UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CAIO BRUNO BEZERRA DE SOUZA

Dynamic Resource Allocation for URLLC and eMBB services in NFV-MEC 5G Networks

Recife

2023

CAIO BRUNO BEZERRA DE SOUZA

Dynamic Resource Allocation for URLLC and eMBB services in NFV-MEC 5G Networks

This dissertation has been submitted to the Postgraduate Program in Computer Science of the Informatics Center of the Federal University of Pernambuco as a partial requirement to obtain the Master Degree in Computer Science.

Main Research Field : Computer Networks and Distributed Systems

Advisor: Andson Marreiros Balieiro

Co-Advisor: Marcos Rocha de Moraes Falcão

Recife

2023

Catalogação na fonte
Bibliotecária Mônica Uchôa, CRB4-1010

S729d Souza, Caio Bruno Bezerra de
Dynamic resource allocation for URLLC and eMBB services in NFV-MEC 5G networks / Caio Bruno Bezerra de Souza. – 2023.
172 f.: il., fig., tab.

Orientador: Andson Marreiros Balieiro.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2023.
Inclui referências.

1. 5G. 2. URLLC. 3. eMBB. 4. MEC. 5. NFV. 6. Alocação de recursos. I. Balieiro, Andson Marreiros (Orientador). II. Título.

004 CDD (23. ed.) UFPE - CCEN 2024 – 002

Caio Bruno Bezerra de Souza

“Dynamic Resource Allocation for URLLC and eMBB services in NFV-MEC 5G Networks”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Redes de Computadores e Sistemas Distribuídos.

Aprovado em: 29/09/2023.

BANCA EXAMINADORA

_____

Prof. Dr. Jamilson Ramalho Dantas
Centro de Informática / UFPE

_____

Prof. Dr. José Neuman de Souza
Departamento de Computação / UFC

_____

Prof. Dr. Andson Marreiros Balieiro
Centro de Informática / UFPE
(**Orientador**)

I dedicate this work to my parents, Jose Severino de Souza and Maria Juzene Bezerra.

ACKNOWLEDGEMENTS

First, I would like to thank God for providing me with strength and wisdom throughout my life. I am grateful to my parents, Jose Souza and Maria Bezerra who have always encouraged and supported me at every moment of my life. I leave a special thanks to my advisor Dr. Andson Balieiro who dedicated countless hours to solve my doubts even with his little free time. I also thank my co-advisor Dr. Marcos Falcão who has always helped me since the beginning of this research project. I must also express my gratitude to professors Dr. José Neuman de Souza and Dr. Jamilson Ramalho Dantas for agreeing to evaluate this work. Finally, I would also like to thank Universidade Federal de Pernambuco for the high quality of the teaching offered.

ABSTRACT

The Fifth Generation of mobile networks (5G) seeks to support a diversity of applications categorized into three types: enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC), and Ultra Reliable Low Latency Communications (URLLC), being their coexistence a major challenge. Multi-access Edge Computing (MEC), Network Function Virtualization (NFV) and Network Slicing (NS) emerge as complementary paradigms that shall support both eMBB and URLLC by offering fine-grained on-demand distributed resources closer to the User Equipment (UE) with a shared utilization of physical infrastructure. In this work, we have addressed the combination of MEC, NFV, NS and dynamic virtual resource allocation in order to overcome the problem of resource dimensioning in the network edge core. Thus, we have designed an analytical model to evaluate how requests are managed by the virtualization resources of a single MEC node, with a primary focus on meeting the requirements of both eMBB and URLLC services. We proposed a CTMC-based model to characterize dynamic virtual resource allocation and incorporated five performance metrics, which are relevant not only for URLLC and eMBB services (e.g., availability and response time) but also for service providers (e.g., power consumption), integrating practical factors like resource failures, service prioritization, and setup (repair) times into the formulation. This model enables an understanding of how the 5G network core behaves in serving different service categories by applying service prioritization to efficiently share processing resources. Some of our key findings include the idea that higher eMBB arrival rates decrease availability and increase response times up to 300 ms, while URLLC availability remains stable. Moreover, the container setup rates and failure rates substantially affect both availability and response times, with higher setup rates enhancing availability by up to 30% and reducing response times by 60%. Also, the number of containers emerges as a significant factor, enhancing both availability and response times, while buffer sizes mainly impact response times. In brief, our work advances in the current state of the art of the MEC-NFV domain by providing valuable insights for the design of MEC-NFV architecture, business models, and mechanisms to address communication constraints.

Keywords: 5G; URLLC; eMBB; MEC; NFV; resource allocation.

RESUMO

A Quinta Geração de redes móveis (5G) busca suportar diversas aplicações categorizadas em três tipos: largura de banda móvel melhorada (eMBB), comunicação do tipo máquina massiva (mMTC) e comunicação com baixa latência e confiabilidade muito alta (URLLC), em que a coexistência delas é um grande desafio. A computação de borda multiacesso (MEC), virtualização de funções de rede (NFV) e o fatiamento de rede (NS) surgem como paradigmas complementares para assistir tanto serviços eMBB quanto URLLC, oferecendo recursos distribuídos sob demanda e de maneira otimizada, mais próximos do equipamento do usuário (UE), com utilização compartilhada da infraestrutura física. Este trabalho explora a integração de MEC, NFV, NS e alocação dinâmica de recursos virtuais para endereçar o problema de dimensionamento na rede de borda. Para isso, utiliza-se um modelo analítico para avaliar como as solicitações são gerenciadas pelos recursos de virtualização em um único nó MEC, com ênfase nos requisitos dos serviços eMBB e URLLC. Um modelo baseado em CTMC foi proposto para caracterizar a alocação dinâmica de recursos virtuais e a derivaçao de cinco métricas de desempenho é realizada, as quais são relevantes não apenas para serviços URLLC e eMBB (e.g., disponibilidade e tempo de resposta), mas também para provedores de serviços (e.g., consumo de energia). Além disso, o modelo integra fatores práticos como falhas nos recursos, priorização de serviços e tempos de configuração e reparo na formulação. Desta forma, o modelo permite compreender como o núcleo da rede 5G se comporta no atendimento a diferentes categorias de serviços, aplicando a priorização de serviços para compartilhar eficientemente os recursos de processamento. Algumas descobertas incluem a ideia de que taxas mais altas de chegada eMBB diminuem a disponibilidade e aumentam os tempos de resposta para até 300 ms, enquanto a disponibilidade para URLLC permanece estável. Além disso, as taxas de configuração de contêineres e as taxas de falhas afetam substancialmente a disponibilidade e os tempos de resposta, com taxas de configuração mais altas aumentando a disponibilidade em até 30% e reduzindo os tempos de resposta em 60%. Ademais, o número de contentores surge como um fator significativo, melhorando tanto a disponibilidade como os tempos de resposta, enquanto os tamanhos dos buffers afetam principalmente os tempos de resposta. Em resumo, nosso trabalho avança no estado da arte atual do domínio MEC-NFV, fornecendo insights valiosos para o dimensionamento da arquitetura MEC-NFV, modelos de negócios e mecanismos para lidar com alocação de recursos sob diferentes restrições de comunicação.

Palavras-chave: 5G; URLLC; eMBB; MEC; NFV; alocação de recursos.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

3GPP	3rd Generation Partnership Project
5G	Fifth Generation Mobile Network
AF	Application Function
AMF	Access and Mobility Function
AR	Augmented Reality
AUSF	Authentication Server Function
CAGR	Compound Annual Growth Rate
CTMC	Continuous-time Markov Chain
DN	Data Network
DRA	Dynamic Resource Allocation
DSS	Dynamic Spectrum Sharing
eMBB	Enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
FWA	Fixed Wireless Access
IoT	Internet of Things
KVM	Kernel-Based Virtual Machine
M2M	Machine-to-Machine
MEC	Multi-access Edge Computing
MIMO	Multiple-Input, Multiple-Output
MME	Mobility Management Entity
MMTC	Massive Machine Type Communication
mmWave	Millimeter-wave
NEF	Network Exposure Function
NFV	Network Function Virtualization
NFVI	NFV Infrastructure

NFVO	NFV Orchestration
NR	New Radio
NS	Network Slicing
OFDM	Orthogonal Frequency Division Multiplexing
ONAP	Open Network Automation Platform
OSM	Open Source MANO
OTT	Over the Top
QoS	Quality of Service
RAN	Radio Access Networks
SDN	Software-Defined Networking
SLA	Service Level Agreement
SMF	Session Management Function
TSN	Time-Sensitive Networking
UE	User Equipment
ULCL	Uplink Classifier
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communications
VNF	Virtual Network Functions
VR	Virtual Reality

LIST OF SYMBOLS

α	Setup Rate
γ	Failure Rate
μ_U	URLLC Service Rate
μ_E	eMBB Service Rate
λ_U	URLLC Arrival Rate
λ_E	eMBB Arrival Rate
π	Steady-state Probability Vector
Ω	Feasible Space State
P_{idle}^{CT}	Idle Container Energy Consumption
P_{setup}^{CT}	Container Energy Consumption during Setup
P_{busy}^{CT}	Busy Container Energy Consumption
K	Buffer Size for eMBB users
k	Buffer Size for URLLC users
C	Total Number of Containers
i	Admitted URLLC services
j	Admitted eMBB services
l	URLLC containers during processing state
m	eMBB containers during processing state

CONTENTS

1 INTRODUCTION

The emergence of the Fifth Generation Mobile Network (5G) technology represents a pivotal advancement in the realm of mobile networks. This transformative technology introduces a robust cloud-native core network with Network Slicing (NS) capabilities, empowering the creation of innovative services within three primary categories: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and Massive Machine Type Communication (MMTC)[3GPP 2020][3GPP 2022].

Several advancements were introduced in 3GPP Release 15 [3GPP 2020], including the deployment of 5G New Radio (NR) with support for higher frequency bands, massive Multiple-Input, Multiple-Output (MIMO), and flexible waveforms. eMBB was a central focus, aiming to boost data rates and enhance performance for applications such as high-definition video streaming and virtual reality. The release also marked the transition to a Service-based Architecture, redefining the core network for greater flexibility and scalability. Network Slicing, another key innovation, allowed the creation of customized virtual networks for diverse applications and furthermore, the introduction of Standalone (SA) 5G Core Network enabled full 5G core deployment without reliance on legacy 4G infrastructure. Lastly, Multi-access Edge Computing (MEC) was introduced, positioning compute and storage resources closer to the network's edge, and thus enabling low-latency processing for various services [Ghosh et al. 2019].

The 3GPP Release 16 brought emphasis on URLLC, which is vital for applications requiring ultra-low latency and high reliability, including industrial automation and mission-critical communications. Additionally, it introduced improvements in network management and orchestration, streamlining network operations and dynamic resource allocation, alongside energy efficiency was also a key focus, addressing the environmental impact and the imperative for sustainable network operations [Ghosh et al. 2019].

Among the three main service categories, URLLC may be the most challenging, as it presents the most critical requirements in terms of latency and reliability [Siddiqui et al. 2023]. In this context, the combination of MEC and Network Function Virtualization (NFV) (MEC-NFV) is fundamental for URLLC because it allows virtualized network functions and applications to be hosted closer to the end-user, reducing the latency and enhancing the reliability. NFV and MEC also offer significant advantages to eMBB services since NFV enables eMBB to dynamically allocate and scale resources based on demand, optimizing network capacity and

reducing latency during high-traffic periods. Moreover, content and applications can be cached and processed at the network edge, further lowering latency and ensuring faster response times.

To enable the coexistence of eMBB and URLLC services, the concept of NS is pivotal [Setayesh and Bahrami 2022]. It plays a fundamental role in enabling the shared utilization of physical infrastructure within dynamic on-demand networking platforms, allowing for the creation of multiple virtual networks. This concept leverages the virtualization of both edge and core network functions, making effective use of well-established virtualization technologies. Notably, it not only leads to significant cost reductions but also elevates network scalability to new heights. Though new wireless services will be available in 5G, there remain several challenging issues to be addressed. One is to handle the dynamic resource allocation to different QoS requirements in the MEC. The current body of knowledge lacks comprehensive mathematical models and solutions pertaining to coexistence mechanisms between eMBB and URLLC. Existing studies predominantly concentrate on analysis, system-level design, or framework development, not addressing the challenges posed by simultaneously accommodating eMBB and uRLLC traffic.

Multiple works have addressed the coexistence of different service categories within 5G networks, but predominantly concentrating only on radio resource allocation within the Radio Access Networks (RAN) ([Bairagi et al. 2021], [Zhang et al. 2021], [Kim and Park 2020], [Huang et al. 2021]). However, there exists a notable gap when it comes to considering factors that influence resource provisioning in the MEC-NFV domain. Notably, prior research often presupposes fault-free cloud environments ([Bairagi et al. 2021] and [Li and Jin 2021]) or with instantaneous provisioning times ([Zhang et al. 2021], [Kim and Park 2020], [Tong et al. 2020], [Ma et al. 2021]) which may not align with the realities of 5G networks. Furthermore, studies often do not consider that there are service subcategories that differ widely ([Li and Jin 2021], [Liu et al. 2022], [Abdelhadi et al. 2022], [Emara et al. 2021]) and that the overhead caused by the virtualization and dynamic resource allocation impact on them. For instance, The boot-up process of a Virtual Network Functions (VNF) instance plays a pivotal role in cost-performance analyses for both edge and core 5G networks. During installation, energy is consumed, and resources are allocated, yet services remain unattended. This has repercussions not only in terms of energy efficiency but also in the context of potential Service Level Agreement (SLA) violations if the VNF takes too long to start processing critical traffic flows.

The aim of our study is to address the combination of MEC, NFV, and dynamic virtual resource allocation within the context of coexisting 5G service categories: URLLC and eMBB,

aiming at the challenge of resource dimensioning in compact MEC-NFV nodes. Considering the MEC-NFV architecture, a model to evaluate how requests will be managed by the virtualization resources of a single MEC node was designed, with a primary focus on meeting the URLLC users (their requirements) coexisting with eMBB ones. We propose a CTMC-based model to characterize the dynamic virtual resource allocation, incorporating five performance metrics, which will be relevant not only for URLLC and eMBB services (e.g., availability and response time) but also for service providers (e.g., power consumption). In addition, to make the model more practical, we have integrated factors like resource failures, service prioritization, and setup (repair) times into the formulation, as they can incur significant impacts on the 5G applications' requirements. In general, our work describes and classifies the relevant works in the field of MEC-NFV resource allocation focusing on mathematical models. Then, we describe the main benefits and drawbacks related to the virtualization layer elements that compose the MEC-NFV environment. Moreover, the MEC-NFV node model incorporates dynamic scaling capabilities and service prioritization to accommodate the two 5G service categories and finally, we evaluate the impact of multiple parameters of a single MEC-NFV node on metrics such as average response time, energy consumption and service availability. Some of our key findings include the idea that higher eMBB arrival rates decrease availability and increase response times, while URLLC availability remains stable. Moreover, the container setup rates and failure rates substantially affect both availability and response times, with higher setup rates enhancing availability and reducing response times. Also, the number of containers emerges as a significant factor, enhancing both availability and response times, while buffer sizes mainly impact response times.

1.1 RESEARCH QUESTIONS

Given the problems presented, this work aims to answer the following research questions:

- What are the main works in the area of MEC-NFV resource allocation with a focus on mathematical models and their characteristics?
- What are the main benefits and drawbacks related to the virtualization layer elements that compose the MEC-NFV environment?
- Is it possible to model a MEC-NFV node incorporating dynamic scalability and service

prioritization capabilities to accommodate two distinct categories of 5G services and from the creation of this model formulate essential performance metrics closely linked to URLLC and eMBB services?

- What is the impact of varying the sizing of different parameters of a MEC-NFV node on metrics such as average response time, energy consumption and service availability?

1.2 OBJECTIVES

The main objective of this work is to analyze the coexistence of different services (eMBB and URLLC) in MEC/NFV-based 5G networks considering the dynamic resource allocation. The following specific aims have been defined to achieve this objective:

- To classify the main works in the field of MEC-NFV resource allocation focusing on the mathematical models and coexistence of different user types in 5G networks.
- To model and validate the virtual resource allocation in the MEC-NFV node considering different user types, scenarios, and virtualization layer aspects.
- To Evaluate the impact of varying the sizing of different parameters of a MEC-NFV node on metrics such as average response time, energy consumption, and service availability.

1.3 DOCUMENT ORGANIZATION

This dissertation is organized as follows: Chapter 2 examines the technical background, which includes an overview of 5G technology, the fundamental 5G enabling technologies that underpin the current 5G architecture and resource allocation and includes an overview of the main mathematical tools used in this document, namely queueing theory. In Chapter 3, a review of the current literature on the topic of resource allocation for MEC-NFV is described. It includes the main features of these works and a short classification. Moreover, Chapter 4 describes a CTMC-based analytical representation for a single node NFV-MEC, assuming a virtual environment featured with containers that are able to process both URLLC and eMBB requests. Chapter 5 describes the model validation and a result analysis obtained by extensive discrete-event simulations. Finally, Chapter 6 provides our concluding remarks and highlights future work directions.

2 TECHNICAL BACKGROUND

This chapter presents the background and fundamental concepts, enhancing the reader's comprehension of this document. In Section 2.1, it offers an overview of 5G technology, with a particular focus on two of the three service categories investigated in this study: eMBB and URLLC. This section also delves into the intricacies of the 5G architecture and its constituent elements. Following that, Section 2.2 provides insights into the fundamental 5G enabling technologies that underpin the current 5G architecture and resource allocation. Finally, the basic principles of queuing theory are explored in Section 2.3.

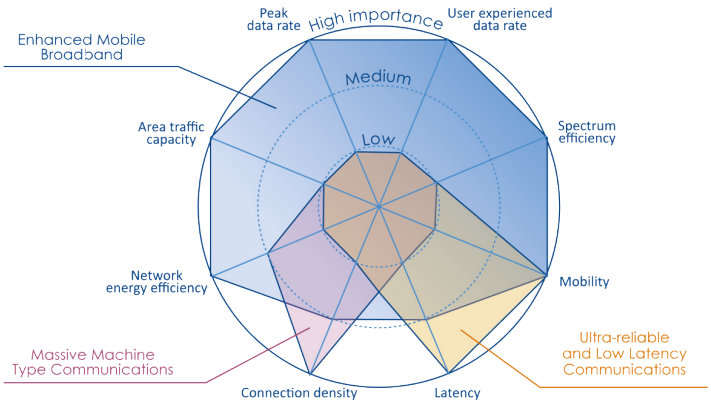
2.1 THE 5TH GENERATION MOBILE NETWORK (5G)

The 5G has emerged as a groundbreaking technology with a promise of ultra-fast speeds, low latency, massive connectivity, and high reliability [Sarrigiannis et al. 2020]. However, since these characteristics are inherently conflicting, a group of telecommunications organizations responsible for defining the standards for the 5G known as the 3rd Generation Partnership Project (3GPP), has divided the applications in three case groups: eMBB, URLLC, and MMTC that are designed to address different performance needs [Ali et al. 2021]. This section provides an overview of the 5G technology, focusing on two of the three service categories that are studied in this work: the eMBB and URLLC.

The eMBB service category focuses on providing significantly higher data rates, increased network capacity, and enhanced user experiences compared to the previous generations of mobile networks [Setayesh and Bahrami 2022]. It enables, for instance, seamless streaming of high-definition videos, Augmented Reality (AR), Virtual Reality (VR), and other bandwidth-intensive applications. On the other hand, the URLLC category is designed to support mission-critical applications that require ultra-reliable and near-instantaneous communication with stringent latency requirements. It is suitable for applications such as autonomous vehicles, industrial automation, remote surgery, and critical infrastructure monitoring [Feng et al. 2019]. Lastly, the third service category known as the MMTC focuses on connecting a massive number of devices and sensors in the Internet of Things (IoT) ecosystem, enabling seamless communication between a vast array of devices, ranging from smart city infrastructure, industrial sensors to wearable devices. mMTC is characterized by high device density, scalable connectivity,

and energy-efficient communication, i.e., Massive Machine-to-Machine (M2M) communication [Mehmeti and Porta 2022]. In Figure 1, we can see the importance of the network capabilities related to each case group. The following subsections will delve into the details of the service categories to be explored in this work, i.e., eMBB and URLLC.

Figure 1 – 5G network capabilities importance.



Source: ETSI

2.1.1 eMBB

Among the features introduced by the 3GPP in Release 15, the eMBB service category stands out as a significant advancement, aiming to provide higher data rates, increased network capacity, and enhanced user experiences compared to previous generations. The purpose of eMBB services is to serve high data rates applications such as AR, and VR with acceptable reliability [Sohaib et al. 2023].

The Radio Sector of the International Telecommunication Union (ITU) has listed in its report M.2412 three deployment options that characterize the eMBB category, which are hotspot, dense urban, and rural. The first focuses on small coverage per site and high user throughput or user density in buildings while the second comprises high user density and traffic loads targeting pedestrian and vehicular users in city centers and dense urban areas, with outdoor and outdoor-to-indoor coverage. The last one in turn deals with rural environments with larger and continuous wide area coverage, supporting pedestrian, vehicular and high-speed vehicular users [Stallings 2021]. Regarding eMBB services, Fixed Wireless Access (FWA) is one

that has been enhanced by 5G networks. For instance, in North America and Western Europe, about 70% of FWA service providers offer it over 5G [Ericsson 2023].

At the physical layer, eMBB leverages advanced technologies such as Millimeter-wave (mmWave) frequencies and MIMO. While the first offers wider bandwidths and higher data rates through higher frequency allocations, the second denotes a large number of antennas that facilitate spectral efficiency, improved signal quality, and increased network capacity [Setayesh and Bahrami 2022]. As for the primary modulation scheme, eMBB is likely to use Orthogonal Frequency Division Multiplexing (OFDM), which partitions the available spectrum into multiple subcarriers, also allowing parallel data transmission [Sohaib et al. 2023]. This not only provides resilience against selective fading but also enables flexible resource allocation which is the main focus of our work.

MEC and NFV technologies also offer significant advantages to various eMBB applications, including intelligent video acceleration and AR-based ones[Antevski et al. 2020]. MEC facilitates rapid data exchange, high computing power, and low latency in localized areas. Additionally, it also may provide specialized MEC services, such as localization for AR applications, enhancing the user experience in settings like museums or sports events. NFV, on the other hand, enables dynamic scaling and optimizes resource utilization based on the application load. For video streaming and AR applications, this load may vary according to the event type and the number of users. More details about MEC and NFV technologies are given in Sections 2.2.2 and 2.2.3, respectively.

2.1.2 URLLC

In addition to the eMBB, the 3GPP Release 15 also introduces the URLLC service category for mission-critical applications that necessitate near-instantaneous and ultra-reliable communication [Filippou et al. 2020]. In this context, the concept of latency refers to the delay between the transmission and reception of a data packet whereas reliability pertains to the system's ability to deliver data packets with minimal errors/losses, ensuring integrity. Another perspective of reliability is defined as the probability of successful transmission of a Layer 2/3 packet within a required maximum time. For instance, according to the ITU report M.2410, $1 - 10^{-5}$ is the minimum required success probability for transmission of a Layer 2 protocol data unit (PDU) of 32 bytes within 1 ms in urban macro-URLLC test environments [Stallings 2021]. Additionally, both latency and reliability are closely linked to another per-

formance metric: Availability, which denotes continuous accessibility of the communication. In mission-critical scenarios, uninterrupted connectivity is of utmost importance, and URLLC should be simultaneously robust and resilient even in challenging environments or during high network congestion periods [Feng et al. 2019]. Table 1 exemplifies some 5G applications and their characteristics.

Table 1 – Examples of eMBB and URLLC Applications.

Work	Use Case	Latency (ms)	Data Rates	Service Category
[Siddiqui et al. 2023]	Factory Automation	0.25 - 10	1 Mbps	URLLC
[Siddiqui et al. 2023]	Smart Transportation Systems	10 - 100	10 - 700 Mbps	URLLC
[Siddiqui et al. 2023]	Robotics and Telepresence	1	100 Mbps	URLLC
[Siddiqui et al. 2023]	Health Care Management	1 - 10	101 Mbps	URLLC
[Mahdi et al. 2021]	AR/VR 120 FPS	< 8	1.5 Gbps	eMBB
[Ericsson 2016]	FWA	4	1 - 5 Gbits	eMBB
[Sugito et al. 2020], [Raca et al. 2020]	8K 120-Hz Video Streaming	< 20	85 - 110 Mbps	eMBB
[Stallings 2021]	Smart Office	< 10	27 Mbps per user	eMBB

Source: The author (2023)

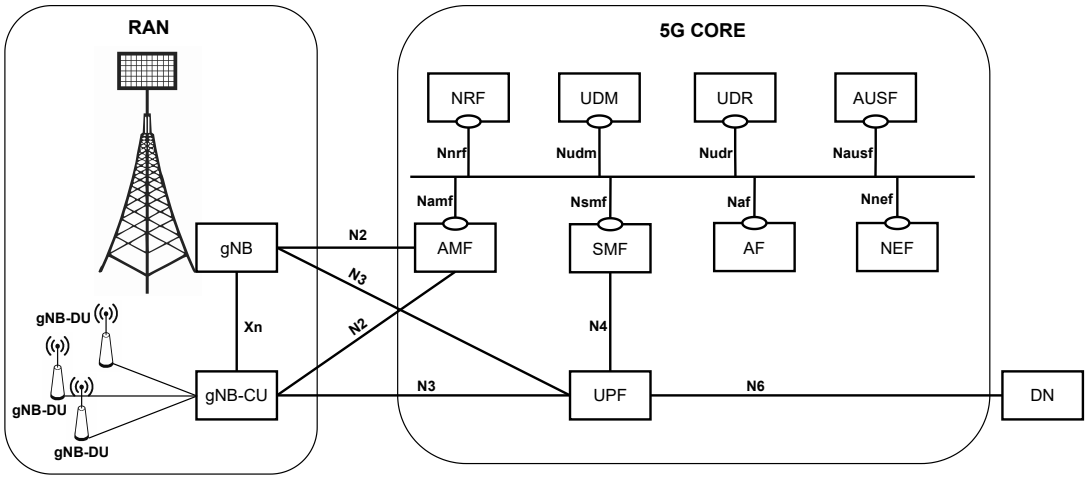
To address the complex requirements of URLLC, 3GPP Release 16 incorporates advanced techniques such as Time-Sensitive Networking (TSN) and NS and moreover, architectural and design paradigms, such as MEC and NFV. Some of these concepts will be further elaborated in Section 2.2. Furthermore, the integration of eMBB and URLLC is expected to encompass a wide range of use cases, however, it also poses great challenges in terms of the coexistence between the two categories [Bairagi et al. 2021]. One of the primary challenges is achieving a balance between their divergent requirements, which is the main topic of this work. In particular, we leverage resource allocation to tackle some of these challenges, considering part of the already existing 5G architecture, which is detailed in the following lines (Subsection 2.1.3).

2.1.3 5G Architecture

The 5G network architecture comprises two essential components: the RAN and the Core [Pana and Babalola 2022]. The RAN is responsible for connecting user devices to the core network, encompassing base stations, spectrum bands, antennas, and associated equipment that enable wireless communication between mobile devices and network infrastructure [Pana and Babalola 2022]. The 5G RAN introduces several new features compared to previous generations, including both sub-6 GHz and mmWave bands, to provide extensive coverage and massive MIMO, besides beamforming and Dynamic Spectrum Sharing (DSS) to enhance per-

formance and optimize radio resource utilization [Pérez and López 2023]. Generally, the lion's share of 5G research concentrates on the RAN, however, our work focuses on the less explored 5G core, which encompasses a variety of functions that are instrumental to the network operation and are subject to the same stringent requirements as the RAN. These functions can be broadly categorized into control plane functions and user plane functions [Du et al. 2023], which are described in Sections 2.1.3.1 and 2.1.3.2, respectively. Figure 2 shows the RAN and 5G Core Network with their network functions and communication interfaces.

Figure 2 – 5G network architecture.



Source: The author

2.1.3.1 5G Core: Control Plane Functions

Control Plane Functions ensure seamless connectivity, efficient routing, and the enforcement of Quality of Service (QoS) policies. By effectively managing the control plane, the 5G Core enables reliable and secure communication while maintaining the necessary control and coordination for a wide range of applications and services [Tang et al. 2022]. Fig.2 presents a subset of functions that compose the 5G core control plane. Additional functions have been defined by the 3GPP and may be consulted in [3GPP 2022].

One of the control plane functions is the Access and Mobility Function (AMF), which is responsible for managing access and mobility-related aspects of user devices within the network. It maintains a non-access stratum (NAS) signaling connection with the UE and manages the UE registration procedures such as user authentication, ensuring seamless connectivity and handover between different network access points [Tang et al. 2022].

Another well-known control plane function is the Session Management Function (SMF), It engages with the separated data plane by managing all the Protocol Data Unit (PDU) sessions. This function encompasses the setup, adjustments, and termination of sessions, including IP allocation to the UE, QoS management and policy enforcement, all in coordination with the User Plane Function (UPF). In the 5G core, a series of Next-Generation (NG) tunnels, coupled with numerous radio bearers on the radio interface, collectively constitute a PDU session [Chakraborty and Corici 2020].

In a broader scope, the Authentication Server Function (AUSF) plays a pivotal role in enabling subscriber authentication during initial registration or re-registration within the 5G network. Moreover, the AUSF assumes the responsibility of furnishing security parameters to safeguard the steering of roaming information and to ensure the protection of data involved in the UE update procedure [Tang et al. 2022].

The Unified Data Management Function (UDM) provides subscriber data to handle tasks such as authorization, registration, and mobility management. It is responsible for creating authentication credentials for User Equipment (UE) authentication in the network. Additionally, the UDM retains the context provided by the serving Access and Mobility Management Function (AMF) for a specific UE, along with the serving Session Management Function (SMF) for the UE's Packet Data Unit (PDU) session. Within its realm, the UDM securely stores subscription data for each individual UE, encompassing both 3GPP and Non-3GPP access information [Koonampilli et al. 2021].

The Unified Data Repository (UDR) serves as a hub for storing and fetching subscription and policy information. Subscription data encompass a wide range of content, spanning 3GPP and Non-3GPP context data, data relevant to PDU session management, as well as the essential keys for authentication credential generation [Koonampilli et al. 2021].

The Network Repository Function (NRF) undertakes NF service registration and maintains NF profiles along with accessible NF instances. It stands as a depository for these services, enabling each Network Function (NF), or service consumer, to uncover the array of services provided by other NFs, the service providers. Comprehensive specifics are contained within the NF profiles housed in the NRF, including details such as NF classification, location, capability, endorsed NF services, and service instance addresses [Tang et al. 2022].

The Network Exposure Function (NEF) is another function with a significant role that provides an interface for external applications and services to access network information and capabilities. It facilitates the exposure of network resources and functions through APIs,

enabling third-party developers to create innovative applications and services that can leverage the capabilities of the 5G network [Kaloxylos 2018].

The 3GPP defines the Application Function (AF) as a versatile functional entity capable of delivering diverse services. These encompass a wide array of services including voice, video, messaging, and applications geared towards the IoT. To support the provision of a service, an AF can influence traffic routing or quality of service by interacting with 5G core network functions, ensuring the streamlined transmission and efficient processing of user data [Lentisco et al. 2023].

2.1.3.2 5G Core: User Plane Functions

The 5G User Plane, on the other hand, is responsible for forwarding and processing user data packets. It handles routing, forwarding, and traffic management, ensuring optimized data transmission, low latency, and high-quality user experiences. The UPF is the main function of the 5G core user plane and associated to theData Network (DN), RAN, and UE compose the user plane of the 5G system [Stallings 2021].UPF and DN are detailed in the following lines.

