INTERNATIONAL JOUNAL OF: SCIENCE RESEARCH AND TECHNOLOGY

TIMBOU-AFRICA ACADEMIC PUBLICATIONS
INTERNATIONAL JOURNAL, NOVEMBER, 2022 EDITIONS
VOL. 11 NO. 9 ISSN: 2623-7861

# OPTIMAL 5G RESOURCE ALLOCATION FOR ULTRA-RELIABLE LOW LATENCY COMMUNICATION (URLLC) AND ENHANCED MOBILE BROADBAND (eMBB) USE CASES

## ABSTRACT

Due to the development of wireless networks for communication, a new issue has emerged. Users have become more organized (especially in 5G) by applications to ensure relatively easy spectrum access. The coexistence of eMBB and URLLC traffic has given rise to puncturing issues. The burstiness of URLLC traffic, which affects eMBB traffic by forcing its packets to wait until the spectrum is briefly free of URLLC

**ABDULHAKEEM-ALUGO, A. A[1], MOHAMMED, A. S[1], & DAUDA, U. S[2]**

[1]Telecommunication Engineering Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria.
[2]Electrical and Electronics Engineering Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria

## INTRODUCTION

When it comes to mobile networks, the fifth generation (5G) technology, which has reportedly been introduced in several regions of the world while many others are still awaiting its introduction, has shown promise. Even better, a variety of application areas have been suggested, some of which are currently being used like; self-driving cars, unmanned aerial vehicles (UAVs), and smart cities. In the situations of these applications, the benefit of 5G technology would result in an exponential improvement.

Wide area network (WAN) technology used in the 5G network supports all communication profiles used in industrial settings. Agility is a primary goal in the design of factories according to the industry 4.0 paradigm. Rearrangeable production line modules, automated guided vehicles, autonomous robots, linked worker solutions, and even drones are a few of the key technologies that enable such agility in factories.

In accordance with International Telecommunication Union (ITU) standards, these applications have now been divided and classified into three use cases or service types. They are:

**traffic, is the main cause of this puncturing. In addition to Q-learning, a joint power and resource block allocation was done in order to get around this issue. The scheduling of resources was done using Q-learning in order to get the best multiplexing possible without puncturing eMBB resources. As a result, a scheduling pattern was created that enhanced reliability by increasing throughput and reducing latency. The suggested algorithm was implemented using MATLAB 5G and Deep Learning toolboxes. The algorithm (OLRT-Q) was compared to three other algorithms and there were some favourable conclusions. According to analysis, a 16% throughput boost over LRT-Q and a 47.7% increase over LR-Q at 2 Mbps (the case with the highest load) were recorded. At 1.5 Mbps, we noticed an increase of 12.2% and 33.16% in performance over LRT-Q and LR-Q, respectively. It outperformed LRT-Q and LR-Q in the case of a 1 Mbps load scenario by 9.44% and 19.09%, respectively. There is an improvement in throughput by 13.36% compared to LR-Q and a 9.58% increase compared to LRT-Q under the lowest load scenario permitted by the standard (0.5 Mbps).**

Keywords: ***eMBB, Latency, OLRT-Q, Reliability, Throughput, URLLC.***

1) eMBB (Enhanced Mobile Broadband)
2) uRLLC (Ultra Reliable Low-Latency Communications)
3) mMTC (Massive Machine-type Communications) (Massive Machine-type Communications) (Sanou, 2018)

The ITU has gradually implemented a number of standards in the information technology (IT) sector to control how people interact with technology while taking into account maximum performance. The ITU also established standards before 5G was available to control how consumers interacted with technology and applications in general.

These numerous use cases are supported by the various quality of service (QoS) needs for 5G wireless networks in terms of data rate, dependability, and latency. Network slicing in cloud radio access networks (C-RANs) allows mobile network operators (MNOs) to virtualize network resources, such as transmit power and physical resource blocks (PRBs), in the shared physical network. Given the restricted resources in the remote radio heads (RRHs), it is essential to allocate resources among network slices efficiently to enhance system throughput while still meeting the QoS requirements of the users (Pocovi et al., 2018).

For 5G wireless systems to function effectively, they must be able to support both mobile broadband and ultra-reliable low latency communications (or non-bursty and bursty traffic) on the same network without favouring one over the other.

On the one hand, gigabit per second data rates (with a bandwidth of several hundred MHz) and modest latency is supported by broadband traffic, or eMBB. On the flip side, URLLC traffic demands very high dependability (99.999%) and extremely short delays – low latency (0.25–0.3 msec/packet) (Anand et al., 2018).

To meet these diverse needs, the 3rd Generation Partnership Project (3GPP) standards body has developed an original superposition/puncturing framework for multiplexing URLLC and eMBB traffic in 5G cellular systems. The Key Performance Indicators (KPIs) or Quality of Service (QoS) criteria for each of these standards are used as a metric to assess that specific use case or service category.

The key system capacity concerns for eMBB users are: Extreme Throughput, Enhanced Spectral Efficiency, and Extended Coverage.

Users of uRLLC are more focused on mission-critical applications, and some key KPIs for these applications include: low latency; high reliability; high availability; and location precision.

Users of massive-machine type communications (mMTC) are primarily concerned with Extreme Density, which includes: Energy Optimization, High Connection Density, Low Complexity, and Extended Coverage.

Studying these three use cases, it has been suggested that there is a strong possibility that a user would only be required to use one or, at most, two of them at any given time because each of them has quite different applications and attributes. In the time-frequency domain or spectrum, whether we are thinking about the orthogonal or non-orthogonal slicing of resources in the wireless network, there is often a designated single frequency channel for mMTC. This means that eMBB and uRLLC users typically share other network blocks, especially in non-orthogonal slicing because in orthogonal slicing there are dedicated blocks for uRLLC traffic that are not utilized by any other sort of traffic, whether it is idle or not (Pocovi et al., 2018).

