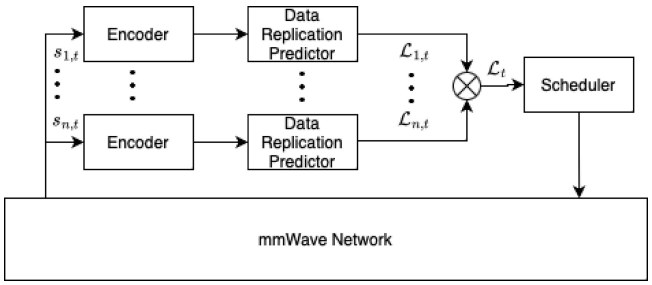


FIGURE 2. System Diagram.

current time-slot which is executed in the mmWave network. Fig. 2 shows the system diagram.

It should also be noted that while training a neural network is computationally expensive and takes considerable time, deploying a trained neural network is quite feasible in real-world scenarios. Network state observations can be collected by the gNBs over a period of time, for example a day, and can then be sent back to a central location for offline training. Once trained, the parameters of the neural network can be sent to the gNBs for real-time implementation of the trained neural network. Moreover, since the network infrastructure is static and traffic patterns can be reliably predicted for different times of the day, it makes sense to harness deep learning to *learn* the local information and make decisions accordingly. For practical deployments, several different models can be trained according to different degrees of traffic conditions. Telecom operators already log extensive traffic profiles which they use to, for example, selectively switch off some components in the gNB during low traffic hours to save power, and devote more resources during peak traffic hours. Similarly, gNBs are deployed after extensive site analysis to determine the ideal placement for a given location. Therefore, it actually is very practical to deploy Deep Learning solutions in wireless networks: the topology of the local area is known and unchanging, the traffic patterns can be analyzed and divided into different categories, and the placements of gNBs in the given area are also known. Moreover, fine-tuning schemes can be implemented on top of the basic deep learning framework, which tunes the trained model in real-time according to current conditions.

B. SCHEDULER

Since XR applications are constrained by strict deadlines, we are explicitly interested in deadline-driven scheduling, and not throughput or rate maximization as is more commonly done. For the single server case, the Earliest Deadline First (EDF) policy has been proven to be optimal in wireline networks [42]. However, the same result cannot be directly extended to wireless networks due to uncertain channel conditions of the wireless links. Shakkottai [43] extended this to wireless networks where users are served by a single server, and showed that a Feasible Earliest Deadline First (FEDD) policy is optimal where the EDF policy is implemented only over channels which are in a *good* state. Maximum weighted link scheduling for wireless networks has also been well studied for the problem of weighted sum-rate maximization [44], [45], [46]. Note that, for networks with fixed link capacities, the maximum weighted link scheduling problem reduces to the classical maximum weighted matching problem and can be solved in polynomial time. However, no solution is known for the general case when the link rates depend on the power allocation of all other links. Moreover, the results do not extend to deadline-driven scheduling in multi-connectivity enabled wireless networks.

In our multi-connectivity enabled mmWave network, the scheduling agent has to find, given $\mathcal{L}_t$, a feasible scheduling policy $\mathcal{P}$ such that the number of expired frames in the network are minimized over an infinite time horizon. We propose a Maximal Weight Matching policy, which we dub β-MWM, that aims to explicitly take into account the strict deadlines of the XR traffic, while also considering channel quality and connectivity. The wireless network forms a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with bi-partition $(\mathcal{M}, \mathcal{N})$ and the mmWave links forming the edges of the graph. Note that without loss of generality, we may assume that $\mathcal{G}$ is a complete weighted bipartite graph (we may add edges of zero weight as necessary); we may also assume that $\mathcal{G}$ is balanced as we can add dummy vertices as necessary. The problem can then be expressed as the following maximization problem:

$$\begin{aligned} \max \quad & \sum_{(m,n)} w(m,n)x(m,n), \\ \text{subject to:} \quad & \sum_{n} x(m,n) = 1 \quad \forall m \in \mathcal{M}, \\ & \sum_{m} x(m,n) = 1 \quad \forall n \in \mathcal{N}, \\ & x(m,n) \in \{0,1\} \quad \forall m \in \mathcal{M}, n \in \mathcal{N}. \end{aligned} \tag{22}$$

The classical solution to this maximum weighted bipartite matching problem in Eq. (22) is given by the Hungarian or Kuhn-Munkres algorithm [47], [48].

The weight function, $w : \mathcal{E} \rightarrow \mathcal{R}$, is of utmost importance for Maximal Weight Matching because it determines

which edges (links) in the graph (network) are chosen to be activated. Since XR traffic has strict deadline constraints, our aim is to do deadline driven scheduling. However, for wireless networks, and especially for mmWave networks which are prone to blockages, channel capacity and connectivity status also needs to be taken into account for a good scheduling decision. We design two weight functions, and compare the performance of the $\beta-$MWM scheduler for both functions.

The first weight function, $w_1(m,n)$ is given by:

$$w_1(m,n) = \begin{cases} \beta\left(\frac{1}{d_n}\right) + (1-\beta)\left(\frac{C_{m,n}}{L_{eff,n,t}}\right) & \text{if } m \in \mathcal{L}_{n,t}, \\ 0 & \text{if } m \notin \mathcal{L}_{n,t}, \end{cases} \quad (23)$$

where d_n is the deadline of the head of line (HOL) frame of UE n, $C_{m,n}$ is the current capacity of the mmWave link between gNB m and UE n, C_{max} is the maximum achievable link capacity, and $L_{eff,n,t} = |\mathcal{L}_{n,t}| \leq L$. The first term in Eq. (23) makes the weight inversely proportional to the strict deadline of the HOL frame, thereby giving a larger weightage to UEs with earlier deadlines. The second term in Eq. (23) makes the weight directly proportional to the channel capacity of the link, while also giving higher weightage to UEs with low connectivity, i.e., where $|\mathcal{L}_{n,t}| < L$. Moreover, $\beta \in (0,1)$ is a scaling parameter that marks the relative significance of the traffic deadlines and channel quality. For $\beta > 0.5$, a higher emphasis is placed on delivering UEs with earlier deadlines. At $\beta = 1$, the policy becomes equivalent to the Earliest Deadline First (EDF) policy. For $\beta < 0.5$, more weight is given to channel quality and fairness for UEs with low connectivity.