The UPF plays a pivotal role in processing and forwarding user data. It establishes connections with external data networks and serves as a steadfast point of reference for User Equipment (UE) concerning external networks as the UEs move. Additionally, the UPF undertakes the task of marking packets with QoS indicators, ensuring that these packets are accorded suitable treatment within the 5G Core and RAN networks [Koonampilli et al. 2021].

Finally, the DN represents the external network infrastructure that carries user data traffic. It encompasses the physical and logical components such as switches, routers, and transmission links that enable the transmission of data packets between an external network and the 5G network.

In summary, the 5G core incorporates various control and user plane functions that enable the delivery of advanced services in the 5G ecosystem. These functions bear the responsibility of delivering services tailored to various demand categories while remaining susceptible to potential failures and overloads. This underscores the significance of employing tools that enable the dimensioning of the network environment. This dimensioning hinges on the workload generated by the different service categories within the system and the subsequent adaptive adjustment of service scales to address these demands.

Next, we provide insights on the key 5G enabling technologies that provide ground to the

current 5G architecture and for resource allocation (Section 2.2).

2.2 5G CORE ENABLING TECHNOLOGIES

NS, MEC, NFV, and Software-Defined Networking (SDN) are at the forefront of 5G innovation, providing the foundation for advanced network architectures and capabilities. Please note that these are independent technologies and thus, each of them utilizes its own orchestrator and management entities such as the SDN controller, NFV orchestrator, network slicing manager, and mobile edge platform manager. In this respect, a synergy between these control entities is needed to jointly optimize network resources. In this section, each technology is individually detailed to cover its principles, functionalities, and contributions to the deployment of 5G networks, especially regarding efficient resource allocation in the core network.

2.2.1 Network Slicing (NS)

NS has emerged as a key concept in dynamic on-demand networking platforms, enabling the creation of multiple virtual networks on a shared physical infrastructure that permits a better use of the resources to the operator. [Setayesh and Bahrami 2022]. This concept leverages the virtualization of edge and core network functions, utilizing established virtualization technologies such as Kernel-Based Virtual Machine (KVM) and Docker. The application of network slicing has been observed in multiple practical mobile network works [Baba et al. 2022], including the proposal of virtual cloud Evolved Packet Core (EPC) [Taleb et al. 2015].

However, the significance of NS extends beyond the core network, as the end-to-end guarantees required by 5G necessitate the implementation of end-to-end slicing. This development has prompted the creation of Management and Orchestration (MANO) frameworks [Yousaf et al. 2019] and the emergence of open-source implementations such as Open Source MANO (OSM) [Conțu et al. 2022] and Open Network Automation Platform (ONAP) [Rodriguez and Guillemin 2019].

Combined with other technologies (e.g., MEC and NFV), NS yields the fundamental principle for the coexistence between eMBB and URLLC [Setayesh and Bahrami 2022], aside from substantial cost reductions and enhance network scalability. In fact, the 3GPP has recognized the relevance of MEC in facilitating mobile network slicing extensions and enhanced multi-tenancy, hence, the synergy between MEC and NS, akin to the relationships between NFV and

SDN, is expected to play a pivotal role in 5G networks.

2.2.2 Multi-Access Edge Computing (MEC)

In general, MEC is related to the idea of bringing computing capabilities closer to the network edge (i.e., end user), reducing latency while also lowering network congestion probability [Khan 2017]. By leveraging its proximity to users, it impacts all three 5G service categories, but is especially relevant for URLLC, due to its stringent requirements. By deploying computing resources at the edge, MEC enables near-instantaneous communication and ultra-low latency for critical and time-sensitive applications. In addition, it also reduces the round-trip time for data transmission by minimizing the physical distance between the user and the processing resources, ensuring greater reliability. Fig. 3 illustrates the use of MEC to host applications and network functions in 4G and 5G networks.

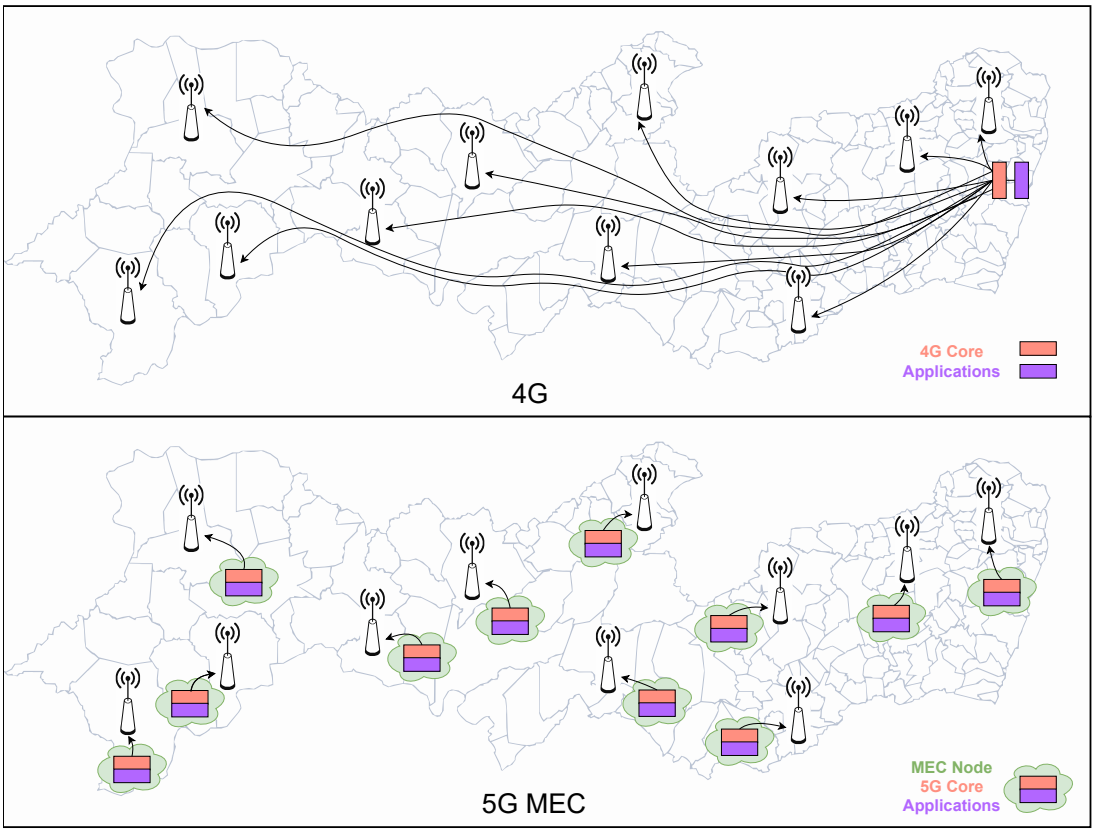
According to the Research and Market report [Research and Market 2023], the global MEC market size is estimated to reach USD 55.41 billion by 2030, exhibiting a Compound Annual Growth Rate (CAGR) of 49.1% over the forecast period (2023-2030). It is driven by the increasing adoption of Over the Top (OTT) media streaming services and rising demand for personalized content and encourages mobile and telecommunication operators to improve their infrastructure.

By using MEC, applications may be implemented as software entities that run on top of a virtualization infrastructure placed at the network edge [ETSI 2022]. The European Telecommunications Standards Institute (ETSI) has played a key role in the adoption of MEC by defining standards for the MEC technology such as the MEC framework illustrated in Fig. 4. It is composed of entities grouped into system, host, and network levels, which are briefly described as follows.

At the MEC system level, the management includes an orchestrator as its core component, which has an overview of the MEC system (e.g. hosts, topology, available resources, and services). It is responsible for a variety of tasks such as selecting MEC host(s) for application deployment based on constraints (e.g., latency and available resources) and triggering application setup, termination, and relocation when needed and feasible [ETSI 2022].

At the MEC host level, the management deals with MEC specific functionalities of a particular MEC host and the applications running on it. The MEC host, in turn, consists of a MEC platform and a virtualization infrastructure. The former is a collection of functionality

Figure 3 – Arrangement of 4G and 5G MEC network elements.



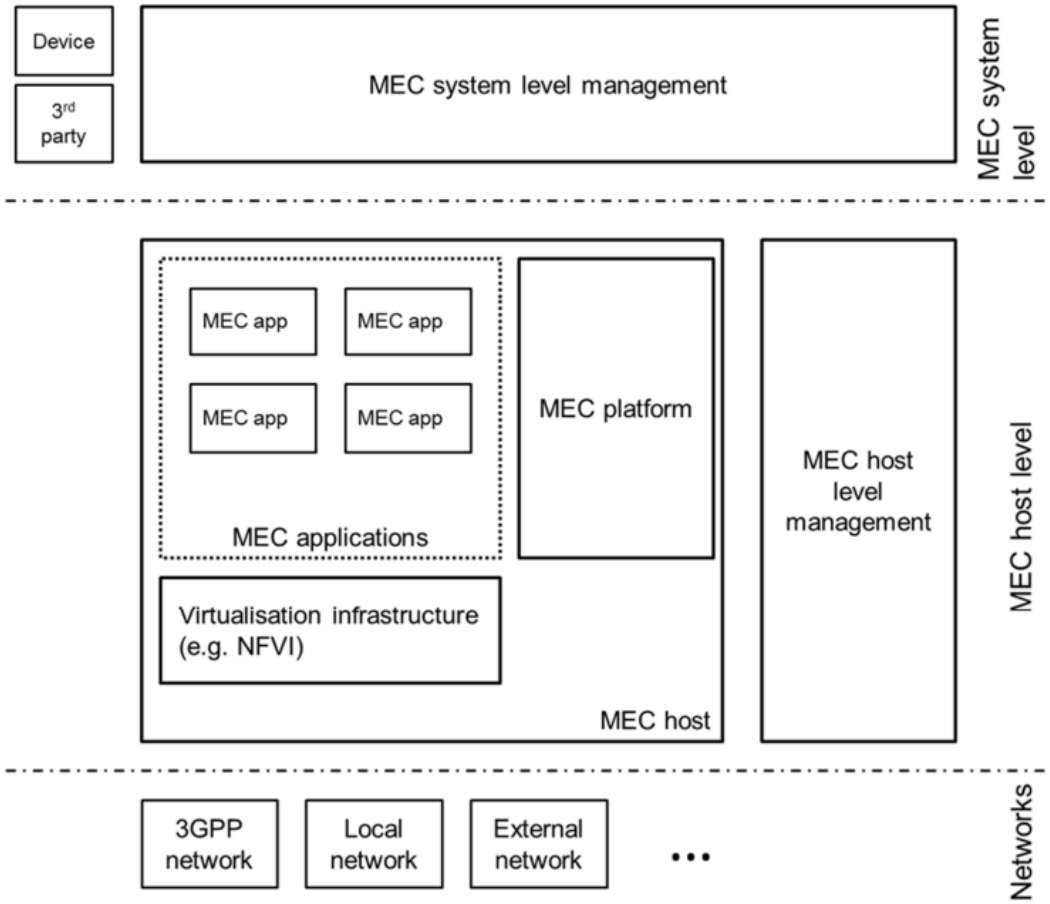
Source: The author

required to run MEC applications on a particular virtualization infrastructure and enable them to provide and consume MEC services. The latter provides compute, storage, and network resources to the MEC applications and presents a data plane that receives traffic rules from the MEC platform and routes the traffic among applications, services, local and external networks. The MEC applications, in turn, are embedded into virtual machines or containers, on top of infrastructure provided by the MEC host. The network level concludes the MEC architecture composition, which encompasses the networks used to provide connectivity to the MEC nodes including local, external and 3GPP ones, for example [ETSI 2022].

Since that MEC and NFV (see Section 2.2.3) are complementary concepts, the ETSI has also designed a variant reference architecture that leverages the combination of MEC and NFV [ETSI 2022]. This integration with NFV provides a standardized framework for virtualizing and managing network functions, allowing MEC to harness the advantages of virtualization and dynamic resource allocation, and to re-use NFV components to fulfil a part of the MEC management and orchestration tasks.

Moreover, being MEC a key-technology to support 5G services (e.g. URLLC), the 5G

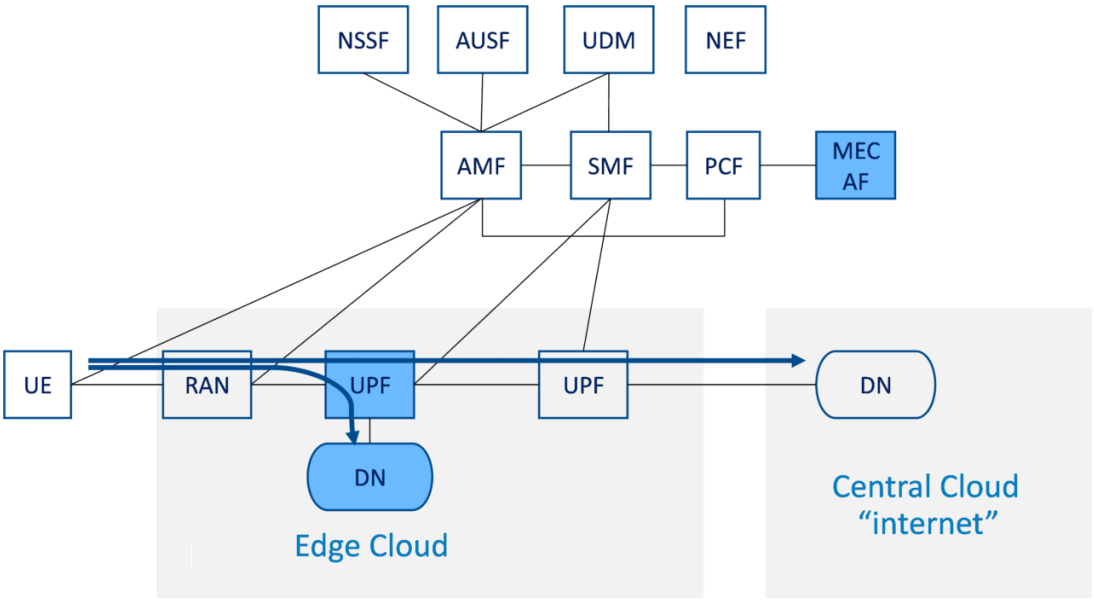
Figure 4 – Multi-access Edge Computing Framework



Source: [ETSI 2022]

system specifications have presented a set of new functionalities that enables the integration of edge computing in 5G networks [Kekki and Featherstone 2018]. For example, 3GPP allows the mapping of MEC onto AF in the 5G core (MEC as an AF) that can use the services and information from other NFs based on the defined policies. Thus, MEC can request the core to select a local UPF near the RAN for handling the PDU sessions of the target UE(s) and controlling the traffic forwarding from the local UPF according to the traffic filters received from MEC (AF) [Weissberger 2021]. Fig. 5 illustrates this scenario where by using the same uplink session, the UE may obtain content from both the local and central servers seamlessly via Uplink Classifier (ULCL). More information about 3GPP specifications for 5G system and MEC integration may be found in [3GPP 2022][Sprecher Nurit and et al. 2020].

Figure 5 – MEC as an Application Function (AF)



Source: [Weissberger 2021]

2.2.3 Network Functions Virtualization (NFV)

NFV is an industry-driven initiative aimed at virtualizing network functions, such as switches, routers, and NATs, by utilizing virtual machines and/or containers on standard servers instead of proprietary single-purpose network devices [Zhao et al. 2021]. The adoption of VNF offers two significant advantages: (1) enables the flexibility of relocating network functions to different locations without the need for additional hardware, thereby reducing operational complexities and (2) facilitate task optimization, scheduling, and resource allocation in scenarios where computing resources are limited (e.g., at the network Edge) [Xue and Jiang 2022]. This is particularly important as each service request may have distinct requirements in terms of the specific service and processing needed.

Regarding the mobile context, NFV plays a pivotal role in supporting the cloudification trend by decoupling mobile network functions, such as the Mobility Management Entity (MME), from dedicated hardware [Mijumbi et al. 2016]. Few Deployment tools based on this concept such as Open Source Mano and OpenBaton orchestration suites have already been developed to facilitate the seamless integration and management of MEC entities in the network infrastructure [Kekki and Featherstone 2018]. These provide essential functionalities for orchestrating and automating the deployment, scaling, and lifecycle management of computing resources, enabling efficient and dynamic allocation of computing and networking capabilities

at the network edge. The fundamental concept of the relationship between NFV and MEC is to dedicate a single VNF to each service request, following a service-on-demand model. Hence, the combination of NFV and MEC holds the potential for increased scalability, as it enables on-demand resource scaling physically close to the end user. As per the ETSI, it is already accepted that MEC can utilize the NFV Infrastructure (NFVI) as the virtualization platform to execute edge applications alongside other VNFs [Yu 2016]. Consequently, MEC applications are regarded as VNFs, and certain aspects of edge orchestration can be delegated to the NFV Orchestration (NFVO) [Kekki and Featherstone 2018].

2.2.4 Software-Defined Networking (SDN)

Traditional networking architectures struggle to efficiently manage the complexity and scale of the 5G network. SDN offers a solution by decoupling the control plane from the underlying infrastructure, enabling centralized network management and programmability. This allows operators to dynamically allocate and optimize network resources, improving network efficiency, scalability, and service agility [Blanco et al. 2017].

The data plane encompasses network orchestration and device control logic, while the control plane is encapsulated by a logically centralized controller that communicates with the data plane through south and northbound APIs. The centralization aspect of SDN contributes to efficient flow management, service discovery, and orchestration, particularly in the context of multi-tier MEC infrastructure [Scotece et al. 2023]. SDN is also strongly related to the concept of NS, i.e., with SDN, operators can efficiently manage and orchestrate network slices, allocating resources based on the unique needs of different applications or user groups. This enables the coexistence of diverse services, such as eMBB and URLLC, all within a single physical infrastructure, leading to optimized resource utilization [Blanco et al. 2017].

Furthermore, SDN enables dynamic network programmability, empowering operators to respond quickly to changing service demands and traffic patterns [Scotece et al. 2023]. By abstracting the control plane, SDN allows operators to centrally manage and configure the network, making it easier to deploy new services and optimize the network in real time. This facilitates the implementation of advanced network functionalities, such as NFV. In brief, with SDN as a fundamental building block, it is possible to efficiently handle 5G networks.

The role of 5G enabling technologies, such as NS, MEC, NFV, and SDN, is to spearhead innovation in 5G networks by providing the foundational capabilities and architectures needed

for advanced services. Each of these technologies operates independently with its orchestrator and management systems, collectively contributing to 5G deployment. NS allows for dynamic network creation, MEC brings computing closer to end users to reduce latency, particularly crucial for URLLC applications, and NFV and SDN offer network virtualization and management capabilities. The integration of MEC and NFV, guided by ETSI, provides standardized virtualization and dynamic resource allocation, collectively shaping the landscape of 5G networks and services. These technologies are instrumental in optimizing resource allocation and enabling the diverse, high-performance services that 5G promises.

Following that, we present the fundamental mathematical framework that can be used for optimizing resource allocation and aiming for the wide range of high-performance services that 5G aims to deliver (Section 2.3).

2.3 QUEUEING THEORY

Queueing theory, derived from probability theory, is a vital field of study that addresses the phenomenon of waiting in queues. In the context of URLLC and eMBB, queueing theory assumes paramount significance. It offers a rigorous mathematical framework to analyze and optimize the performance of queueing systems.

Within the field of mobile communications, queueing theory plays a role in understanding and managing the flow of data through various network components, such as base stations, routers, and data centers. By modeling the arrival process, service times, and system capacity, queueing theory enables the evaluation of relevant performance metrics like packet delay, throughput, and queue length. Furthermore, it facilitates the development of strategies to minimize latency, improve resource allocation, and enhance overall network performance.

This section aims to provide a comprehensive overview of queueing theory and its application to the analysis of Continuous-Time Markov processes in the domain of URLLC and eMBB mobile communications. By understanding the fundamental principles, researchers can effectively model and analyze the performance of queueing systems, leading to advancements in the design and management of future mobile communication networks.

2.3.1 Concept and Notation

A queueing system can be described in terms of various components and parameters, including the following:

- The arrival process: This process characterizes the rate at which customers arrive at the queue and the distribution of inter-arrival times between successive customer arrivals [Gross et al. 2008]. The most commonly used Markovian arrival process is the Poisson process, where inter-arrival times are exponentially distributed.
- The service time: It represents the time spent by a customer being served.
- The number of servers: This parameter indicates the count of servers available in the system to handle customer requests.
- The system capacity: It specifies the maximum number of customers allowed in the system, including both those waiting for service in a buffer and those currently being served.
- Population size: The total number of customers that can enter the system, which can be finite or infinite.
- The service discipline: This parameter defines the policy for determining the order in which customers are served. Common disciplines include First Come, First Served (FCFS), Last Come, First Served (LCFS), and Static Priorities (SP), where customers are served based on predefined priorities.
- The preemption discipline can be applied in conjunction with LCFS or Static Priorities. This discipline interrupts or preempts the customer currently being served if a higher priority customer enters the queue [Bolch et al. 2006].

Kendall's notation is widely used to represent queueing systems and provides a compact representation of their characteristics. The symbols A/S/m/N/K/SD are commonly used in the notation, where A denotes the inter-arrival times distribution, S represents the service time distribution, m indicates the number of servers, N denotes the system capacity, K signifies the population size, and SD represents the service discipline [Cooper 1981]. The letter M (Markovian) is used to denote exponential distributions for inter-arrival times and service times.

In some cases, a queueing system may assume infinite system capacity, infinite population size, or FCFS service discipline, allowing for a shorter notation.

2.3.2 Types of Queues

There are several types of queueing systems that are commonly studied and analyzed. The main types are:

- M/M/1 queue: This single-server queue is commonly employed to model systems where a single server serves the customers. In an M/M/1 queue, both the inter-arrival times and service times follow exponential distributions. There are no limitations on the population size or the system capacity, and the adopted service discipline is FCFS [Jain 1991]. The state of the system is represented by the number of customers in it, and the two main parameters are the arrival rate of customers and the service rate [Gross et al. 2008].

- M/M/m queue: The M/M/m queue is a multi-server model where the arrival rate distribution is Poisson, the service times are exponentially distributed, and there are m identical servers, each with the same service capacity. In this system, if at least one server is idle, an arriving customer is immediately served. Otherwise, the customer may have to wait in a buffer before being served. The buffer size is infinite, meaning there is no limit on the number of customers it can accommodate [Jain 1991].

- M/M/m/N queue: This system is similar to the M/M/m queue but has a limited number of users denoted by N, representing the system capacity [Gross et al. 2008]. When m and N have the same value, resulting in M/M/N/N, it indicates that the system has no buffer to hold blocked or interrupted users. Alternative notations, such as M/M/m/0 (Erlang's loss system), are used to represent this special case. These systems are characterized by m identical servers, Poisson input, exponential service times, no waiting positions ($N = 0$), and an unlimited number of customers. Consequently, after reaching full capacity, all new arrivals are blocked. In such cases, the authors consider the effective arrival rate as the difference between the total arrival rate and the blocked arrival rate.

Furthermore, depending on the system being modeled, a priority discipline can be adopted. Prioritized queueing systems prioritize customers, regardless of arrival time. In particular, our work utilizes customer prioritization, being the URLLC requests the higher priority type whereas

eMBB is served with lower priority. Hence, our research employs the basic M/M/m/N queue with some adaptations to address the proposed scenarios. For instance, we assume that waiting jobs are served according to FCFS and we consider that server resources may have different provisioning capacities, one for the URLLC and one for the eMBB. In addition, the processing resource is built to simulate a delayed start which we call the setup time. We refer to this model as an M/M/m/N setup/failure queue.

2.3.3 Continuous-Time Markov Process

In the analysis of queueing systems, Continuous-Time Markov processes are widely employed. These processes possess the Markovian property, wherein the future behavior of the system solely depends on its current state, independent of its past history [Kemeny 1960]. Continuous-Time Markov processes provide a flexible and powerful framework for modeling and analyzing complex queueing systems in mobile communications. In particular, this work uses the continuous-time version which is known as Continuous-time Markov Chain (CTMC). A CTMC is defined by:

- A finite state space Ω
- A transition rate matrix Q with dimensions equal to the state space Ω
- An initial state S such that $X_0 = S$, or a probability distribution for the initial state

For $i \neq j$, the elements q_{ij} are non-negative and represent the rate at which the process transitions from state i to state j. The elements q_{ii} can be chosen as zero, but a common convention sets them such that each row of Q sums to zero for mathematical convenience. Most properties of CTMCs can be derived directly from results about their discrete counterparts, Poisson processes, and exponential distributions [Norris 1998]. Moreover, the analysis of the stationary behavior of a CTMC provides the probability distribution to which the process converges for large time units. In brief, CTMCs are powerful tools for forecasting the stationary state probability (π) of a system. The stationary distribution can be obtained by solving $\pi Q = 0$, subject to the constraint that the sum of the elements equals 1 [Kemeny 1960].

2.4 CHAPTER SUMMARY

In this chapter, a technical background on the 5G and its core enabling technologies was provided. We started by discussing the three service categories of 5G: eMBB, URLLC, and mMTC, focusing on the first two, and exploring their characteristics and applications. Next, we delved into the 5G network architecture, which comprises RAN and the Core. We highlighted the control plane functions, such as the AMF and AF as well as the user plane one. We then introduced the key enabling technologies for 5G, including NS, MEC, NFV, and SDN. These technologies provide the foundation for advanced network architectures, efficient resource allocation, and improved service delivery in 5G networks. Lastly, we discussed queueing theory and its application to the analysis of queueing systems in the MEC-NFV context, covering its concept and notation, different types of queues, and the use of CTMCs for modeling and analyzing queueing systems. Overall, this chapter provided the necessary technical background and insights into the technologies and concepts that will be further explored and applied in the subsequent chapters, focusing on resource allocation for URLLC and eMBB services coexisting in 5G networks.

3 RELATED WORK

This chapter provides a comprehensive survey of the primary analytical models proposed in the existing literature to address the MEC-NFV in the context of the 5G network. The focus is particularly placed on the distinct characteristics addressed by each model, including the specific problem(s) they aim to solve, the types of services involved, and the mathematical tools employed. Additionally, it aims to elucidate the contribution of the present work in relation to the previously developed models. The structure of this chapter is as follows: Section 3.1 discusses the major subcategories of resource allocation problems within the MEC-NFV context. Section 3.1 provides an overview of the key assumptions made by each work regarding the virtual environment. Finally, Section 3.3 delineates the performance metrics adopted by the related works, while presenting a general classification of them.

3.1 ADDRESSED PROBLEMS

A body of existing literature on radio and computational resource issues related to the MEC-NFV architecture encompasses various problem classes, including resource scheduling, Dynamic Resource Allocation (DRA), and resource dimensioning [Li et al. 2021]. In this section, we provide a summary of the main studies in these fields, focusing on the addressed problems, the network segments involved (RAN or Core functions), and the 5G service categories considered. Additionally, since all of the following works are analytical in nature, we also extract the mathematical tools utilized to build their models.

The first three works address the radio resources sharing between two 5G service categories, namely eMBB and URLLC. In [Bairagi et al. 2021], the authors tackle the challenge of sharing radio resources between eMBB and URLLC, involving a trade-off between latency, reliability, and spectral efficiency, using Combinatorial Programming as the main mathematical tool. In [Zhang et al. 2021], a dynamic joint scheduling approach for URLLC and eMBB traffic is proposed at the sub-frame level, incorporating a queuing mechanism to monitor and control the latency of each URLLC packet in real-time to ensure strict requirements. Moreover, in [Kim and Park 2020], the proposal involves an overlapping scheme of puncturing a portion of resources scheduled to an eMBB packet for URLLC packets, resulting in damage to eMBB packets. This work extends the method provided by ITU-R to reflect the puncturing of URLLC

on eMBB and considers additional delays due to retransmissions, utilizing Queueing Theory as the underlying mathematical framework.

The subsequent works in [Tong et al. 2020], [Ma et al. 2021], and [Huang et al. 2021] explore end-to-end characteristics, encompassing both RAN and Core segments. However, they only consider a single service category (URLLC). In particular, [Tong et al. 2020] develops a DRA algorithm that minimizes end-to-end delay while ensuring a minimum service rate and maximum reliability, considering VNF mapping in both the core network and access network to minimize end-to-end delay and ensure network slice reliability. Similarly, in [Ma et al. 2021], the author discusses how to meet the reliability and latency requirements in URLLC using stochastic network calculus (SNC), focusing on mathematical bounds. The paper constructs a tandem model that describes communication in the 5G network and analyzes parameters influencing the delay. Lastly, in [Huang et al. 2021], the paper proposes an NFV-enabled 5G paradigm for industry applications, guaranteeing URLLC through service chain acceleration and dynamic blockchain-based spectrum resource sharing among various applications running on NFV-based equipment.

The remaining works are focused on core network functions and do not consider RAN characteristics. Moreover, it is important to note that none of these works address two or more 5G service categories simultaneously; they are generally dedicated to a single category or agnostic towards a specific category. For instance, in [Emara et al. 2021], an analytical model based on Continuous-Time Markov Chain (CTMC) is proposed along with an optimization problem to determine the optimal number of virtual resources to maximize task execution capacity. The paper focuses on jointly considering contention-based communications for task offloading and parallel computing, as well as the occupation of failure-prone MEC processing resources, without focusing on a specific service category. Similarly, in [Abdelhadi et al. 2022], [Liu et al. 2022], and [Li and Jin 2021], no specific categories are specified. In [Abdelhadi et al. 2022], the authors propose a spatiotemporal framework employing stochastic geometry and CTMC to analyze the intertwined communication and computation performance of edge computing systems. They study the influence of various parameters on task response delay using the incorporated framework. In [Liu et al. 2022], the proposal encompasses an online task offloading and resource allocation approach for edge-cloud orchestrated computing, aiming to minimize the average latency of tasks using a mixed-integer optimal decision approach. Lastly, in [Li and Jin 2021], the paper focuses on the task offloading strategy issue in MEC systems to improve experience quality and increase energy efficiency,

employing a task offloading strategy. The paper establishes a model and formulates a joint optimization problem for the task offloading strategy, investigating the influence of parameters on the task offloading strategy and obtaining optimal results.

The last set of works includes two of our research group's previous studies on mobile dynamic resource allocation. In both works, we focus on the core network and a single service category (URLLC). In [Falcao et al. 2022], we propose an analytical CTMC framework to evaluate a hybrid virtual MEC environment that combines the strengths of Virtual Machines (VMs) and Containers to meet URLLC constraints and provide cloud-like Virtual Network Function (VNF) elasticity. Similarly, in [Souza et al. 2021], we leverage another CTMC-based model to analyze MEC-NFV node configuration, allowing resource pre-initialization to mitigate the negative effects of VNF failures and setup rates. In [Falcão et al. 2023] we design a CMTC model that allows a service provider to properly dimension a MEC-enabled UAV node under availability, power consumption, reliability and latency perspectives. In this dissertation, we once again focus on the DRA problem in the core network segment, utilizing CTMC as the main mathematical tool. However, this work introduces a key aspect that differentiates it from the previously discussed literature. We address two 5G service categories, eMBB and URLLC, in a single model while also considering other key distinctions described in the following sections. Table 2 provides a summary of the related work contributions in terms of the addressed problems, network segments, 5G service categories, and the mathematical branches adopted to model the problems.

