



HAL
open science

Optimal multi-connectivity strategies for delay-sensitive industrial IoT traffic

Mariam El, Salah Eddine Elayoubi, Antonia Maria Masucci

► To cite this version:

Mariam El, Salah Eddine Elayoubi, Antonia Maria Masucci. Optimal multi-connectivity strategies for delay-sensitive industrial IoT traffic. IEEE Globecom 2024, Dec 2024, Cape Town, South Africa. hal-04688847

HAL Id: hal-04688847

<https://inria.hal.science/hal-04688847v1>

Submitted on 5 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Optimal multi-connectivity strategies for delay-sensitive industrial IoT traffic

Mariam El Hassan*, Salah Eddine Elayoubi[†], Antonia Maria Masucci*

* Orange, Châtillon, France

[†] CentraleSupélec, Université Paris-Saclay, CNRS, L2S, Gif-sur-Yvette, France

Abstract—In this paper, we tackle the problem of multi-connectivity for latency critical industrial IoT services in 5G and Beyond. Sending multiple copies of the same packet on different radio interfaces is a popular mean for reducing delay. However, systematic redundancy may lead to congestion and more dynamic scheduling schemes such as Join the Shortest Queue (JSQ) are not guaranteed to be optimal. In order to derive the optimal multi-connectivity scheme, we first develop a novel discrete-time queuing model for the system, that integrates the heterogeneity of radio conditions and can consider any multi-connectivity policy. We then formulate an associated Markov Decision Process (MDP) that learns, online, the optimal state/action combination. Our simulations demonstrate that the derived optimal scheme outperforms classical ones like systematic redundancy and JSQ and is able to adapt dynamically to different traffic conditions. Interestingly, our proposed scheme yields a clear advantage when the objective is to ensure a hard latency constraint compared with the case where the average delay is to be minimized.

I. INTRODUCTION

Industrial IoT (IIoT) service in 5G networks and beyond presents challenging latency requirements. Ultra-Reliable and Low Latency Communications (URLLC) service is expected to be a pillar for IIoT and, among its enabling technologies, multi-connectivity is a promising technique to improve reliability and latency [1], [2]. In this context, redundant scheduling over several available resources is a practical way for reducing queuing delays by dynamically selecting the base station with the smallest instantaneous load or replicating the packet on several base stations. In practice, a redundant coverage is ensured in almost all locations, especially in dense areas [3].

The literature on multi-connectivity and scheduling redundancy mainly focuses on cloud computing applications. For instance, Join-the-Shortest-Queue (JSQ) model corresponds to selecting the server that has the lowest number of packets waiting in his queue [4], [5]. In the context of 5G networks, [6] studied the performance of JSQ scheme compared to systematic redundancy, and showed that, while JSQ has a better performance in medium load scenarios, systematic redundancy is better when load is low. Similar results have been obtained in [7]. [8] proposed a dual connectivity control solution where packet duplication can be dynamically controlled to cope with low SINR situations. [9] evaluated by extensive system level simulations the advantage of dropping the yet unscheduled packet replicas once one of the replicas is received.

When dealing with policy optimization, literature focuses on devising scheduling policies in multi Radio Access Technologies (RAT) networks. For instance, [10] controls URLLC

packet duplication in Heterogeneous Network (hetNet) with dual connectivity. [11] studies dual connectivity based on various system parameters. [12] and [13] propose a smart RAT access strategy in a HetNet coexisting cellular and WiFi.

We propose in this paper a framework for deriving the optimal multi-connectivity policy for IIoT systems served by 5G networks. Our contributions are twofold.

First, we develop a novel discrete-time queuing model that integrates the specificity of the 5G radio interface, the heterogeneous radio conditions, and the stochastic traffic profiles. Most of the models of the literature consider continuous-time queuing models that are not adequate for IIoT traffic and the time-slotted nature of 5G networks. For instance, an $M/GI/\infty$ model is used in [14], a $M/M/m/K$ model in [15] and an $M/G/1$ model in [16]. The work in [17] proposed a discrete-time model for 5G networks, but it was limited to one cell and does not consider multi-connectivity. When it comes to queuing models for multi-connectivity, besides from works that are specific to cloud systems ([4], [5]), only [6] proposes a model for multi-connectivity in 5G, that is yet based on an $M/G/1$ model and does not consider the radio interface parameters. To the best of our knowledge, our work is the first that proposes a discrete-time queuing model for multi-connectivity of IIoT traffic in 5G systems. Our model computes, for each policy, the outage probability, i.e. the probability that the packet is not received within the delay budget.