The second weight function, $w_2(m,n)$ is given by:

$$w_2(m,n) = \begin{cases} \frac{1}{L_{eff,n,t}}\left(\frac{C_{m,n}^{(1-\beta)}}{d_n^{\beta}}\right) & \text{if } m \in \mathcal{L}_{n,t}, \\ 0 & \text{if } m \notin \mathcal{L}_{n,t}, \end{cases} \quad (24)$$

Eq. (24) implies that the utility for delivering $C_{m,n}^{(1-\beta)}$ packets on time is $(L_{eff,n,t} d_n^{\beta})^{-1}$, with β once again determining the trade-off between prioritizing deadlines or channel capacities.

VI. SIMULATION RESULTS AND DISCUSSION

We do comprehensive performance evaluation by simulating the mmWave network using Python. 11 gNBs are deployed in a hexagonal grid with an inter-site distance of 100 m and 35 UEs are dropped randomly into the area. We use a connectivity threshold of 300 m, i.e., if a UE is within 300 m of a gNB and not blocked, the gNB is considered to be a *candidate* gNB. The gNB density is sufficiently high, such that in case of blockages, a UE always has other candidate gNBs to switch to. An outage is defined as an event when all gNBs in $\mathcal{L}_{n,t}$ are concurrently blocked - this will lead to an interruption of the data plane while the UE initiates a switch to other available gNBs. In order to mimic a system that is not capacity-limited, we use a per-gNB bandwidth of 400 MHz. Additionally, the system operates in discrete time slots of $125\mu s$, which is equivalent to an OFDM slot that can be used for transmitting downlink or uplink data [49]. Traffic arrivals, scheduling decisions, and blockages operate at this granularity. However, channel state updates are done at a larger time scale, once every second, because the path-loss is only affected by large-scale shadow fading, a change in which occurs on the order of seconds [5]. We simulate downlink XR traffic for the UEs and evaluate the performance for varying degrees of association (K), degrees of data replication (L), dynamic blocker densities (γ_B), and handover delays (Δ_K and Δ_L). Since XR traffic requires low latency and expires after a strict deadline, we use the percentage of frames delivered within the deadline as our primary performance metric. This captures the system performance better than other metrics such as average throughput because it explicitly takes into account only the successful traffic which was delivered within the deadline. We perform our simulation over a mobility period of 15 minutes. To account for the randomness in the experiments, each experiment configuration is run two hundred times and the results are averaged. The rest of the simulation parameters are presented in Table 5.

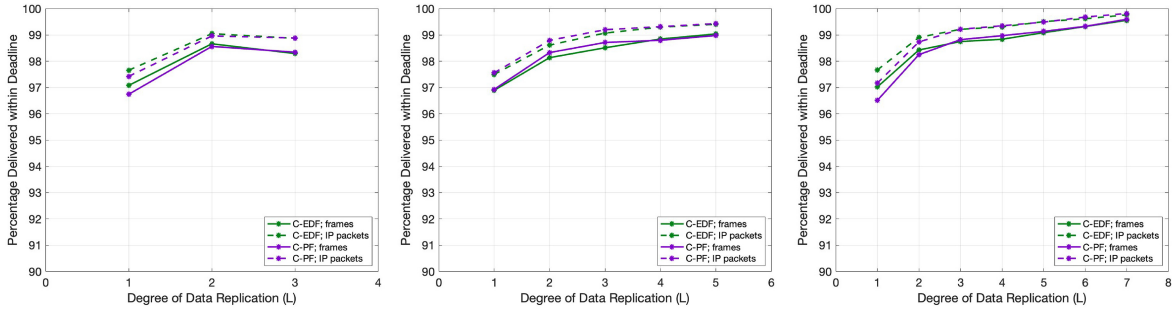
TABLE 5. Simulation parameters.

Parameters	Values
Carrier Frequency, f	73 GHz
Max Spectral Efficiency, ρ_{max}	4.8 bps/Hz [35]
Transmitter Antenna Gain, G_T	10 dBi
Receiver Antenna Gain, G_R	10 dBi
Transmit Power, P_T	24 dBm
Loss Factor, δ	3 dB
Velocity of Dynamic Blockers, V	1 m/s
Height of Dynamic Blockers,h_B	1.8 m
Height of UE, h_R	1.4 m
Height of gNB, h_T	5 m
Expected Blockage Duration	500 ms [50]

For the selection of $\mathcal{L}_{n,t}$ from $\mathcal{K}_{n,t}$ for each UE n, the baseline algorithm we use is the Best Channel Quality Indicator (BEST-CQI) algorithm where $\mathcal{L}_{n,t}$ is selected based on channel quality alone. BEST-CQI is an algorithm that selects $\mathcal{L}_{n,t}$ by starting with an ordered set of SNRs and picking the gNBs corresponding to the L highest SNR values.

A. PERFORMANCE EVALUATION OF MULTI-CONNECTIVITY ARCHITECTURE

We first evaluate the feasibility of our multi-connectivity architecture and prove that it does indeed provide benefits in terms of shielding the UEs from adverse effects of blockages and minimizing data plane interruptions. In



(a) $K = 3$ (b) $K = 5$ (c) $K = 7$

FIGURE 3. Effect of the degree of association (K) and the degree of data replication (L) on the percentage of frames and IP packets successfully delivered within deadline, with $L \leq K$, $\Delta_K = 20$ ms, $\Delta_L = 10$ ms and blocker density $\gamma_B = 0.01$ bl/m^2.

a multi-connectivity setting, it is not sufficient to just select UEs for scheduling based on some priority value. Once a UE is selected, another selection decision needs to be made to match it to a gNB because multiple gNBs are available to each UE for data transmission. A centralized scheduler would enhance the system performance, though at the cost of much higher overhead in terms of information exchange and delays in relaying the control decision. For the purpose of our simulation, we assume an omniscient, centralized scheduler that is able to operate with zero delays.