Table 2 – Problem, Network Segment, Service Types and Mathematical Tools

Work	Problem	Network Segment	5G service Types	Mathematical Tools
[Bairagi et al. 2021]	Scheduling and DRA	RAN	URLLC, eMBB	Combinatorial Programming
[Zhang et al. 2021]	Scheduling and DRA	RAN	URLLC, eMBB	Queueing Theory
[Kim and Park 2020]	Scheduling and DRA	RAN	URLLC, eMBB	Queueing Theory
[Emara et al. 2021]	DRA	CORE (MEC)	n/a	CTMC, Stochastic Geometry
[Tong et al. 2020]	DRA	RAN and CORE	URLLC	Graph Theory
[Ma et al. 2021]	Delay Bound	RAN and CORE	URLLC	Queueing Theory, SNC
[Abdelhadi et al. 2022]	Offloading	CORE (MEC)	n/a	CTMC, Sthochastic Geometry
[Liu et al. 2022]	Offloading and DRA	CORE (MEC)	n/a	n/a
[Li and Jin 2021]	DRA	CORE (MEC)	n/a	Queueing Theory
[Huang et al. 2021]	DRA	RAN, CORE	URLLC	SNC
[Falcao et al. 2022]	DRA	CORE (MEC)	URLLC	CTMC
[Souza et al. 2021]	DRA	CORE (MEC)	URLLC	CTMC
[Falcão et al. 2023]	DRA	CORE (MEC)/UAV	URLLC	CTMC
This Work	DRA	CORE (MEC)	URLLC, eMBB	CTMC

Source: The author (2023)

3.2 MODEL ASSUMPTIONS

There is no current consensus on the size, computational power, or virtualization technology that is more appropriate for the MEC-NFV architecture with the objective of providing services to the 5G networks. [Santoyo-Gonzalez and Cervello-Pastor 2018]. The decision on these aspects may be based on technical and business parameters such as available site facilities, supported applications and their requirements, estimated user load, operation and deployment costs [Kekki and Featherstone 2018]. However, container-based virtualization has been getting momentum, although the literature still offers works that are agnostic towards a given virtualization technology, which denotes a certain lack of commitment to the feasibility of their propositions. More importantly, the current virtualization technology lacks appropriate adjustments to accommodate the data volume and specific requirements associated with 5G service categories. Therefore, it is crucial to consider events that may hinder the communication process, such as container failures and setups. In this section, we further evaluate the works described in Section 3.1, with a focus on the key considerations regarding container usage in the MEC-NFV architecture, aiming to provide more realistic analytical models.

The primary challenge of utilizing containers in the MEC-NFV infrastructure of mobile communications lies in their maturity for this domain. Containerization introduces multiple security risks, as all containers within an OS share a single kernel. Consequently, a breach in the OS kernel can compromise all dependent containers. Furthermore, isolating faults within containers is not trivial, and a fault can be replicated across subsequent instances. In addition to failures, we evaluate two other phenomena: the VNF instantiation, which represents the delay until a VNF is ready to process a request after being turned off, and a repair time, which denotes the duration taken for a VNF to recover from a failure event. However, it is still common to find works that do not consider these aspects, as observed in [Bairagi et al. 2021] and in [Li and Jin 2021]. Neglecting these factors can be problematic, as they directly impact the main objectives of the research. For instance, if a resource dimensioning strategy fails to account for the possibility of resource failures, the resulting node size is likely to be underestimated. Moreover, some studies do consider failure events but do not associate them with repair times, as seen in [Liu et al. 2022], [Huang et al. 2021], and in [Zhang et al. 2021]. This omission may impact metrics such as resource availability and power consumption. Finally, in this dissertation, we adopt the considerations from the aforementioned set of previous works: [Abdelhadi et al. 2022], [Falcao et al. 2022], [Falcão et al. 2023]and [Souza et al. 2021], as

they provide a satisfactory approach to address the evaluated events, encompassing all three phenomena. Table 3 summarizes the assumptions made by each evaluated work regarding these aspects.

Table 3 – Model Assumptions

Work	Instantiation Time	Failure	Repair Time
[Bairagi et al. 2021]	✗	✗	✗
[Zhang et al. 2021]	✗	✓	✗
[Kim and Park 2020]	✗	✓	✓
[Emara et al. 2021]	✗	✓	✓
[Tong et al. 2020]	✗	✓	✓
[Ma et al. 2021]	✗	✓	✓
[Abdelhadi et al. 2022]	✗	✓	✗
[Liu et al. 2022]	✗	✓	✗
[Li and Jin 2021]	✗	✗	✗
[Huang et al. 2021]	✗	✓	✗
[Falcao et al. 2022]	✓	✓	✓
[Souza et al. 2021]	✓	✓	✓
[Falcão et al. 2023]	✓	✓	✓
This Work	✓	✓	✓

Source: The author (2023)

3.3 EVALUATION METRICS

Since the introduction of 3GPP Release 16 [3GPP 2020], significant attention has been given to potential architecture enhancements aimed at supporting URLLC services through MEC and NFV. In addition to the fundamental metrics of latency and reliability, the literature explores various other metrics, such as resource availability, which is crucial for resource provisioning and dimensioning schemes, as well as energy-related metrics, which are of particular interest to infrastructure providers. Furthermore, it is important to note that the interpretation of metrics may vary depending on the network segment being analyzed. In this section, we provide an overview of the principal metrics examined in some of the works discussed in this chapter. Specifically, we focus on four metrics: Availability, Reliability, Energy Consumption, and Latency, although certain works may encompass additional metrics.

In the studies characterizing the 5G network RAN ([Bairagi et al. 2021], [Zhang et al. 2021], and [Kim and Park 2020]), the definitions of latency and reliability differ from those applicable to the Core network or edge. Among these works, [Bairagi et al. 2021] concentrates solely on latency, while [Zhang et al. 2021] examines reliability exclusively. Notably, [Kim and Park 2020]

is the sole study to simultaneously consider both reliability and latency, which aligns with the requirements of URLLC. The remaining works primarily focus on the backhaul, which leads to differences in the interpretation of certain performance metrics compared to the RAN. Latency-related metrics are commonly evaluated in all of these works, as seen in [Abdelhadi et al. 2022]. However, it is more common to find evaluations involving two or more metrics simultaneously. For instance, in [Emara et al. 2021], the authors explore availability and reliability while imposing an energy constraint per device. Nevertheless, since no dedicated formulation for the energy metric is provided, it is not considered as a distinct metric, but rather as a constraint. In contrast, [Ma et al. 2021], [Tong et al. 2020], and [Huang et al. 2021] focus solely on reliability and latency. Furthermore, [Liu et al. 2022] and [Li and Jin 2021] both evaluate the energy-related metric alongside latency.

The subsequent set of works represents a more comprehensive approach to metrics, as they address three or more metrics. For example, [Souza et al. 2021] evaluates three metrics: availability, energy, and latency. Similarly, [Falcao et al. 2022] [Falcão et al. 2023] consider all four metrics. In our work, we evaluate three metrics, excluding reliability, which is adopted as an input parameter (failure rate) and its value consequently is reflected in the system performance when a homogeneous virtualization technology is employed. Therefore, this work focuses on availability, power consumption, and latency analysis, taking into account both eMBB and URLLC service types. Table 4 summarizes the related works based on their metrics.

Table 4 – Evaluation Metrics

Work	Availability	Reliability	Energy	Latency
[Bairagi et al. 2021]	✗	✗	✗	✓
[Zhang et al. 2021]	✗	✓	✗	✗
[Kim and Park 2020]	✗	✓	✗	✓
[Emara et al. 2021]	✓	✓	✗	✗
[Tong et al. 2020]	✗	✓	✗	✓
[Ma et al. 2021]	✗	✓	✗	✓
[Abdelhadi et al. 2022]	✗	✗	✗	✓
[Liu et al. 2022]	✗	✗	✓	✓
[Li and Jin 2021]	✗	✗	✓	✓
[Huang et al. 2021]	✗	✓	✗	✓
[Falcao et al. 2022]	✓	✓	✓	✓
[Souza et al. 2021]	✓	✗	✓	✓
[Falcão et al. 2023]	✓	✓	✓	✓
This Work	✓	✗	✓	✓

Source: The author (2023)

3.4 CHAPTER SUMMARY

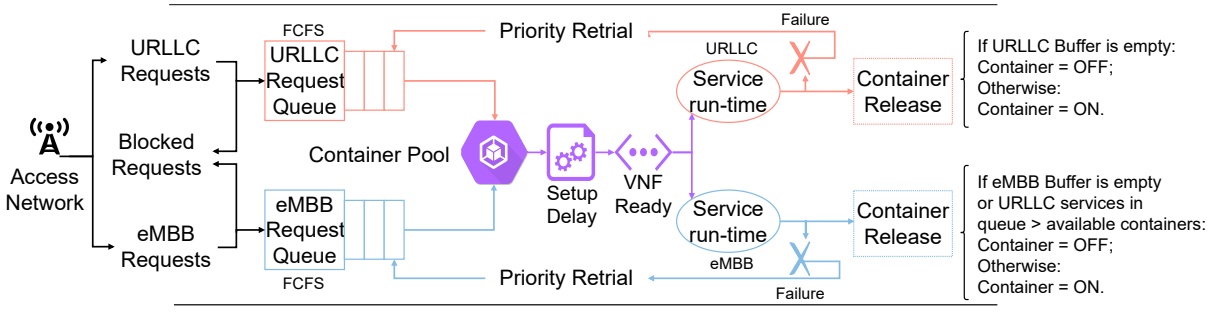
This chapter provided a comprehensive analysis of the analytical models proposed in the existing literature for addressing MEC-NFV nodes in the context of the 5G network. The chapter focused on the distinct characteristics addressed by each model, including the specific problem(s) they aim to solve, the types of services involved, and the mathematical tools employed. The chapter also highlighted the contribution of the present work in relation to previously developed models.

4 SYSTEM MODEL

Building upon the characteristics discussed in Chapter 3, this chapter describes a CTMC-based analytical representation for a single node NFV-MEC, assuming a virtual environment featured with containers that are able to process both URLLC and eMBB requests and that are prone to events such as container setup, failures, and repair times. This chapter is structured as follows. First, this introduction outlines the main events related to the computing model and failure/repair characteristics. Sections 4.1-4.7 detail the formulation/modeling itself, providing the equations and main conditions divided into groups to facilitate comprehension. Lastly, Section 4.8 describes the formulation for the adopted performance metrics, and Section 4.9 Summarizes this chapter.

Analytical models serve as valuable tools for efficiently evaluating large-scale distributed MEC infrastructure projects since simulation and testbeds, which require thousands of Edge Nodes, may not always be feasible. In this study, we assess the performance of a single isolated MEC node, as illustrated in Figure 6, where both eMBB (blue flow) or URLLC (red flow) requests originated from UEs are processed by the RAN, are passed on to the MEC node and handled by containerized VNFs, which are scaled accordingly. This model was designed in isolation from RAN, Core, and Central Cloud, i.e., rather than accounting for multiple network path subparts; hence, the only uncertainty is related to the virtual components themselves, i.e., setup, failure, and repair events.

Figure 6 – Edge Node



Source: The author (2023)

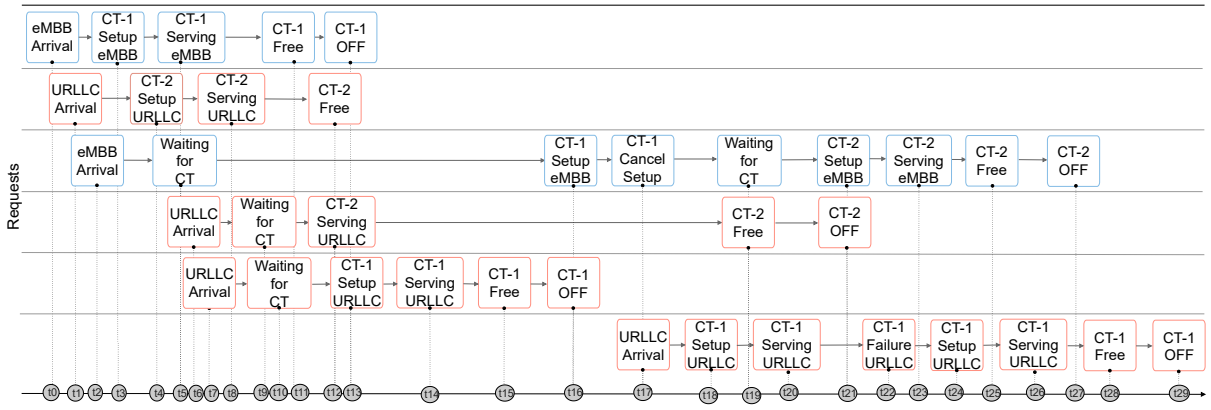
The system consists of a finite number of containers and buffer positions that can be allocated to each type. In our model, each VNF runs equally and independently on a single container, and a centralized control unit determines if requests are admitted or blocked. A request admission occurs if there are enough resources, i.e., if either containers or buffer

positions are available, thus, if admitted, each request may be processed or queued, depending on the resource availability.

With regards to the auto-scaling mechanism for the VNFs, a dynamic VNF auto-scaling strategy was embedded into our formulation to help cope with the sudden load increase caused by the intensive requests, especially caused by the URLLC service category. In other words, before the proper processing phase, the containerized-VNF must be initialized, which incurs a delay called setup time. In addition, the possibility of failure during service and its respective repair time is also embedded in our formulation. In this case, the containerized VNF is restarted, and the request is either reallocated to another available container or, if there are no available resources, it is placed back in its respective service queue with higher priority than new requests. In both cases, the service processing is restarted.

Moreover, following the main sources and as they are latency-sensitive, we enable URLLC services to be prioritized over eMBB. In terms of this prioritization, the following policy has been adopted: (1) If there are both URLLC and eMBB services to be served, URLLC services have higher priority, thus, the containers that are being released or activated are allocated first to URLLC services. (2) In the case where there is a URLLC service waiting in queue for available resources and an eMBB service has been completed, the released container is restarted to be used by the URLLC service. However, if there are other available containers, the current one will be allocated to a sequential eMBB service or deactivated if the eMBB queue is empty. (3) Preemption of the lower-priority service (eMBB) that is being processed is not allowed. A step-by-step description of some of these events can be found in the following lines that explain Fig. 7.

Figure 7 – Figura de Exemplo

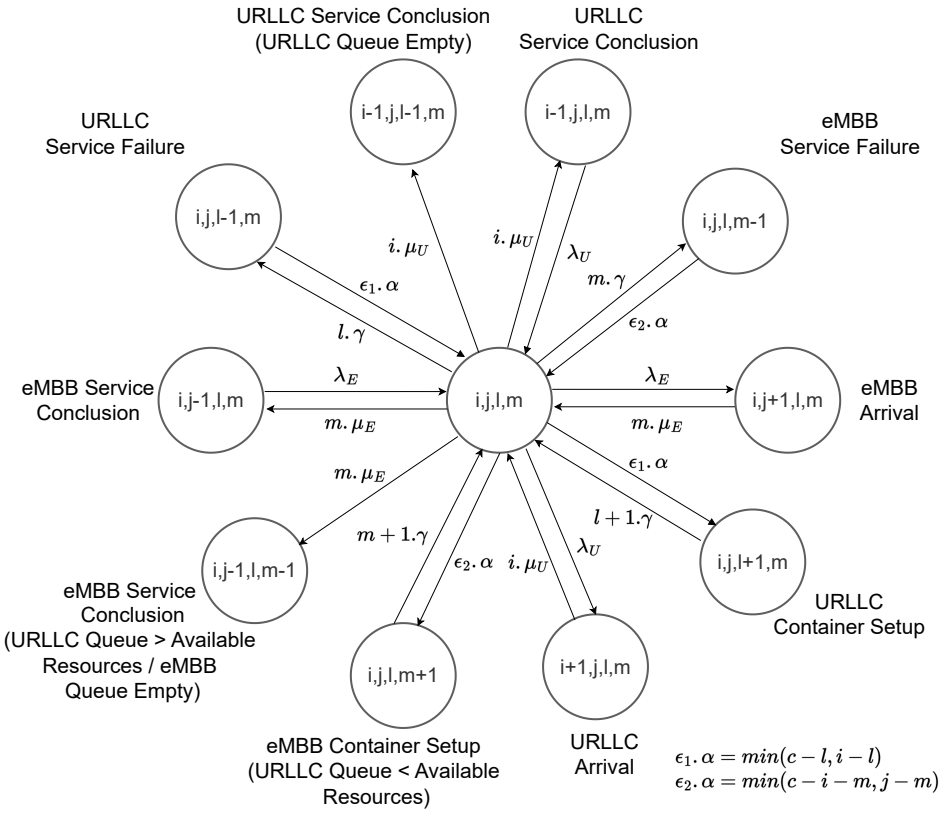


Source: The author (2023)

Fig. 7 describes a small MEC node comprised of only two containers. The first three events are regular service requests, being t0 the 1st eMBB request, $t1$ the 1st URLLC request and $t2$ another eMBB (2o eMBB). However, since there are only two containers ($CT1$ and $CT2$), only the 1st eMBB and 1st URLLC requests are allocated to each available container ($CT1$ and $CT2$) in $t3$ and $t4$, respectively, while the 2nd eMBB service is placed in a dedicated buffer. This triggers a setup phase due to each container initialization, hence a waiting period is set until the resource is ready; the service is only processed if the setup is successful, such as in $t5$ ($CT1$ finishes setup phase and starts processing the 1st eMBB service) and $t8$ ($CT2$ finishes setup phase and starts processing the 1st URLLC service). Moreover, during the setup intervals, other two URLLC arrivals happen in $t6$ (2nd URLLC) and $t7$ (3rd URLLC), being placed in the URLLC buffer since both containers are currently processing other requests. Up to $t10$, the system is processing an eMBB request in $CT1$ and a URLLC in $CT2$ while holding a single eMBB and two URLLC services in their buffers. Furthermore in Fig. 7, in $t11$, CT1 completes processing the first eMBB service, becoming available. The same happens in $CT2$ in $t12$, where the first URLLC service is completed, however, in this case, $CT2$ uses fast allocation to start serving the second URLLC service that was buffered. On the other hand, in $t13$ $CT1$ is reinitialized to begin serving the third URLLC, which was also buffered. This happens since $CT1$ switches its image and internal components from eMBB to an URLLC service. Only then, in $t14$, $CT1$ starts processing the third URLLC service that ends right after in $t15$. Now in $t16$, again $CT1$ needs to transition from attending an URLLC service to start another eMBB service, which was in the buffer. However, in $t17$, a new URLLC arrival cancels the setup phase for the $CT1$ due to the higher priority given to URLLC services. Hence, in $t18$, $CT1$ begins another setup phase, but this time to address the newly arrived URLLC service (4th URLLC service). The last set of events in Fig. 7 begins at $t19$, where $CT2$ finishes processing the 2nd URLLC service, while at $t20$, the 4th URLLC service starts to be processed by $CT1$. Furthermore, at $t21$, $CT2$ transitions to begin processing the 2nd eMBB service, which was buffered, incurring a new setup period, only to properly begin processing it at $t23$. At $t22$, a failure occurs during the fourth URLLC service in $CT1$, triggering a new setup period at $t24$. At $t25$, $CT2$ finishes processing the 2nd eMBB service. At $t26$, $CT1$, which experienced a failure and was restarted, begins serving the 4th URLLC service. At $t27$, $CT2$ shuts down since there are no services left to process. At $t28$, $CT1$ finishes processing the last service (4th URLLC), and since there are no services left to process, it also shuts down at $t29$.

Following the above description, the system is modeled using an M/N/c/k+K queue with two types of users, prioritization, failure, initialization time, FCFS service discipline, and a limited buffer for each user type. The model states are represented by the tuple $\pi(i, j, l, m)$, where $i, j, l, m \in N$, with i and j denoting the number of URLLC and eMBB services, and l and m denoting the number of active containers for each user type, with $l+m$ being smaller or equal to the maximum number of containers (c). The service arrivals follow a Poisson process with rate λ_u for URLLC services and λ_e for eMBB. The service is provided by the c available containers, with an exponentially distributed service time with rates μ_u for URLLC and μ_e for eMBB. Similarly, the failure occurrence and container initialization time follow exponential distributions with rates γ and α, respectively. Fig. 8 summarizes all possible CTMC transitions and states of the proposed system, with its respective parameters. Hence, each of the following sections 4.1-4.7 describes a group of states derived from this figure.

Figure 8 – Generic CTMC state



Source: The author (2023)

The possible state space is given by $\Omega = \{(i, j, l, m|0 \leq i \leq k, 0 \leq j \leq K, 0 \leq l \leq c, 0 \leq m \leq c, (m + l) \leq c,$ since $, 3 \leq c, c < k, c < K, l \leq i$ and $m \leq j\}$. To derive system evaluation metrics, the probability of states with the system in stationary state $\pi(i, j, l, m)$

needs to be found, which can be done by solving the linear system formed by 4.0.1 normalization condition and (4.1.1 - 4.7.24) the flow balance equations (inflow equal to outflow).

$$\Sigma_{(i,j,l,m)\in\Omega}\pi(i,j,l,m) = 1. \tag{4.0.1}$$

4.1 STATE $(0,0,0,0)$ (EMPTY SYSTEM)

This section describes the single equation that represents the empty system (Eq. 4.1.1), i.e., a state without URLLC and eMBB users in the system, and no active containers. In summary, state $(0,0,0,0)$, where $i = j = l = m = 0$. Fig. 9 shows this state and its neighboring states and Table 5 presents the related events.

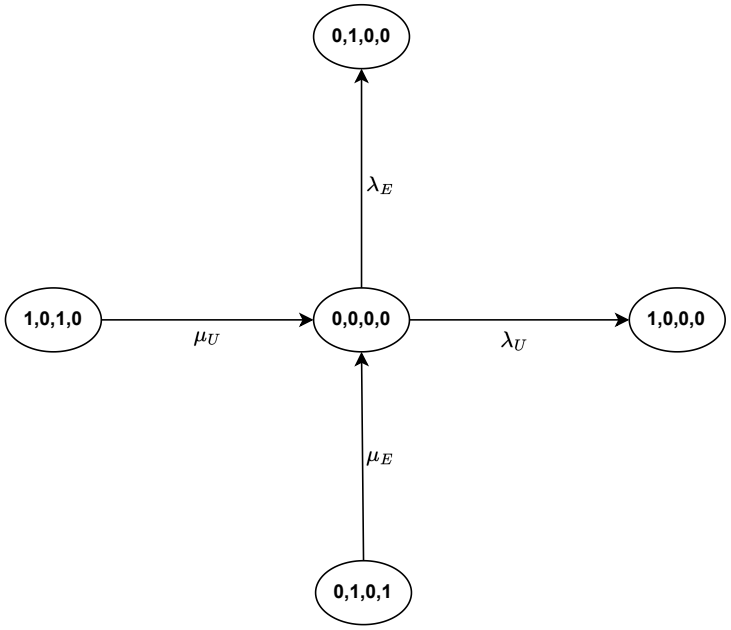
Table 5 – Events related to the states $i = 0$, $j = 0$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✗
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✗

Source: The author (2023)

$$[\lambda_U + \lambda_E]\pi(0,0,0,0) = \mu_U\pi(1,0,1,0) + \mu_E\pi(0,1,0,1) \tag{4.1.1}$$

Figure 9 – State $(0,0,0,0)$, with $i=0$, $j=0$ and $l=m=0$



Source: The author (2023)

4.2 STATES $(i,j,0,0)$, $with\ 0 \leq i \leq k$, $0 \leq j \leq K$, $l=0$, $and\ m=0$

This section presents the equations that refer to the states with at least one user (URLLC or eMBB), no active containers, and the number of URLLC and eMBB users lower than or equal to his respective limits (k and K), i.e., states $(i,j,0,0)$, with $0 \leq i \leq k$, $0 \leq j \leq K$, $l=0$, and $m=0$. These states may be divided into the following groups defined by Equations 4.2.1 - 4.2.10.

States with at least one URLLC user and without eMBB users in the system, with the amount of URLLC users lower than his respective limit (k) and no active containers. In summary, states $(i,0,0,0)$, with $0<i<k$, $j=0$ and $l=m=0$, with balance equation, state diagram, and related events given by Eq. 4.2.1, Fig. 10, and Table 6, respectively.

$$[\lambda_E + \lambda_U + min(i,c)\alpha]\pi(i,0,0,0) = \lambda_U\pi(i-1,0,0,0) + \gamma\pi(i,0,1,0) + \mu_E\pi(i,1,0,1) \tag{4.2.1}$$

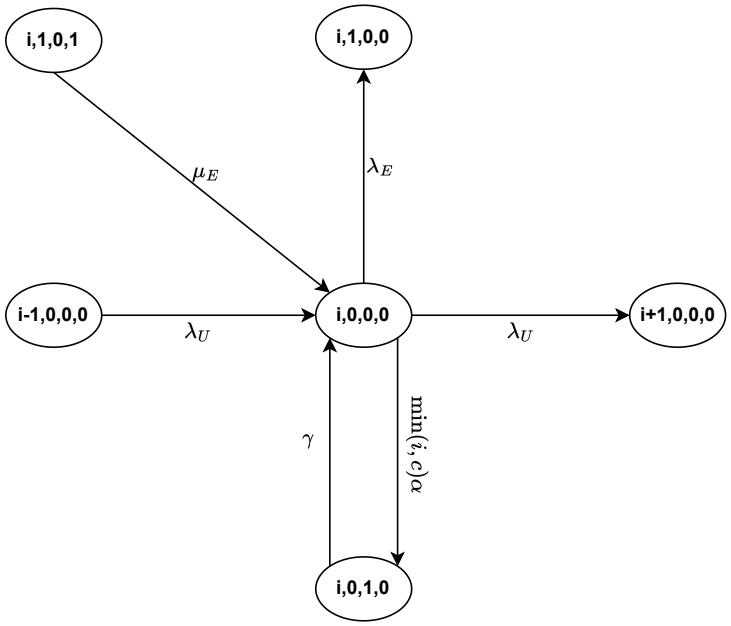
States with at least one eMBB user and without URLLC users in the system, with the amount of eMBB users lower than his respective limit (K) and no active containers (see Fig. 11). In summary, states $(0,j,0,0)$, with $i=k$, $1<j<K$ and $l=m=0$ as in Table 7 and Eq. 4.2.2.

Table 6 – Events related to the states$(i,0,0,0)$, with $0<i<k$ and $j=l=m=0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 10 – State $(i,0,0,0)$, with $i<k$, $j=0$ and $l=m=0$



Source: The author (2023)

$$[\lambda_E+\lambda_U+min(j,c)\alpha]\pi(0,j,0,0)=\lambda_E\pi(0,j-1,0,0)+\gamma\pi(0,j,0,1)+\mu_U\pi(1,j,1,0) \tag{4.2.2}$$

State in which the limit for URLLC users has been achieved without eMBB users in the system, with no active containers (see Fig. 12), i.e., state $(k,0,0,0)$, with $i=k$, $j=0$ and $l=m=0$. It follows the Eq. 4.2.3 and its related events are listed in Table 8.

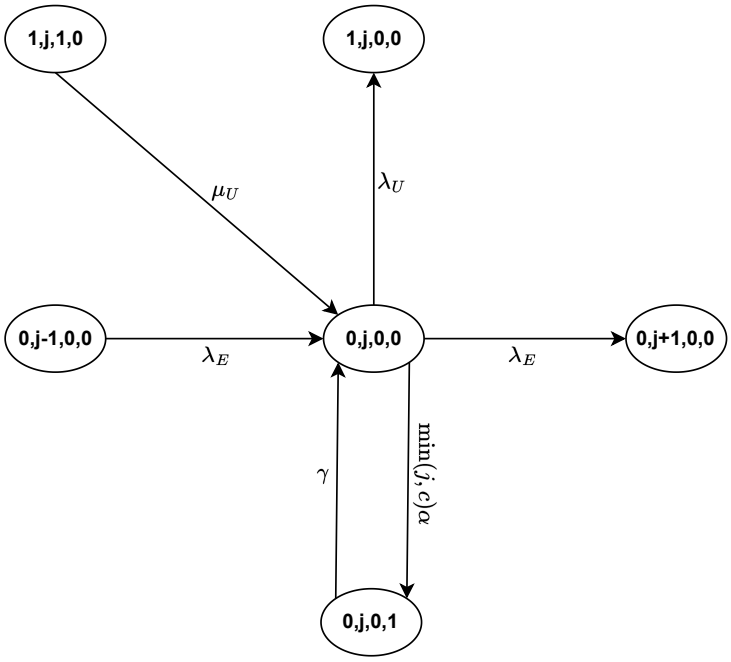
$$[\lambda_E+min(k,c)\alpha]\pi(k,0,0,0)=\lambda_U\pi(k-1,0,0,0)+\gamma\pi(k,0,1,0)+\mu_E\pi(k,1,0,1) \tag{4.2.3}$$

Table 7 – Events related to the states $(0, j, 0, 0)$, with $i = k$, $1 < j < K$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 11 – State $(k, 0, 0, 0)$, with $i = k$, $j = 0$ and $l = m = 0$



Source: The author (2023)

State in which the limit for eMBB users has been achieved without URLLC users in the system, with no active containers, as shown in Fig. 13. In summary, state $(0, K, 0, 0)$, with $i = 0$, $j = K$ and $l = m = 0$, whose related events and balance equation are presented in Table 9 and Eq. 4.2.4, respectively.

$$[\lambda_U + min(K, c)\alpha]\pi(0, K, 0, 0) = \lambda_E\pi(0, K-1, 0, 0) + \gamma\pi(0, K, 0, 1) + \mu_U\pi(1, K, 1, 0) \tag{4.2.4}$$

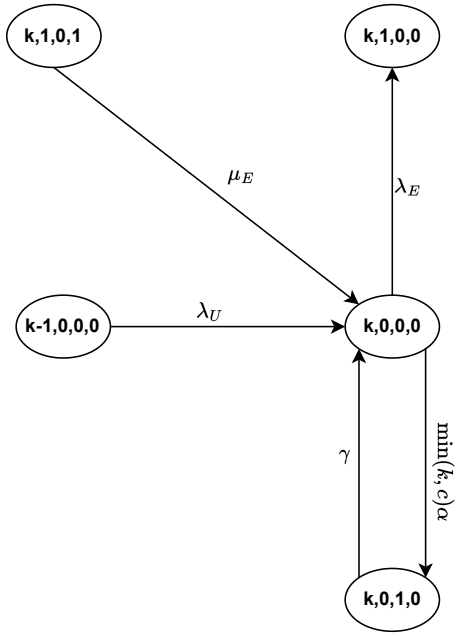
States in which the limit for URLLC and eMBB users has been achieved, with no active containers (see Fig. 14). Note that the URLLC services have higher priority than eMBB ones in

Table 8 – Events related to the states $(k, 0, 0, 0)$, with $i = k$, $j = 0$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 12 – State $(k, 0, 0, 0)$, with $i = k$, $j = 0$ and $l = m = 0$



Source: The author (2023)

the active container allocation. In brief, states $(k, K, 0, 0)$, with $i = k$, $j = K$ and $l = m = 0$ as in Table 10 and Eq. 4.2.5.

$$[min(k, c)\alpha]\pi(k, K, 0, 0) = \lambda_U\pi(k-1, K, 0, 0) + \lambda_E\pi(k, K-1, 0, 0) + \gamma\pi(k, K, 1, 0) + \gamma\pi(k, K, 0, 1) \quad (4.2.5)$$

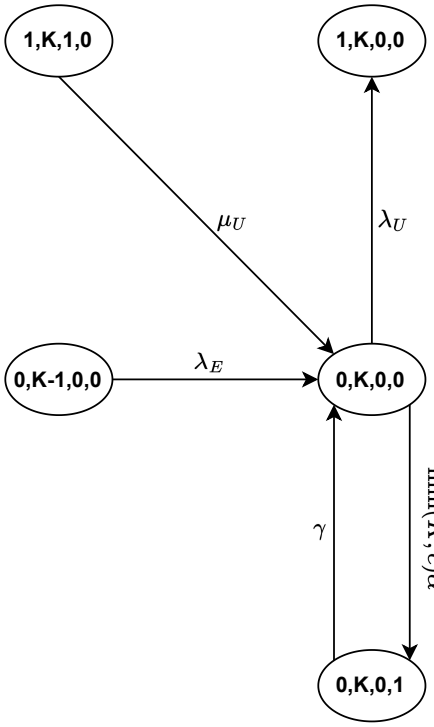
States with at least one and no more than $k-1$ URLLC users, K eMBB users, no active containers and the number of available containers are insufficient to be activated to process URLLC services in queue (see Fig. 15). Eq. 4.2.6 describes these states $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$ and $(c <= i)$. Table 11 denotes their related events.