Second, we propose an online policy optimization framework that implements our discrete-time model in the form of a Markov Decision Process (MDP) that associates to each state of the system the optimal scheduling decision (duplicate or not the packet and on which interface). We propose two flavours of the solution: one that aims at minimizing the average packet delay, and another flavour that aims at minimizing the outage. Our results show that the former does not yield significant gains compared to static JSQ schemes, while the latter has a clear performance advantage. This result highlights the importance of a dynamic optimization of packet duplication in the framework of IIoT traffic with a strict delay constraint.

The remainder of this paper is organized as follows. Section II develops the queuing model for the system and derives the outage probability for different multi-connectivity policies. Section III formulates an MDP, based on the queuing model, that finds the optimal policy. Section IV presents numerical results and compares the optimal policy to the state of the art multi-connectivity policies. Section V concludes this work.

II. QUEUING MODEL

We consider a set of users served by n cells, and generating a delay-sensitive traffic. An agent connected to all cells is responsible for dispatching/duplicating the packets to the n cells, following some policy that we will optimize next.

A. Radio and traffic models

For each cell, the packets are served within a 5G OFDM resources grid, organized into Physical Resource Blocks (PRB) of w Hz and minislot duration δ ms (that might be as small as 2 OFDM symbols). Packets arrive to the system following some stochastic process and have to be served within d minislots, otherwise they are considered in outage. Even if packets for URLLC are usually of equal, small size, different packets may occupy different amounts of resources, as heterogeneous radio conditions between a base station and its users may result in different Channel Quality Indicators (CQI), reflected by the corresponding Modulation and Coding Scheme (MCS).

Due to the centralized location of the entity and the stringent delay requirements, the agent does not know the instantaneous CQI, but has the following limited information:

- 1) The distribution of the radio conditions, based on statistics collected from the base stations. Let the J MCS be sorted in ascending order of efficiency, and e_j be the efficiency of MCS i , with $e_j \in [\underline{e} = e_1, \bar{e} = e_J]$. The probability of having MCS j , for cell i , is given by $\beta_j^{(i)}$.
- 2) The traffic statistics, with the distributions of the number of packets generated per slot. Let V_t be the number of packets arriving during slot t ; V_t is distributed following the discrete law $g_v = Pr(V_t = v)$, defined on $[0, \bar{V}]$. Each packet is of size b bits.
- 3) The number of packets waiting in each queue. Let $N_t^{(i)}$ be the number of packets at queue i at time slot t .

Let $M^{(i)}$ be the amount of PRBs available at cell i . We now derive the distribution of the number of packets, $L_t^{(i)}$, that can be served by cell i at a given time slot. If the radio conditions of all users were homogeneous, $L_t^{(i)}$ would be constant and equal to the number of available RBs ($M^{(i)}$), divided by the number of RBs needed for serving one packet. However, the spectral efficiency of each user depends on its instantaneous channel realization. For a given packet, numbered k , that is served using MCS m_k , the amount of PRBs it requires is equal to

$$r_k = \frac{b}{e_{m_k} \delta w}. \quad (1)$$

The distribution of r_k depends on the distribution of the radio conditions. The probability that a packet, served by cell i , needs r PRBs is

$$\alpha_r^{(i)} = \sum_{j=1}^J \beta_j^{(i)} \mathbb{I}_{\lceil \frac{b}{e_j \delta w} \rceil}, \quad r \in \left[\lceil \frac{b}{\bar{e} \delta w} \rceil, \lceil \frac{b}{\underline{e} \delta w} \rceil \right] \quad (2)$$

with $\lceil x \rceil$ the largest integer larger or equal to x and \mathbb{I}_C is the indicator function equal to 1 if condition C is verified and to 0 otherwise.

Knowing these probabilities, we compute the distribution of the number of packets, $L^{(i)}$ that can be served, at a given time, by the $M^{(i)}$ available resources:

$$Pr(L^{(i)} \geq l) = Pr\left(\sum_{k=1}^l r_k^{(i)} \leq M^{(i)}\right) \quad (3)$$

where the $r_k^{(i)}$ are i.i.d. variables distributed following (2).