We compare the performance of two centralized schedulers:

- Centralized Earliest Deadline First (C-EDF): The UE which has the HOL frame with the earliest deadline in the network is matched to the best available Data Replication gNB.
- Centralized Proportional Fair (C-PF): The UE priority function is given by [51]:

$$P = \frac{T}{R},$$

where T is the current channel capacity of the UE-gNB link, and R is the historical average data rate of the UE. The UE with the highest priority is matched to the best available Data Replication gNB.

1) EFFECT OF DEGREE OF ASSOCIATION (K) AND REPLICATION (L)

Fig. 3 shows how the percentage of frames and IP packets delivered successfully within their deadline varies with the degree of data replication (L), for different values of the degree of association (K). First, note that the percentage of IP packets delivered within the deadline is always more than the frames delivered within the deadline. Frame delivery is only counted as successful if the *entire* frame is delivered successfully within the deadline. This shows why the percentage of frames delivered within the deadline is a better QoS metric for deadline-driven XR applications because it only counts the useful throughput. Next, from Fig. 3 we observe that there is a huge spike in performance when we go from single connectivity ($L = 1$) to dual connectivity

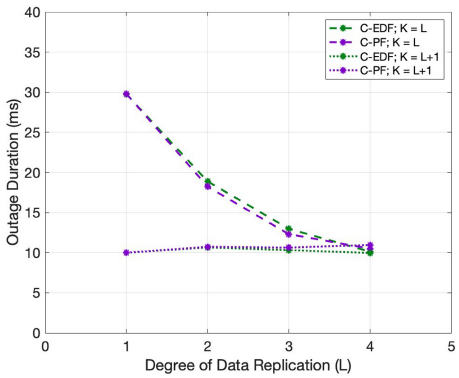


FIGURE 4. Effect of the relative values of K and L on the average outage duration, with $\gamma_B = 0.05$ bl/m^2, $\Delta_K = 20$ ms and $\Delta_L = 10$ ms.

($L = 2$). The availability of an extra gNB in dual connectivity ensures that the scheduler has a backup to fall back on in case of sudden service disruption due to blockages. As we further increase the degree of data replication from $L = 2$ to $L = 5$, we see diminishing returns in terms of performance improvement. This is due to the fact that the extra backup gNBs only become useful when there are several concurrent blockages. For example, when $L = 4$, the fourth gNB will only be useful in the scenario when the first three gNBs are concurrently blocked. Since the outage probability decreases exponentially with the number of gNBs, as shown in Fig. 5, we see corresponding diminishing returns as L increases.

From Fig. 3, we note that with $K = 3$, there is a dip in performance going from $L = 2$ to $L = 3$. However, this is well within the confidence intervals ($\pm 0.23\%$) and the broader trend of performance increasing with multi-connectivity remains true. In fact, from Fig. 3(c) we can see that we boost performance from 96.5% when $L = 1$ to 99% when $L = 5$. This is a significant improvement in performance given the fact that one of the main QoS criteria for XR applications is to deliver 99% of a UE's traffic within the deadline [3]. Moreover, we see that the prime benefit of increasing K is that it allows us to potentially replicate the data at a larger number of gNBs, since $L \leq K$. However, if we fix L, there is no benefit to be gained in further increasing K beyond $K = L + 1$. For example, with

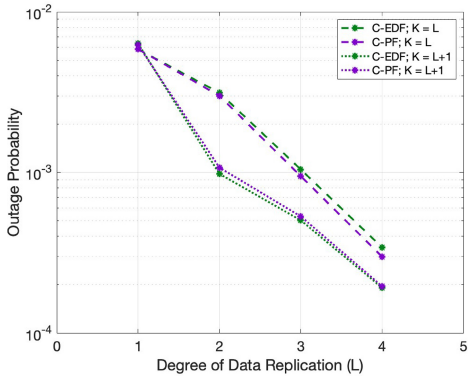


FIGURE 5. Effect of the relative values of K and L on the outage probability, with $\gamma_B = 0.05$ bl/m^2, $\Delta_K = 20$ ms and $\Delta_L = 10$ ms.

$L = 2$, we see a similar performance, disregarding the minor variations which are within the confidence intervals, as K is increased from 3 to 7.

We now turn our attention towards a discussion and comparison of the performance of our two schedulers: C-EDF and C-PF. The decision to use a centralized scheduler is a deliberate one and stems from our multi-connectivity architecture where the *selection* of a Serving gNB plays a critical role in the subsequent scheduling decision and system performance. Hence, the scheduling problem is fundamentally different from single-connectivity scenarios, where the only decision that needs to be made is the scheduling decision. Thus, it is imperative that the *selection* and *scheduling* decisions be made jointly in order to gain better performance. Even so, neither C-EDF nor C-PF is optimal. Simple examples can be crafted that show both schedulers taking sub-optimal decisions.

Moreover, we acknowledge that our schedulers operate under ideal assumptions that will not hold in real-world scenarios, namely the availability of instantaneous channel state and traffic information at the scheduler and the instantaneous relaying and execution of the scheduling decision at the gNBs. However, our results can be used to gauge the performance of schedulers that better emulate real-world conditions and operate in a distributed manner.

From Figs. 3–7, we see that both C-EDF and C-PF have similar performance, with C-PF performing better at higher blocker densities. C-PF performs well because it jointly optimizes over the UE's historical data rate and the available channel capacities; however, its drawback is that it does not explicitly take into account the traffic deadlines nor does it attempt to do delay-aware scheduling. On the other hand, C-EDF attempts delay-aware scheduling but does not take a joint gNB selection and scheduling decision; instead, it does scheduling and selection sequentially which is sub-optimal. Hence, we can see that there is a need for new scheduling algorithms that are optimized for the multi-connectivity paradigm, i.e., which do deadline-driven scheduling in conjunction with gNB selection. We address this problem with our β-MWM scheduler, whose performance is evaluated in Section VI-C.