As a result, numerous research on multiplexing eMBB and uRLLC use cases have been conducted because, more often than not, their packets co-exist or at the very least interact in the wireless network architecture. It has been observed that many compromises arise as a result of this multiplexing and coexistence, and these tradeoffs need to be handled.

It is important to note that several approaches have been put out for dealing with the tradeoffs that arise when the two use cases coexist. It could be tempting to want to give up one for the other because their demands are distinct from one another. For instance, we observe that puncturing happens with respect to the resources of eMBB traffic as uRLLC users are given higher priority during resource allocation in order to meet their latency demands. This finally results in throughput degradation as it affects eMBB users.

In essence, the aim of this work is to multiplex URLLC and eMBB for optimal allocation of 5G radio resources. The research's goals are to: Develop a pattern for scheduling throughput, latency, and dependability utilizing Q-learning in an effort to accomplish this goal and to compare and contrast throughput and latency results with industry standards in order to access the quality of research.

**METHODOLOGY**

This section describes how a multi-agent Q-learning algorithm is employed to increase throughput, reliability, and minimize latency, all of which are directly related to the delivery of packet data rates.

**SOFTWARE**

The MATLAB simulator will be used for this work, and a discrete-level simulator built on the MATLAB 5G toolbox will be used for simulations. The 5G and Deep Learning Toolboxes will be employed. Usually, the need to write codes when using toolboxes is largely eliminated and makes the work easier and more efficient. However, the 5G and Deep Learning Toolboxes didn't appear on MATLAB until its 2019 version.

**METHOD EMPLOYED**

Before moving forward, there are a few issues that need to be resolved in order for us to achieve our aim. The eMBB packets would be modeled with Poisson traffic and the URLLC packets need to be modeled with both Poisson & CBR traffics since it is a bursty kind of packet.

We would then multiplex packets of both forms of traffic on a 5G time-frequency grid which will enable us to address the puncturing issue that typically occurs when eMBB and URLLC packets are multiplexed.

On a number of gNodeBs that support eMBB and URLLC users, our suggested algorithm would be put to the test using the 5G-NR Rel. 15 standard. The flexible resource distribution offered by the 5G-NR standard is made possible by Transmit Time Intervals (TTIs) of varying length. Resolution in the time direction is based on OFDM symbol slots of 2, 4, 7, or 14. The standard encourages the rapid transfer of messages that are perfectly suited for URLLC communication by using the greatest resolution possible, which is TTI of 2 OFDM symbols. To address the high throughput demands of eMBB users, larger resolutions—like TTI of 14 OFDM symbols—are utilized. The total downlink bandwidth, $B$ MHz, is split into $N_{RB}$ resource blocks, each of which consists of 12 consecutive subcarriers, in terms of spectrum distribution. Additionally, as mentioned in (Specification, 2018), a Resource Block Group is formed by contiguous resource blocks (RBG).

Let a set of $K$ RBGs with a size of $[N_{RB} / K]$ resource blocks be denoted by the symbol, ƙ. To reduce the number of states in our Q-learning technique, we would use RBG as our allocation unit in the frequency direction. Additionally, each $k^{th}$ RBG is given a transmission power, $p_{k,j}$, via the $j^{th}$ gNodeB. The Q-learning algorithm, which would be referenced later aims to improve transmission power assignments and RBG allocation.

According to our system model, each gNodeB has an equivalent amount of transmission buffers to the attached users. The downlink scheduler of each TTI distributes resources to users who are active or who have pending data transfers. In particular, the scheduler implements joint power and RBG allocation while taking the QoS requirements of URLLC and eMBB users into account. In contrast to eMBB users, whose traffic is exclusively modeled using Poisson arrivals, URLLC users' traffic is represented using a combination of CBR and Poisson arrivals.

Link capacity between the User Equipment, UE $i$ and gNodeB $j$ is computed thus:

$$C_{i,j} = \sum_{k=1}^{K} \omega_k log_2(1 + \frac{p_{k,j} x_{k,i,j} g_{k,i,j}}{\omega_k N_0 + \sum_{\substack{m \in J \\ m \neq j}} p_{k,m} x_{k,i,m} g_{k,i,m}}) \quad (1)$$

$N_o$ is the single-sided power spectral density of additive white Gaussian noise, and $w_k$ is the bandwidth of the kth RBG. $x_{k,i,j}$ is the link allocation indicator of the RBG, $p_{k,j}$ is the transmit power of the $j^{th}$ gNodeB on the $k^{th}$ RBG, and $g_{k,i,j}$ is the channel co-efficient.

The link allocation indicator of link (k,i,m) is denoted by $x_{k,i,m}$, the channel coefficient is $g_{k,i,m}$ and the transmit power of the $m^{th}$ interfering gNodeB is $p_{k,m}$.

"Equation (1)" shows that reducing interference is essential for boosting throughput. Ineffective power management will significantly affect edge users, reducing throughput as a result.

As indicated in "(2)", there are three components that make up packet latency:

$$T = T^q + T^{tx} + T^{harq} \quad (2)$$

where $T^{tx}$ stands for transmission delay, $T^q$ for queuing time, and $T^{harq}$ for HARQ re-transmission round-trip delay. $T^{harq}$ = 4.TTI is our default assumption in line with (Esswie et al., 2018). During HARQ, a retransmitted packet has a greater priority than a brand-new packet.