The distribution of $L^{(i)}$ can be derived numerically, or by using the probability generating function (PGF) of $x_l = \sum_{k=1}^l r_k^{(i)}$, exploiting the property $G_{x_l}(z) = (G_{r^{(i)}}(z))^l$, where $G_y(z)$ is the PGF of discrete variable y . In the following, we will use $\eta_l^{(i)}$ as the probability that cell i is able to serve l packets at a given time slot:

$$\eta_l^{(i)} = Pr(L^{(i)} = l), \quad l \in \left[1, \bar{L} = \left\lceil \frac{M^{(i)}}{\lceil \frac{b}{\bar{e} \delta w} \rceil} \right\rceil \right] \quad (4)$$

B. Queue dynamics for a given policy

When a packet arrives to the system, it may be duplicated and then dispatched to one or multiple cells, based on the observed system state, and the number of new arriving packets. The system state at time t , $\mathbf{S}_t = (N_t^{(i)}) \in \mathcal{S} = \mathbb{N}^n$, is defined as the number of packets awaiting in each of the cell queues. We now consider a given policy, defined by a vector $\pi = (\pi(\mathbf{S}, V), \mathbf{S} \in \mathcal{S}, V \leq \bar{V})$. π associates to each state in \mathcal{S} an action from \mathcal{A} , the power set of $\{1, 2, \dots, n\}$. In other words, if $i \in \pi(\mathbf{S}, V)$, the received V packets are sent to queue i , and each of these packets is duplicated $|\pi(\mathbf{S}, V)|$ times, $|\cdot|$ denoting the cardinality of a set.

The queue lengths, under policy π , evolve as:

$$N_{t+1}^{(i)} = \left(N_t^{(i)} + V_t \mathbb{I}_{i \in \pi(\mathbf{S}_t, V_t)} - L_t^{(i)} \right)^+, \quad (5)$$

where $(x)^+ = \max(0, x)$. V_t is the number of packets arriving during slot t , $N_t^{(i)}$ is the the backlogged packets at queue i from previous slots. If the action corresponding to the system state at time t , under event V_t , is to use cell i , the number of packets (new and backlogged) will be equal to $N_t^{(i)} + V_t$; otherwise, it will stay equal to $N_t^{(i)}$. The number of packets that can be served during the slot is equal to $L^{(i)}(t)$, a random variable that is distributed following (4) and the remaining packets after the service stay in the queue until the next slot.

C. Steady-state for a given policy

Under a given policy π , the system evolves following a Markov chain, and the transition matrix can be built based on equation (5), using the distributions of the number of generated packets g_v and the service distribution (4).

In steady-state, let $\mathbf{p}(\pi)$ be the vector grouping the probabilities of individual states under policy π , denoted by $p(\mathbf{S}, \pi)$. We write the following set of linear equations:

$$\begin{cases} \mathbf{p}(\pi) = \mathbf{p}(\pi) \mathbf{P}(\pi) \\ \mathbf{p} \mathbf{p}' = 1 \end{cases} \quad (6)$$

where $\mathbf{P}(\pi)$ is the transition matrix and $'$ is the transpose operator. For defining \mathbf{P} , we first describe the behaviour of

an individual queue under the policy. The probability that the length of queue i moves from N to K in one slot is:

$$P_{N,K}^{(i)}(\pi) = \begin{cases} \sum_{l>(N-K)+\eta_l^{(i)}} g_{K+l-N} \mathbb{I}_{i \in \pi(\mathbf{S}_t, K+l-N)}, & K > 0 \\ \sum_{l \geq N} \sum_{v \leq l-N} g_v \mathbb{I}_{i \in \pi(\mathbf{S}_t, v)}, & \text{otherwise} \end{cases} \quad (7)$$

The global transition probabilities can thus be computed as the products of the individual queues' transition probabilities:

$$P_{(N^{(1)}, \dots, N^{(n)}), (K^{(1)}, \dots, K^{(n)})}(\pi) = \prod_{i=1}^n P_{N^{(i)}, K^{(i)}}^{(i)}(\pi) \quad (8)$$

D. Performance evaluation

The traffic being delay sensitive, the main performance metric is the delay outage, expressed as the probability that the packets cannot be served within the delay budget, expressed in a maximal number of slots d .