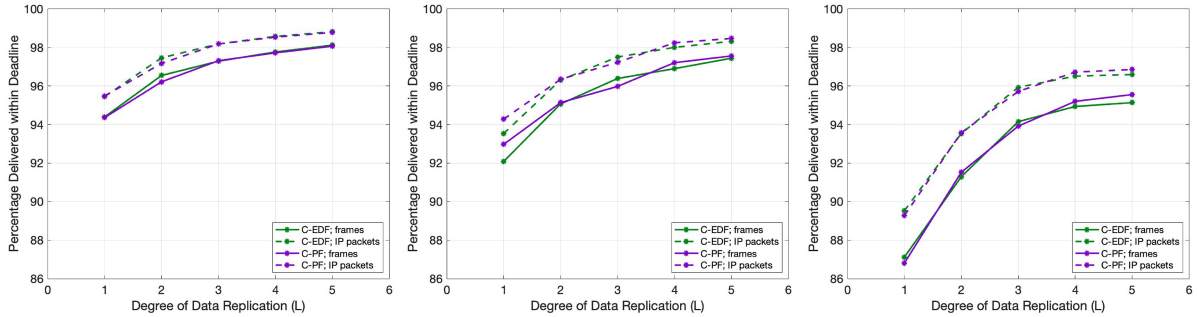
Next, we illustrate how our architecture minimizes data plane interruptions. We are interested in the average outage duration, which is the amount of time it takes a UE to recover from an outage event by resuming the data plane connection with another gNB. At 60 fps, the average frame inter-arrival time is 17 ms, so depending on the link capacity available after the interruption, at most one frame is dropped when $\Delta_K = 20$ ms and $\Delta_L = 10$ ms. From Fig. 4, we note that when $K = L$, which is the case when the Association and Data Replication tiers are collapsed into one, i.e., data is replicated at all the associated gNBs, the average outage duration is upper-bounded by $(\Delta_K + \Delta_L)$ ms. However, the power of our multi-tier architecture is displayed when $K > L$. Consider the simplest case, when $K = L + 1$. With one extra gNB in $\mathcal{K}_{n,t}$, the average outage duration falls to approximately Δ_L ms. Fig. 4 also shows that this benefit does not increase with L because the response time to the outage is determined by whether an extra gNB is available in $\mathcal{K}_{n,t}$ when all gNBs in $\mathcal{L}_{n,t}$ get blocked. However, from Fig. 5, we observe that increasing L decreases the outage *probability*. Thus, from Figs. 4 and 5 we can conclude that for the same value of L, $K = L+1$ gives better performance than $K = L$, if this option is available.

2) EFFECT OF DYNAMIC BLOCKER DENSITY γ_B

Fig. 6 illustrates the effect of dynamic blocker density (γ_B) on the percentage of frames and IP packets delivered within the deadline. We observe that a higher blocker density results in a significant loss of performance, especially at low levels of multi-connectivity. Moreover, as the blocker density is increased the boost in performance from a higher degree of data replication also increases. This is because a higher blocker density results in more frequent blockages, which is reflected in a higher out-of-service probability. Consequently, the benefit to be gained by having backup gNBs also increases as the density of blockers is increased.

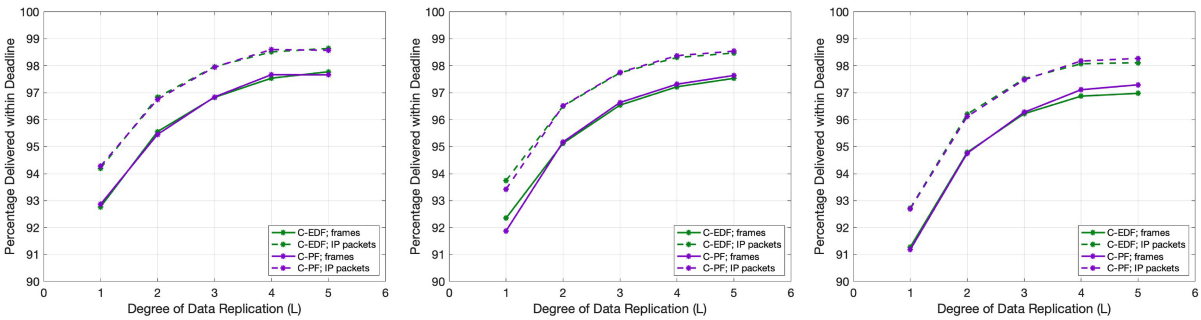
3) EFFECT OF HANDOVER DELAYS, Δ_K AND Δ_L

Handover Delays, Δ_K and Δ_L, are vital for performance evaluation because they affect the response time to blockages and determine the duration of data plane interruptions. Recall that, given the gNB density is high enough to ensure that there are always candidate gNBs available, a UE experiences a maximum data plane interruption of Δ_L ms if all the gNBs in $\mathcal{L}_{n,t}$ are blocked concurrently, and a maximum data plane interruption of $(\Delta_K + \Delta_L)$ ms if all the gNBs in $\mathcal{K}_{n,t}$ are blocked concurrently. From Fig. 7, we see that the system performance decreases as Δ_K and Δ_L are increased, which is due to the higher out-of-service duration as a result of higher handover delays. However, this decrease in performance is less at higher values of L, because the out-of-service probability decreases exponentially as L is increased. For example, from Fig. 7(a) and 7(c) we observe that at $L = 1$, performance decreases from 92.8% to 91.2% - a decrease of 1.6% - when handover delays increase.



(a) $\gamma_B = 0.03$ bl/m^2 (b) $\gamma_B = 0.05$ bl/m^2 (c) $\gamma_B = 0.1$ bl/m^2

FIGURE 6. Effect of the dynamic blocker density (γ_B) on the percentage of frames and IP packets successfully delivered within deadline, with $K = 5$, $\Delta_K = 20$ ms and $\Delta_L = 10$ ms.



(a) $\Delta_K = 20$ ms, $\Delta_L = 10$ ms (b) $\Delta_K = 50$ ms, $\Delta_L = 30$ ms (c) $\Delta_K = 80$ ms , $\Delta_L = 50$ ms

FIGURE 7. Effect of Association Handover Delay, Δ_K, and Data Replication Handover Delay, Δ_L, on the percentage of frames and IP packets successfully delivered within deadline, with $K = 5$ and $\gamma_B = 0.05$ bl/m^2.