The transmission delay of user $i$ linked to gNodeB $j$ can be calculated by dividing the packet length $Li,j$ by the connection capacity $Ci,j$, as illustrated below:

$$T_{i,j}^{tx} = \frac{L_{i,j}}{C_{i,j}} \qquad (3)$$

The ideal power allocation and, consequently, interference mitigation have a substantial impact on transmission delay in addition to throughput, as demonstrated in "(3)". Transmission rate, on the other hand, has an impact on the Radio Link Control (RLC) layer. There is less segmentation as the rate rises. The transmission latency is thereby decreased. Additionally, giving a user access to more RBGs enlarges the assigned transport block, reducing the bandwidth.

The queuing time in "(2)" corresponds to the scheduling delay of the MAC scheduler. Therefore, the scheduler must schedule URLLC traffic as soon as it arrives and limit the number of HARQ re-transmissions in order to provide URLLC users with a 1ms response time. We assume that there can be only one HARQ re-transmission in order to achieve the smallest delay possible. On the other hand, limiting re-transmissions may lead to a higher Packet Drop Rate (PDR) and hence lower reliability. For edge users in particular, the effects of such poor reliability might be disastrous. In our suggested algorithm, transmission power control based on RBG achieves great reliability while minimizing latency.

It is important to note that improving the latency and reliability of URLLC users is projected to have an effect on eMBB users' throughput performance as seen in "(1)". This calls for a resource allocation strategy that balances URLLC and eMBB KPIs. In the section that follows, we go over our proposed method for power and resource block allocation, which is based on Q-learning and aims to jointly maximize throughput for eMBB users and latency and reliability for URLLC users.

**Q-learning Algorithm (OLRT-Q)**

The suggested algorithm (OLRT-Q) is based on decentralized reinforcement learning and uses a Q-learning algorithm to execute resource allocation on each gNodeB. A Markov Decision Process (MDP), which includes agents, states, actions, a reward function, and a policy, is how Q-learning is formally described. The foundation of Q-learning is interaction with the environment and learning through trial-and-error rewards given for acceptable or desired behaviors. In more detail, an agent decides on a course of action, carries it out, and is rewarded based on how well the action was completed. Until the agent discovers an action selection strategy that maximizes its total discounted reward, this cycle is repeated. To determine the quality of the visited state action pair, Q-learning employs an iterative update, as follows:

$$Q^{new}\left(s^{(t)},a^{(t)}\right) \leftarrow (1-\alpha).Q\left(s^{(t)},a^{(t)}\right)+\alpha\,.\left[r^{(t)}+\gamma.\max_{a} Q^{old}\left(s^{(t+1)},a\right)\right] \quad (4)$$

Where $Q^{new}\left(s^{(t)},a^{(t)}\right)$ is the Q-value of the state-action pair $\left(s^{(t)},a^{(t)}\right)$ at the t[th] iteration, $\alpha$ is the learning rate, $\gamma$ is the discount factor, and $r^{(t)}$ is the instantaneous reward. Since Q-values are stored in a Q-table that is indexed by states and actions, the size of the Q-table is governed by the state-action space.

OLRT-Q, the proposed algorithm is a Q-learning algorithm with a reward function that aims to improve the latency and reliability of URLLC users as well as the throughput of eMBB users. The algorithm will refer to the combined power and resource block allocations that are carried out by agents, or gNodeBs, as actions. To keep the size of the Q-table manageable, we would arrange 8 consecutive resource blocks into an RBG, and the agent would allocate RGBs. States in the suggested algorithm, which represent the effects of other agents' actions, are driven by observations from the environment. It is particularly difficult to reduce user interference when trying to increase throughput, reliability, and decrease latency. As a result, the states are made to represent the typical Signal to Interference Noise Ratio (SINR) attained by users connected to each gNodeB thus:

$$S_{k,j}=\begin{cases}S_0 & \overline{\gamma_{k,J}}\geq\gamma_{th}\\ S_1 & Otherwise\end{cases} \quad (5)$$

$\overline{\gamma_{k,J}}$ is the estimated value of the $k^{th}$ RBG's SINR value on average, and is defined as:

$$\overline{\gamma_{k,J}}=\beta\,\overline{\gamma_{k,J}^{U}}+(1-\beta)\overline{\gamma_{k,J}^{E}} \quad (6)$$

$\overline{\gamma_{k,J}^{U}}$ is the average SINR of URLLC users and $\overline{\gamma_{k,J}^{E}}$ is the average SINR of eMBB users. $\beta$ is the priority controlling factor given to URLLC and eMBB users, $\gamma_{th}$ is a threshold SINR value. $\gamma_{th}$ usually is chosen to maintain high probability of decoding. The reward function is finally formulated to reward actions that achieve the proposed objectives:

$$\rho_{k,j}^{U}=\begin{cases}1-\max\limits_{i\in\tilde{U}}(T_{i,j}^{q})^2 & \overline{\gamma_{k,J}}\geq\gamma_{th}\\ -1 & Otherwise\end{cases} \quad (7)$$

$$\rho_{k,j}^{E}=\frac{2}{\pi}\tan^{-1}\left(\overline{C_{k,J}^{E}}\right) \quad (8)$$

$$\rho_{k,j}=\beta\,\rho_{k,j}^{U}+(1-\beta)\rho_{k,j}^{E} \quad (9)$$

where $\rho_{k,j}^{U}$ is the reward for URLLC users on the $k^{th}$ RBG, $\rho_{k,j}^{E}$ is the reward for eMBB users, $\rho_{k,j}$ is the overall reward for the $j^{th}$ gNodeB. The average throughput of eMBB users is represented by $C_{k,j}^{E}$, whereas the final packet queuing delay of the ith URLLC user ($i \in \tilde{U}$) is represented by $T_{i,j}^{q}$. "Equation (8)" addresses the KPIs of both URLLC and eMBB users by modifying parameter $\beta$. By rewarding the agent with a value proportionate to the queuing delay as long as its reliability reaches a predetermined threshold, such as the SINR


TIJSRAT
ISSN: 2623-7861

threshold, "(6)" specifically aims to reduce the latency and reliability of URLLC users. In fact, the reward value is determined by the URLLC packets on the queue for the shortest time. This means that the suggested algorithm will attempt to shorten the longest potential queue time. Additionally, increasing the average SINR has a significant impact on total latency since it reduces packet segmentation and transmission delay. Overall, "(6)" motivates the MAC scheduler to immediately assign URLLC users to better RBGs, producing minimal latency and high dependability.