We start by computing the outage for a given queue. Let $N^{(i)}$ be the number of packets in queue i at a given slot. The outage probability knowing $N^{(i)}$ is the probability that d slots are not sufficient to serve all the packets:

$$o^{(i)}(N^{(i)}) = Pr \left[\sum_{k=1}^{N^{(i)}} r_k^{(i)} > dM^{(i)} \right] \quad (9)$$

This is computed based on the distribution of $r_k^{(i)}$, as for the distribution of the number of served packets (4).

The outage probability for a given policy is then:

$$o(\pi) = \sum_{\mathbf{S}=(N^{(i)}) \in \mathcal{S}} p(\mathbf{S}, \pi) \sum_{v=0}^{\bar{V}} g_v \prod_{i \in \pi(\mathbf{S}, v)} o^{(i)}(N^{(i)} + v) \quad (10)$$

The optimal policy is the one that minimizes the outage:

$$\pi^* = \arg \min_{\pi} [o(\pi)] \quad (11)$$

III. ONLINE OPTIMAL CONTROL

The above performance model computes the outage for any multi-connectivity policy and allows finding the optimal one. However, in an online setting, the network operator needs to adjust its policy in a dynamic way until converging to the optimal one. Inspired from the discrete queuing problems in [18, 3.7], the system is modeled as a MDP, that is a natural choice when the system model is known.

A. Reward definition

Every transition from a state \mathbf{S}_t to a state \mathbf{S}_{t+1} , due to an action a_t leads to a reward $R_t = R(\mathbf{S}_t, a_t, \mathbf{S}_{t+1})$, that maps $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ to numerical rewards. It reflects how advantageous is this transition at the short-term, and is related to the packet delay. The total discounted gain G_t to be maximized [19] is a measure of the long-term value of a sequence of rewards:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}, \quad (12)$$

where γ is the discount factor, that reflects how much the system is far-sighted.

The definition of the reward will depend on the performance metric of interest. Even if the outage is the main metric for URLLC, we also consider in this paper a reward related to the average delay, for comparison purposes:

1) *Reward related to mean service time*: A classical metric is the average packet service time T_{avg} that is the mean time between the instant where the packet arrives at the scheduler queue and the instant it is served. The reward is then defined as the negative mean service time.

2) *Reward related to outage probability, as defined in section II-D*: It is computed by:

$$R(N^{(1)}, \dots, N^{(n)}, a) = - \sum_{v=0}^{\bar{V}} g_v \prod_{i \in \pi(\mathbf{S}, v)} o^{(i)}(N^{(i)} + v), \quad (13)$$

where $o^{(i)}(l)$ is the outage of cell i when it has l packets.

B. The value iteration algorithm

As we have derived the system dynamics, i.e. the transition probability between states and the corresponding expected reward, it is possible to use Dynamic Programming (DP) algorithms in order to derive the optimal policy. We make use of the Value Iteration algorithm, known for its ability to handle large state spaces efficiently.

1) *Bellman Equations*: Each state-action couple is evaluated by a scalar quantity that tells how valuable it is to take a particular action a from the state \mathbf{S} , when following a policy π . This state-action value function, $Q_{\pi}(\mathbf{S}, a)$, represents the expected total discounted gain that the agent may get after taking the particular action a from the state \mathbf{S} onward:

$$Q_{\pi}(\mathbf{S}, a) = E_{\pi}[G_t | \mathbf{S}_t = \mathbf{S}, a_t = a] \quad (14)$$

From this definition of the value function, and using the recursive property $G_t = R_t + \gamma G_{t+1}$, it is possible to write Bellman equations [19]. In the policy evaluation step, Bellman equations calculate the Q values of all state-action pairs. The second step is to improve the policy by selecting the action that maximises the Q value at each state. At convergence, the value function is expressed as:

$$Q^*(\mathbf{S}, a) = \sum_{\mathbf{S}' \in \mathcal{S}} Pr(\mathbf{S}' | \mathbf{S}, a) \left[R(\mathbf{S}, a, \mathbf{S}') + \gamma \max_{a' \in \mathcal{A}} Q_*(\mathbf{S}', a') \right] \quad (15)$$

where $Pr(\mathbf{S}' | \mathbf{S}, a)$ is the probability of transition from state \mathbf{S} to state \mathbf{S}' after action a , thus getting reward $R(\mathbf{S}, a, \mathbf{S}')$.