However, at $L = 2$, the performance decreases from 95.5% to 94.8% - a smaller decrease of 0.7%.

B. DQN DATA REPLICATION PREDICTOR

We now move on to the implementation and evaluation of the Data Replication Predictor DQN. For this evaluation, we only use the Centralized Proportional Fair (C-PF) scheduler described earlier, and compare the performance of the Data Replication Predictor DQN with the following algorithms:

- Best Channel Quality Indicator (BEST-CQI): L gNBs with the best channel quality, based on most recent measurements, are selected to be in $\mathcal{L}_{n,t}$.
- Nearest Neighbor (NN): L gNBs nearest to the UE, based on its current position, are selected to be in $\mathcal{L}_{n,t}$.
- Trajectory Estimate Replication (TER): Assuming that the UE's current position and past mobility is known, the trajectory of the UE for time window T is estimated. gNBs closest to the estimated future position are selected to be in $\mathcal{L}_{n,t}$.

  Since we use a low-speed mobility model, modeling pedestrian traffic, we use a larger time window for trajectory estimation. For high-speed mobility, such as for vehicular traffic, the trajectory estimation window can be reduced. Trajectory estimation based association and replication is also a good method because it pre-emptively removes a gNB from $\mathcal{L}_{n,t}$ when the UE is about to leave its coverage area and adds a gNB to $\mathcal{L}_{n,t}$ whose coverage area the UE is about to enter.

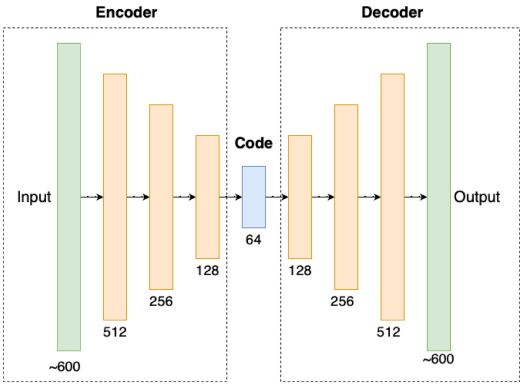
We use a blocker density (γ_B) of 0.01 bl/m^2, and Δ_K and Δ_L of 20 ms and 10 ms, respectively. Other system parameters remain the same.

1) TRAINING ENCODER

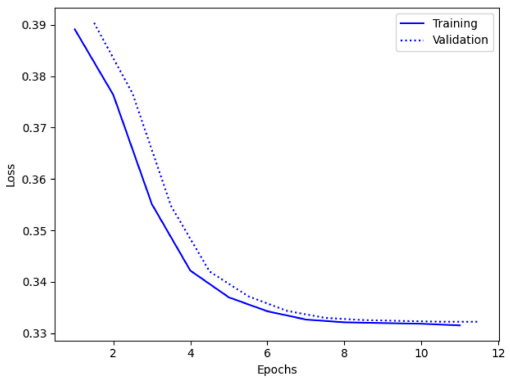
The autoencoder comprises three fully connected hidden layers of size 512, 256, and 128 respectively, followed by a code layer of size 64, as shown in Fig. 8(a). Thus our state is encoded into a 64-dimensional vector, which represents an order-of-magnitude decrease in the number of dimensions. To train the autoencoder, a dataset comprising 2 million UE states is generated. The autoencoder is implemented using Keras, with the SGD optimizer, MAE loss function, ReLU activation function, batch size of 32, and a learning rate of $1e-4$. The loss converges in 11 epochs, as shown in Fig. 8(b). Once the autoencoder is trained, we only use the Encoder part to encode our state and use it as an input to the Predictor DQN. We compare the performance of our Predictor DQN with a BEST-CQI heuristic which simply selects $\mathcal{L}_{n,t}$ to be the set of gNBs with best channel quality to UE n at time t.

2) TRAINING PREDICTOR DQN

For the Predictor DQN, we use two fully connected hidden layers, each of size 1000, as shown in Fig. 9. Since we



(a) Architecture



(b) Training and Validation Loss

FIGURE 8. AutoEncoder.

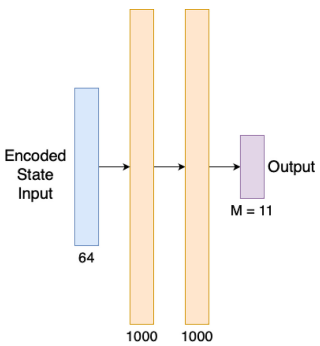


FIGURE 9. Predictor Deep Q-Network.

train the DQN for $L = 1$, the output layer is of size M, i.e., the maximum number of gNBs a UE can associate with. The DQN is trained with the SGD optimizer, Huber loss function, ReLU activation function, and a learning rate of $5e-4$. We use an experience replay memory of size $10,000$. For the DQN algorithm, we use a batch size of 32, a discount factor of 0.99, and a decaying epsilon for exploration-exploitation trade-off which decays from an initial value of 1 to a terminal value of 0.01. The DQN is trained once every 10 time slots in the network (time periods where there is no traffic in the network are automatically excluded and not counted towards this). The convergence of the DQN depends on the hyper-parameters - we empirically

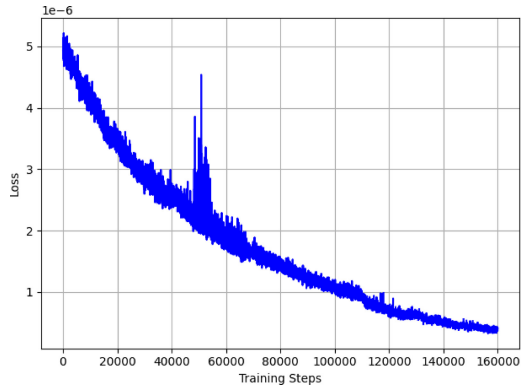


FIGURE 10. Convergence of Loss Function during training.