Table 9 – Events related to the states $i = 0$, $j = K$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 13 – State $(0, K, 0, 0)$, with $i = 0$, $j = K$ and $l = m = 0$



Source: The author (2023)

$$[\lambda_U + min(i,c)\alpha]\pi(i,K,0,0) = \lambda_U\pi(i-1,K,0,0) + \lambda_E\pi(i,K-1,0,0) + \gamma\pi(i,K,1,0) + \gamma\pi(i,K,0,1) \quad (4.2.6)$$

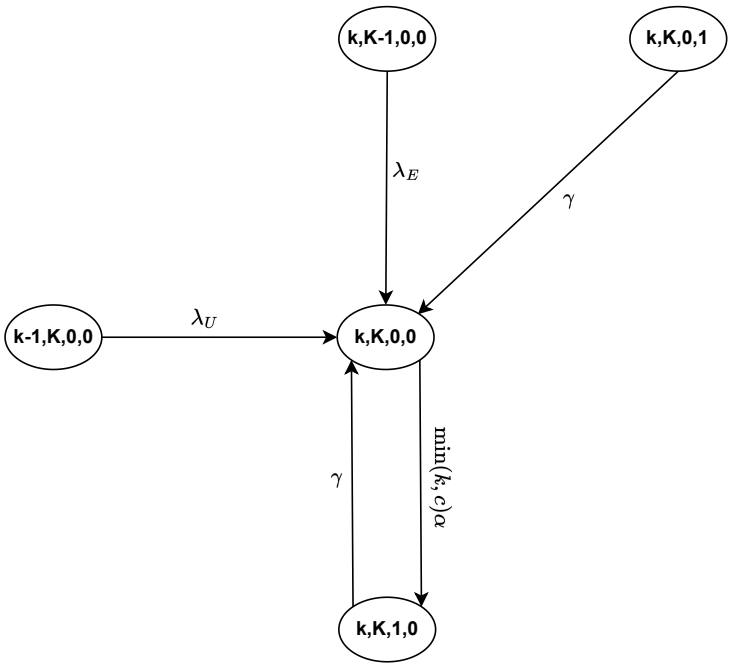
States with at least one and no more than $k-1$ URLLC users, K eMBB users, no active containers and the number of available containers are sufficient to be activated to process URLLC services in queue (see Fig. 16). In short, states $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$, and $(c > i)$. These states follow the balance equation given in Eq.4.2.7 and their

Table 10 – Events related to the states $(k, K, 0, 0)$, with $i = k$, $j = K$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 14 – States $(k, K, 0, 0)$, with $i = k$, $j = K$ and $l = m = 0$



Source: The author (2023)

related events are listed in Table 12.

$$[\lambda_U + min(i,c)\alpha + min(c-i,K)\alpha]\pi(i,K,0,0) = \lambda_U\pi(i-1,K,0,0) + \lambda_E\pi(i,K-1,0,0) + \gamma\pi(i,K,1,0) + \gamma\pi(i,K,0,1) \quad (4.2.7)$$

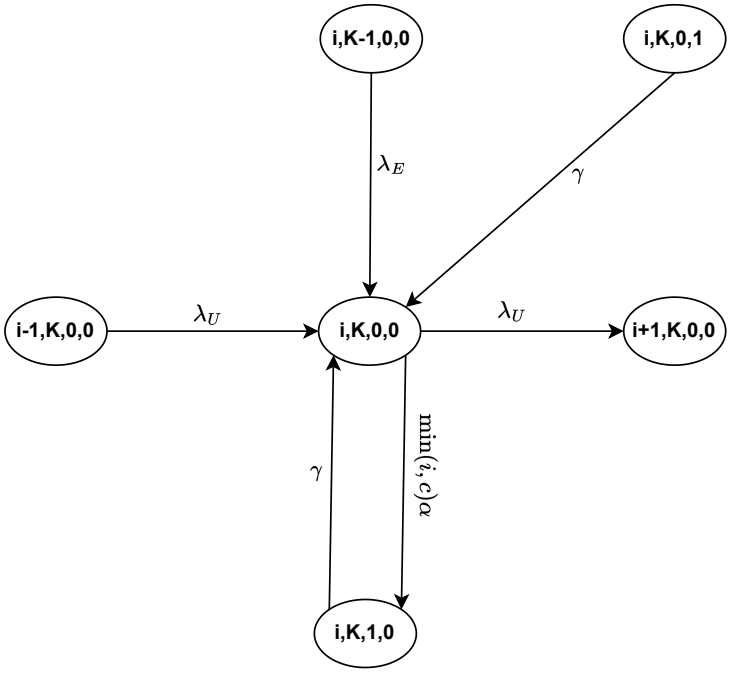
States in which the limit for URLLC users has been achieved and the number of eMBB users is more than 0 and lower than his limits, with no active containers (see Fig. 17). In summary, states $(k, j, 0, 0)$, with $i = k$, $0 < j < K$ and $l = m = 0$, whose balance equation and related event are denoted in Eq. 4.2.8 and Table 13.

Table 11 – Events related to the states $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$ and $(c <= i)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 15 – States $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$ and $(c <= i)$



Source: The author (2023)

$$[min(k,c)\alpha + \lambda_E]\pi(k,j,0,0) = \lambda_E\pi(k,j-1,0,0) + \lambda_U\pi(k-1,j,0,0) + \gamma\pi(k,j,1,0) + \mu_E\pi(k,j+1,0,1) + \gamma\pi(k,j,0,1) \quad (4.2.8)$$

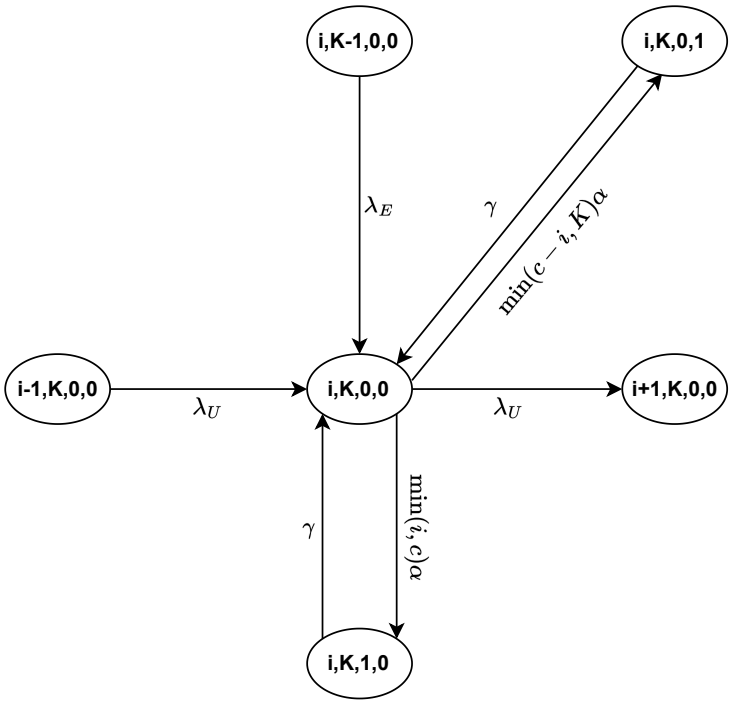
States with at least one user of each type, but with the number of URLLC and eMBB users lower than k and K, respectively, no active containers and the number of available containers are insufficient to be activated to process URLLC services in queue $(c <= i)$. In essence, states $(i, j, 0, 0)$, with $0 < i < k$, $0 < j < K$, $l = m = 0$ and $(c <= i)$ as in Table 14. The state diagram and balance equation that denote these states are shown in Fig. 18 and Eq. 4.2.9.

Table 12 – Events related to the states $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$ and $(c > i)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 16 – States $(i, K, 0, 0)$, with $0 < i < k$, $j = K$, $l = m = 0$ and $(c > i)$



Source: The author (2023)

$$[\lambda_U + \lambda_E + min(i,c)\alpha]\pi(i,j,0,0) = \lambda_U\pi(i-1,j,0,0) + \lambda_E\pi(i,j-1,0,0) + \gamma\pi(i,j,1,0) + \mu_E(i,j+1,0,1) + \gamma\pi(i,j,0,1) \quad (4.2.9)$$

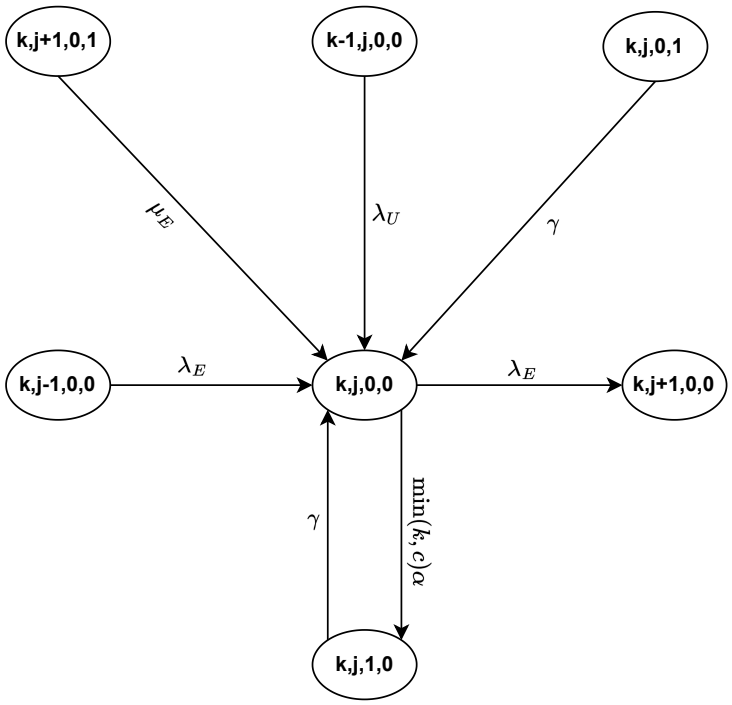
States with at least one user of each type, but with the number of URLLC and eMBB users lower than k and K, respectively, no active containers and the number of available containers are sufficient to be activated to process URLLC services in queue $(c > i)$. In short, states $(i, j, 0, 0)$, states with $0 < i < k$, $0 < j < K$, $l = m = 0$ and $(c > i)$. Fig 19 illustrates the

Table 13 – Events related to the states $(k, j, 0, 0)$, with $i = k$, $0 < j < K$ and $l = m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 17 – States $(k, j, 0, 0)$, with $i = k$, $0 < j < K$ and $l = m = 0$



Source: The author (2023)

state diagram of these states and Table 15 summarizes their related events. Additionally, their balance equation is given by Eq. 4.2.10.

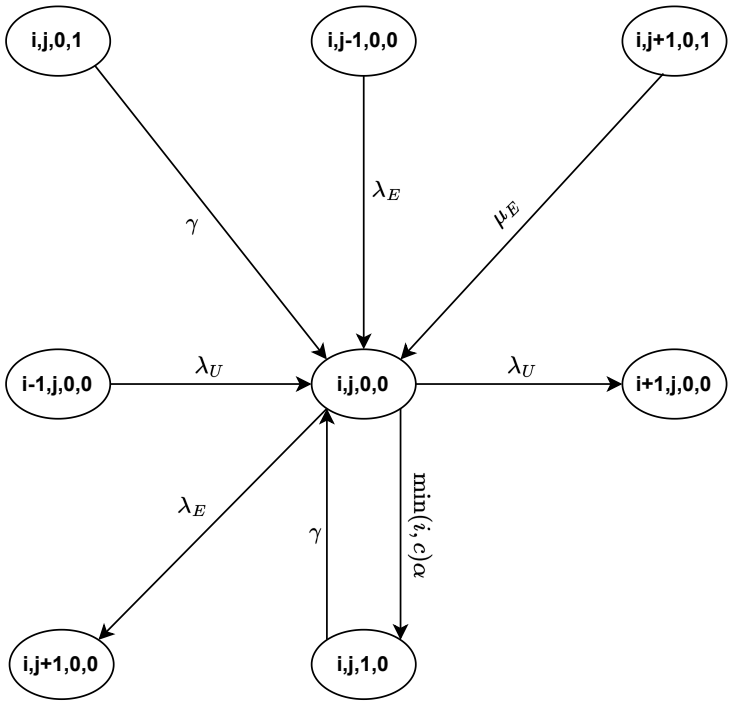
$$[\lambda_U + \lambda_E + min(i,c)\alpha + min(c-i,j)\alpha]\pi(i,j,0,0) = \lambda_U\pi(i-1,j,0,0) + \lambda_E\pi(i,j-1,0,0) + \gamma\pi(i,j,1,0) + \gamma\pi(i,j,0,1) \quad (4.2.10)$$

Table 14 – Events related to the states $(i, j, 0, 0)$, with $0 < i < k$, $0 < j < K$, $l = m = 0$ and $(c <= i)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 18 – States $(i, j, 0, 0)$, with $0 < i < k$, $0 < j < K$, $l = m = 0$ and $(c <= i)$



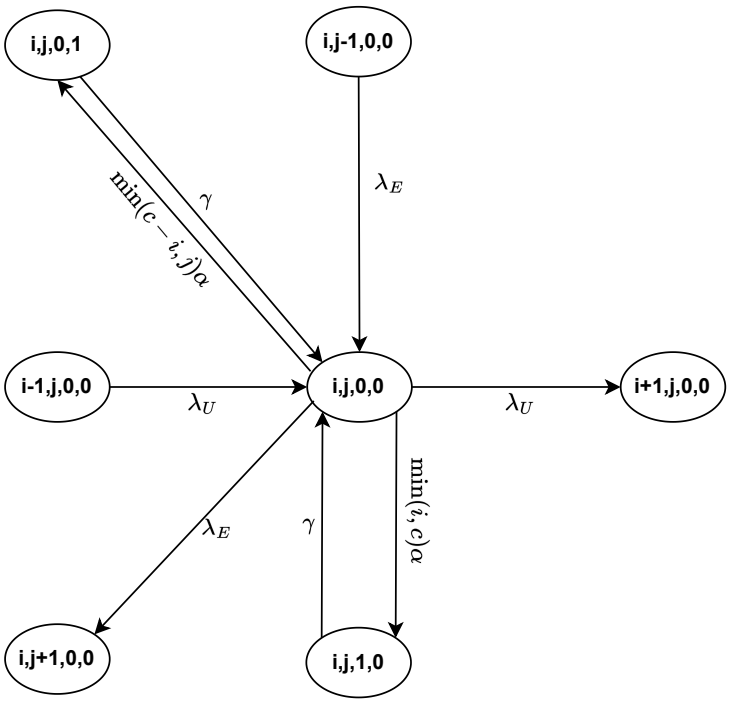
Source: The author (2023)

Table 15 – Events related to the states $(i, j, 0, 0)$, states with $0 < i < k$, $0 < j < K$, $l = m = 0$ and $(c > i)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 19 – States $(i, j, 0, 0)$, with $0 < i < k$; $0 < j < K$ and $l = m = 0$; $(c > i)$



Source: The author (2023)

4.3 STATES $(i, 0, l, 0)$, $with\ 0 < i \leq k$, $0 < l \leq c$, $and\ j = m = 0$

This section describes the states with at least one URLLC user being served by a container and no eMBB users in the system, i.e., states $(i, 0, l, 0)$, with $0 < i \leq k$, $0 < l \leq c$, $k > c$, and $j = m = 0$. They are grouped as follows.

States in which the number of URLLC users is lower than the number of available containers to be activated and all users are being served, i.e., the number of URLCC users matches the number of active containers, with no more than $c - 1$ active containers (see Fig. 20). In summary, states $(i, 0, i, 0)$, with $0 < i < c$ and $l = i$, whose balance equation and related events are denoted in Eq. 4.3.1 and Table 16.

Table 16 – Events related to the states $(i, 0, i, 0)$, with $0 < i < c$ and $l = i$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

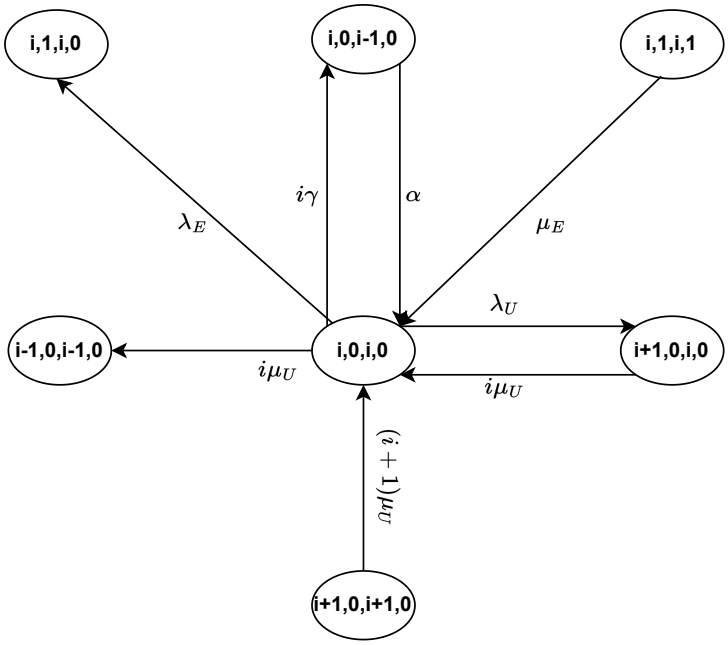
$$
\begin{aligned}
&[\lambda_U + \lambda_E + i(\mu_U + \gamma)]\pi(i, 0, i, 0) \\
&= \mu_E \pi(i, 1, i, 1) + \alpha\pi(i, 0, i-1, 0) + (i+1)\mu_U \pi(i+1, 0, i+1, 0) + i\mu_U \pi(i+1, 0, i, 0)
\end{aligned} \tag{4.3.1}
$$

State in which all containers are busy, serving URLLC users, and there are no URLLC users waiting in line. In summary, states $(c, 0, c, 0)$, with $i = c$ and $l = c$, whose balance equation, state diagram, and related events are given by Eq. 4.3.2, Fig. 21, and Table 17, respectively.

$$
[\lambda_U + \lambda_E + c(\mu_U + -)] - (c, 0, c, 0) = \alpha - (c, 0, c-1, 0) + c\mu_U - (c+1, 0, c, 0) \tag{4.3.2}
$$

States in which all containers are busy, serving URLLC users, and there are also URLLC users waiting in line, with at least two rooms-queue. In summary, states $(i, 0, c, 0)$, with $c <$

Figure 20 – States $(i, 0, i, 0)$, with $0 < i < c$ and $l = i$



Source: The author (2023)

Table 17 – Events related to the states $(c, 0, c, 0)$, with $i = c$ and $l = c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

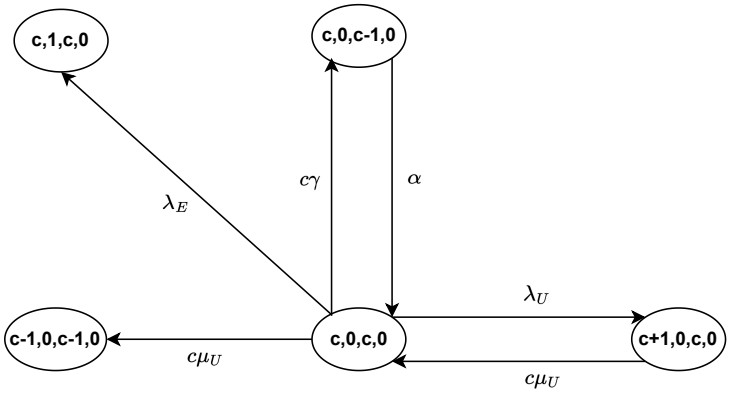
Source: The author (2023)

$i < k$, $l = c$ and $k - c > 1$ as in Table 18. The state diagram and balance equation that denote these states are shown in Fig. 22 and Eq. 4.3.3. Table 18 presents the events related to these states.

$$[\lambda_U+\lambda_E+c(\mu_U+\gamma)]\pi(i,0,c,0) = \lambda_U\pi(i-1,0,c,0)+\alpha\pi(i,0,c-1,0)+c\mu_U\pi(i+1,0,c,0) \tag{4.3.3}$$

State in which all containers are busy, serving URLLC users, and there are k URLLC users (maximum number of URLLC users) in the system waiting in line. In summary, states $(k, 0, c, 0)$, with $i = k$ and $j = c$. Fig 23 illustrates the state diagram of these states and Table

Figure 21 – States $(c, 0, c, 0)$, with $i = c$ and $l = c$



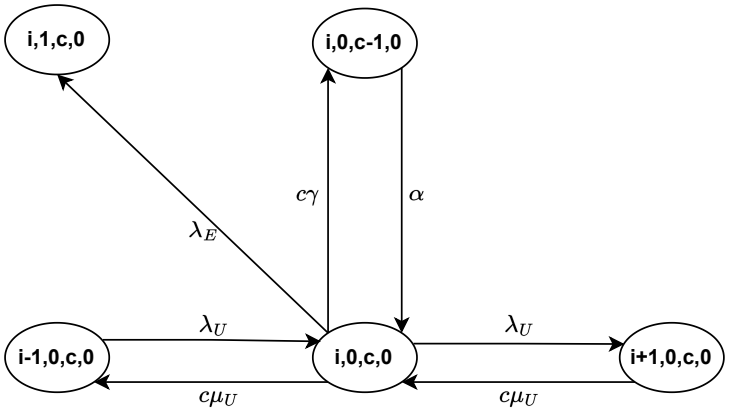
Source: The author (2023)

Table 18 – Events related to the states $(i, 0, c, 0)$, with $c < i < k$, $l = c$ and $k - c > 1$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 22 – States $(i, 0, c, 0)$, with $c < i < k$, $l = c$ and $k - c > 1$



Source: The author (2023)

19 summarizes their related events. Additionally, their balance equation is given by Eq 4.3.4.

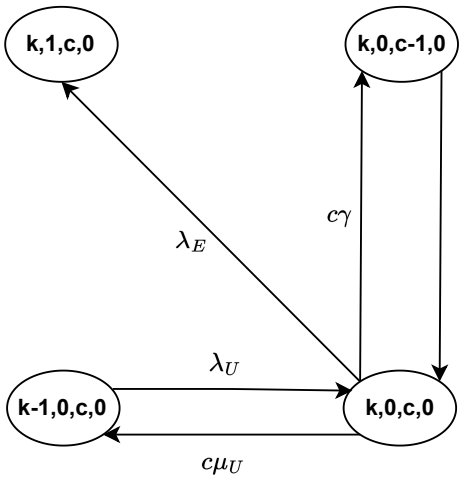
$$[\lambda_E + c(\mu_U + \gamma)]\pi(k, 0, c, 0) = \lambda_U \pi(k - 1, 0, c, 0) + \alpha\pi(k, 0, c - 1, 0) \quad (4.3.4)$$

Table 19 – Events related to the states $(k, 0, c, 0)$, with $i = k$ and $j = c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 23 – State $(k, 0, c, 0)$, with $i = k$ and $j = c$



Source: The author (2023)

States that model k URLLC users in the system and at least one container and not more than $c - 1$ are active serving these users (see Fig. 4.3.5). In summary, states $(k, 0, l, 0)$, with $i = k$, $l < k$ and $0 < l < c$. These states follow the balance equation given in Eq. 4.3.5 and their related events are listed in Table 20.

$$
\begin{aligned}
&[\lambda_E + l(\mu_U + \gamma) + min(c - l, k - l)\alpha]\pi(k, 0, l, 0) \\
&= \lambda_U\pi(k{-}1, 0, l, 0){+}\mu_E\pi(k, 1, l, 1){+}min(c{-}l{+}1, k{-}l{+}1)\alpha\pi(k, 0, l{-}1, 0){+}(l{+}1)\gamma\pi(k, 0, l{+}1, 0)
\end{aligned}
\tag{4.3.5}
$$

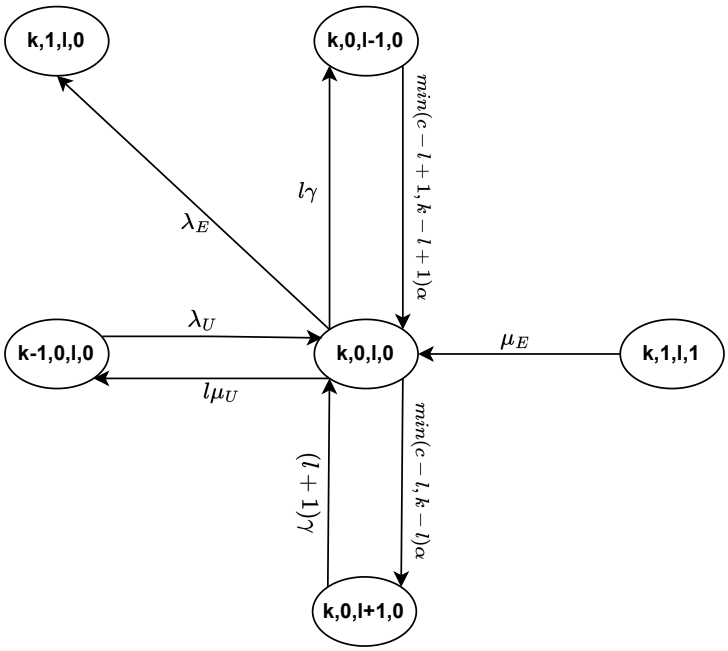
States that model the system with more URLLCs users than active containers, having at least two and no more than $k - 1$ URLLC users and at least one and no more than $c - 1$ active containers, are shown in Fig. 25. In summary, states $(i, 0, l, 0)$, with $1 < i \leq k - 1$ and

Table 20 – Events related to the states $(k, 0, l, 0)$, with $i = k$, $l < k$ and $0 < l < c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 24 – States $(k, 0, l, 0)$, with $i = k$, $l < k$ and $0 < l < c$



Source: The author (2023)

$0 < l < c$ and $i > l$. It follows the Eq. 4.3.6 and its related events are listed in Table 21.

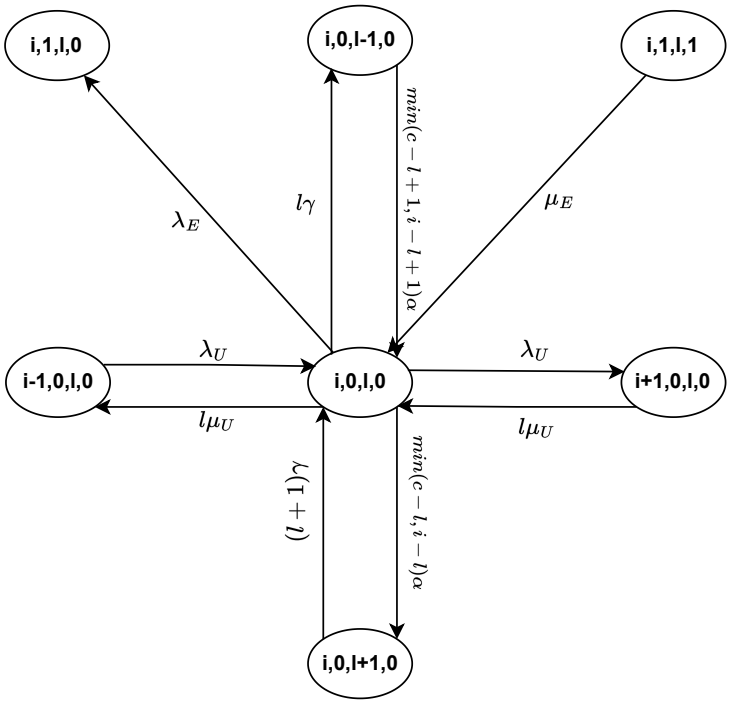
$$
\begin{aligned}
&[\lambda_U + \lambda_E + l(\mu_U + \gamma) + min(c-l, i-l)\alpha]\pi(i,0,l,0) \\
&= \lambda_U\pi(i-1,0,l,0) + l\mu_U\pi(i+1,0,l,0) + \mu_E\pi(i,1,l,1) + min(c-l+1, i-l+1)\alpha\pi(i,0,l-1,0) \\
&\qquad + (l+1)\gamma\pi(i,0,l+1,0) \quad (4.3.6)
\end{aligned}
$$

Table 21 – Events related to the states $(i, 0, l, 0)$, with $1 < i \leq k - 1$ and $0 < l < c$ and $i > l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 25 – States $(i, 0, l, 0)$, with $1 < i \leq k - 1$ and $0 < l < c$ and $i > l$



Source: The author (2023)

4.4 STATES $(0, j, 0, m)$, *with* $0 < j \leq K$, $0 < m \leq c$, *and* $i = l = 0$

In this section, we describe the equations that refer to the states with at least one eMBB user being served by a container and no URLLC users in the system. In summary, these states are denoted by $(0, j, 0, m)$, with $0 < j \leq K$, $0 < m \leq c$, and $i = l = 0$.

States in which the number of eMBB users is lower than the number of containers and all users are being served, i.e., the number of eMBB users matches the amount of active containers, with no more than $c - 1$ active containers. In summary, states $(0, j, 0, j)$, with

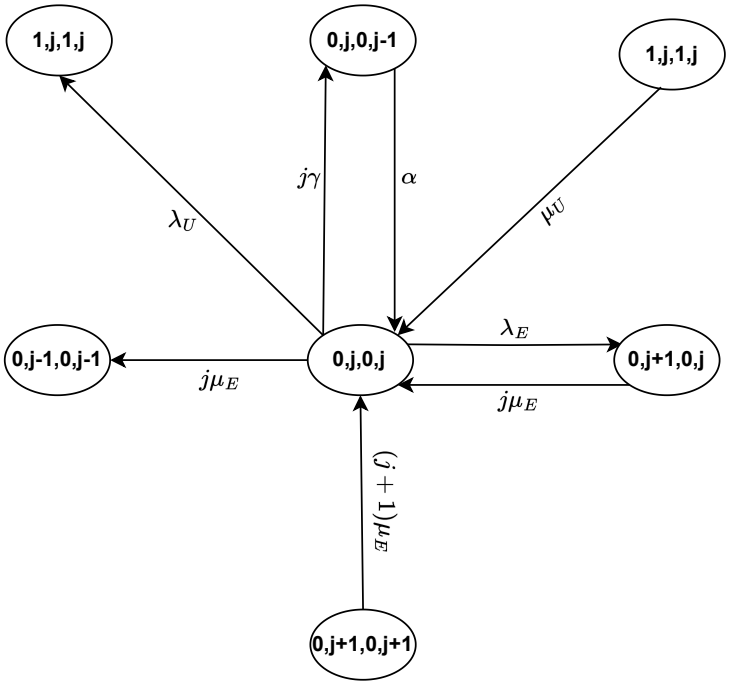
$0 < j < c$ and $m = j$ as in Table 22. The state diagram and balance equation that denote these states are shown in Fig. 26 and Eq. 4.4.1.

Table 22 – Events related to the states $(0, j, 0, j)$, with $0 < j < c$ and $m = j$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 26 – States $(0, j, 0, j)$, with $0 < j < c$ and $m = j$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma)]\pi(0, j, 0, j) \\
&= \mu_U \pi(1, j, 1, j) + \alpha\pi(0, j, 0, j-1) + (j+1)\mu_E \pi(0, j+1, 0, j+1) + j\mu_E \pi(0, j+1, 0, j)
\end{aligned}
\tag{4.4.1}
$$

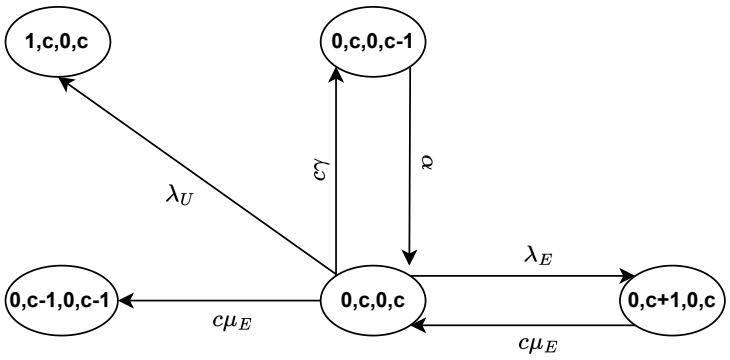
State in which all containers are active and serving eMBB users, without services waiting in line (see Fig. 27). In summary, states $(0, c, 0, c)$, with $j = c$ and $m = c$. It follows the Eq. 4.4.2 and its related events are listed in Table 23.