2) *Optimal policy*: The Bellman equations in (15) present a system of $|\mathcal{S}| \times |\mathcal{A}|$ unknowns, corresponding to the Q values of every state-action pair. This system of equations permits to find the value function of the optimal policy by iteratively updating the value of each state-action pair until the state-action values converge to their optimal values, based on a convergence threshold ϵ . Then, it is possible to define the optimal policy as:

$$\pi^*(a | \mathbf{S}) = \begin{cases} \frac{1}{|\mathcal{A}_{max}(\mathbf{S})|} & \text{if } a \in \mathcal{A}_{max}(\mathbf{S}) \\ 0 & \text{otherwise} \end{cases}, \quad (16)$$

$\forall (\mathbf{S}, a) \in \mathcal{S} \times \mathcal{A}$, where $\mathcal{A}_{max}(\mathbf{S}) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q(\mathbf{S}, a)$ is the set of actions that maximize the Q function. The following algorithm summarizes the value iteration process:

Algorithm 1: Value Iteration Algorithm

Data: Initialize $Q(\mathbf{S}, a) = 0 \forall (\mathbf{S}, a) \in \mathcal{S} \times \mathcal{A}$

Result: $Q^*(\mathbf{S}, a) \forall (\mathbf{S}, a) \in \mathcal{S} \times \mathcal{A}$ and policy π^*

```

1 repeat
2   for each  $(\mathbf{S}, a) \in \mathcal{S} \times \mathcal{A}$  do
3      $Q_{prev}(\mathbf{S}, a) \leftarrow Q(\mathbf{S}, a)$ ;
4     update  $Q(\mathbf{S}, a)$  as in (15)
5   end
6 until  $\sum_{\mathcal{S} \times \mathcal{A}} |Q_{prev}(\mathbf{S}, a) - Q(\mathbf{S}, a)| < \epsilon$ ;
7 for each  $(\mathbf{S}, a) \in \mathcal{S} \times \mathcal{A}$  do
8   update  $\pi^*(a|\mathbf{S})$  as in (16)
9 end
10 return  $Q, \pi^*$ ;

```

IV. NUMERICAL EXPERIMENTS

In this section, we first evaluate the performance of different policies by the means of the queuing model developed in section II, and show that there is no classical scheme that fits all situations. We then implement the online policy optimization framework of section III on a 5G simulator.

A. Performance evaluation of selected policies

1) *Dual connectivity policies:* We start by the case of two cells, and implement the queuing model for three policies:

a) *JSQ:* this policy is popular in cloud computing systems [20] and in 5G systems [6]. The idea of JSQ is to use the server that has the shortest queue for minimizing delay. However, it does not have the guarantee of the lowest service time since packets have different service times depending on the corresponding CQIs, unknown to the traffic steering agent.

b) *RED:* The other approach (REDundancy) consists on systematically sending the packets on both RATs. It thus ensures, on one packet level, to be served faster. However this -packet selfish- policy adds load to all queues.

c) *Hybrid policy:* Our queuing model can be applied to any multi-connectivity policy. We thus also consider a hybrid policy, that acts as RED if the queue lengths of both cells are close (difference no more than c packets), and applies JSQ otherwise. The idea behind this policy is that, when the queue lengths are very close, the uncertainty about the packet resource consumption may dominate the effect of queue length. However, for a large difference between the queues, it is safer to use the shortest one.

We consider a 5G system with a subcarrier spacing $f = 15kHz$, and a 2-symbol time slot. URLLC packets arrive following a Poisson process and are of size 32 bytes. We consider two cases: the symmetric case, where both cells have the same amount of spectrum dedicated to URLLC service (35 PRBs each), and asymmetric case, where a cell has 40 PRBs and the other 30 PRBs dedicated to URLLC. We consider an

MCS distribution that is issued from the system level simulator described in the next section.

Figure 1 compares the performances of the three policies, for different arrival rates of packets. We observe that, in general, RED outperforms JSQ for low loads, while JSQ is better for higher loads. Here the traffic load is defined as the ratio of the average number of new packets per slot, to the sum of the average number of served packets in each cell per slot. We also implemented the hybrid policy for two values of c ($c = 2$ and $c = 5$). We observe that it has an intermediate performance between JSQ and RED, and that it is close to JSQ for large c . The third observation is that, even if RED is better than JSQ, the flipping point depends on the configuration.

These results call for an online optimization framework that derives the optimal policy for different situations, and where the policy may change with the state and the arrival events.