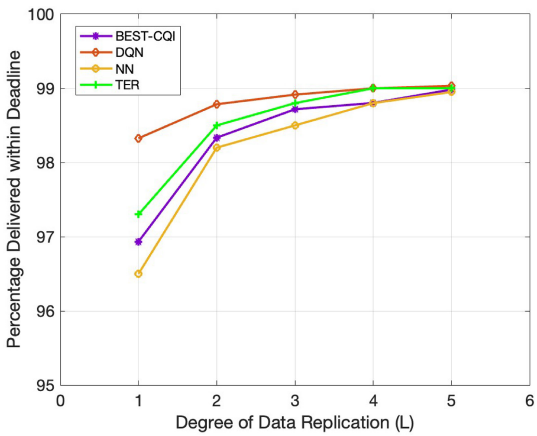
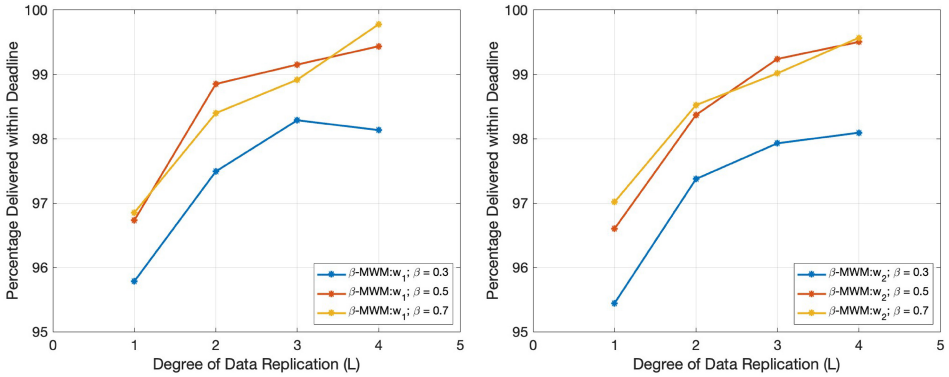


FIGURE 11. Effect of $\mathcal{L}_t$ selection algorithm on the percentage of frames and IP packets delivered within deadline, with $K = 5$.

chose the hyper-parameters which resulted in better convergence. For our final model, it took nearly 160,000 training steps for the DQN to converge. Since one training step was executed after every 10 time-slots, the system ran for a total of nearly 1.6 million time-slots. The plot of the loss function converging during training is shown in Fig. 10.

Note that the DQN is designed such that its architecture remains the same irrespective of load: the input to the DQN is an encoded state which is of a fixed size equal to the size of the code, and the output is equal to the number of gNBs in the system. However, the DQN predicts $\mathcal{L}_{n,t}$ for each UE individually, so N UEs require N executions of the DQN, where the n^{th} execution corresponds to UE n's encoded state being fed into the DQN and the output being observed to determine $\mathcal{L}_{n,t}$. Thus, with respect to space complexity, the DQN is static with respect to load - as increasing the number of UEs does not increase the number of trainable parameters in the DQN. However, with regards to time complexity, the DQN Predictor scales linearly as N executions of the DQN are required for N UEs. However, since the DQN executions are independent of each other, they can be executed in parallel, but this would require the use of multiple GPUs.

Fig. 11 shows how the percentage of frames delivered within the deadline varies with L, with $K = 5$. We see



(a) Weight Function w_1 given by Eq.23 (b) Weight Function w_2 given by Eq.24

FIGURE 12. Performance of the $\beta-$MWM scheduler, with the two weight functions given in Eqs. (23)-(24). Here, $K = 5$, $\gamma_B = 0.02$ bl/m^2, $\Delta_K = 20$ ms, $\Delta_L = 10$ ms.

a significant improvement in the performance of the DQN Predictor, especially at lower values of L. At $L = 1$, we see a 1.3% performance boost when we use the DQN - this may seem an insignificant improvement, but it is critical for reaching the 99% on-time delivery threshold set by 3GPP for XR application [3]. TER performs better than NN and BEST-CQI, but worse than the DQN Predictor, for lower values of L. This is because the DQN is able to make better decisions as it also takes other information, such as UE traffic levels and gNB loads, into account. For higher values of L, the algorithms all converge because it becomes likelier that the best gNBs are selected in the Data Replication set. Moreover, the performance improvements given by the DQN suggest that we can operate at a lower level of multi-connectivity to achieve performance similar to other heuristics. For example, the DQN is able to achieve 98.9% on-time delivery with $L = 3$, while the BEST-CQI algorithm is only able to achieve that with $L = 5$. Thus, with the DQN predictor, we incur significantly lower overhead costs (in terms of data replication) to achieve a similar level of performance.

C. β-MWM SCHEDULER

We showed earlier that the C-EDF and C-PF schedulers were not necessarily optimized for the multi-connectivity paradigm. We now proceed to illustrate performance results for our proposed scheduler, $\beta-$MWM, which was described in detail in Section V. We use a dynamic blocker density (γ_B) of 0.02 bl/m^2, $\Delta_K = 20ms$ and $\Delta_L = 10ms$. We evaluate the performance of the $\beta-$MWM scheduler for both weight functions given in Eq. (23) and Eq. (24).

Fig. 12 shows the comparison of the performance of $\beta-$MWM scheduler for the two weight functions, with varying β. Recall that $\beta \in (0, 1)$. At $\beta = 0$, the MWM scheduler ignores the traffic deadlines, and instead uses only the channel capacity as the utility function for the scheduler. At $\beta = 1$, the MWM scheduler collapses into the earliest deadline first policy, albeit one that chooses a serving gNB randomly for the UE with the earliest deadline. From Fig. 12 we see that performance is worse for lower values of β, especially at $\beta = 0.3$, irrespective of which weight function is chosen. As β is increased, system performance increases significantly but there are diminishing returns with increasing β, i.e., as β increases from 0.3 to 0.5, we see a significant boost in performance, but only a small jump when β further increases from 0.5 to 0.7. This shows that while it is important to take deadline constraints into account, other network conditions, such as channel quality and connectivity status, cannot be ignored for optimal scheduling.