Table 23 – Events related to the states $(0, c, 0, c)$, with $j = c$ and $m = c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 27 – State $(0, c, 0, c)$, with $j = c$ and $m = c$



Source: The author (2023)

$$[\lambda_U + \lambda_E + c(\mu_E + \gamma)]\pi(0, c, 0, c) = \alpha\pi(0, c, 0, c-1) + c\mu_E\pi(0, c+1, 0, c) \quad (4.4.2)$$

States in which all containers are busy, serving eMBB users, and there are also eMBB users waiting in line, with at least two rooms-queue (see Fig. 28). Eq. 4.4.3 describes these states $(0, j, 0, c)$, with $c < j < K$, $m = c$ and $K - c > 1$. Table 24 denotes their related events.

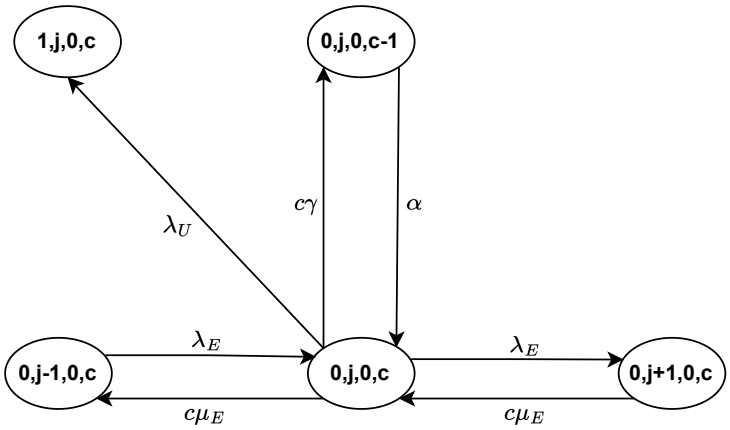
$$[\lambda_U+\lambda_E+c(\mu_E+\gamma)]\pi(0, j, 0, c) = \lambda_E\pi(0, j-1, 0, c)+c\mu_E\pi(0, j+1, 0, c)+\alpha\pi(0, j, 0, c-1) \quad (4.4.3)$$

Table 24 – Events related to the states $(0, j, 0, c)$, with $c < j < K$, $m = c$ and $K - c > 1$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 28 – States $(0, j, 0, c)$, with $c < j < K$, $m = c$ and $K - c > 1$



Source: The author (2023)

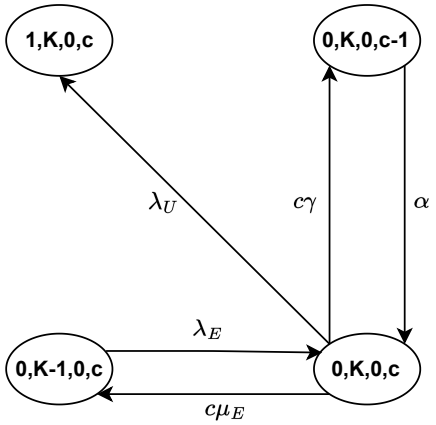
State in which all containers are busy, serving eMBB users, and there are K eMBB users (maximum number of eMBB users) in the system. Fig. 29 denotes this state $(0, K, 0, c)$, with $J = K$ and $m = c$, whose related events and balance equation are presented in Table 25 and Eq. 4.4.4, respectively.

Table 25 – Events related to the states $(0, K, 0, c)$, with $J = K$ and $m = c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 29 – State $(0, K, 0, c)$, with $J = K$ and $m = c$



Source: The author (2023)

$$[\lambda_U + c(\mu_E + \gamma)]\pi(0, K, 0, c) = \lambda_E\pi(0, K - 1, 0, c) + \alpha\pi(0, K, 0, c - 1) \quad (4.4.4)$$

States that model K eMBB users in the system and at least one container and not more than $c - 1$ are actively serving these users (see Fig. 30). In short, $(0, K, 0, m)$, with $j = K$ and $0 < m < c$, as in Table 26 and Eq. 4.4.5.

Table 26 – Events related to the states $j = K$ and $0 < m < c$.

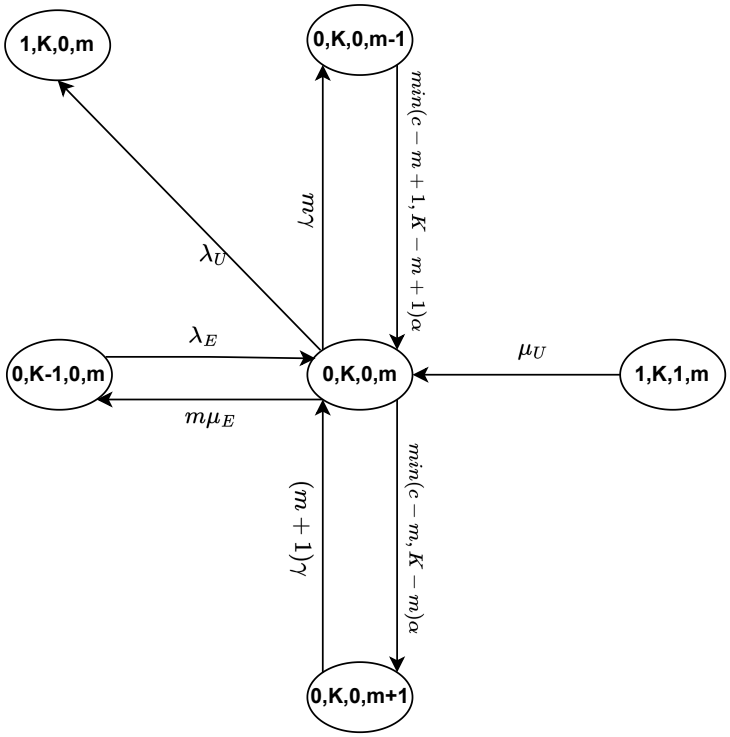
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

$$\begin{aligned}&[\lambda_U + m(\mu_E + \gamma) + min(c - m, K - m)\alpha]\pi(0, K, 0, m)\\ &= \lambda_E\pi(0, K - 1, 0, m) + \mu_U\pi(1, K, 1, m) + min(c - m + 1, K - m + 1)\alpha\pi(0, K, 0, m - 1)\\ &\qquad + (m + 1)\gamma\pi(0, K, 0, m + 1) \quad (4.4.5)\end{aligned}$$

States that model the system with more eMBB users than active containers, having at least two and no more than $K - 1$ eMBB users and at least one and no more than $c - 1$ active

Figure 30 – States $(0, K, 0, m)$, with $j = K$ and $0 < m < c$



Source: The author (2023)

containers serving them. The Eq. 4.4.6 describes these states $(0, j, 0, m)$, with $1 < j < K$ and $0 < m < c$. Fig 31 illustrates the state diagram of these states and Table 27 summarizes their related events.

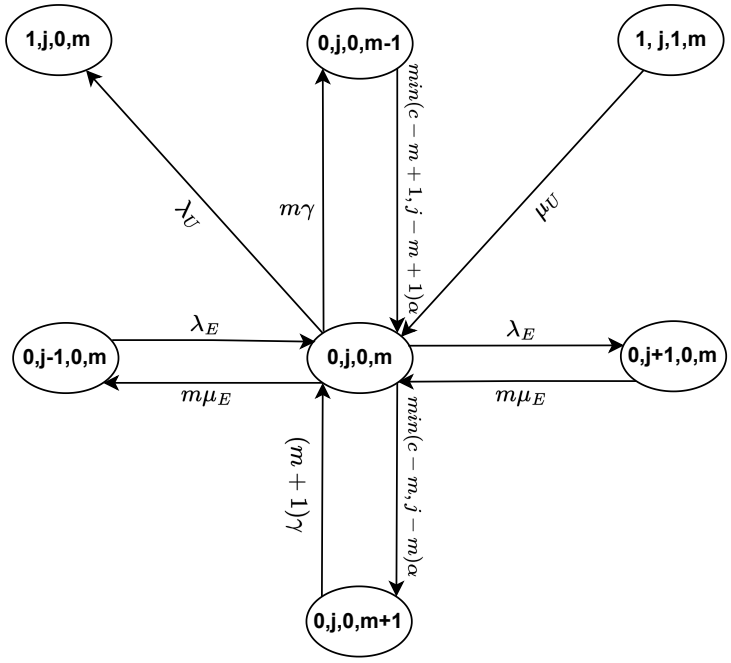
Table 27 – Events related to the states $(0, j, 0, m)$, with $1 < j < K$ and $0 < m < c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + m(\mu_E + \gamma) + min(c-m, j-m)\alpha]\pi(0, j, 0, m) \\
&\quad = \lambda_E \pi(0, j-1, 0, m) + m\mu_E \pi(0, j+1, 0, m) + \mu_U \pi(1, j, 1, m) \\
&\quad + min(c-m+1, j-m+1)\alpha\pi(0, j, 0, m-1) + (m+1)\gamma\pi(0, j, 0, m+1) \quad (4.4.6)
\end{aligned}
$$

Figure 31 – States $(0, j, 0, m)$, with $1 < j < K$ and $0 < m < c$



Source: The author (2023)

4.5 STATES $(i, j, l, 0)$, *with* $0 < i \leq k$, $0 < j \leq K$, $1 \leq l \leq c$, *and* $m = 0$

This section explores the states in which there are users of both types and there is at least one active container serving URLLC services while no one is processing eMBB services. In brief, we describe the states $(i, j, l, 0)$, with $0 < i \leq k$, $0 < j \leq K$, $1 \leq l \leq c$, and $m = 0$, which are divided into the following subsets.

States in which all URLLCs users are being served, with at least one container available to be activated, and the number of eMBB users not achieving the limit, i.e., states $(i, j, i, 0)$, with $0 < i < c$, $0 < j < K$, $0 < l < c$ and $i = l$. Their balance equation, state diagram, and related events are given by Eq. 4.5.1, Fig. 32, and Table 28, respectively.

$$
\begin{aligned}
&[\lambda_U + \lambda_E + i(\mu_U + \gamma) + min(c - i, j)\alpha]\pi(i, j, i, 0) \\
&= \lambda_E\pi(i, j{-}1, i, 0){+}i\mu_U\pi(i{+}1, j, i, 0){+}(i{+}1)\mu_U\pi(i{+}1, j, i{+}1, 0){+}\alpha\pi(i, j, i{-}1, 0){+}\gamma\pi(i, j, i, 1)
\end{aligned}
\tag{4.5.1}
$$

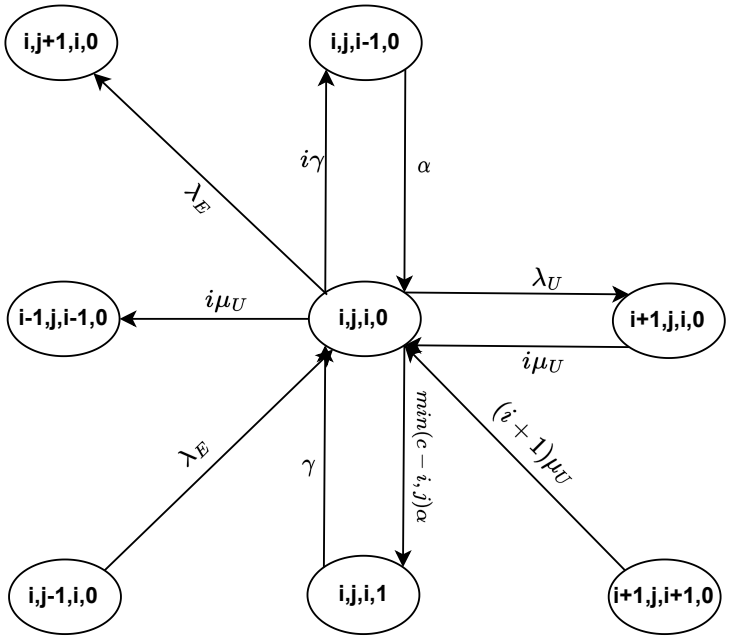
States in which the number of URLLC and eMBB users is lower than their respective limits, with some URLLC ones being served, others waiting in line, available containers to be activated, and the number of available containers equal or less than the URLLC services in the

Table 28 – Events related to the states $(i, j, i, 0)$, with $0 < i < c$, $0 < j < K$, $0 < l < c$ and $i = l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 32 – States $(i, j, i, 0)$, with $0 < i < c$, $0 < j < K$, $0 < l < c$ and $i = l$



Source: The author (2023)

queue ($i - l >= c - l$). In summary, states $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i - l >= c - l$. Fig 33 illustrates the state diagram of these states and Table 33 summarizes their related events. Additionally, their balance equation is given by Eq 4.5.2.

$$[\lambda_U + \lambda_E + l(\mu_U + \gamma) + min(c - l, i - l)\alpha]\pi(i, j, l, 0)$$
$$= \lambda_U \pi(i - 1, j, l, 0) + \lambda_E \pi(i, j - 1, l, 0) + l\mu_U \pi(i + 1, j, l, 0)$$
$$+min(c-l+1, i-l+1)\alpha\pi(i, j, l-1, 0)+\gamma\pi(i, j, l, 1)+(l+1)\gamma\pi(i, j, l+1, 0)+\mu_E\pi(i, j+1, l, 1) \quad (4.5.2)$$

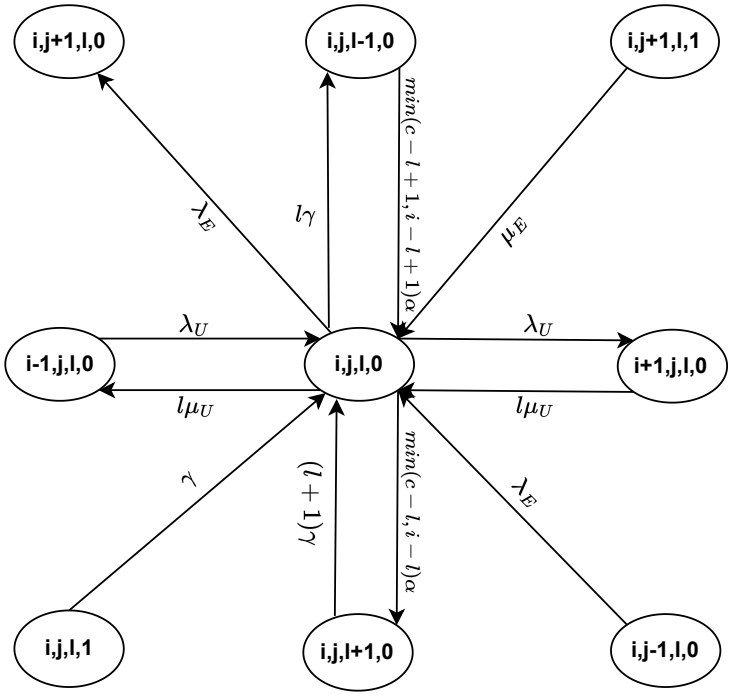
States in which the number of URLLC and eMBB users is lower than their respective

Table 29 – Events related to the states $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i - l >= c - l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 33 – States $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i - l < c - l$



Source: The author (2023)

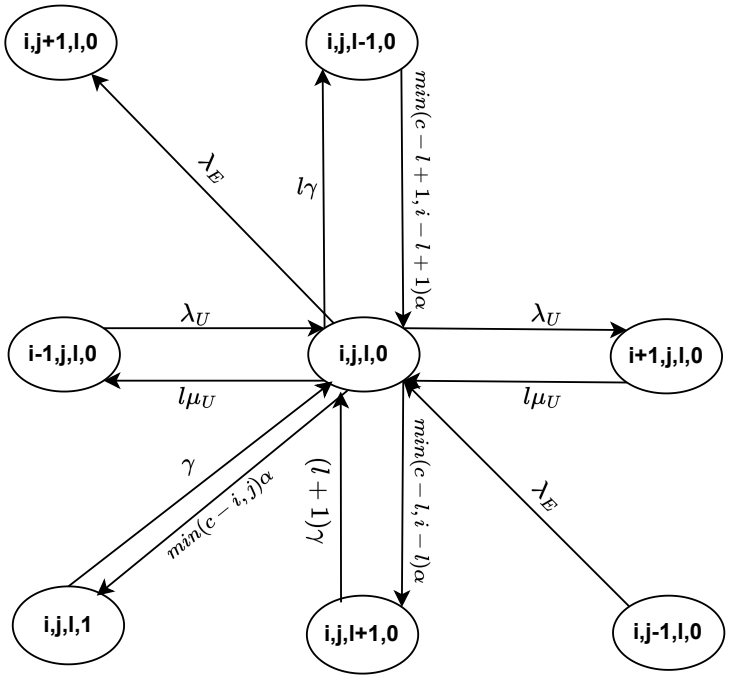
limits, with some URLLC ones being served, others waiting in line, available containers to be activated, and the number of available containers are more than sufficient to process URLLC services in the queue $(i - l < c - l)$, whose the balance equation that denote these states are shown in Fig. 34. In summary, states $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i - l < c - l$. It follows the Eq. 4.5.3 and its related events are listed in Table 30.

Table 30 – Events related to the states $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i-l < c-l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 34 – States $(i, j, l, 0)$, with $0 < i < k$, $0 < j < K$, $0 < l < c$, $i > l$ and $i - l < c - l$



Source: The author (2023)

$$[\lambda_U + \lambda_E + l(\mu_U + \gamma) + min(c-l, i-l)\alpha + min(c-i, j)\alpha]\pi(i, j, l, 0)$$
$$= \lambda_U\pi(i-1, j, l, 0) + \lambda_E\pi(i, j-1, l, 0) + l\mu_U\pi(i+1, j, l, 0) + min(c-l+1, i-l+1)\alpha\pi(i, j, l-1, 0)$$
$$+ \gamma\pi(i, j, l, 1) + (l+1)\gamma\pi(i, j, l+1, 0) \quad (4.5.3)$$

States in which all URLLCs users are being served, with at least one container available to be activated, and the number of eMBB users has achieved the limit (see Fig. 35), i.e., states $(i, K, i, 0)$, with $0 < i < c$ and $j = K$, $0 < l < c$, $i = l$ as in Table 31 and Eq. 4.5.4. For these states, only eMBB users require container initialization, but a new eMBB arrival is not

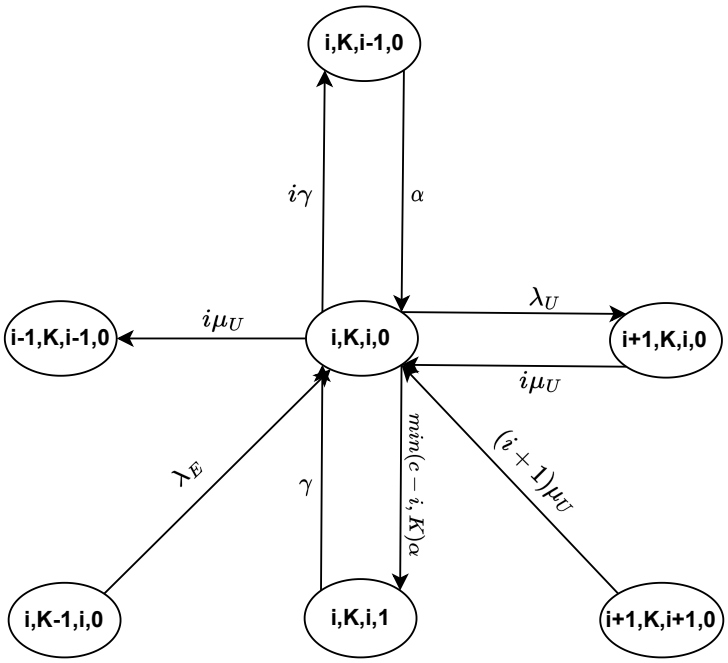
allowed.

Table 31 – Events related to the states $(i, K, i, 0)$, with $0 < i < c$ and $j = K$, $0 < l < c$, $i = l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 35 – States $(i, K, i, 0)$, with $0 < i < c$ and $j = K$,$0 < l < c$, $i = l$



Source: The author (2023)

$$[\lambda_U + i(\mu_U+) + min(c-i,K)\alpha]\pi(i,K,i,0)$$
$$= \lambda_E\pi(i,K-1,i,0) + i\mu_U\pi(i+1,K,i,0) + (i+1)\mu_U\pi(i+1,K,i+1,0)$$
$$+ \alpha\pi(i,K,i-1,0) + \pi(i,K,i,1) \quad (4.5.4)$$

States in which the number of URLLC is lower than the limit, with some URLLC ones being served, others waiting in line, K eMBB users in the system, available containers to

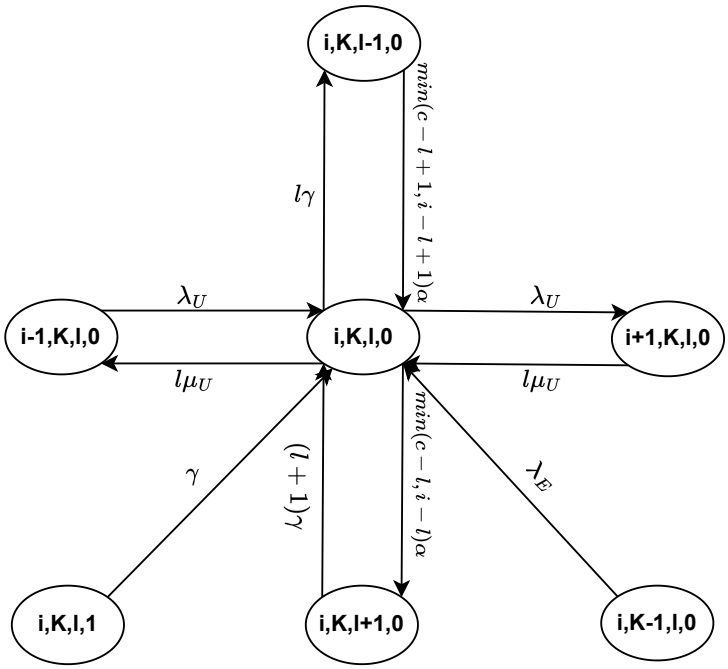
being activated, and the number of available containers insufficient to be activated to process URLLC services in queue $(c <= i)$, Fig 36 illustrates the state diagram of these states. In summary, states $(i, K, l, 0)$, with $0 < i < k$, $j = K$, $0 < l < c$, $i > l$ and $(c <= i)$ whose balance equation and related event are denoted in Eq. 4.5.5 and Table 32.

Table 32 – Events related to the states $(i, K, l, 0)$, with $0 < i < k$, $j = K$, $0 < l < c$, $i > l$ and $(c <= i)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 36 – States $(i, K, l, 0)$, with $0 < i < k$, $j = K$, $0 < l < c$, $i > l$ and $(c <= i)$



Source: The author (2023)

$$[\lambda_U + l(\mu_U + \gamma) + min(c-l, i-l)\alpha]\pi(i, K, l, 0)$$
$$= \lambda_U \pi(i-1, K, l, 0) + \lambda_E \pi(i, K-1, l, 0) + l\mu_U \pi(i+1, K, l, 0)$$
$$+ min(c-l+1, i-l+1)\alpha\pi(i, K, l-1, 0) + \gamma\pi(i, K, l, 1) + (l+1)\gamma\pi(i, K, l+1, 0) \quad (4.5.5)$$

States in which the number of URLLC is lower than the limit, with some URLLC ones being served, others waiting in line, K eMBB users in the system, available containers to be activated and the number of available containers are sufficient to be activated to process URLLC services in queue $(c > i)$. In summary, $(i, K, l, 0)$, with $0 < i < k$, $j = K$, $0 < l < c$, $i > l$, $m = 0$ and $(c <= i)$ as in Table 33. The state diagram and balance equation that denote these states are shown in Fig. 37 and Eq. 4.5.6.

Table 33 – Events related to the states $(i, K, l, 0)$, with $0 < i < k$, $j = K$, $0 < l < c$, $i > l$, $m = 0$ and $(c <= i)$.

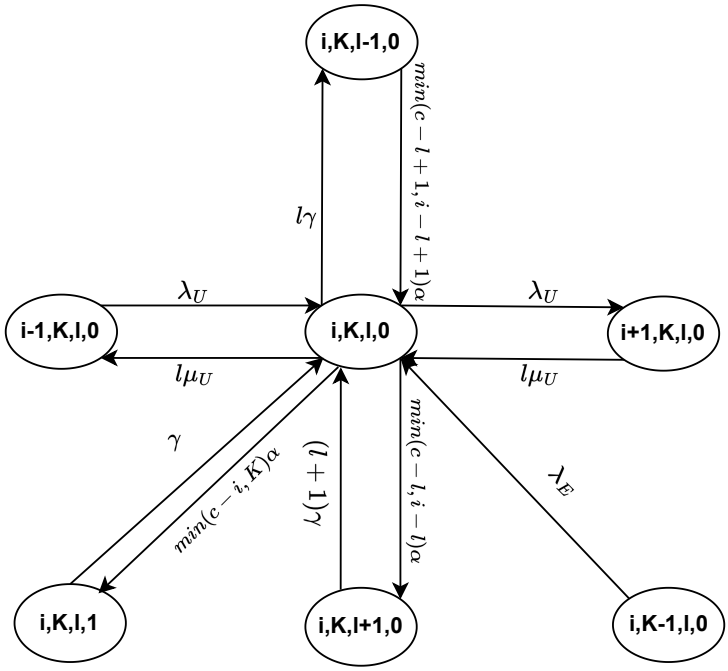
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$[\lambda_U + l(\mu_U + \gamma) + min(c-l, i-l)\alpha + min(c-i, K)\alpha]\pi(i, K, l, 0)$$
$$= \lambda_U \pi(i-1, K, l, 0) + \lambda_E \pi(i, K-1, l, 0) + l\mu_U \pi(i+1, K, l, 0)$$
$$+ min(c-l+1, i-l+1)\alpha\pi(i, K, l-1, 0) + \gamma\pi(i, K, l, 1) + (l+1)\gamma\pi(i, K, l+1, 0) \quad (4.5.6)$$

States in which all containers are being used to process URLLC services and there is no URLLC user waiting in line. Besides that, the system does not achieve its capacity for eMBB users (see Fig. 38). Eq. 4.5.7 describes these states $(c, j, c, 0)$, with $c = i$, $0 < j < K$, $l = c$ and $m = 0$. Table 34 denotes their related events.

Figure 37 – States $(c, j, c, 0)$, with $c = i$, $0 < j < K$, $l = c$ and $m = 0$.



Source: The author (2023)

Table 34 – Events related to the states $(c, j, c, 0)$, with $c = i$, $0 < j < K$, $l = c$ and $m = 0$.

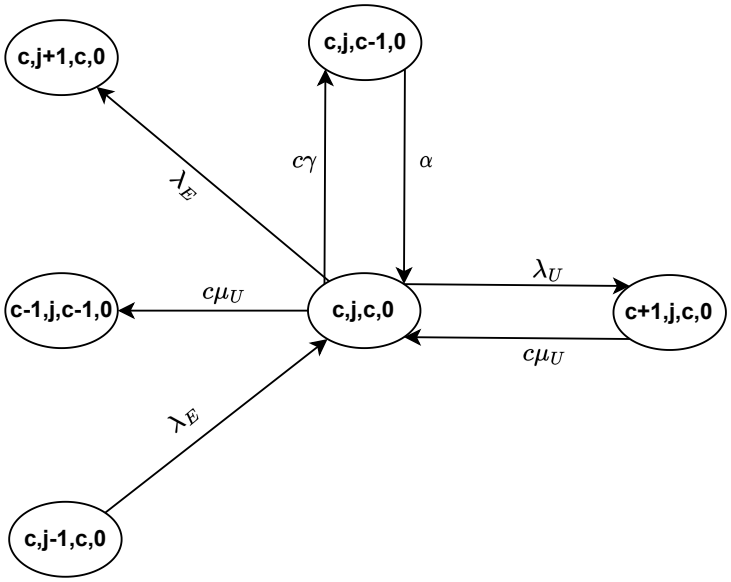
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

$$[\lambda_U + \lambda_E + c(\mu_U + \gamma)]\pi(c, j, c, 0) = \lambda_E\pi(c, j-1, c, 0) + c\mu_U\pi(c+1, j, c, 0) + \alpha\pi(c, j, c-1, 0) \quad (4.5.7)$$

States in which all containers are being used to process URLLC services and there are URLLC users waiting in line, but the limit is not achieved, Fig 39 illustrates the state diagram of these states. Besides that, the system does not achieve its capacity for eMBB users. In summary, states $(i, j, c, 0)$, with $c < i < k$, $0 < j < K$, $l = c$ and $m = 0$. It follows the Eq. 4.5.8 and its related events are listed in Table 35.

Figure 38 – States $(c, j, c, 0)$, with $c = i$, $0 < j < K$, $l = c$ and $m = 0$.



Source: The author (2023)

Table 35 – Events related to the states $(i, j, c, 0)$, with $c < i < k$, $0 < j < K$, $l = c$ and $m = 0$.

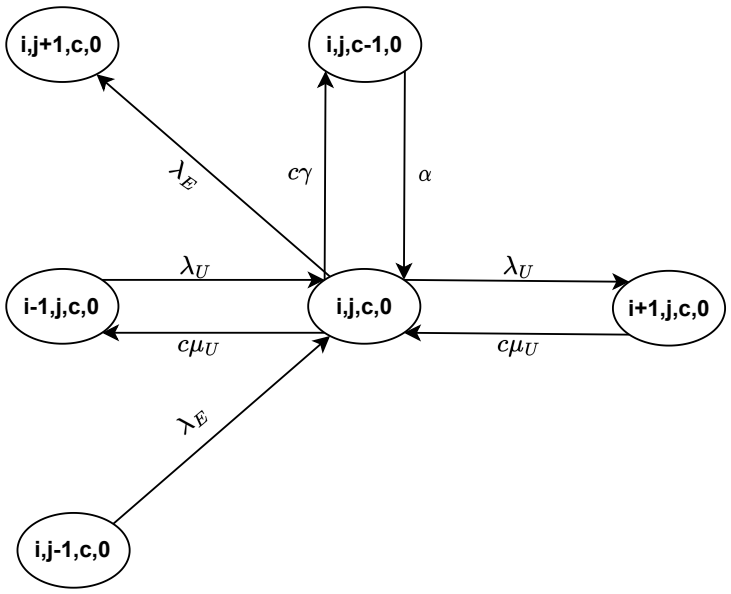
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$[\lambda_U + \lambda_E + c(\mu_U + \gamma)]\pi(i, j, c, 0)$$
$$= \lambda_U \pi(i-1, j, c, 0) + \lambda_E \pi(i, j-1, c, 0) + c\mu_U \pi(i+1, j, c, 0) + \alpha\pi(i, j, c-1, 0) \quad (4.5.8)$$

States in which the limit of URLLC users has been achieved, with someone in line and containers available to be activated. Besides that, the system did not achieve its capacity for eMBB users (see Fig. 40). In summary, states $(k, j, l, 0)$, with $k > l$, $i = k$, $0 < j < K$, $0 < l < c$ and $m = 0$ as in Table 36 and Eq. 4.5.9.

Figure 39 – States $(i, j, c, 0)$, with $c < i < k$, $0 < j < K$, $l = c$ and $m = 0$.