"Equation (7)" aids eMBB users in increasing throughput, which raises the reward value almost to one. Using the parameter $\beta$ in "(8)", we can obtain the balance between the opposing KPIs.

The proposed algorithm's steps for each agent (gNodeB) are thus represented:

1: **Initialization:** Q-table ← 0, $\alpha$, $\gamma$ and $\epsilon$.
2: **for** TTI t = 1 to T **do**
3: **Step 1:** Agent (that is, gNodeB) receives uplink report (that is, SINR) from its attached users.
4: **Step 2:** Compute the reward as in Eq. (6), (7), and (8).
5: **Step 3:** Update the Q-value of the current state-action pair as in Eq. (4).
6: **Step 4:** Observe and transit to next state as in Eq. (5).
7: **Step 5:** Select the next action based on $\epsilon$ –greedy policy.
8: **Step 6:** Repeat at Step 1.
9: **end for**

## SIMULATION PARAMETERS

Table1 is a summary of network and Q-learning parameters used to implement our simulations in this work.

**TABLE 1: NETWORK PARAMETERS**

| Parameters | Values |
|---|---|
| **Network environment** | 5 gNodeBs and 500-meter inter-site distance in the 3GPP Urban Macro (UMa) network |
| **PHY configuration** | Subcarrier spacing of 15 kHz<br>Each resource block has 12 subcarriers.<br>K = 13 (number of RBGs)<br>TTI of 2 OFDM symbols (0.1429 ms)<br>15 dB Tx/Rx antenna gain |


TIJSRAT
ISSN: 2623-7861

| | |
|---|---|
| | 40 dBm maximum transmission power (Specification, 2017) |
| **Carrier configuration** | B = bandwidth of 20 MHz<br>100 resource blocks ($N_{RB}$)<br>4 GHz carrier frequency |
| **HARQ** | Round-trip latency of 4 TTI<br>HARQ is asynchronous with 6 processes<br>1 HARQ re-transmission maximum |
| **Propagation** | $128.1 + 37.6\ log_{10}(D\,[Km])$<br>5dB Noise Figure<br>5dB Penetration loss<br>Shadowing = Log-Normal Shadowing (8dB) |
| **User distribution** | Stationary distribution<br>Uniform or even distribution<br>50 URLLC packets (10packets / cell)<br>25 eMBB packets (5packets / cell) |
| **Traffic model** | Payload size = 32 byte<br>URLLC = Poisson (80%) + CBR (20%)<br>eMBB = Poisson (100%) |
| **URLLC Load/Cell** | [0.5: 0.5: 1] Mbps |
| **eMBB Load/Cell** | 0.5 Mbps |
| **Q-Learning** | $\alpha = 0.5$<br>$\gamma = 0.9$<br>$\varepsilon = 0.05$<br>$\beta = 0.1$<br><br>where $\alpha$ is learning rate, $\gamma$ is discount factor, $\varepsilon$ is exploration probability and $\beta$ is controlling factor. |

**RESULTS AND DISCUSSION**

In addition to the previously mentioned techniques, the network and Q-learning parameters in Table1 will be tested and implemented. The outcomes of the tests and simulations will then be examined and contrasted with those from our benchmark in literature, i.e. (Elsayed & Erol-kantarci, 2019).

We run simulations using our discrete-level simulator, based on the MATLAB 5G toolbox, as mentioned earlier. In our simulations, we look at 5 gNodeBs, each of which has 10

URLLC and 5 eMBB users. While eMBB traffic is entirely based on Poisson arrivals, URLLC traffic is a mix of 20% CBR and 80% Poisson arrivals. The payload size is 32 bytes for all users. Additionally, whereas eMBB traffic loads are set at 0.5 Mbps, URLLC traffic loads per cell range from 0.5 to 1 Mbps. Ten simulation runs' worth of simulation results for 5000 TTIs are combined, averaged, and presented with a 95% confidence interval. In this work, we use TTI of 2 OFDM symbols, the best time resolution, as our scheduling interval. The action space of Q-learning based algorithms consists of power and RBG allocations. 13 RBGs are utilized for a system bandwidth of 20 MHz, with the first 12 RBGs containing 8 resource blocks in a row and the last RBG containing 4 resource blocks in a row. According to (Specification, 2017) the maximum transmission power of the gNodeB is set at 40 dBm, and the power allocation, $\rho_{k,j}$, is selected from a range of 0 to 3 dBm. Finally, a SINR threshold of $\gamma_{th}$ = 20 dB is used to maintain a high possibility of successful reception.

Performance of the proposed method is evaluated using URLLC and eMBB traffic KPIs, that is, latency and reliability for URLLC and throughput for eMBB.