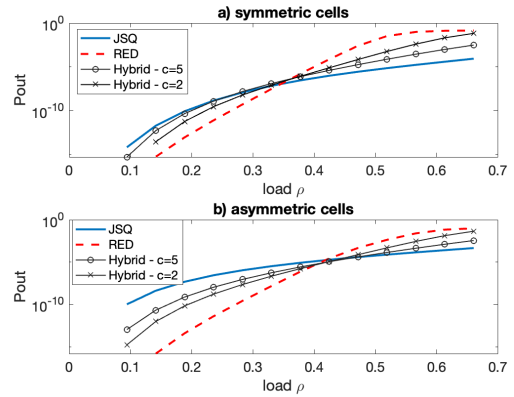


Fig. 1: Comparison of dual-connectivity policies.

2) *Extension to multi-connectivity:* While dual connectivity is the most common case, multi-connectivity can arise in some situations. We illustrate in Figure 2 the case of three cells covering the same area. In addition to the JSQ and RED policies, we consider the ALQ (for Avoid the Longest Queue) policy, where the new packets are sent to 2 queues among the three, avoiding the most loaded one. We observe that RED has the worst performance, even for low loads, as the traffic of the three cells is served by each of the cells, and ALQ has an intermediate performance between JSQ and RED.

B. Optimal policy derivation

The previous analysis showed that the optimal policy depends on the traffic load and on the system configuration, and there no one-fits-all optimal policy. An online optimization of the policy is then needed, as has been proposed in section III.

1) *Simulator description:* The online optimization framework is implemented on a system level simulator. The 5G NR AMC table in [21, Table 5.2.2.1-4] is followed. In order to ensure the robustness, only CQI indices from 1 to 5 are considered. The amount of spectrum dedicated to URLLC in each cell is 25 MHz.

The simulator operates as follows. Packets arrive to the traffic steering agent following a Poisson process. When the

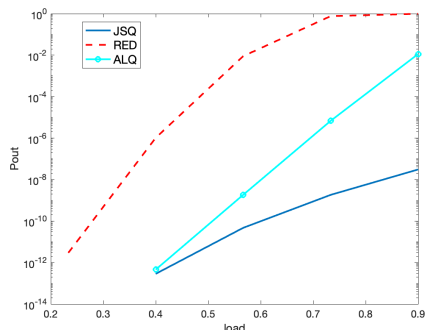


Fig. 2: Policies performance for three cells case.

agent allocates a packet to a cell, it is allocated an MCS following its CQI with respect to the considered cell. Scheduling is then performed following the First Come First Serve (FCFS) discipline. In each time slot, the spectral resources are filled by the packets at the head of the queue, and the unscheduled packets wait for the next time slot. If a packet is lost due to bad radio conditions, despite the robust MCS, it is queued again and retransmitted until it is correctly decoded. For clarity purposes, we illustrate the case of two symmetrical cells (same amount of spectrum, independent radio conditions).

2) Policy illustration:

a) Optimal policy for a reward related to the average delay: We now illustrate the optimal policy obtained by the value iteration algorithm, when the objective is to minimize the mean delay. Fig. 3 (a) and (b) show the obtained policy for a discount factor $\gamma = 0$ and 0.7 respectively. We note that the obtained policy have shown to be the same whatever the value of V_t is, as it is the case for JSQ and RED.

First, for $\gamma = 0$, the Q values are updated based only on the current received reward, without an impact of the future state-action pairs (see eq. (15)). That is, the value of each state-action pair corresponds to the negative waiting time obtained for the corresponding action. Thus, taking the action $A_1 = \{1\}$ will lead to a gain equal to the negative waiting time at the first BS, $-N^{(1)} \times T_{avg}$, the action $A_2 = \{1\}$ leads to $-N^{(2)} \times T_{avg}$ and $A_3 = \{1, 2\}$ to $-\min(N^{(1)}, N^{(2)}) \times T_{avg}$. This explains the so-obtained decision policy for $\gamma = 0$ in Fig. 3 (a). The policy, that maximizes the total discounted gain, considers equally selecting: A_1 and A_3 above the diagonal line, A_2 and A_3 under the diagonal line, A_1, A_2 and A_3 on the diagonal line. This is because the agent with $\gamma = 0$ has a very limited vision that only considers the current reward, which has the same value with redundancy, or without redundancy while selecting the RAT with the minimum number of packets.