Source: The author (2023)

Table 36 – Events related to the states $(k, j, l, 0)$, with $k > l$, $i = k$, $0 < j < K$, $0 < l < c$ and $m = 0$.

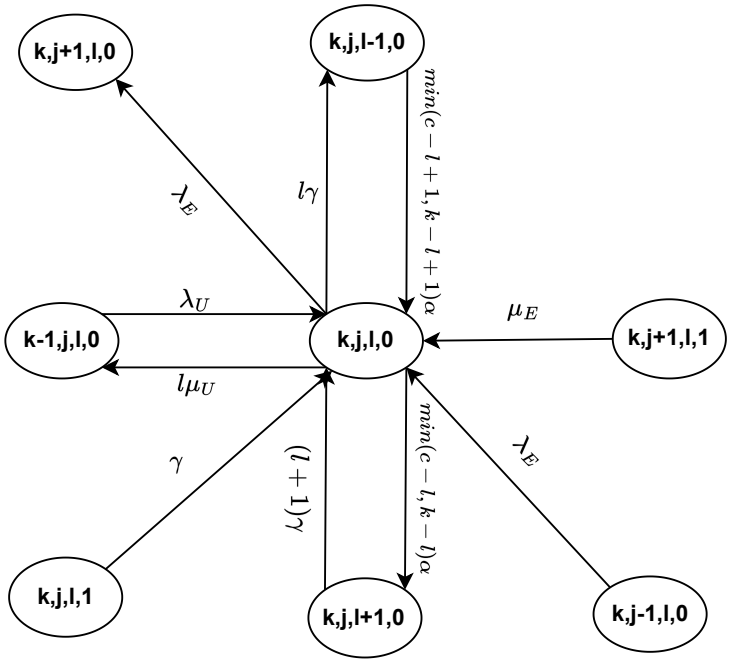
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✓	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_E + l(\mu_U + \gamma) + min(c - l, k - l)\alpha]\pi(k, j, l, 0) \\
&\quad = \lambda_U \pi(k - 1, j, l, 0) + \lambda_E \pi(k, j - 1, l, 0) + \gamma\pi(k, j, l, 1) + (l + 1)\gamma\pi(k, j, l + 1, 0) \\
&\qquad + min(c - l + 1, k - l + 1)\alpha\pi(k, j, l - 1, 0) + \mu_E \pi(k, j + 1, l, 1) \quad (4.5.9)
\end{aligned}
$$

States in which the limit of URLLC users has been achieved and all containers are being used to process URLLC services. Besides that, the system did not achieve its capacity for eMBB users. In summary, states $(k, j, c, 0)$, with $i = k$, $0 < j < K$, $l = c$ and $m = 0$. Whose balance equation, state diagram, and related events are given by Eq. 4.5.10, Fig. 41, and Table 37, respectively.

Figure 40 – States $(k, j, l, 0)$, with $k > l$, $i = k$, $0 < j < K$, $0 < l < c$ and $m = 0$.



Source: The author (2023)

Table 37 – Events related to the states $(k, j, c, 0)$, with $i = k$, $0 < j < K$, $l = c$ and $m = 0$.

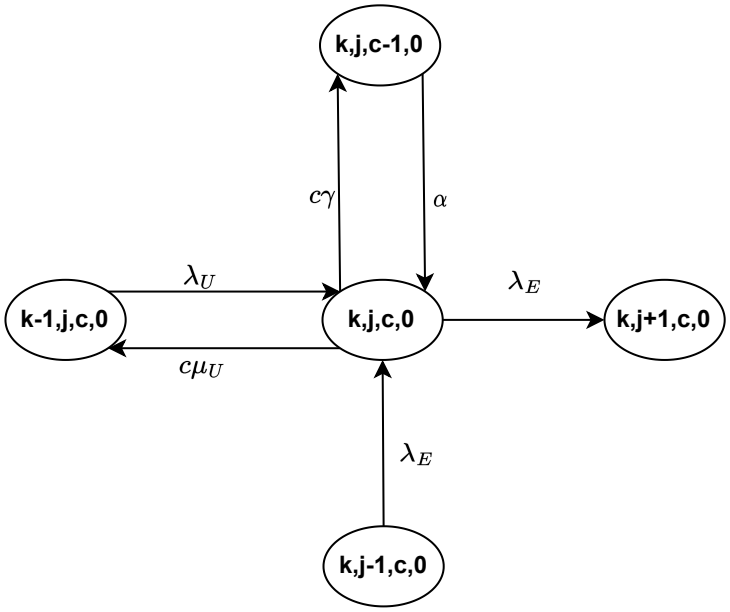
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + c(\mu_U + \gamma)]\pi(i, j, c, 0) \\
&= \lambda_U \pi(i-1, j, c, 0) + \lambda_E \pi(i, j-1, c, 0) + c\mu_U \pi(i+1, j, c, 0) + \alpha\pi(i, j, c-1, 0)
\end{aligned} \tag{4.5.10}
$$

States in which the limits for URLLC users and eMBB users have been achieved and there are containers to be activated. Eq. 4.5.11 and Fig. 42 describes these states $(k, K, l, 0)$, with $i = k$, $j = K$, $0 < l < c$ and $m = 0$. Table 38 denotes their related events.

Figure 41 – States $(k, j, c, 0)$, with $i = k$, $0 < j < K$, $l = c$ and $m = 0$.



Source: The author (2023)

Table 38 – Events related to the states $(k, K, l, 0)$, with $i = k$, $j = K$, $0 < l < c$ and $m = 0$.

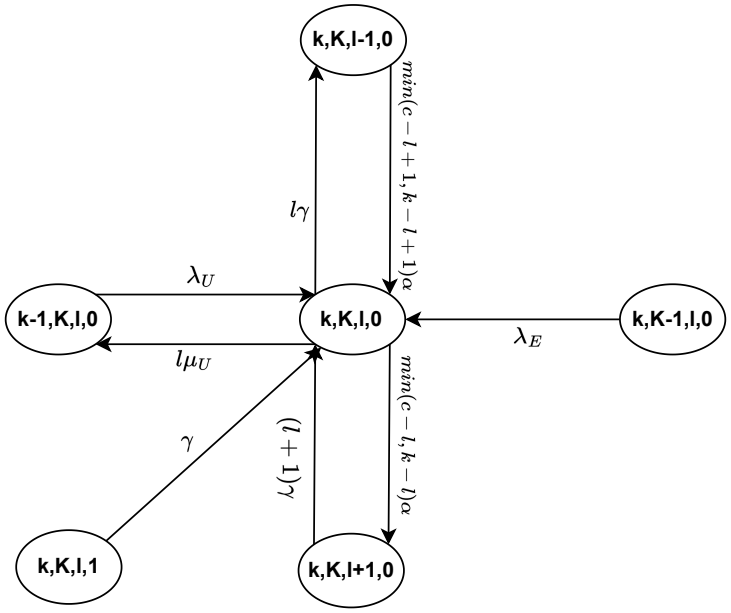
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✓	✗
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$[l(\mu_U + \gamma) + min(c - l, k - l)\alpha]\pi(k, K, l, 0) =$$
$$\lambda_U \pi(k - 1, K, l, 0) + \lambda_E \pi(k, K - 1, l, 0) + min(c - l + 1, k - l + 1)\alpha\pi(k, K, l - 1, 0)$$
$$+ (l + 1)\gamma\pi(k, K, l + 1, 0) + \gamma\pi(k, K, l, 1) \quad (4.5.11)$$

State in which the limit for eMBB users has been achieved, but all containers of the system are processing URLLC services with no one waiting in line. In summary, states $(c, K, c, 0)$, with $i = c$, $j = K$, $l = c$ and $m = 0$ as in Table 39. The state diagram and balance equation that denote these states are shown in Fig. 43 and Eq. 4.5.12.

Figure 42 – States $(k, K, l, 0)$, with $i = k$, $j = K$, $0 < l < c$ and $m = 0$



Source: The author (2023)

Table 39 – Events related to the states $(c, K, c, 0)$, with $i = c$, $j = K$, $l = c$ and $m = 0$.

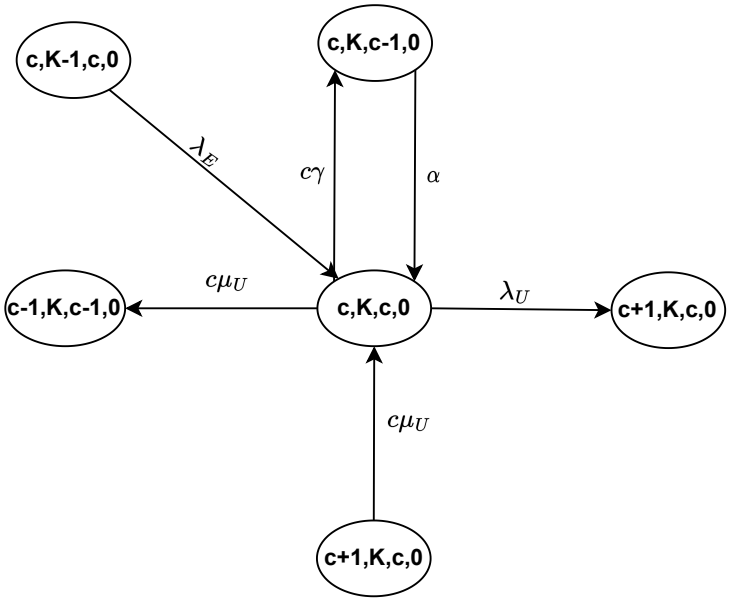
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

$$[\lambda_U + c(\mu_U + \gamma)]\pi(c, K, c, 0) = \lambda_E\pi(c, K-1, c, 0) + \alpha\pi(c, K, c-1, 0) + c\mu_U\pi(c+1, K, c, 0) \quad (4.5.12)$$

States in which the limit for eMBB users has been achieved, all containers of the system are processing URLLC services, with ones waiting in line. Besides that, the system may still admit URLLC users (see Fig.44). In summary, states $(i, K, c, 0)$, with $0 < i < k$, $j = K$, $l = c$, $m = 0$, $c < i < k$ and $k - c > 1$. These states follow the balance equation given in Eq. 4.5.13 and their related events are listed in Table 40.

Figure 43 – States $(c, K, c, 0)$, with $i = c$, $j = K$, $l = c$ and $m = 0$



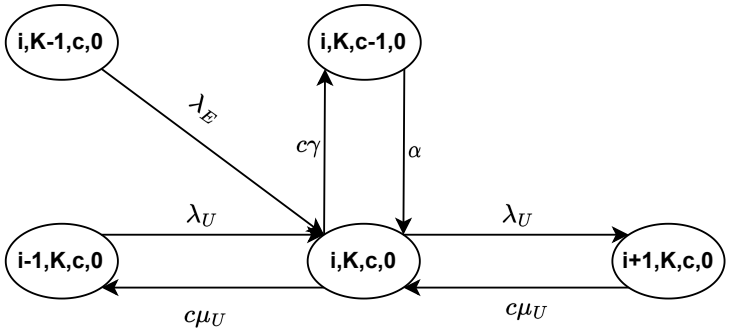
Source: The author (2023)

Table 40 – Events related to the states $(i, K, c, 0)$, with $0 < i < k$, $j = K$, $l = c$, $m = 0$, $c < i < k$ and $k - c > 1$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 44 – States $(i, K, c, 0)$, with $0 < i < k$, $j = K$, $l = c$, $m = 0$, $c < i < k$ and $k - c > 1$



Source: The author (2023)

$$[\lambda_U + c(\mu_U + \gamma)]\pi(i, K, c, 0)$$
$$= \lambda_U \pi(i-1, K, c, 0) + \lambda_E \pi(i, K-1, c, 0) + \alpha\pi(i, K, c-1, 0) + c\mu_U \pi(i+1, K, c, 0) \quad (4.5.13)$$

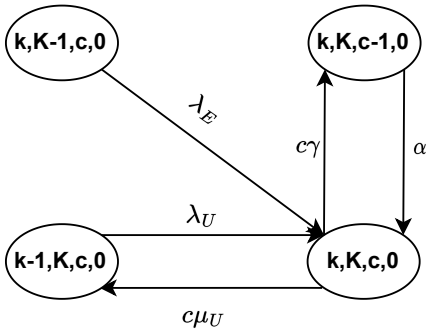
State in which is full and all containers are processing URLLC services. In summary, states $(k, K, c, 0)$, with $m = 0$, $i = k$, $j = K$, $l = c$ and $m = 0$. The balance equation, state diagram, and related events are given by Eq. 4.5.14, Fig. 45, and Table 41, respectively.

Table 41 – Events related to the states $(k, K, c, 0)$, with $m = 0$, $i = k$, $j = K$, $l = c$ and $m = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✗
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 45 – State $(k, K, c, 0)$, with $m = 0$, $i = k$, $j = K$, $l = c$ and $m = 0$



Source: The author (2023)

$$[c(\mu_U + \gamma)]\pi(k, K, c, 0)$$
$$= \lambda_U \pi(k-1, K, c, 0) + \lambda_E \pi(k, K-1, c, 0) + \alpha\pi(k, K, c-1, 0) \quad (4.5.14)$$

4.6 STATES $(i, j, 0, m)$, $with\ 0 < i < k$, $0 < j < K$, $1 \leq m \leq c$, $and\ l = 0$

In this section, the equations that refer to the states in which there are users of both types and there is at least one active container attending eMBB services while no one is processing URLLC services are described. In summary, this section is grouped by the states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $1 \leq m \leq c$, and $l = 0$.

States in which all eMBB users are being served, with at least one container available to be activated, insufficient number of available containers to process URLLC services in queue $(c - m < i)$, and the number of URLLC users not achieving the limit (see Fig. 46), i.e., states $(i, j, 0, j)$, with $0 < j < c$ and $0 < i < k$. Table 42 summarizes their related events. Additionally, their balance equation is given by Eq. 4.6.1. For these states, only URLLC users require container initialization.

Table 42 – Events related to the states $(i, j, 0, j)$, with $0 < j < c$ and $0 < i < k$.

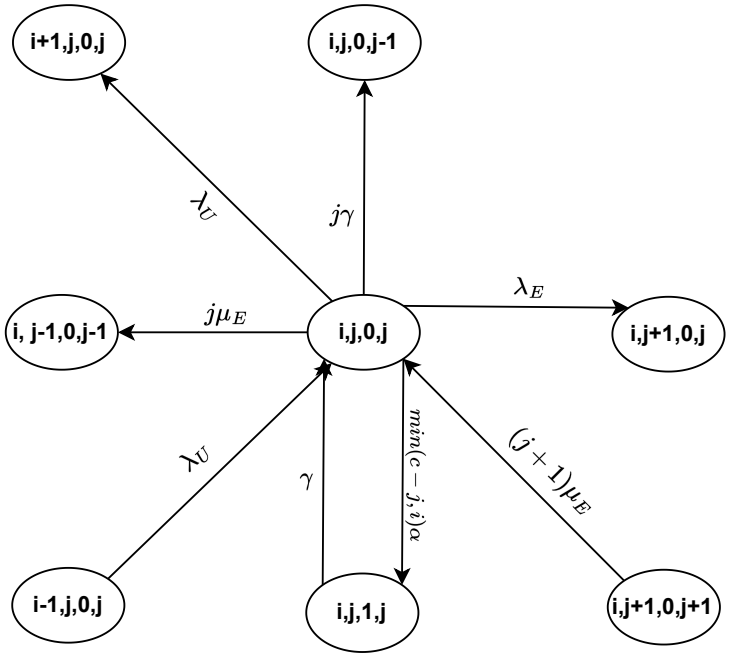
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

$$[\lambda_U + \lambda_E + j(\mu_E + \gamma) + min(c - j, i)\alpha]\pi(i, j, 0, j) = \lambda_U \pi(i - 1, j, 0, j) + (j + 1)\mu_E \pi(i, j + 1, 0, j + 1) + \gamma\pi(i, j, 1, j) \quad (4.6.1)$$

States in which all eMBB users are being served, with at least one container available to be activated, the number of available containers to be activated is sufficient to process URLLC services in queue $(c - m >= i)$, and the number of URLLC users not achieving the limit, Fig 47 illustrates the state diagram of these states, i.e., states $(i, j, 0, j)$, with $0 < j < c$ and $0 < i < k$ whose balance equation and related event are denoted in Eq. 4.6.2 and Table 43.

Figure 46 – States $(i, j, 0, j)$, with $0 < i < k$, $j > 0$, $j = m = c$, and $(c - m) < i$



Source: The author (2023)

Table 43 – Events related to the states $(i, j, 0, j)$, with $0 < j < c$ and $0 < i < k$.

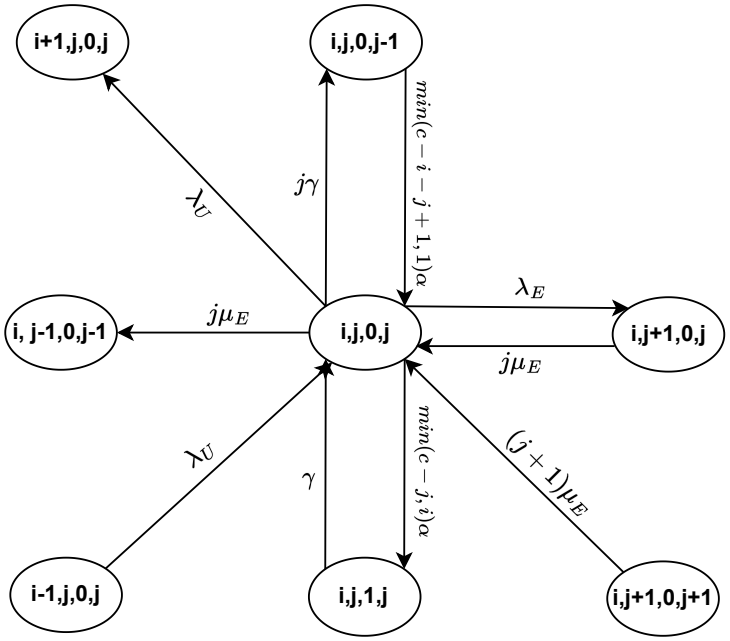
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + min(c - j, i)\alpha]\pi(i, j, 0, j) = \\
&\qquad \lambda_U \pi(i - 1, j, 0, j) + j\mu_E \pi(i, j + 1, 0, j) + (j + 1)\mu_E \pi(i, j + 1, 0, j + 1) \\
&\qquad\qquad + \gamma\pi(i, j, 1, j) + min(c - i - j + 1, 1)\alpha\pi(i, j, 0, j - 1) \quad (4.6.2)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB ones being served, others waiting in line, available containers to be activated, and an insufficient number of available containers to be activated to process URLLC services in queue $(c - m < i)$. The state diagram that denotes these states is shown in Fig.

Figure 47 – States $(i, j, 0, j)$, with $0 < i < k$, $0 < j = c <$, $m = j$ and $(c - m) >= i$



Source: The author (2023)

48. In summary, states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$, and $c - m < i$. It follows the Eq. 4.6.3 and its related events are listed in Table 44.

Table 44 – Events related to the states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$, and $c - m < i$.

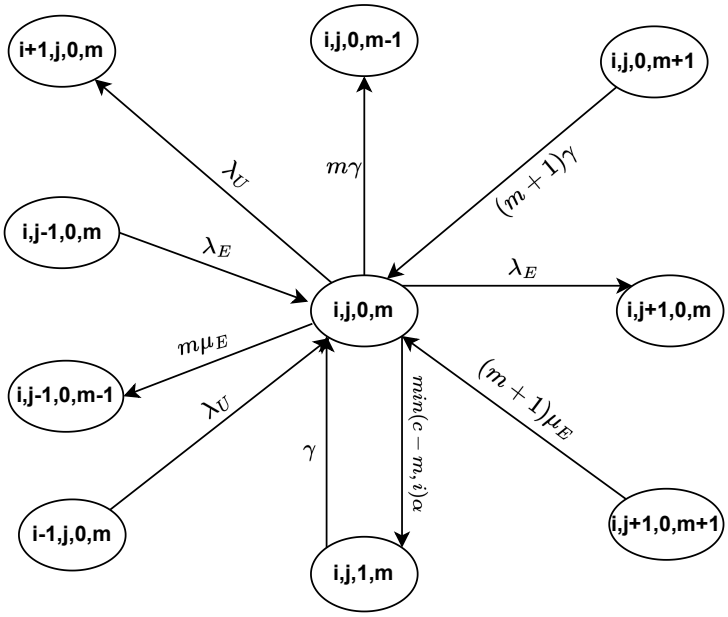
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + m(\mu_E + \gamma) + min(c - m, i)\alpha]\pi(i, j, 0, m) \\
&\quad = \lambda_U \pi(i - 1, j, 0, m) + \lambda_E \pi(i, j - 1, 0, m) + (m + 1)\mu_E \pi(i, j + 1, 0, m + 1) \\
&\qquad + \gamma\pi(i, j, 1, m) + (m + 1)\gamma\pi(i, j, 0, m + 1) \quad (4.6.3)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB ones being served, others waiting in the queue, available containers to be

Figure 48 – States $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$ and $(c - m) < i$



Source: The author (2023)

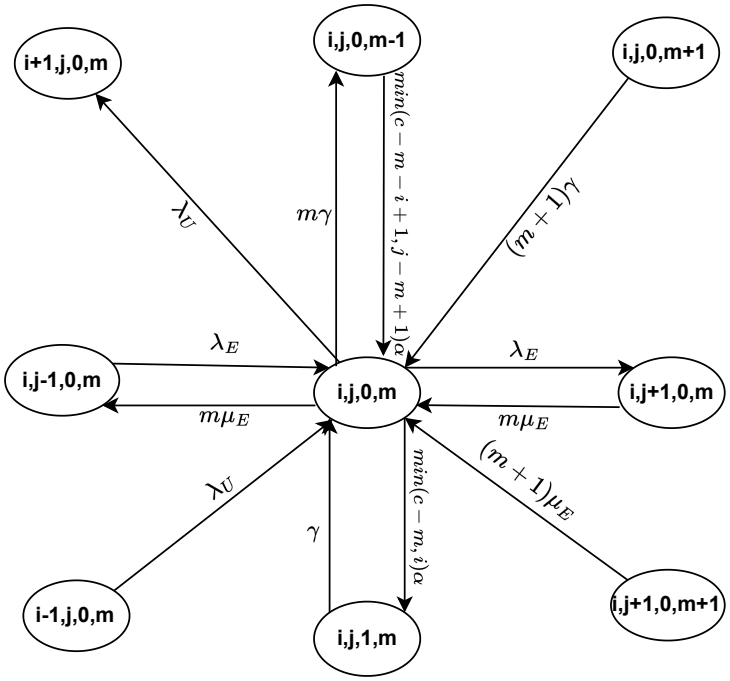
activated, and the number of available containers to be activated are sufficient to process URLLC services in queue $(c - m = i)$. In summary, states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$, whose and balance equation, state diagram, and related events given by Eq. 4.6.4, Fig. 49, and Table 45, respectively.

Table 45 – Events related to the states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, and $j > m$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 49 – States $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$ and $(c - m) = i$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + m(\mu_E + \gamma) + min(c-m, i)\alpha]\pi(i, j, 0, m) \\
&\quad = \lambda_U \pi(i-1, j, 0, m) + \lambda_E \pi(i, j-1, 0, m) + m\mu_E \pi(i, j+1, 0, m) \\
&\quad + (m+1)\mu_E \pi(i, j+1, 0, m+1) + \gamma\pi(i, j, 1, m) + (m+1)\gamma\pi(i, j, 0, m+1) \\
&\quad + min(c-m-i+1, j-m+1)\alpha\pi(i, j, 0, m-1) \quad (4.6.4)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB ones being served, others waiting in line, available containers to be activated, and the number of available containers to be activated are more than enough to process URLLC services in queue $(c - m > i)$. In summary, states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, and $j > m$ as in Table 46. The state diagram and balance equation that denote these states are shown in Fig. 50 and Eq. 4.6.5.

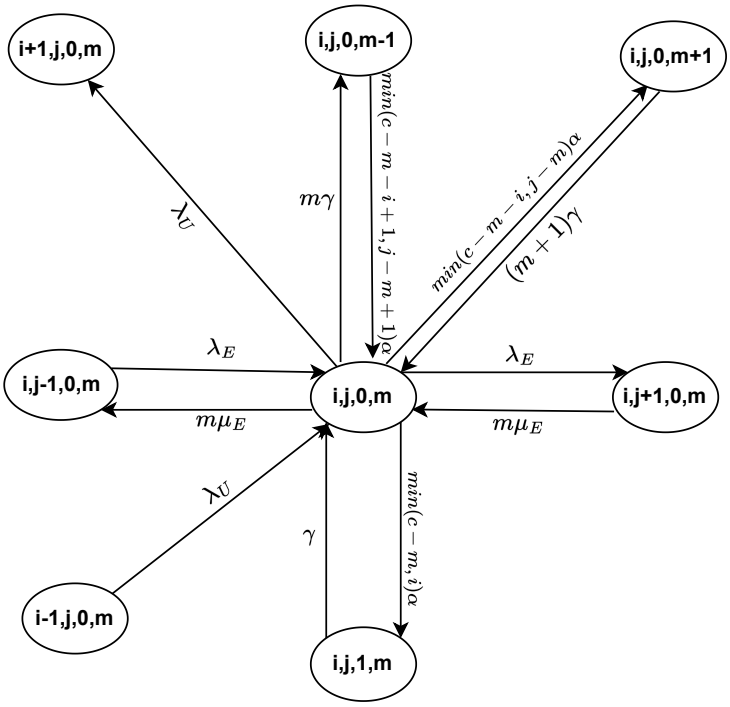
$$
\begin{aligned}
&[\lambda_U + \lambda_E + m(\mu_E + \gamma) + min(c-m, i)\alpha + min(c-m-i, j-m)\alpha]\pi(i, j, 0, m) \\
&\quad = \lambda_U \pi(i-1, j, 0, m) + \lambda_E \pi(i, j-1, 0, m) + m\mu_E \pi(i, j+1, 0, m) + \gamma\pi(i, j, 1, m) \\
&\quad + (m+1)\gamma\pi(i, j, 0, m+1) + min(c-m-i+1, j-m+1)\alpha\pi(i, j, 0, m-1) \quad (4.6.5)
\end{aligned}
$$

Table 46 – Events related to the states $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, and $j > m$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 50 – States $(i, j, 0, m)$, with $0 < i < k$, $0 < j < K$, $0 < m < c$, $j > m$ and $(c - m) > i$



Source: The author (2023)

States in which all eMBB users are being served, with at least one container available to be activated and the number of URLLC users has achieved the limit (see Fig. 51), i.e., states $(k, j, 0, j)$, with $0 < j < c$ and $i = k$, $j = m$, whose balance equation and related event are denoted in Eq. 4.6.6 and Table 47. For these states, only URLLC users require container initialization, but a new URLLC arrival is not allowed.

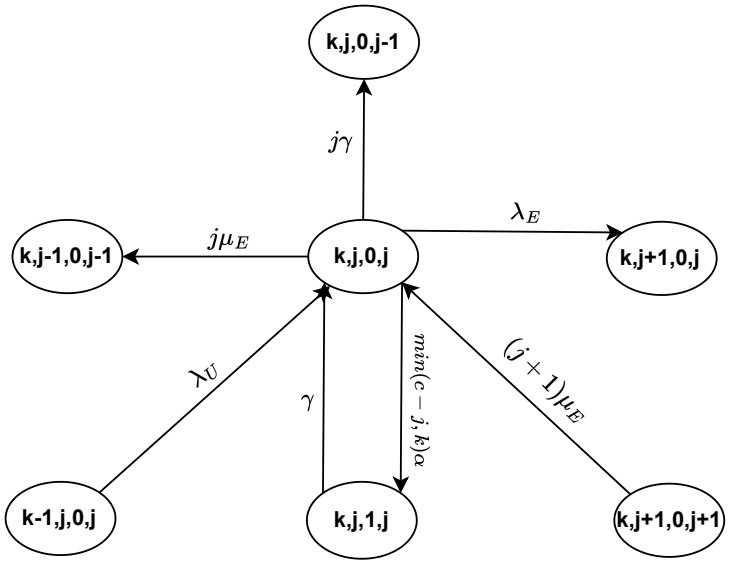
$$[\lambda_E + j(\mu_E + \gamma) + min(c - j, k)\alpha]\pi(k, j, 0, j) = \lambda_U \pi(k - 1, j, 0, j) + (j + 1)\mu_E \pi(k, j + 1, 0, j + 1) + \gamma\pi(k, j, 1, j) \quad (4.6.6)$$

Table 47 – Events related to the states $(k, j, 0, j)$, with $0 < j < c$ and $i = k$ and $j = m$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 51 – States $(k, j, 0, j)$, with $0 < j < c$ and $i = k$ and $j = m$



Source: The author (2023)

States in which the number of eMBB is lower than the limit, with some eMBB ones being served, others waiting in line, k URLLC users in the system, and available containers to be activated, Fig 52 illustrates the state diagram of these states. In summary, states $(k, j, 0, m)$, with $0 < j < K$,$i = k$,$0 < m < c$, and $j > m$ as in Table 48 and Eq. 4.6.7.

$$
\begin{aligned}
&[\lambda_E + m(\mu_E + \gamma) + min(c - m, k)\alpha]\pi(k, j, 0, m) \\
&\quad = \lambda_U \pi(k - 1, j, 0, m) + \lambda_E \pi(k, j - 1, 0, m) + (m + 1)\mu_E \pi(k, j + 1, 0, m + 1) + \\
&\qquad \gamma\pi(k, j, 1, m) + (m + 1)\gamma\pi(k, j, 0, m + 1) \quad (4.6.7)
\end{aligned}
$$

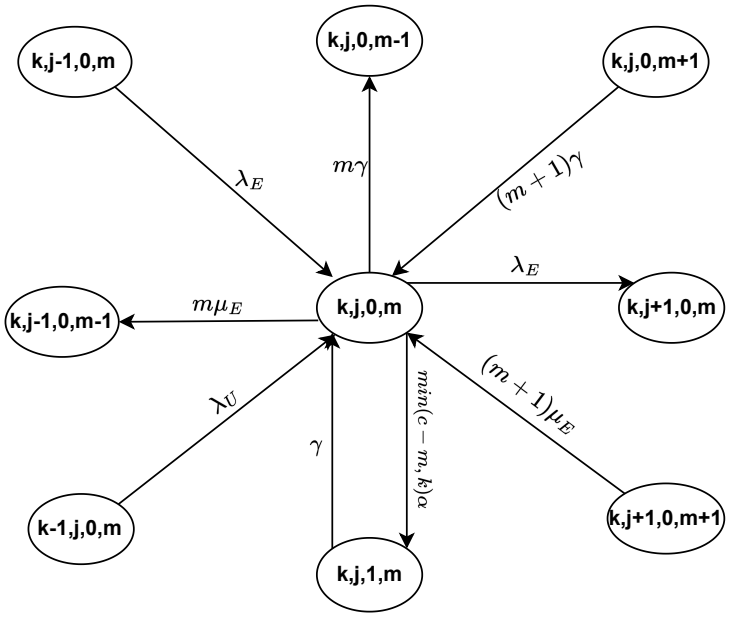
States in which all containers are being used to process eMBB services and there is no eMBB user waiting in line. Besides that, the system does not achieve its capacity for URLLC

Table 48 – Events related to the states $(k, j, 0, m)$, with $0 < j < K$,$i = k$,$0 < m < c$, and $j > m$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 52 – States $(k, j, 0, m)$, with $0 < j < K$,$i = k$,$0 < m < c$, and $j > m$



Source: The author (2023)

users (see Fig. 53). Eq. 4.6.8 describes these states $(i, c, 0, c)$, with $0 < i < k$, $l = 0$, $m = c$, $j = c$. Table 49 denotes their related events.

$$[\lambda_U + \lambda_E + c(\mu_E + \gamma)]\pi(i, c, 0, c) = \lambda_U \pi(i - 1, c, 0, c) \quad (4.6.8)$$

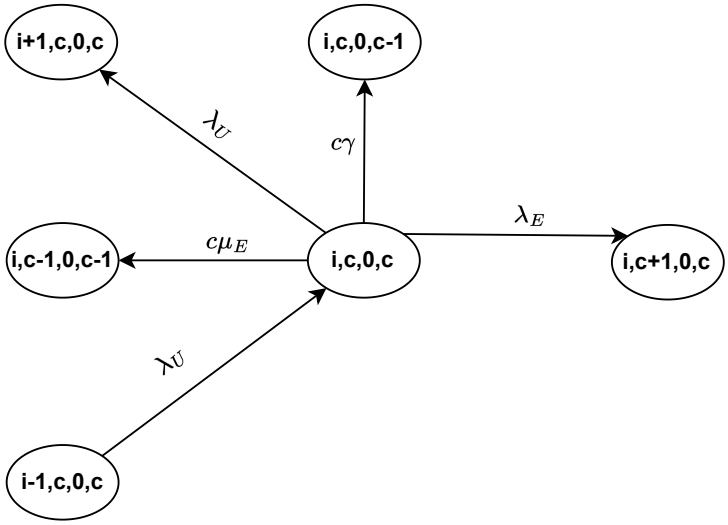
States in which all containers are being used to process eMBB services and there are eMBB users waiting in line, but the limit is not achieved. Besides that, the system does not achieve its capacity for URRLC users. The state diagram of these states is shown in Fig. 54. In summary, states $(i, j, 0, c)$, with $l = 0$, $m = c$, $K > j > c$, $0 < i < k$, $K - c > 1$. It follows the Eq. 4.6.9 and its related events are listed in Table 50.