**RESULTS FOR eMBB KPI (THROUGHPUT)**

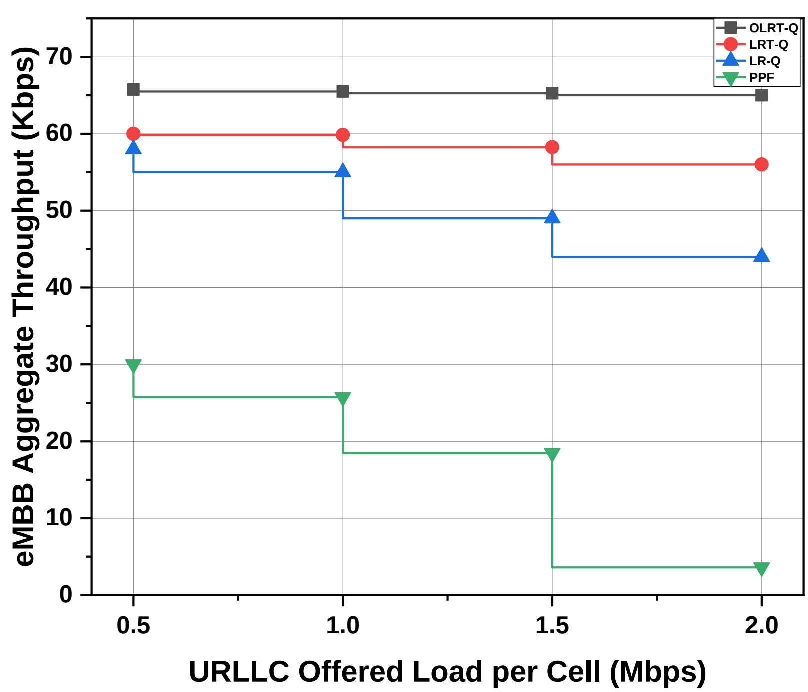


Figure 1: eMBB Users Cumulative Throughput [Mbps] against Traffic Load of URLLC


TIJSRAT
ISSN: 2623-7861

The aggregate throughput of eMBB users under various URLLC traffic loads per cell, ranging from 0.5 Mbps to 1 Mbps, is depicted on the graph in Figure1. In fact, improving the throughput performance of eMBB users should have an impact on increasing the traffic load on URLLC. The PPF algorithm (Pocovi et al., 2018) has a very low throughput because it hardly takes into account eMBB users, as shown in Figure1. As a result, it is clear that throughput, an eMBB KPI, has declined. Additionally, as the authors heavily took into account eMBB users, Figure1 depicts the LR-Q algorithm, which has a substantially higher throughput than PPF. In comparison to the two algorithms previously described, the LRT-Q algorithm was able to achieve better improvement since it used the Q-learning algorithm. In fact, compared to LR-Q and the PPF approach, the throughput of the LRT-Q is increased by 29% and 21 times, respectively, in the scenario with the highest load. Even with a 0.5 Mbps proposed load, the LRT-Q has twice the throughput of PPF.

OLRT-Q, our method, produced a noticeable change when compared to the other three algorithms. It is important to note that as the URLLC loads per cell rise from 0.5 Mbps to 1 Mbps, there is a slight, essentially negligible decrease in throughput at each cell. Although it is difficult to tell if there is an increase or decrease in URLLC loads per cell when they move from 0.5 Mbps to 1 Mbps because the values appear to be the same, we can still see that there has been improvement over the preceding algorithms, PPF, LR-Q, and LRT-Q.

According to analysis, we have a 16% throughput boost over LRT-Q and a 47.7% increase over LR-Q at 2 Mbps (the case with the highest load). Additionally, at 1.5 Mbps, we saw increases of 12.2% and 33.16 percent in performance over LRT-Q and LR-Q, respectively. We outperformed LRT-Q and LR-Q in the case of a 1 Mbps load scenario by 9.44% and 19.09%, respectively. There is an improvement in throughput of 13.36% compared to LR-Q and a 9.58% increase compared to LRT-Q under the lowest load scenario permitted by the standard, or 0.5 Mbps.

This analysis shows that, when compared to our baseline methods, the OLRT-Q, which was created by incorporating Q-learning in addition to power and resource block allocation, performs better in all compartments for throughput enhancement.

## RESULTS FOR AVERAGE URLLC LATENCY FOR URLLC LOADS: 0.5 AND 1 MBPS

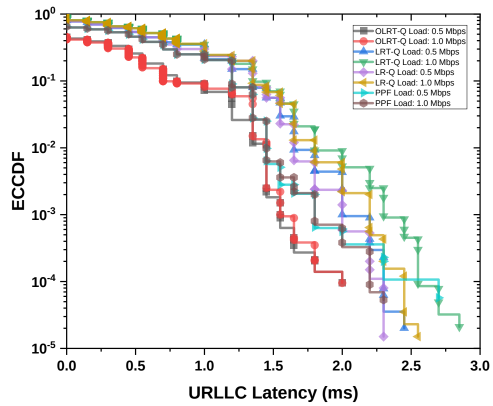


Figure.2: Average URLLC Latency [ms]; URLLC Loads: 0.5 and 1 Mbps; eMBB Load: 0.5 Mbps

Since latency is the KPI that needs to be adjusted and is one of the KPIs of URLLC users, the graph in Figure2 depicts the average delay of URLLC users with URLLC loads of 0.5 and 1 Mbps while eMBB load is set at 0.5 Mbps. The Empirical Complementary Cumulative Distribution Function (ECCDF), shows how the load changes over time by cumulating the latencies of URLLC users in this scenario. It compares the varied latencies for the four algorithms OLRT-Q, LRT-Q, LR-Q, and PPF. However, comparing the main algorithm presented in this work, OLRT-Q, to each of the other algorithms in separate graphs might make Figure2 easier to read.