As the value of γ increases, the impact of the excess load by applying RED starts to appear. The impact of being far-sighted can be seen from Fig. 3 (b), where the agent prefers JSQ over redundancy. The value iteration algorithm, when considering the mean service time reward, converges thus to JSQ.

b) Optimal policy for a reward related to the outage probability: Fig. 4 shows the obtained optimal policy for

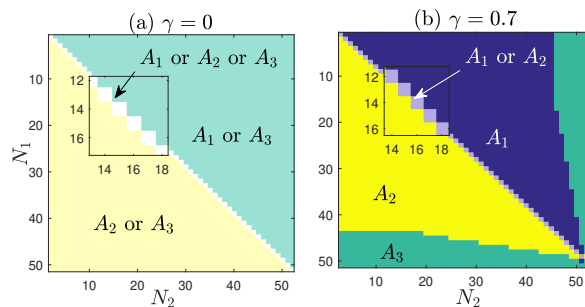


Fig. 3: Optimal policy (reward related to mean service time).

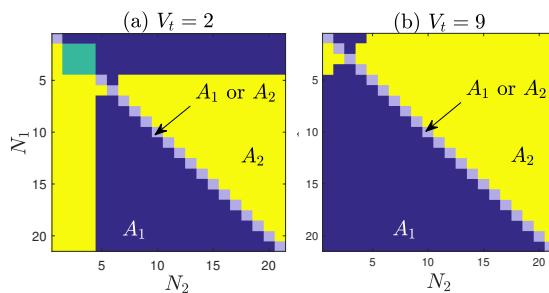


Fig. 4: Optimal policy with reward related to outage.

$V_t = 2$ and $V_t = 9$ arrivals, respectively, and with $\gamma = 0.7$, selected to ensure enough far sight. We first note that the optimal policy depends on V_t here. At low loads (small $N^{(i)}$), the behaviour is close to JSQ. However, when the load increases, the behaviour may seem counter-intuitive as the queue with largest $N^{(i)}$ is chosen. Indeed, when the system is already loaded, outage is inevitable for the new packets, and these packets are sent to the more loaded RAT, in order to preserve the capacity of the other queue for future arrivals, leading to an implicit admission control.

C. Performances evaluation and comparison

Fig. 5 shows the performances of the obtained optimal policy π^* , that correspond the reward related to outage. These performances are depicted in terms of (a) average delay (in slots) and (b) outage probability (delay exceeds the target of 1ms), for different values of the system load ρ . For simplification, the mean service time is shown in numbers of service time slots. The performances of the baseline policies, RED and JSQ are shown as well for comparison and analysis. It is worth noting that the optimal policy obtained by using the mean service time related reward gives the same performances of JSQ, as concluded in the policy illustration analysis.

1) Performances in terms of mean service time: As expected, at higher loads (Fig. 5 (c), $\rho > 0.1$), the JSQ policy outperforms RED in terms of mean packet service time, since the overall system load is lower and the mean service time is thus lower. However, this doesn't apply at low system load (Fig. 5 (a), $\rho \leq 0.1$). This is explained by the fact that, using RED at such loads, permits to serve every packet on the cell with the best radio conditions. Though packets waiting

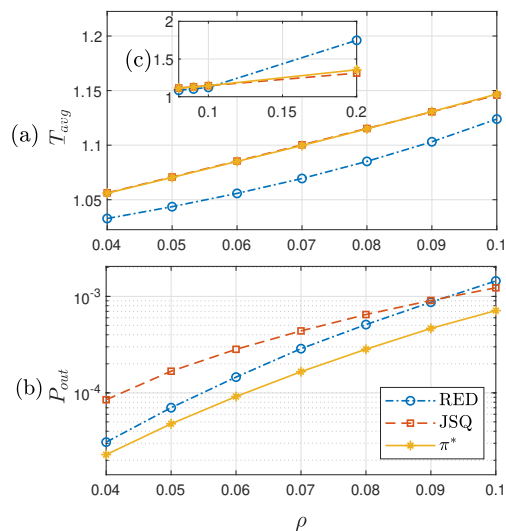


Fig. 5: Performances of the optimal and baseline policies for a reward related to outage probability

time might be slightly longer due to the added load, this is compensated at relatively low load.

For the optimal policy π^* , it can be seen that its performances are nearly the same as the JSQ scheme at low load, where the added service time is due to the RAT conditions. However, with added load, the performances of π^* start to degrade due to the added waiting times, since packets are not always sent to the RAT with the minimum expected waiting time. These performances remain better than the RED policy that adds more waiting time due to the duplicated load.