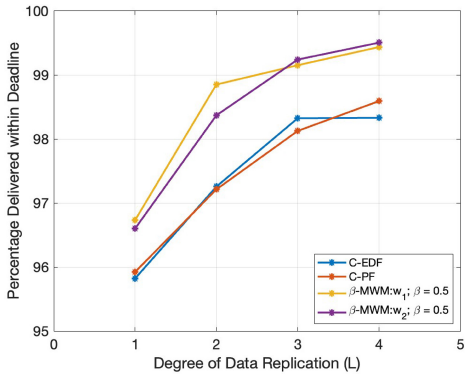


FIGURE 13. Performance Comparison of $\beta-$MWM scheduler with C-EDF and C-PF. Here, $K = 5$, $\gamma_B = 0.02$ bl/m^2, $\Delta_K = 20$ ms, $\Delta_L = 10$ ms.

Next, we compare the performance of the $\beta-$MWM scheduler with the C-EDF and C-PF schedulers. From Fig. 13, we can see that $\beta - MWM$ scheduler outperforms both C-EDF and C-PF. With the $\beta-$MWM scheduler, we are able to achieve the desired 99% successful delivery rate at $L = 3$, even with a comparatively high blocker density of 0.02 bl/m^2. Lastly, we note that both weight functions provide almost similar results, which proves that maximal weight matching, which takes some combination of deadlines, channel capacities, and connectivity status into account can deliver better results than conventional algorithms which only take one metric into account.

VII. CONCLUSION

The world is at the cusp of a new technological revolution, with XR applications poised to fundamentally change how we interact with the world around us. Given the stringent requirements of XR applications, which include strict deadlines as well as high data rates, it is necessary for existing network architectures and protocols to evolve to support these QoS constraints. In this paper, we proposed a multi-tiered multi-connectivity architecture that allows us to shield UEs from data plane interruptions and reduce the response time to blockages. Moreover, we show how existing network functions are not optimized for the multi-connectivity paradigm. To fill this gap, we leveraged Deep Reinforcement Learning to propose an intelligent Data Replication Predictor which gives the optimal selection of gNBs to replicate the data at. Furthermore, we also proposed a heuristic scheduler, $\beta-$MWM, which takes advantage of the multi-connectivity architecture to deliver better performance than conventional scheduling algorithms. However, the work in this paper focused on centralized algorithms which take advantage of global state information, a feature that could be expensive in real-world systems. Thus, future research should focus on expanding the work in this paper to a decentralized framework, where gNBs only exchange information with neighboring gNBs and imperfect, time-delayed channel measurements are available.

REFERENCES

[1] G. Minopoulos and K. E. Psannis, "Opportunities and challenges of tangible XR applications for 5G networks and beyond," *IEEE Consum. Electron. Mag.*, vol. 12, no. 6, pp. 9–19, Nov. 2023.

[2] "Service requirements for the 5G system," 3GPP, Sophia Antipolis, France, 3GPP Rep. TS 22.261, version 16.8.0, Mar. 2019.

[3] "Study on XR (extended reality) evaluations for NR," 3GPP, Sophia Antipolis, France, 3GPP Rep. TS 38.838, version 17.0.0, Dec. 2021.

[4] R. Dangi, P. Lalwani, G. Choudhary, I. You, and G. Pau, "Study and investigation on 5G technology: A systematic review," *Sensors*, vol. 22, no. 1, p. 26, 2022.

[5] T. S. Rappaport, *Wireless Communications: Principles and Practice, 2/E*. Noida, Uttar Pradesh: Pearson Edu. India, 2010.

[6] S. Sun and T. S. Rappaport, "Wideband mmWave channels: Implications for design and implementation of adaptive beam antennas," in *Proc. IEEE MTT-S Int. Microw. Symp. (IMS2014)*, 2014, pp. 1–4.

[7] D. Jiang and G. Liu, "An overview of 5G requirements," in *5G Mobile Communications*. Cham, Switzerland: Springer, 2016, pp. 3–26. [Online]. Available: https://scholar.googleusercontent.com/scholar.bib?q=info:rM-1zxC2JWkJ:scholar.google.com/&output=citation&scisdr=ClHcW1_tEMiUlogQK0A:AFWwaeYAAAAAZS8WM0A5wGD-0Qbupko-gKxvKH0&scisig=AFWwaeYAAAAAZS8WM93fBQtplLBRjNhPIlw3Rs8&scisf=4&ct=citation&cd=-1&hl=en

[8] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[9] "Overall description," 3GPP, Sophia Antipolis, France, 3GPP Rep. TS 36.300, v12.5.0, Mar. 2015.

[10] M. Özkoç, A. Koutsaftis, R. Kumar, P. Liu, and S. Panwar, "The impact of multi-connectivity and handover constraints on millimeter wave and terahertz cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 6, pp. 1833–1853, Jun. 2021.

[11] A. Koutsaftis, R. Kumar, P. Liu, and S. Panwar, "Fast inter-base station ring (FIBR): A new millimeter wave cellular network architecture," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 12, pp. 2699–2714, Dec. 2019.

[12] V. Petrov et al., "Dynamic multi-connectivity performance in ultra-dense urban mmWave deployments," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2038–2055, Sep. 2017.

[13] M. Gapeyenko et al., "On the degree of multi-connectivity in 5G millimeter-wave cellular urban deployments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1973–1978, Feb. 2019.

[14] R. Liu, G. Yu, and G. Y. Li, "User association for ultra-dense mmWave networks with multi-connectivity: A multi-label classification approach," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1579–1582, Dec. 2019.

[15] K. Jin, X. Cai, J. Du, H. Park, and Z. Tang, "Toward energy efficient and balanced user associations and power allocations in multiconnectivity-enabled mmWave networks," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 4, pp. 1917–1931, Dec. 2022.

[16] "Multi-connectivity stage 2," 3GPP, Sophia Antipolis, France, 3GPP Rep. TS 37.340, version 16.1.0, Mar. 2020.