## COMPARING URLLC LATENCY OF OLRT-Q TO PPF

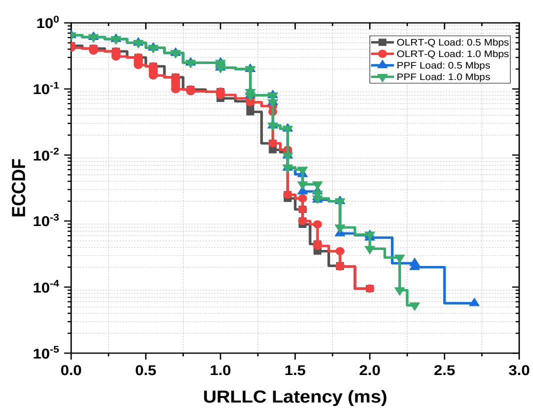


Figure 3: Average URLLC Latency [ms] of OLRT-Q and PPF


ISSN: 2623-7861

TIMBOU-AFRICA ACADEMIC PUBLICATIONS
NOV., 2022 EDITIONS, INTERNATIONAL JOURNAL OF:
SCIENCE RESEARCH AND TECHNOLOGY VOL. 11

The OLRT-Q algorithm, which is used in this work, and PPF, which was used in one of the earlier works on the subject, are the two algorithms that are compared in Figure3. It is crucial to remember that the lower the cumulative frequency, the better the proposed algorithm for that specific metric under consideration. In this instance, the lower latency values are preferable to those with higher latency for URLLC users.

There are a few inferences we may make from the graph. OLRT-Q performs better than PPF in a number of areas. We can say that we have comparable latency values at the $10^{-1}$ percentile by interpreting the ECCDF values as percentiles. We also have an overlap at the $10^{-2}$ percentile, indicating that the values for the latencies there are nearly identical. When compared to the $10^{-3}$ percentile, the values start to diverge. We can now see that each URLLC load at 0.5 and 1 Mbps has a latency difference of about 0.3 ms. Here is where the superiority of our OLRT-Q algorithm becomes apparent. We have a latency difference at the $10^{-4}$ percentile of 0.25 ms at 1 Mbps load and 0.5 ms at 0.5 Mbps load. It is apparent that at this point, the OLRT-Q algorithm now outperforms the PPF much more. This shows that the level of latency that was accomplished in earlier works has been maintained and subsequently improved.

**COMPARING URLLC LATENCY OF OLRT-Q TO LR-Q**

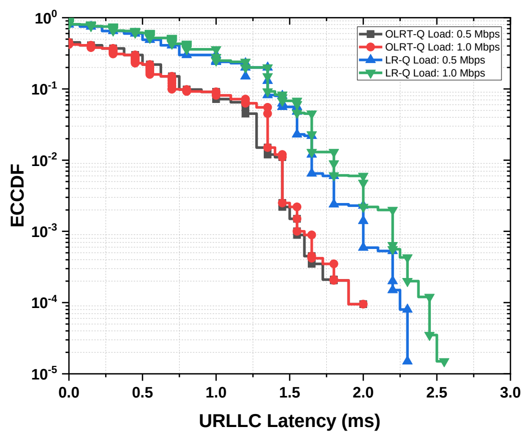


Figure 4: Average URLLC Latency [ms] of OLRT-Q and LR-Q

The two algorithms that are compared in Figure4 are the one that is described in this work, OLRT-Q, and one that was employed in a prior study on the subject, LR-Q.

The graph shows that OLRT-Q performs better than LR-Q in a number of categories. By converting the ECCDF values to percentiles, we can say that the $10^{-1}$ percentile latency values for both URLLC loads of 0.5 Mbps and 1 Mbps differ by roughly 0.5 ms. At the $10^{-2}$



TIJSRAT
ISSN: 2623-7861

percentile, latency values at 0.5 Mbps and 1 Mbps URLLC loads, respectively, differ by approximately 0.35 ms and 0.25 ms. The difference in latency for each of the URLLC loads at 0.5 and 1 Mbps, respectively, is roughly 0.45 ms and 0.60 ms at the $10^{-3}$ percentile. For the $10^{-4}$ percentile, there is a difference in latency of 0.35 ms at 0.5 Mbps load and 0.35 ms at 1 Mbps load. At this stage, our algorithm continues to perform better than the LR-Q. This once more shows that we were able to keep and raise the level of latency that was accomplished in earlier research.

COMPARING URLLC LATENCY OF OLRT-Q TO LRT-Q

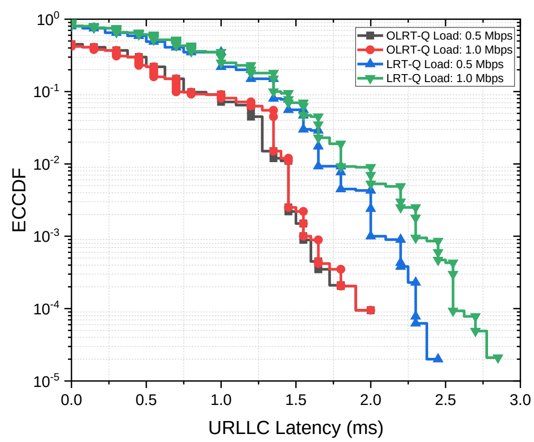


Figure 5: Average URLLC Latency [ms] of OLRT-Q and LRT-Q

The two algorithms that are compared in Figure5 are OLRT-Q, which is the algorithm that is presented in this work, and LRT-Q, which is the most advanced of the algorithms that we have compared to and was used in one of the earlier works on the subject.

We can see from the graph that OLRT-Q performs better than LRT-Q in a number of areas. Using the ECCDF values as percentiles, we can say that for both URLLC loads of 0.5 Mbps and 1 Mbps, there are differences in latency values at the $10^{-1}$ percentile of about 0.25 ms. At the $10^{-2}$ percentile, latency values at 0.5 Mbps and 1 Mbps URLLC loads, respectively, differ by approximately 0.15 ms and 0.25 ms. The difference in latency for each of the URLLC loads at 0.5 and 1 Mbps, respectively, is about 0.5 and 0.7 ms at the $10^{-3}$ percentile. We have a latency difference at the $10^{-4}$ percentile of 0.35 ms at 1 Mbps load and 0.55 ms at 0.5 Mbps load. Now, our algorithm continues to perform better than the LRT-Q. The implication is that OLRT-Q has lower latency than other algorithms, as we also inferred from PPF and LR-Q. This once more shows that we were able to keep and raise the level of latency that was accomplished in earlier research.