2) *Performances in terms of outage probability:* For the outage probability, as can be seen, for a load higher than 0.09, the outage is more probable for RED than JSQ. This is because RED duplicates packets and thus doubles the system load. However, at lower system loads, RED is more beneficial. As remarked before, when analysing mean service times, the majority of packets is directly served and RED doesn't impact this relatively small load. In that case, each packet ensures to be served on the channel where it has the higher efficiency and thus the lower service time.

Interesting performances of the optimal policy appear in this figure, as it is shown to minimize the outage for all loads lower than 0.1, which are the values convenient for URLLC service, i.e. corresponding to outage lower than 10^{-3} .

V. CONCLUSION

In this paper, we have derived the optimal traffic allocation strategy that exploits multi-connectivity for latency-critical services in 5G networks. We modeled the system dynamics as a discrete-time queuing system, controlled by the traffic steering policy, and provided a novel queuing model for computing the outage probability. We then formulated an MDP, with a reward that depends on the objective of the operator (minimize the average delay or the outage probability). We derived the optimal policy in both cases, and compared the obtained

performance with classical multi-connectivity schemes, such as JSQ and systematic replication. Our numerical results show that the resulting scheme converges to the classical JSQ when the objective is to minimize the average packet delay. However, if a URLLC-related metric is considered, such as the delay outage, the proposed policy intelligently manages the bursts of packet arrivals and outperforms the classical schemes, be they static (systematic redundancy) or dynamic based on the queue length (JSQ).

REFERENCES

- [1] M.-T. Suer *et al.*, "Multi-connectivity as an enabler for reliable low latency communications—an overview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, 2019.
- [2] T. Sylla *et al.*, "Multi-connectivity for 5g networks and beyond: A survey," *Sensors*, vol. 22, no. 19, p. 7591, 2022.
- [3] P. Marsch *et al.*, "5G radio access network architecture: Design guidelines and key considerations," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 24–32, 2016.
- [4] L. Flatto and H. McKean, "Two queues in parallel," *Communications on Pure and Applied Mathematics*, no. 3/4, pp. 255–263, 1977.
- [5] G. Brightwell and M. Luczak, "The supermarket model with arrival rate tending to one," *arXiv preprint arXiv:1201.5523*, 2012.
- [6] A. Chagdali *et al.*, "Performance of urllc traffic scheduling policies with redundancy," in *2020 32nd International Teletraffic Congress (ITC 32)*. IEEE, 2020, pp. 55–63.
- [7] N. H. Mahmood *et al.*, "Reliability oriented dual connectivity for urllc services in 5g new radio," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.
- [8] J. Rao and S. Vrzic, "Packet duplication for urllc in 5g dual connectivity architecture," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.
- [9] M. Centenaro *et al.*, "Resource-efficient dual connectivity for ultra-reliable low-latency communication," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.
- [10] Q. Zhao *et al.*, "Hierarchical multi-objective deep reinforcement learning for packet duplication in multi-connectivity for urllc," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2021, pp. 142–147.
- [11] H. Lee, S. Vahid, and K. Moessner, "Machine learning based rats selection supporting multi-connectivity for reliability," in *International Conference on Cognitive Radio Oriented Wireless Networks*. Springer, 2019, pp. 31–41.
- [12] M. Yan *et al.*, "Smart multi-rat access based on multiagent reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4539–4551, 2018.
- [13] —, "Intelligent resource scheduling for 5g radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, 2019.
- [14] A. Anand and G. de Veciana, "Resource allocation and harq optimization for urllc traffic in 5g wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2411–2421, 2018.
- [15] C.-P. Li *et al.*, "5g ultra-reliable and low-latency systems design," in *2017 European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–5.
- [16] B. Shi *et al.*, "Risk-resistant resource allocation for embb and urllc coexistence under m/g/1 queueing model," *IEEE Trans. on Vehicular Technology*, vol. 71, no. 6, 2022.
- [17] M. Abdullah *et al.*, "Performance modeling and dimensioning of latency-critical traffic in 5g networks," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 4307–4312.
- [18] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [19] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [20] E. Anton *et al.*, "Improving the performance of heterogeneous data centers through redundancy," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 3, pp. 1–29, 2020.
- [21] "5G; NR; Physical layer procedures for data, TS 38.214 version 16.5.0 Release 16," 3GPP, Tech. Rep., 04/2021.