Table 49 – Events related to the states $(i, c, 0, c)$, with $0 < i < k$, $l = 0$, $m = c$, $j = c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 53 – States $(i, c, 0, c)$, with $0 < i < k$, $l = 0$, $m = c$, $j = c$



Source: The author (2023)

$$[\lambda_U + \lambda_E + c(\mu_E + \gamma)]\pi(i, j, 0, c) = \lambda_U \pi(i - 1, j, 0, c) + \lambda_E \pi(i, j - 1, 0, c) \quad (4.6.9)$$

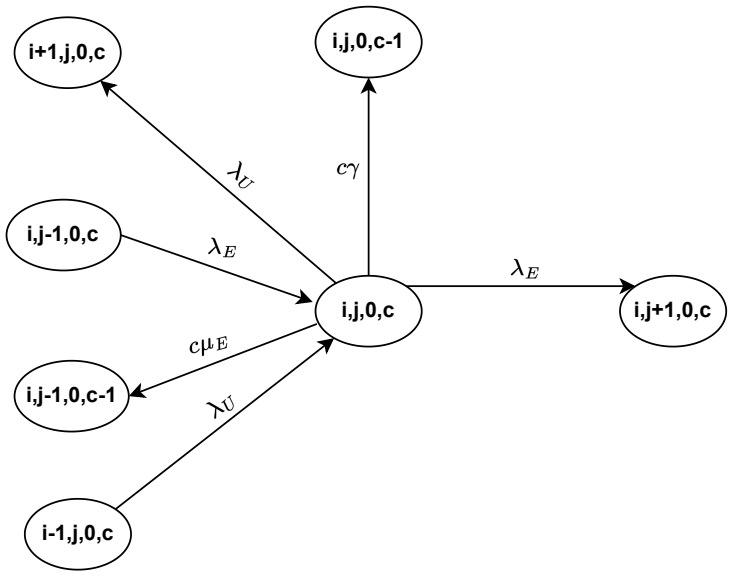
States in which the limit of eMBB users has been achieved, with someone in line, containers available to be activated and the number of available containers to be activated are insufficient to process URLLC services in queue $(c - m) < i$ (see Fig. 55). Besides that, the system did not achieve its capacity for URLLC users. Eq. 4.6.10 describes these states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$, $(c - m) < i$, and its related events are listed in Table 51.

Table 50 – Events related to the states $(i, j, 0, c)$, with $l = 0$, $m = c$, $K > j > c$, $0 < i < k$, $K - c > 1$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 54 – States $(i, j, 0, c)$, with $l = 0$, $m = c$, $K > j > c$, $0 < i < k$, $K - c > 1$



Source: The author (2023)

$$[\lambda_U + m(\mu_E + \gamma) + min(cm, i)\alpha]\pi(i, K, 0, m)$$
$$= \lambda_U\pi(i-1, K, 0, m) + \lambda_E\pi(i, K-1, 0, m) + \gamma\pi(i, K, 1, m + (m+1)\gamma\pi(i, K, 0, m+1) \tag{4.6.10}$$

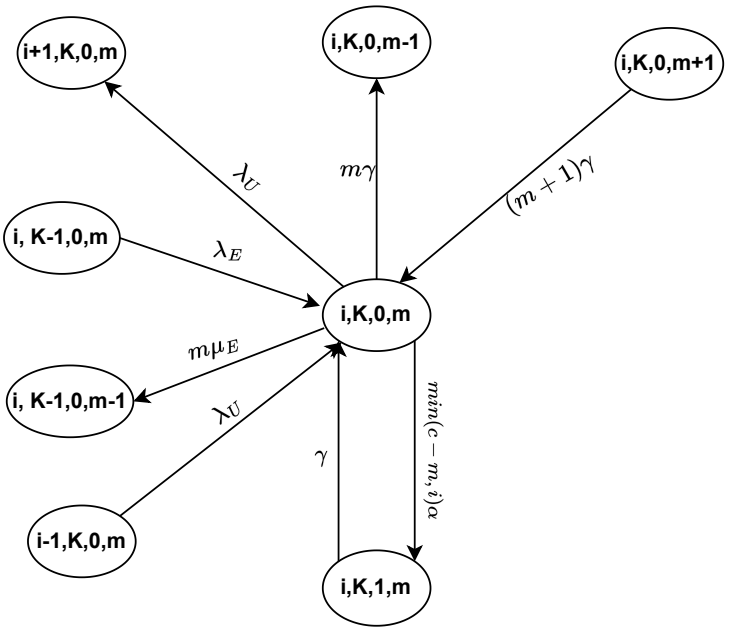
States in which the limit of eMBB users has been achieved, with someone in line, containers available to be activated and the number of available containers to be activated are sufficient to process URLLC services in the queue $(c - m) = i$. Besides that, the system did not achieve its capacity for URLLC users. In summary, states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) = i$. The balance equation, state diagram, and related events of these states are given by Eq. 4.6.11, Fig. 56, and Table 52, respectively.

Table 51 – Events related to the states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$, and $(c - m) < i$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 55 – States $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) = i$



Source: The author (2023)

Example of states: $(2, 5, 0, 2)$, $(1, 5, 0, 3)$, for $c = 4$ and $K = 5$.

$$[\lambda_U + m(\mu_E + \gamma) + min(c - m, i)\alpha]\pi(i, K, 0, m) =$$
$$\lambda_U \pi(i - 1, K, 0, m) + \lambda_E \pi(i, K - 1, 0, m) + \gamma\pi(i, K, 1, m)$$
$$+ (m + 1)\gamma\pi(i, K, 0, m + 1) + min(c - m - i + 1, K - m + 1)\alpha\pi(i, K, 0, m - 1) \quad (4.6.11)$$

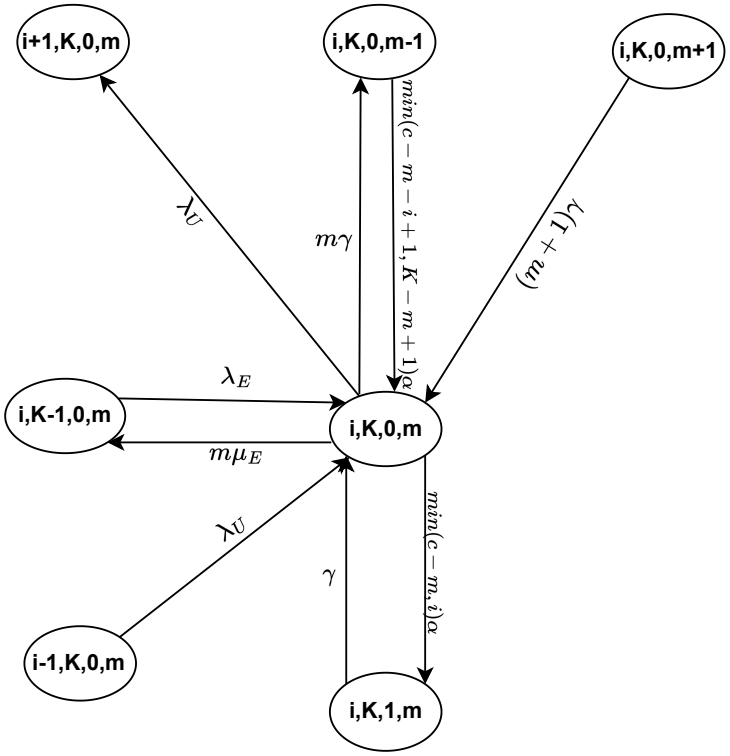
States in which the limit of eMBB users has been achieved, with someone in line, containers available to be activated and the number of available containers to be activated are more than sufficient to process URLLC services in queue $(c - m) > i$ (see Fig. 57). Besides that, the

Table 52 – Events related to the states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) = i$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 56 – States $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) = i$



Source: The author (2023)

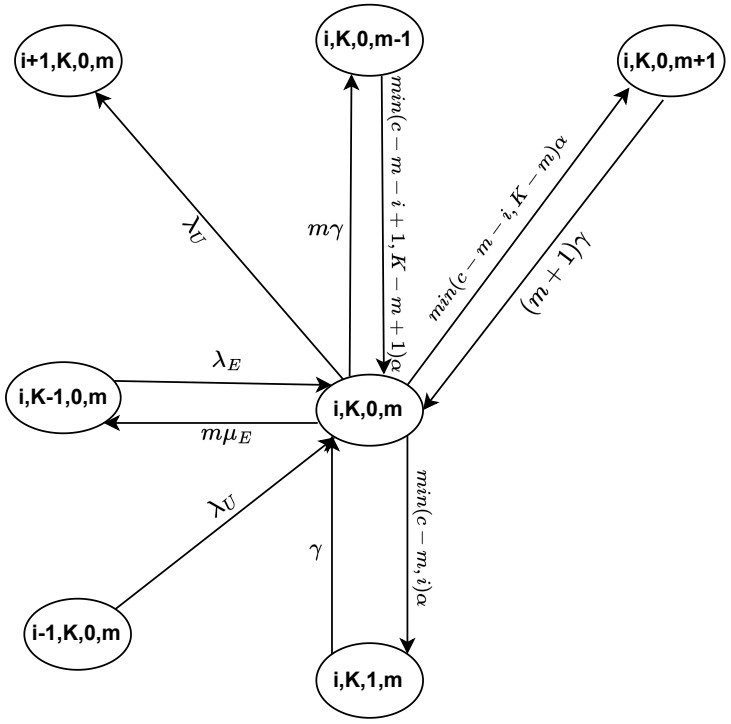
system did not achieve its capacity for URLLC users. In summary, states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) > i$. These states follow the balance equation given in 4.6.12 and their related events are listed in Table 53.

Table 53 – Events related to the states $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) > i$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 57 – States $(i, K, 0, m)$, with $K > m$, $m < c$, $0 < i < k$ and $(c - m) > i$



Source: The author (2023)

$$[\lambda_U + m(\mu_E + \gamma) + min(c-m, i)\alpha + min(c-m-i, K-m)\alpha]\pi(i, K, 0, m)$$
$$= \lambda_U \pi(i-1, K, 0, m) + \lambda_E \pi(i, K-1, 0, m) + \gamma\pi(i, K, 1, m)$$
$$+ (m+1)\gamma\pi(i, K, 0, m+1) + min(c-m-i+1, K-m+1)\alpha\pi(i, K, 0, m-1) \quad (4.6.12)$$

States in which the limit of eMBB users has been achieved and all containers are being used to process eMBB services. Besides that, the system did not achieve its capacity for URLLC

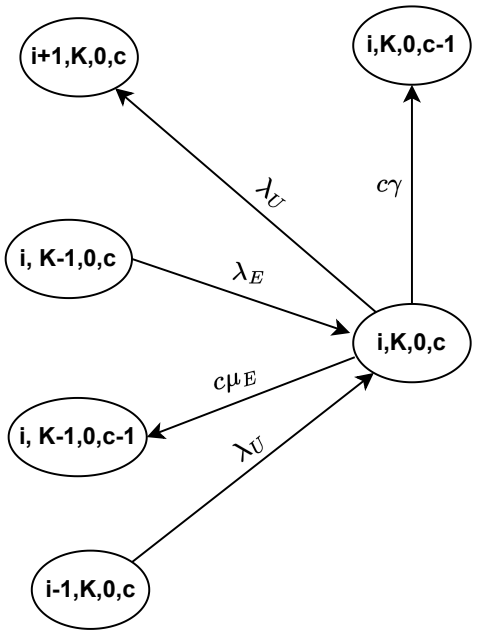
users. In summary, states $(i, K, 0, c)$, with $0 < i < k$, $j = K$, $l = 0$, $m = c$, and $K > c$. Fig 58 illustrates the state diagram of these states and Table 54 summarizes their related events. Additionally, their balance equation is given by Eq. 4.6.13.

Table 54 – Events related to the states $(i, K, 0, c)$, with $0 < i < k$, $j = K$, $l = 0$, $m = c$, and $K > c$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 58 – States $(i, K, 0, c)$, with $0 < i < k$, $j = K$, $l = 0$, $m = c$, and $K > c$



Source: The author (2023)

$$[\lambda_U + c(\mu_E + \gamma)]\pi(i, K, 0, c) = \lambda_U \pi(i - 1, K, 0, c) + \lambda_E \pi(i, K - 1, 0, c) \quad (4.6.13)$$

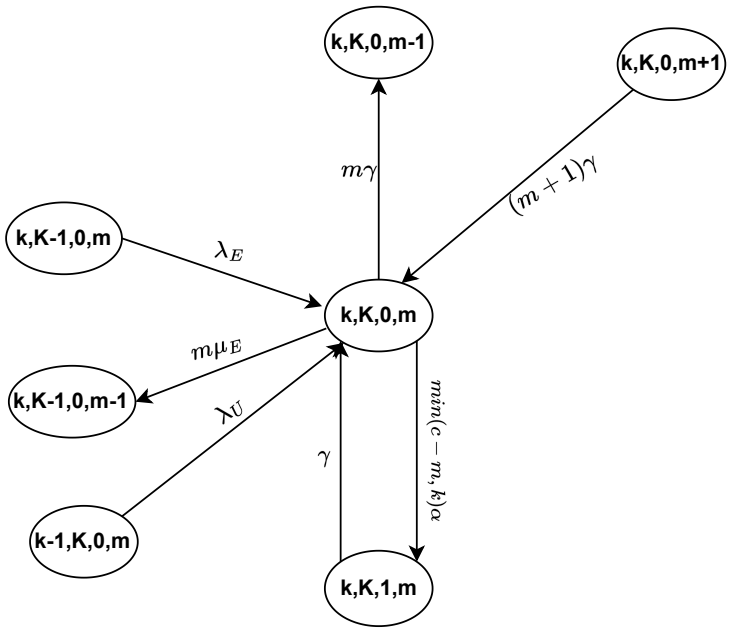
States in which the limits for eMBB users and URLLC users have been achieved and there are containers to be activated. Eq. 4.6.14 describes these states $(k, K, 0, m)$, with $K > c$,

Table 55 – Events related to the states $(k, K, 0, m)$, with $K > c$, $0 < m < c$, $i = k$, $j = K$ and $l = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✗

Source: The author (2023)

Figure 59 – States $(k, K, 0, m)$, with $K > c$, $0 < m < c$, $i = k$, $j = K$ and $l = 0$



Source: The author (2023)

$0 < m < c$, $i = k$, $j = K$ and $l = 0$. The state diagram and related events that denote these states are shown in Fig. 59 and Table 55.

$$
\begin{aligned}
&[m(\mu_E + \gamma) + min(c - m, k)\alpha]\pi(k, K, 0, m) \\
&= \lambda_U \pi(k-1, K, 0, m) + \lambda_E \pi(k, K-1, 0, m) + \gamma\pi(k, K, 1, m) + (m+1)\gamma\pi(k, K, 0, m+1)
\end{aligned}
\tag{4.6.14}
$$

State in which the limit for URLLC users has been achieved, but all containers of the system are processing eMBB services with no online waiting in line (see Fig. 60). In summary, states $(k, c, 0, c)$, with $j = c$, $i = k$, $j < K$, $m = c$, $l = 0$ whose related events and balance

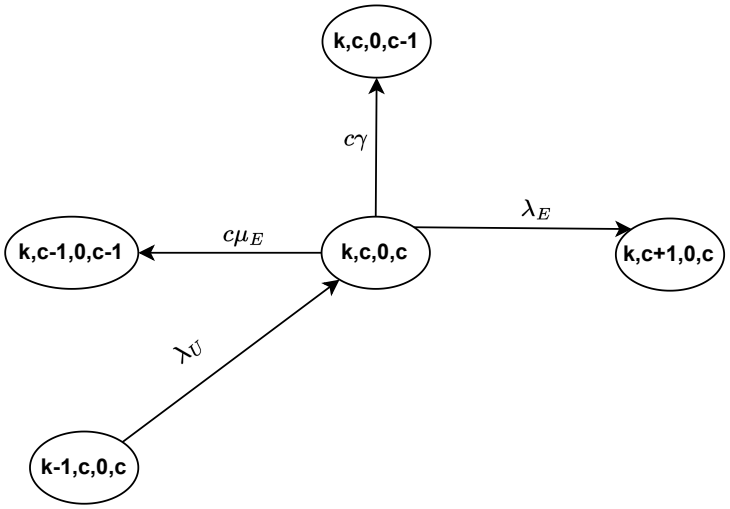
equation are presented in Table 56 and Eq. 4.6.15, respectively.

Table 56 – Events related to the states $(k, c, 0, c)$, with $j = c$, $i = k$, $j < K$, $m = c$, $l = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 60 – States $(k, j, 0, c)$, with $j > c$, $K - c > 1$, $i = k$, $0 < j < K$, $m = c$, $l = 0$



Source: The author (2023)

$$[\lambda_E + c(\mu_E + \gamma)]\pi(k, c, 0, c) = \lambda_U \pi(k - 1, c, 0, c) \quad (4.6.15)$$

State in which the limit for URLLC users has been achieved, all containers of the system are processing eMBB services, with ones waiting in line. Besides that, the system may still admit eMBB users. The diagram that denotes this state is shown in Fig. 61. In summary, states $(k, j, 0, c)$ with $j > c$, $K - c > 1$, $i = k$, $0 < j < K$, $m = c$, $l = 0$ as in Table 57 and Eq. 4.6.16.

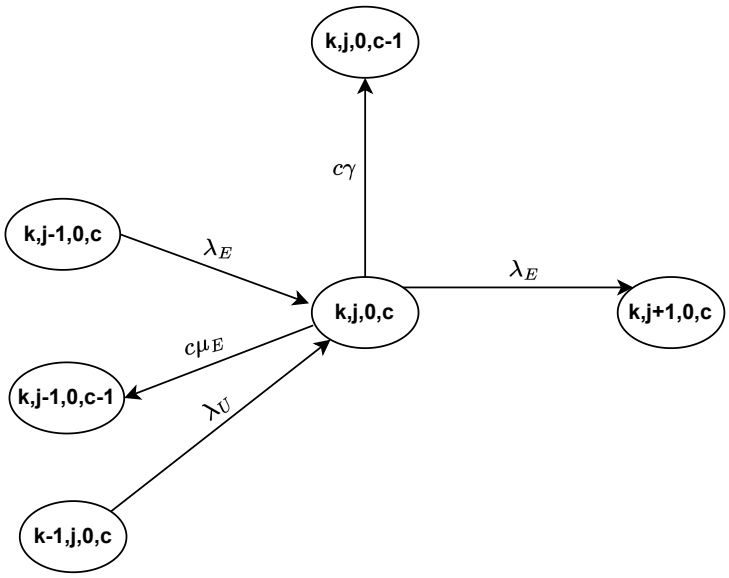
$$[\lambda_E + c(\mu_E + \gamma)]\pi(k, j, 0, c) = \lambda_U \pi(k - 1, j, 0, c) + \lambda_E \pi(k, j - 1, 0, c) \quad (4.6.16)$$

Table 57 – Events related to the states $(k, j, 0, c)$ with $j > c$, $K - c > 1$, $i = k$, $0 < j < K$, $m = c$, $l = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 61 – States $(k, j, 0, c)$, with $j > c$, $K - c > 1$, $i = k$, $0 < j < K$, $m = c$, $l = 0$



Source: The author (2023)

States in which the limit for URLLC and eMBB users has been achieved, and all containers are processing eMBB services. In summary, states $(k, K, 0, c)$, with $i = k$, $j = K$, $m = c$, $l = 0$. Fig 62 illustrates the state diagram of these states and Table 58 summarizes their related events. Additionally, their balance equation is given by Eq. 4.6.17.

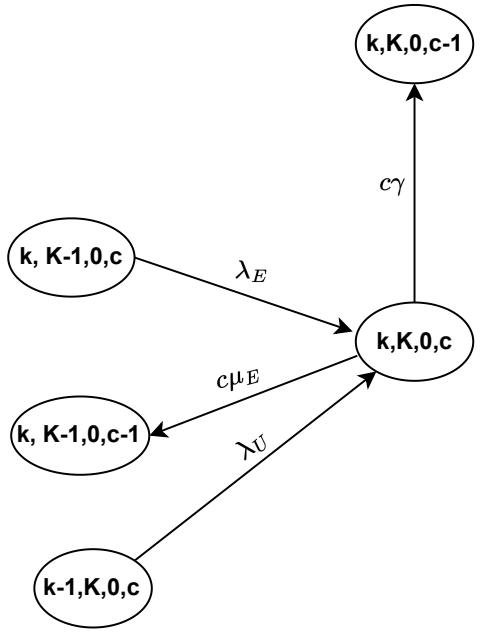
$$[c(\mu_E + \gamma)]\pi(k, K, 0, c) = \lambda_U \pi(k - 1, K, 0, c) + \lambda_E \pi(k, K - 1, 0, c) \quad (4.6.17)$$

Table 58 – Events related to the states $(k, K, 0, c)$, with $i = k$, $j = K$, $m = c$, $l = 0$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✗	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✗
eMBB service conclusion	✗	✗
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✗

Source: The author (2023)

Figure 62 – State $(k, K, 0, c)$, with $i = k$, $j = K$, $m = c$, $l = 0$



Source: The author (2023)

4.7 STATES (i, j, l, m), *with* $0 < i \leq k$, $0 < j \leq K$, $1 \leq m \leq c$, *and* $1 \leq l \leq c$

This section delineates the states in which there are users of both types and there is at least one active container serving eMBB and URLLC requests, i.e., states (i, j, l, m), with $0 < i \leq k$, $0 < j \leq K$, $1 \leq m \leq c$, and $1 \leq l \leq c$, which are divided into the following groups.

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB and URLLC being served, others waiting in line, available containers to be activated and the number of available containers to be activated is insufficient to process URLLC services in queue $(c - m - l < i - l)$. Eq. 4.7.1 describes these states (i, j, l, m), with

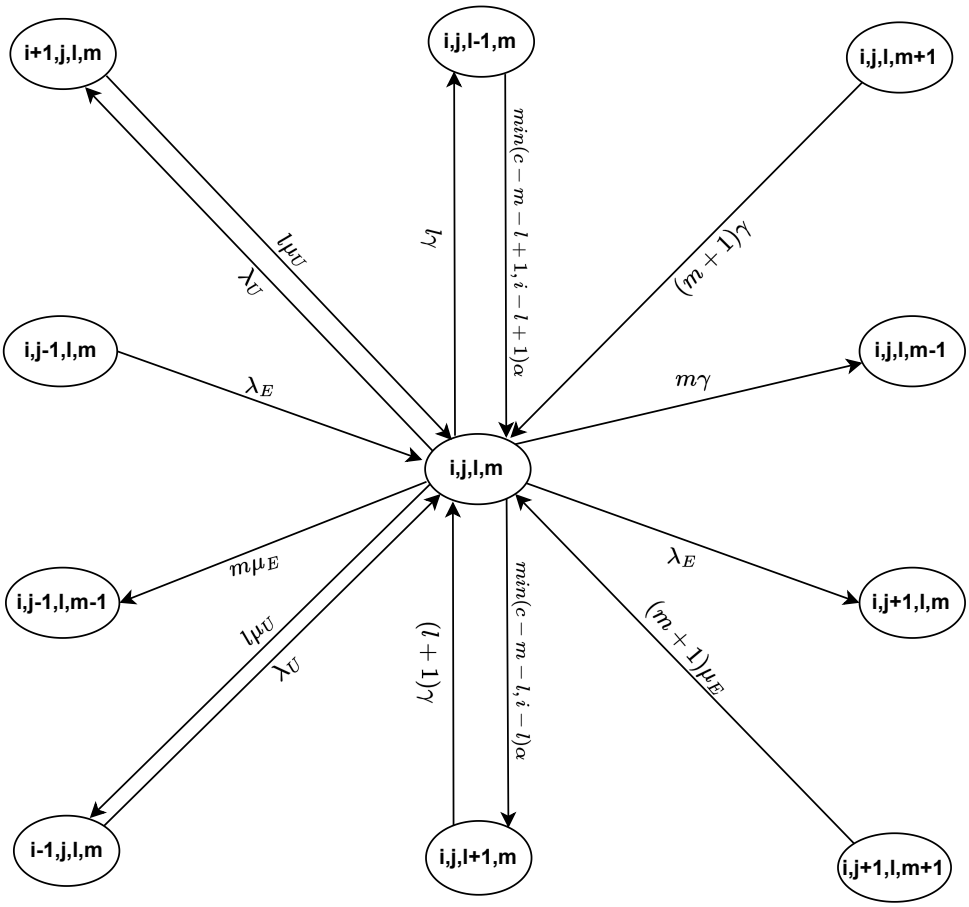
$0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $i > l$. Fig. 63 and Table 59 present their state diagram and related events.

Table 59 – Events related to the states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $i > l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 63 – States (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j > m$ and $(c - m - l) < (i - l)$



Source: The author (2023)

$$
\begin{aligned}
[\lambda_U + \lambda_E + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, i - l)\alpha]\pi(i, j, l, m) = \\
\lambda_U \pi(i - 1, j, l, m) + \lambda_E \pi(i, j - 1, l, m) + (m + 1)\mu_E \pi(i, j + 1, l, m + 1) \\
+ l\mu_U \pi(i + 1, j, l, m) + (l + 1)\gamma\pi(i, j, l + 1, m) + (m + 1)\gamma\pi(i, j, l, m + 1) \\
+ min(c - m - l + 1, i - l + 1)\alpha\pi(i, j, l - 1, m) \quad (4.7.1)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and some URLLC being served, others waiting in line, available containers to be activated, and the number of available containers to be activated are insufficient to process URLLC services in queue $(c - m - l < i - l)$. In summary, states (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j = m$, and $i > l$, which follow the Eq. 4.7.2, Table 60, and Fig.64

Table 60 – Events related to the states (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j = m$ and $i > l$.

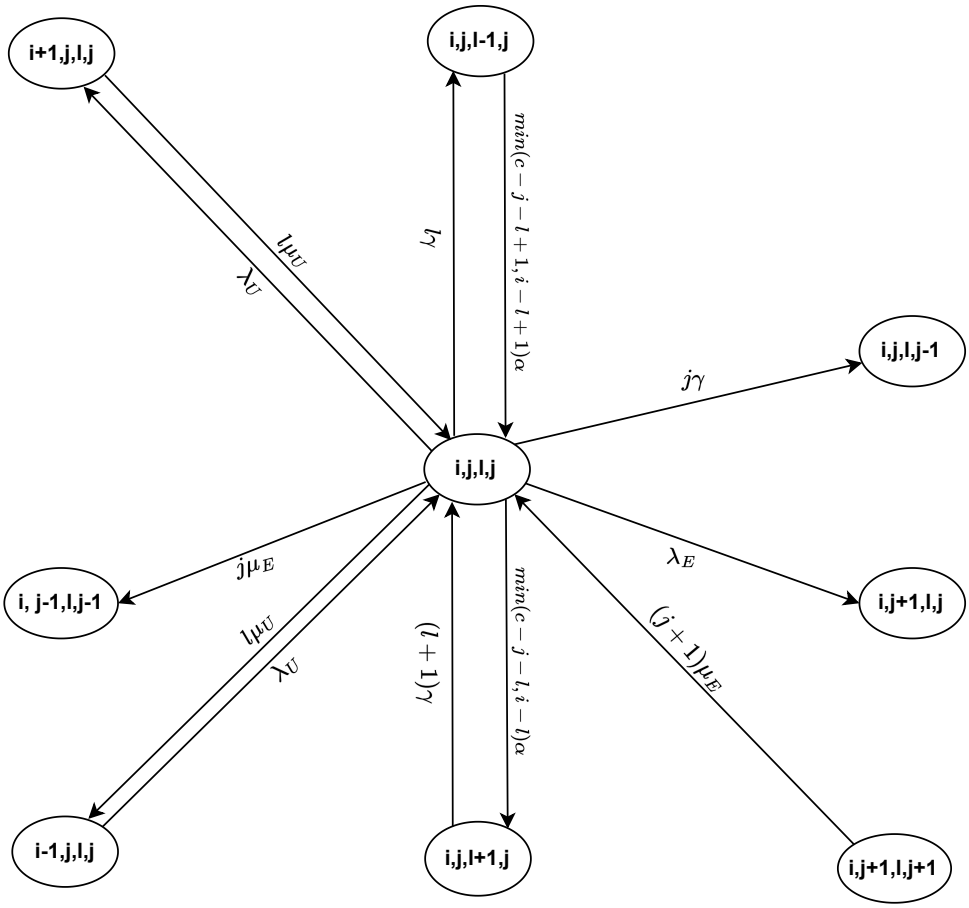
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$
\begin{aligned}
[\lambda_U + \lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - j - l, i - l)\alpha]\pi(i, j, l, j) \\
= \lambda_U \pi(i - 1, j, l, j) + (j + 1)\mu_E \pi(i, j + 1, l, j + 1) + l\mu_U \pi(i + 1, j, l, j) \\
+ (l + 1)\gamma\pi(i, j, l + 1, j) + min(c - j - l + 1, i - l + 1)\alpha\pi(i, j, l - 1, j) \quad (4.7.2)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and some URLLC being served, others waiting in line and no available containers to be activated $(c = m + l)$. In summary, states (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c = m + l)$, which are described by Eq. 4.7.3, Table 61, and Fig. 65.