## RESULTS FOR AVERAGE PACKET DROP RATE

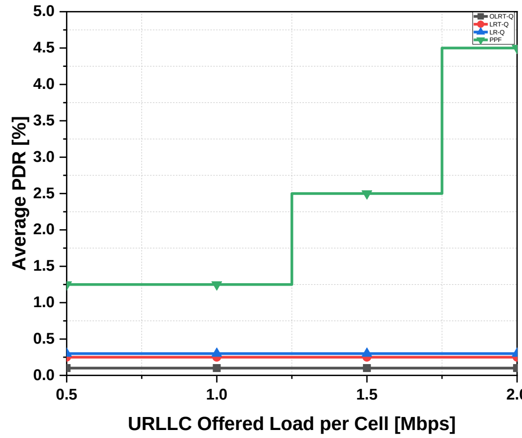


Figure 6: Average Packet Drop Rate [%]; eMBB Load: 0.5 Mbps

Figure6 provides more information on the connection between the reliability and latency KPIs for URLLC users. After the initial re-transmission, latency can be reduced if we permit packets to be dropped.

PPF has a re-transmission issue as a result of ineffective interference handling. Figure6 shows that the PPF drop rate is rising as a result. While LRT-Q and LR-Q were only able to achieve packet drop rates of about 0.25 and 0.3%, respectively, thanks to their limitations on power allocation, OLRT-Q was able to achieve drop rates of less than 0.1%. In our worst-case traffic load, OLRT-Q outperforms PPF, lowering the packet drop rate by 4%. It also outperforms LR-Q and LRT-Q by 3.4% and 2.7%, respectively.

After 3000 TTI, or 428.5 ms, we finally noticed convergence of our OLRT-Q algorithm. The outcomes could not be plotted because of space limitations.

## CONCLUSION

The difficulties with multiplexing URLLC and eMBB use cases on the same spectrum have successfully been addressed by this research work. The primary problem found is the depletion of eMBB users' resources. We have put into practice a technique using the Q-learning algorithm that was successful in eradicating the initial issues and resolving the constraints identified in the studied literature. This approach also prevented us from negatively impacting URLLC users in an attempt to compensate for the constraints of eMBB users. There was an arrangement of resources where they could go beyond the spectrum if necessary.

In a nutshell, we have been able to increase throughput and reliability while drastically lowering packet latency. Thus, we may draw the conclusion that this work has brought the 5G world closer to the established requirements of 99.999% data rate and less than 1 ms latency.

## REFERENCES


*30 top 5G use cases: discover what 5G can really do | 5Gradar*. (n.d.). Retrieved September 7, 2021, from https://www.5gradar.com/features/what-is-5g-these-use-cases-reveal-all

*5G Top 10 Use Cases - Huawei X Labs*. (n.d.). Retrieved September 7, 2021, from https://www.huawei.com/en/technology-insights/industry-insights/outlook/mobile-broadband/xlabs/use-cases/5g-top-10-use-case

Alsenwi, M., Tran, N. H., Bennis, M., Pandey, S. R., Bairagi, A. K., & Hong, C. S. (2021). Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach. *IEEE Transactions on Wireless Communications*, *20*(7), 4585–4600. https://doi.org/10.1109/TWC.2021.3060514

Alsenwi, M., Tran, N. H., Member, S., Bennis, M., Member, S., Kumar, A., Hong, C. S., & Member, S. (2006). *eMBB-URLLC Resource Slicing : A Risk-Sensitive Approach*. 1–4.

Anand, A., Veciana, G. De, & Shakkottai, S. (2018). Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 1970–1978.

Bhatia, R., Gupta, B., Benno, S., Esteban, J., Samardzija, D., Tavares, M., & Lakshman, T. V. (2018). Massive Machine Type Communications over 5G using Lean Protocols and Edge Proxies. *IEEE 5G World Forum, 5GWF 2018 - Conference Proceedings*, 462–467. https://doi.org/10.1109/5GWF.2018.8517086

Chin, Y. K., Kow, W. Y., Khong, W. L., Tan, M. K., Tze, K., & Teo, K. (2020). Q-Learning Traffic Signal Optimization within Multiple Intersections Traffic Network Q-Learning Traffic Signal Optimization within Multiple Intersections Traffic Network. *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, November 2012*, 343–348. https://doi.org/10.1109/EMS.2012.75

Coms, I. (n.d.). *Towards 5G : A Reinforcement Learning-based Scheduling Solution for Data Traffic Management*. 1–14.

Devraj, A. M., & Meyn, S. P. (2017). Zap Q-learning. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 2236–2245.

*Difference between 3G, 4G and 5G? | About Verizon*. (n.d.). Retrieved September 7, 2021, from https://www.verizon.com/about/our-company/5g/difference-between-3g-4g-5g

Elsayed, M., & Erol-kantarci, M. (2019). AI-Enabled Radio Resource Allocation in 5G for URLLC and eMBB Users. *2019 IEEE 2nd 5G World Forum (5GWF)*, 590–595.

Elwin, P., & Hirst, R. (2007). Guide to key performance indicators. *PriceWaterhouseCoopers*, 1–28. http://www.pwc.com/gx/en/corporate-reporting/assets/pdfs/UK_KPI_guide.pdf

Esswie, A. A., Pedersen, K. I., & Member, S. (n.d.). *Multi-User Preemptive Scheduling For Critical Low Latency Communications in 5G Networks*.

Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2019). *A Theoretical Analysis of Deep Q-Learning*. *120*(1995), 1–4. http://arxiv.org/abs/1901.00137

François-lavet, V., Henderson, P., Islam, R., Bellemare, M. G., François-lavet, V., Pineau, J., & Bellemare, M. G. (2018). An Introduction to Deep Reinforcement Learning. (arXiv:1811.12560v1 [cs.LG]) http://arxiv.org/abs/1811.12560. *Foundations and Trends in Machine Learning*, *II*(3–4), 1–140. https://doi.org/10.1561/2200000071.Vincent

fschaich. (2017). *Call:H2020-ICT-2016-2 Project reference: 760809 Project Name: E2E-aware Optimizations and* *advancements for Network Edge of 5G New Radio (ONE5G) Deliverable D2.1 Scenarios, KPIs, use cases and baseline system evaluation*. https://one5g.eu/wp-content/uploads/2017/12/ONE5G_D2.1_finalversion.pdf

Hasselt, H. Van, Guez, A., & Silver, D. (2016). Double DQN.pdf. *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 2094–2100.

*Home - 5G-ACIA*. (n.d.). Retrieved September 7, 2021, from https://5g-acia.org/

Jwo, J. S., Lin, C. S., Lee, C. H., & Lo, Y. C. (2020). An empirical investigation of transfer effects for reinforcement learning. *Computational Intelligence and Neuroscience*, *2020*. https://doi.org/10.1155/2020/8873057

Karamyshev, A., Khorov, E., Krasilov, A., & Akyildiz, I. F. (2020). Fast and accurate analytical tools to estimate network capacity for URLLC in 5G systems. *Computer Networks*, *178*(May). https://doi.org/10.1016/j.comnet.2020.107331

Karimi, A., Pedersen, K. I., Mahmood, N. H., Steiner, J., & Mogensen, P. (2018). Centralized Joint Cell Selection and Scheduling for Improved URLLC Performance. *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 1–6.

Ma, T., Zhang, Y., Wang, F., Wang, D., & Guo, D. (2020). Slicing Resource Allocation for eMBB and URLLC in 5G RAN. *Wireless Communications and Mobile Computing*, *2020*. https://doi.org/10.1155/2020/6290375

Nassar, A., & Yilmaz, Y. (2019). Dynamic network slicing for fog radio access networks. *GlobalSIP 2019 - 7th IEEE Global Conference on Signal and Information Processing, Proceedings*, 1–32. https://doi.org/10.1109/GlobalSIP45357.2019.8969455

Ohnishi, S., Uchibe, E., Yamaguchi, Y., Nakanishi, K., Yasui, Y., & Ishii, S. (2019). Constrained deep Q-learning gradually approaching ordinary Q-learning. *Frontiers in Neurorobotics*, *13*(December), 1–19. https://doi.org/10.3389/fnbot.2019.00103

Pandey, S. R., Alsenwi, M., Kyaw, Y., & Hong, C. S. (2019). *A Downlink Resource Scheduling Strategy for URLLC Traffic*. *April*. https://doi.org/10.1109/BIGCOMP.2019.8679266

Pocovi, G., Pedersen, K. I., Member, S., & Mogensen, P. (2018). *Joint Link Adaptation and Scheduling for 5G Ultra-Reliable Low-Latency Communications*. 6.

Pocovi, G., Pedersen, K. I., & Mogensen, P. (2018). Joint Link Adaptation and Scheduling for 5G Ultra-Reliable Low-Latency Communications. *IEEE Access*, *6*, 28912–28922. https://doi.org/10.1109/ACCESS.2018.2838585

Probes, A. F. M. (n.d.). *Applications — Table of Contents*.

Samarakoon, S., Bennis, M., Member, S., & Saad, W. (n.d.). *Distributed Federated Learning for Ultra-Reliable Low-Latency Vehicular Communications*. 1–14.

Sanou, B. (2018). Setting the scene for 5G: Opportunities & Challenges. In *Itu*. https://doi.org/http://handle.itu.int/11.1002/pub/811d7a5f-en

Siddiqi, M. A., Yu, H., & Joung, J. (2019). 5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices. *Electronics (Switzerland)*, *8*(9), 1–18. https://doi.org/10.3390/electronics8090981

Siddiqi, Yu, Joung, J., Siddiqi, M. A., Yu, H., & Joung, J. (2019). 2019 5G Ultra-Reliable Low-Latency Communication.pdf. *Electronics*, *8*(9), 981. https://www.mdpi.com/2079-9292/8/9/981

Soret, B., Mogensen, P., Pedersen, K. I., & Aguayo-torres, M. C. (2014). *Fundamental Tradeoffs among Reliability , Latency and Throughput in Cellular Networks*. 1391–1396.

Specification, T. (2017). *TS 136 104 - V14.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception (3GPP TS 36.104 version 14.3.0 Release 14)*. *0*.

Specification, T. (2018). *TS 138 214 - V15.2.0 - 5G; NR; Physical layer procedures for data (3GPP TS 38.214 version 15.2.0 Release 15)*. *0*, 0–94.

*Three Use Cases from the 5G Future*. (n.d.). Retrieved September 7, 2021, from https://www.aeris.com/news/post/three-use-cases-from-the-5g-future/

Vt, W., Engineering, C., & Tech, V. (n.d.). *Federated Learning for Ultra-Reliable Low-Latency V2V*

*Communications*.

Wang, G., Feng, G., Qin, S., Wen, R., & Sun, S. (2019). Optimizing Network Slice Dimensioning via Resource Pricing. *IEEE Access*, *7*(March), 30331–30343. https://doi.org/10.1109/ACCESS.2019.2902432

Wang, G., Feng, G., Tan, W., Qin, S., Wen, R., & Sun, S. (2017). *Resource Allocation for Network Slices in 5G with Network Resource Pricing*. *December*. https://doi.org/10.1109/GLOCOM.2017.8254074