[17] J. Sundararajan et al., "Performance evaluation of extended reality applications in 5G NR system," in *Proc. IEEE 32nd Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2021, pp. 1–7.

[18] V. Petrov, M. Gapeyenko, S. Paris, A. Marcano, and K. I. Pedersen, "Extended reality (XR) over 5G and 5G-advanced new radio: Standardization, applications, and trends," 2022, *arXiv:2203.02242*.

[19] C. Perfecto, M. S. Elbamby, J. D. Ser, and M. Bennis, "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, Apr. 2020.

[20] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, and P. Popovski, "Can Terahertz provide high-rate reliable low-latency communications for wireless VR?" *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9712–9729, Jun. 2022.

[21] Z. Mao, C. E. Koksal, and N. B. Shroff, "Optimal Online scheduling with arbitrary hard deadlines in multihop communication networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 177–189, Feb. 2016.

[22] K. S. Kim, C.-P. Li, and E. Modiano, "Scheduling multicast traffic with deadlines in wireless networks," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 2193–2201.

[23] X. Liu, W. Wang, and L. Ying, "Spatial–temporal routing for supporting end-to-end hard deadlines in multi-hop networks," *Perform. Eval.*, vol. 135, Nov. 2019, Art. no. 102007.

[24] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 127–142, Jan. 2019.

[25] H. Deng and I.-H. Hou, "On the capacity requirement for arbitrary end-to-end deadline and reliability guarantees in multi-hop networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 1, pp. 15–16, 2017.

[26] 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, Sep. 2017.

[27] M. A. Javed, P. Liu, and S. S. Panwar, "A multi-connectivity architecture with data replication for XR traffic in mmWave networks," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, 2023, pp. 1–9.

[28] A. Koutsaftis, M. F. Özkoç, F. Fund, P. Liu, and S. S. Panwar, "Fast wireless Backhaul: A multi-connectivity enabled mmWave cellular system," in *Proc. IEEE Global Commun. Conf.*, 2022, pp. 1813–1818.

[29] S. Sun, G. R. MacCartney, and T. S. Rappaport, "A novel millimeter-wave channel simulator and applications for 5G wireless communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–7.

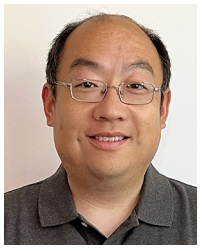
[30] S. Ju, O. Kanhere, Y. Xing, and T. S. Rappaport, "A millimeter-wave channel simulator NYUSIM with spatial consistency and human blockage," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[31] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, "Downlink and uplink cell association with traditional macrocells and millimeter wave small cells," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6244–6258, Sep. 2016.

[32] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2196–2211, Oct. 2015.

[33] S. Sun et al., "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 2843–2860, May 2016.
[34] 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.
[35] M. 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.
[36] I. K. Jain, R. Kumar, and S. Panwar, "Driven by capacity or blockage? a millimeter wave blockage analysis," in *Proc. 30th Int. Teletraffic Congr. (ITC 30)*, vol. 1, 2018, pp. 153–159.
[37] I. K. Jain, R. Kumar, and S. S. Panwar, "The impact of mobile blockers on millimeter wave cellular systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 854–868, Apr. 2019.
[38] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Netw.*, vol. 10, pp. 555–567, Sep. 2004.
[39] "5G-study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, Sophia Antipolis, France, 3GPP Rep. TR 38.901, version 14.3.0, Jan. 2018.
[40] M. A. Javed, P. Liu, and S. S. Panwar, "Predictive data replication for XR applications in multi-connectivity enabled mmWave networks," in *Proc. Int. Balkan Conf. Commun. Netw. (BalkanCom)*, 2023, pp. 1–5.
[41] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[42] S. S. Panwar, D. Towsley, and J. K. Wolf, "Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service," *J. ACM*, vol. 35, no. 4, pp. 832–844, 1988.
[43] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," in *Proc. 2nd ACM Int. Workshop Wireless Mobile Multimedia*, 1999, pp. 35–42.
[44] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.
[45] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proc. 29th IEEE Conf. Decis. Control*, 1990, pp. 2130–2132.
[46] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.
[47] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logist. Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[48] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.
[49] M. Mezzavilla, S. Dutta, M. Zhang, M. R. Akdeniz, and S. Rangan, "5G mmWave module for the NS-3 network simulator," in *Proc. 18th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, 2015, pp. 283–290.
[50] G. MacCartney, T. Rappaport, and S. Rangan, "Rapid fading due to human blockage in pedestrian crowds at 5G millimeter-wave frequencies," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–7.
[51] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

MUHAMMAD AFFAN JAVED (Member, IEEE) received the B.S. degree in electrical engineering from the Lahore University of Management Sciences, Lahore, Pakistan, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from the NYU Tandon School of Engineering in 2017 and 2023, respectively. He is currently working as a Research Software Engineer with AT&T Labs. His research interests include wireless communications and wireless networks, with a focus in enabling low latency communications.



PEI LIU (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Polytechnic University in 2007. He is a Research Assistant Professor with the Electrical and Computer Engineering Department, NYU Tandon School of Engineering. His research interests are in designing and analyzing wireless network protocols with an emphasis on cross-layer optimizations. His current research topics include next-gen communication networks and software defined radios.

SHIVENDRA S. PANWAR (Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, MA, USA. He is currently a Professor with the Electrical and Computer Engineering Department, NYU Tandon School of Engineering. He is also the Director of the New York State Center for Advanced Technology in Telecommunications, the Co-Founder of New York City Media Lab, and a member of NYU Wireless. He has coauthored a textbook titled *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press). His research interests include the performance analysis and design of networks. His current research focuses on cross-layer research issues in wireless networks and multimedia transport over networks. He was a winner of the IEEE Communication Society's Leonard Abraham Prize for 2004, the ICC Best Paper Award in 2016, and the Sony Research Award. He was also co-awarded the Best Paper in 2011 Multimedia Communications Award. He has served as the Secretary for the Technical Affairs Council of the IEEE Communications Society. He is a Fellow of the National Academy of Inventors.