Figure 64 – States (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j = m$ and $(c - m - l) < (i - l)$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(i, j, l, j) \\
&= \lambda_U \pi(i-1, j, l, j) + l\mu_U \pi(i+1, j, l, j) + min(c - j - l + 1, i - l + 1)\alpha\pi(i, j, l-1, j)
\end{aligned}
\tag{4.7.3}
$$

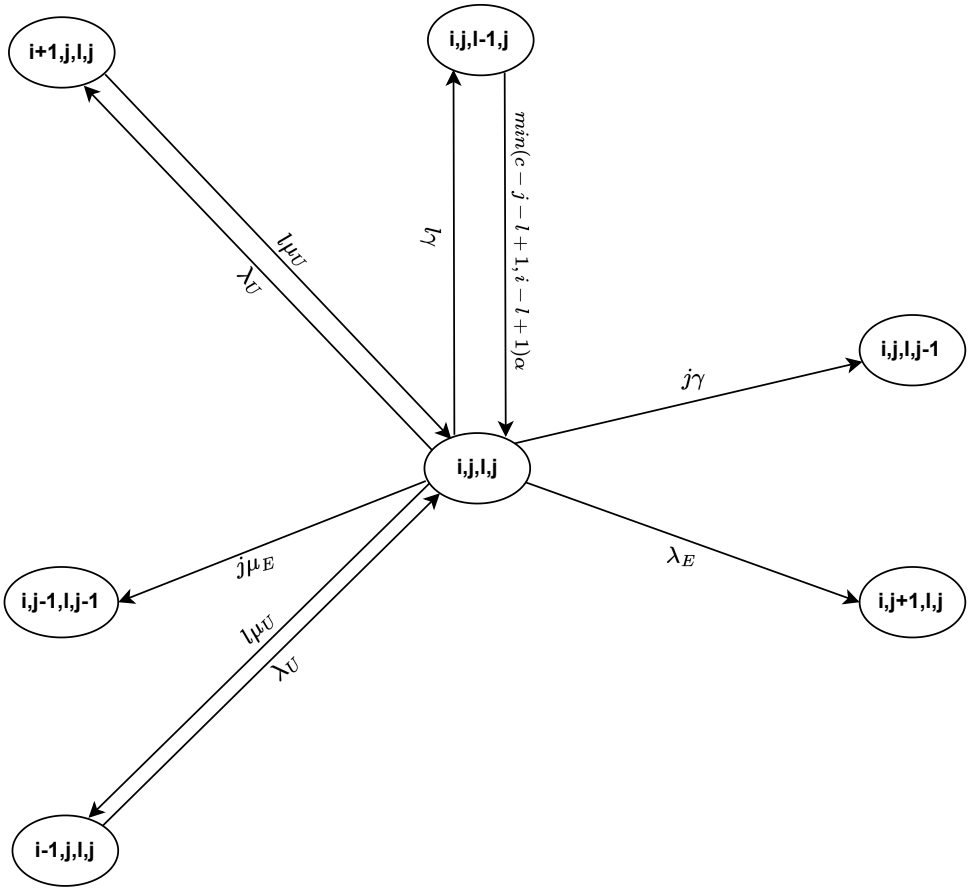
States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB and URLLC being served, others waiting in line and no available containers to be activated $(c = m + l)$. Eq. 4.7.4 describes these states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c = m + l)$. The state diagram and related events that denote these states are shown in Fig. 66 and Table 62.

Table 61 – Events related to the states $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 65 – States (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j = m$ and $(c = m+l)$



Source: The author (2023)

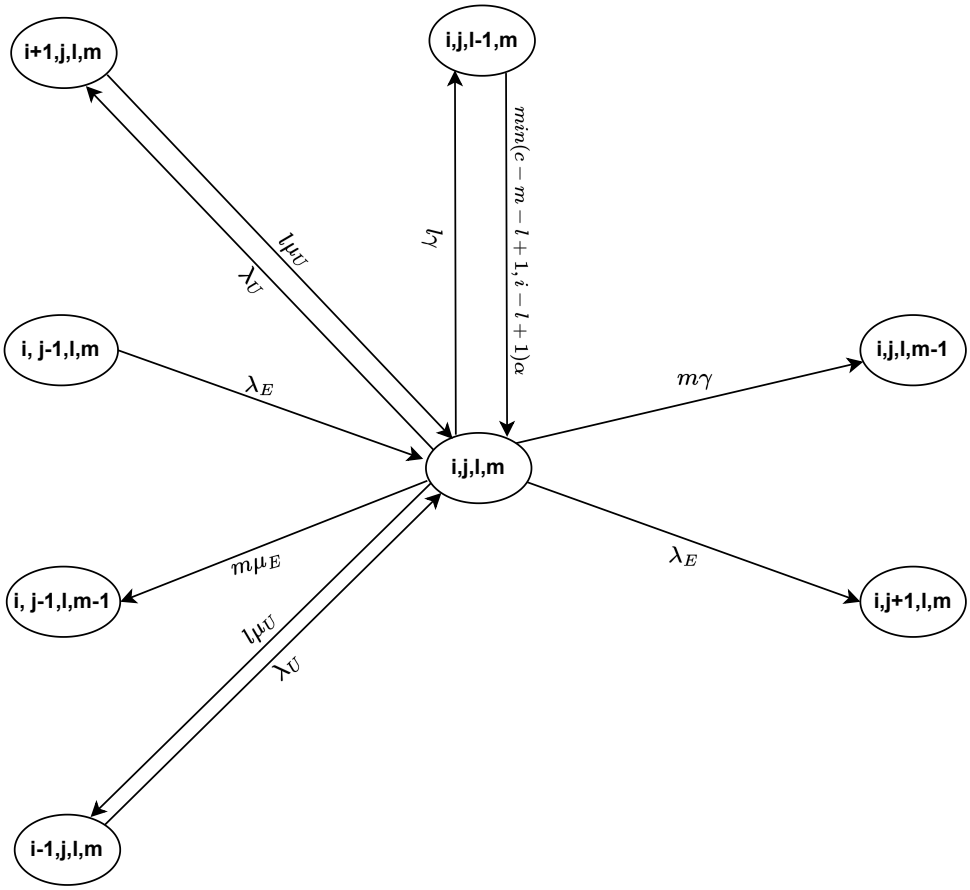
$$[\lambda_U + \lambda_E + m(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(i, j, l, m)$$
$$= \lambda_U \pi(i-1, j, l, m) + \lambda_E \pi(i, j-1, l, m) + l\mu_U \pi(i+1, j, l, m)$$
$$+ min(c - m - l + 1, i - l + 1)\alpha\pi(i, j, l-1, m) \quad (4.7.4)$$

Table 62 – Events related to the states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 66 – States (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j > m$ and $(c = m + l)$



Source: The author (2023)

States in which the limit for eMBB users has been achieved and the number of URLLC users is lower than their limits, with some eMBB and URLLC being served, others waiting in line and no available containers to be activated $(c = m + l)$ (see Fig. 67). In brief, states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c = m + l)$, whose

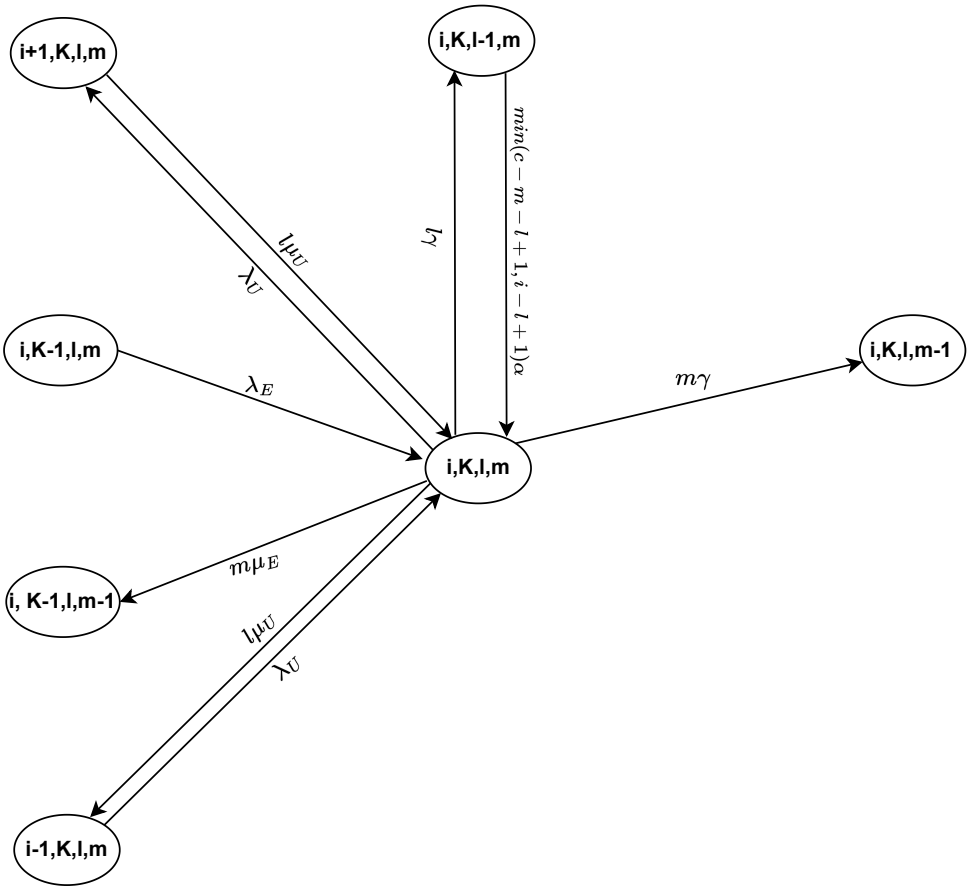
related events and balance equation are presented in Table 63 and Eq. 4.7.5, respectively.

Table 63 – Events related to the states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	
xmark		
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 67 – States (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$, and $(c - m - l < i - l)$



Source: The author (2023)

$$[\lambda_U + m(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(i, K, l, m)$$
$$= \lambda_U \pi(i-1, K, l, m) + \lambda_E \pi(i, K-1, l, m) + l\mu_U \pi(i+1, K, l, m)$$
$$+ min(c - m - l + 1, i - l + 1)\alpha\pi(i, K, l-1, m) \quad (4.7.5)$$

States in which the limit for eMBB users has been achieved and the number of URLLC users is under the limit, with some eMBB and URLLC being served, others waiting in line, available containers to be activated but they are insufficient to process URLLC services in queue $(c - m - l < i - l)$. The diagram that denotes these states states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c - m - l < i - l)$ is is shown in Fig. 68. Table 64 and 4.7.6 present their related events and balance equation.

Table 64 – Events related to the states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c - m - l < i - l)$.

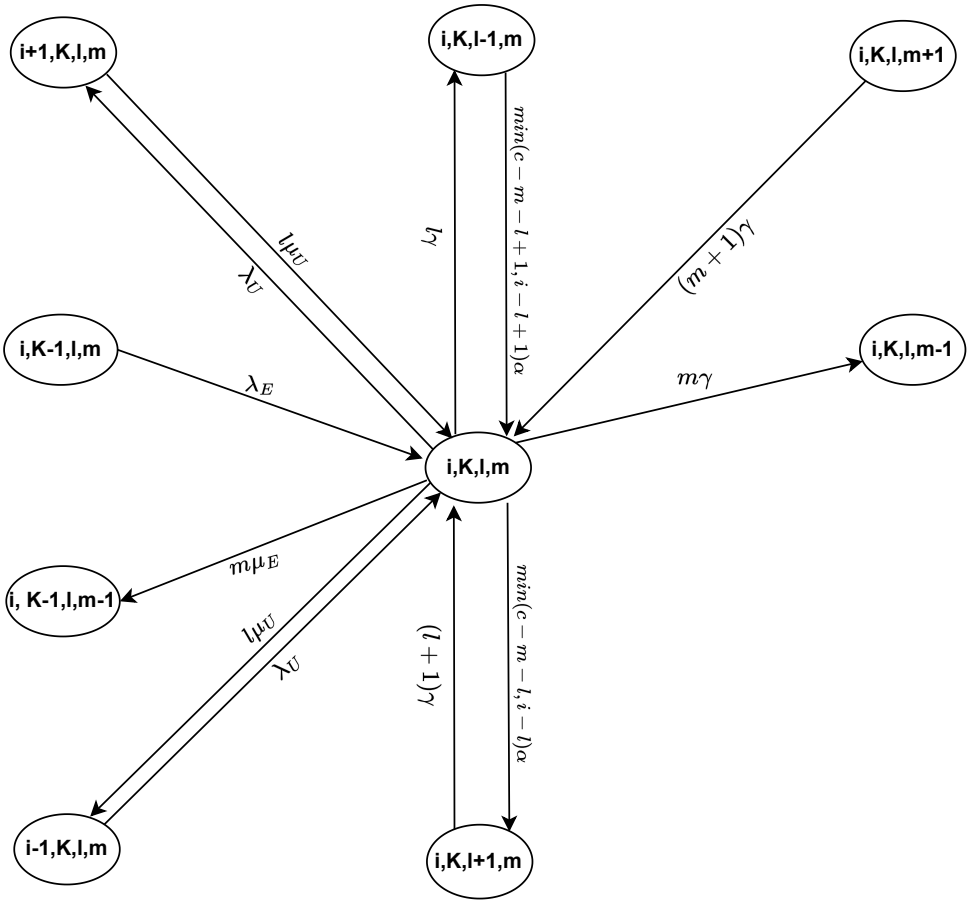
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$[\lambda_U + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, i + K - m - l)\alpha]\pi(i, K, l, m) =$$
$$\lambda_U \pi(i-1, K, l, m) + \lambda_E \pi(i, K-1, l, m) + l\mu_U \pi(i+1, K, l, m)$$
$$+min(c-m-l+1, i-l+1)\alpha\pi(i, K, l-1, m)+(l+1)\gamma\pi(i, K, l+1, m)+(m+1)\gamma\pi(i, K, l, m+1) \quad (4.7.6)$$

States in which the limit for URLLC users has been achieved and the number of eMBB users is lower than their limit, with all eMBB and some URLLC being served, others waiting in line, available containers to be activated, and the number of available containers to be activated are insufficient to process URLLC services in queue $(c - m - l < i - l)$. In summary, states (k, j, l, j), with $i = k$, $0 < j < K$, $j = m$, $0 < l < c$, $0 < m < c$ and $(c - m - l < i - l)$.

Figure 68 – States (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$, and $(c - m - l < i - l)$



Source: The author (2023)

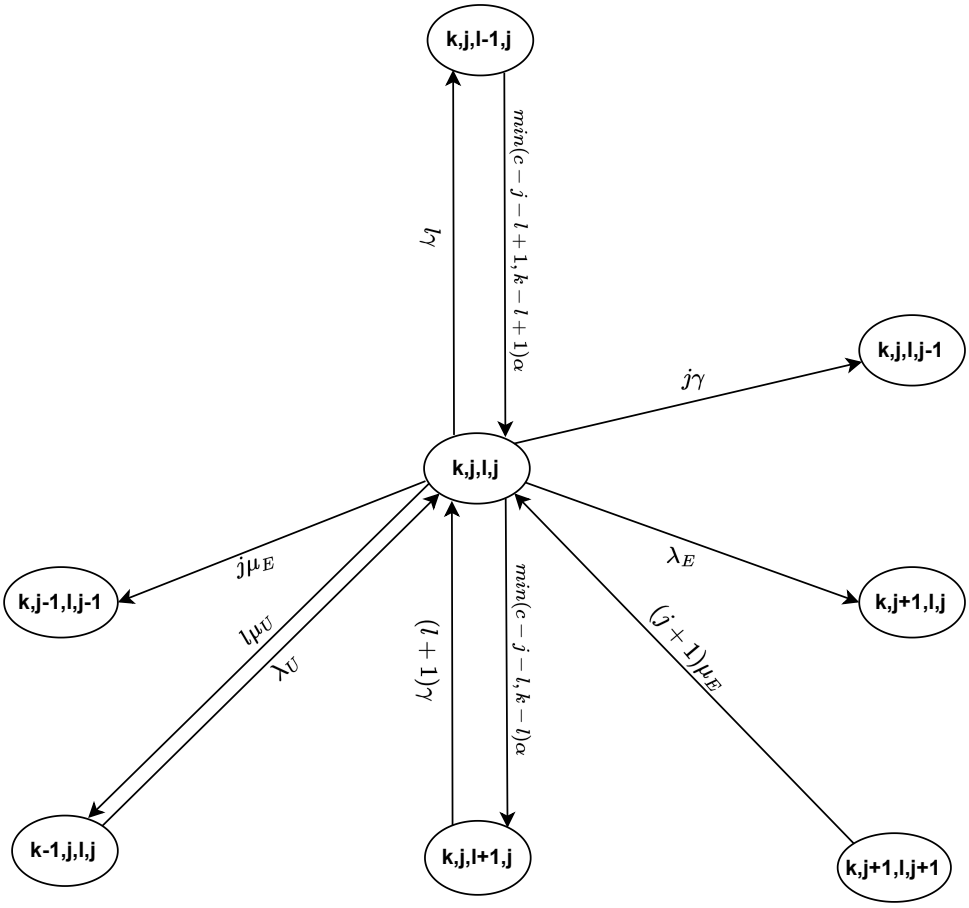
The balance equation, state diagram, and related events are given by Eq. 4.7.7, Fig. 69, and Table 65, respectively.

Table 65 – Events related to the states (k, j, l, j), with $i = k$, $0 < j < K$, $j = m$, $0 < l < c$, $0 < m < c$ and $(c - m - l < i - l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 69 – States (k, j, l, j), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j = m$ and $(c - m - l) < (i - l)$



Source: The author (2023)

$$[\lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - j - l, k - l)\alpha]\pi(k, j, l, j) =$$
$$\lambda_U \pi(k - 1, j, l, m) + (j + 1)\mu_E \pi(k, j + 1, l, j + 1) + (l + 1)\gamma\pi(k, j, l + 1, j)$$
$$+ min(c - j - l + 1, k - l + 1)\alpha\pi(k, j, l - 1, j) \quad (4.7.7)$$

States in which the limit for URLLC users has been achieved and the number of eMBB users is lower than his limits, with all eMBB and some URLLC being served, others waiting in line and no available containers to be activated (see Fig. 70). In short, states (k, j, l, j), with $i = k$, $0 < j < K$, $j = m$, $0 < l < c$, $0 < m < c$ and $(c = m + l)$, whose balance equation and related event are denoted in Eq. 4.7.8 and Table 66.

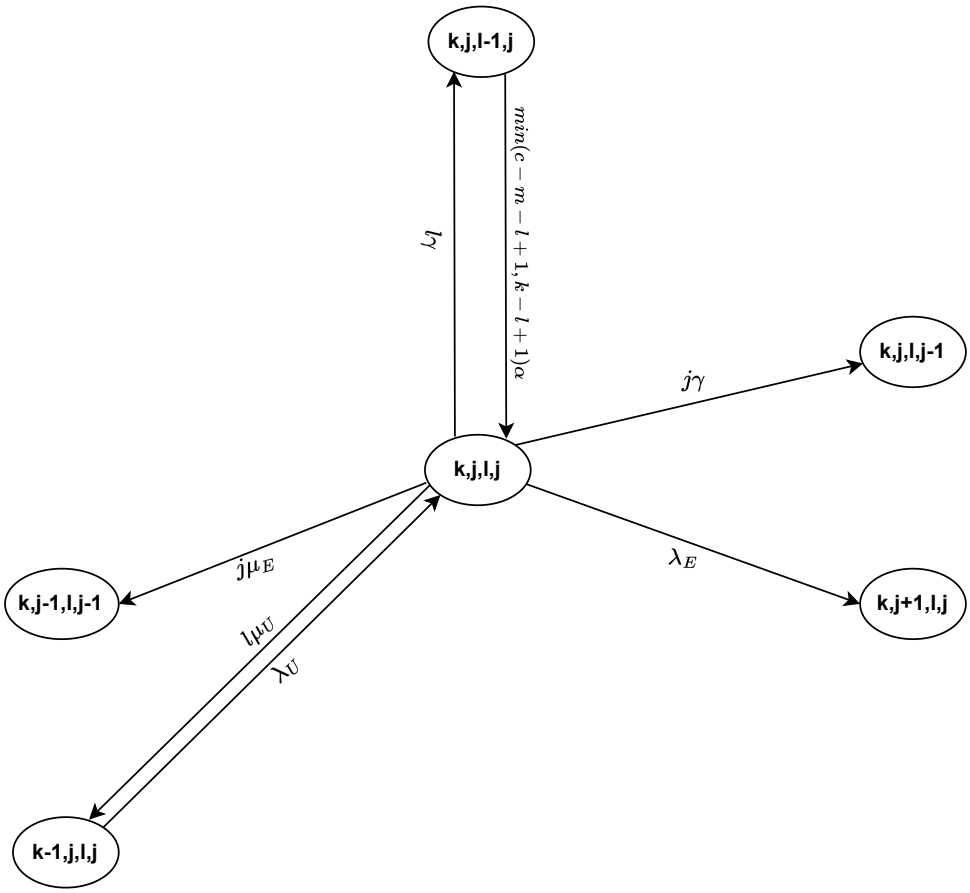
$$[\lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(k, j, l, j)$$
$$= \lambda_U \pi(k - 1, j, l, j) + min(c - m - l + 1, k - l + 1)\alpha\pi(k, j, l - 1, j) \quad (4.7.8)$$

Table 66 – Events related to the states (k, j, l, j), with $i = k$, $0 < j < K$, $j = m$, $0 < l < c$, $0 < m < c$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 70 – States (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $(c - m - l < i - l)$



Source: The author (2023)

States in which the limit for URLLC users has been achieved and the number of eMBB users is lower than their limit, with some eMBB and URLLC being served, others waiting in line, available containers to be activated and the number of available containers to be activated is insufficient to process URLLC services in queue $(c - m - l < i - l)$. In summary, states (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$ and $j > m$. Fig 71 illustrates the

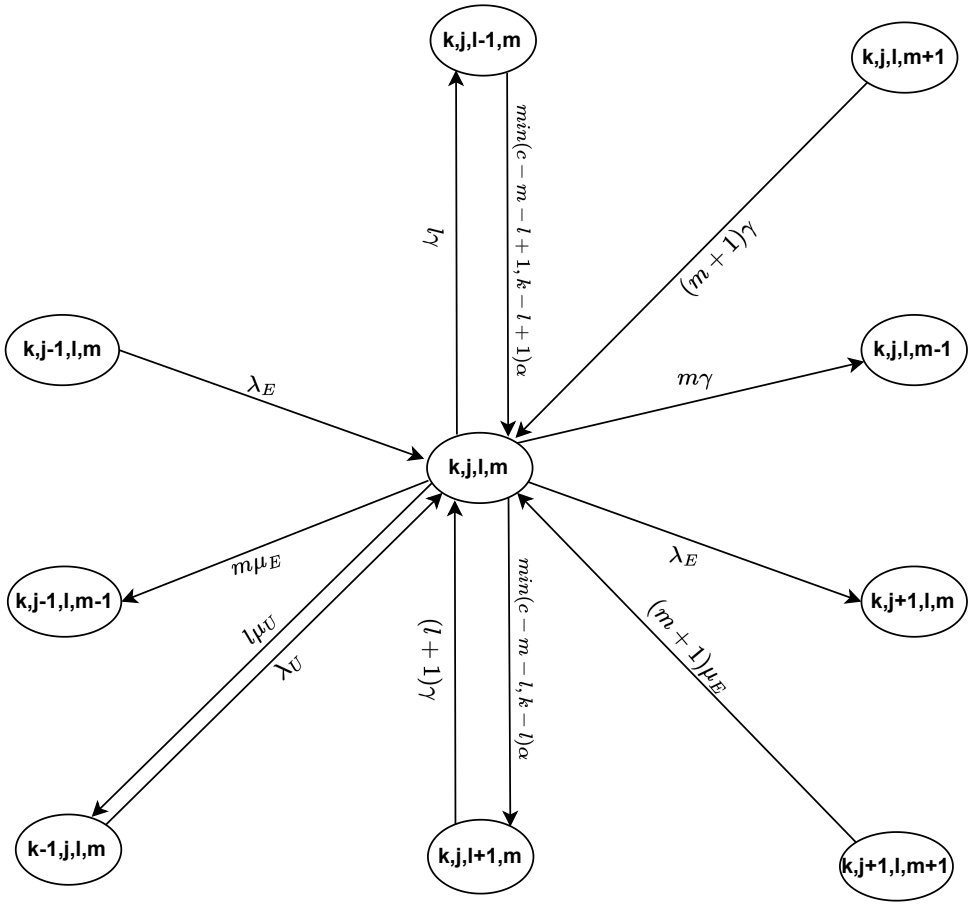
state diagram of these states and Table 67 summarizes their related events. Additionally, their balance equation is given by Eq. 4.7.9.

Table 67 – Events related to the states (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$ and $j > m$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 71 – States (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $(c - m - l < i - l)$



Source: The author (2023)

$$[\lambda_E + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, k - l)\alpha]\pi(k, j, l, m) =$$
$$\lambda_U\pi(k-1, j, l, m)+\lambda_E\pi(k, j-1, l, m)+(m+1)\mu_E\pi(k, j+1, l, m+1)+(l+1)\gamma\pi(k, j, l+1, m)$$
$$+ (m + 1)\gamma\pi(k, j, l, m + 1) + min(c - m - l + 1, k - l + 1)\alpha\pi(k, j, l - 1, m) \quad (4.7.9)$$

States in which the limit for URLLC users has been achieved and the number of eMBB users is lower than their limit, with some eMBB and URLLC being served, others waiting in line and no available containers to be activated (see Fig. 72). In brief, states (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $(c = m+l)$. They follow the Eq. 4.7.10 and their related events are listed in Table 68.

Table 68 – Events related to the states (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $(c = m + l)$.

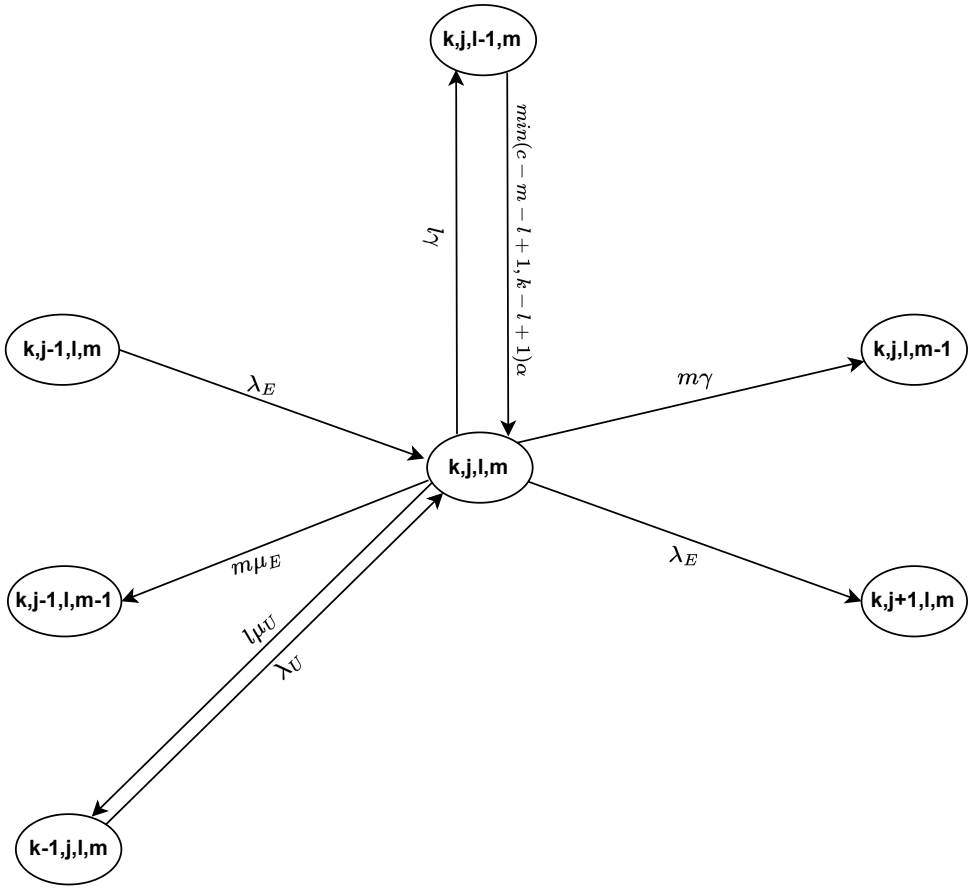
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

$$[\lambda_E + m(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(k, j, l, m)$$
$$= \lambda_U\pi(k-1, j, l, m) + \lambda_E\pi(k, j-1, l, m) + min(c-m-l+1, k-l+1)\alpha\pi(k, j, l-1, m) \quad (4.7.10)$$

States in which the limit for URLLC and eMBB users has been achieved, with some eMBB and URLLC being served, others waiting in line, available containers to be activated and the number of available containers to be activated are insufficient to process URLLC services in queue $(c - m - l < i - l)$. Fig 73 illustrates the state diagram of these states. In summary, state (k, K, l, m), with $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c - m - l < i - l)$, whose balance equation and related event are denoted in Eq. 4.7.11 and Table 69.

Figure 72 – States (k, j, l, m), with $i = k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$ and $(c = m + l)$



Source: The author (2023)

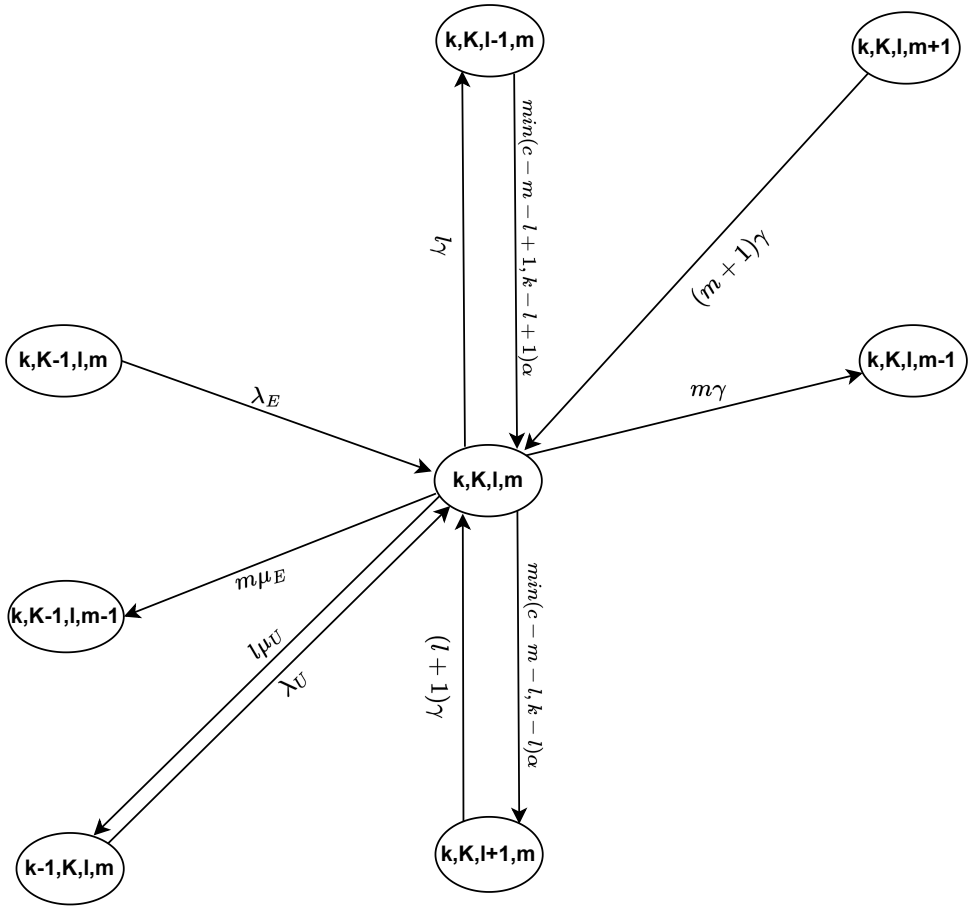
Table 69 – Events related to the states (k, K, l, m), with $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c - m - l < i - l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$
\begin{aligned}
&[m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, k - l)\alpha]\pi(k, K, l, m) \\
&= \lambda_U \pi(k-1, K, l, m) + \lambda_E \pi(k, K-1, l, m) + (l+1)\gamma\pi(k, K, l+1, m) + (m+1)\gamma\pi(k, K, l, m+1) \\
&\quad + min(c - m - l + 1, k - l + 1)\alpha\pi(k, K, l-1, m) \quad (4.7.11)
\end{aligned}
$$

Figure 73 – States (k, K, l, m), with $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c - m - l < i - l)$



Source: The author (2023)

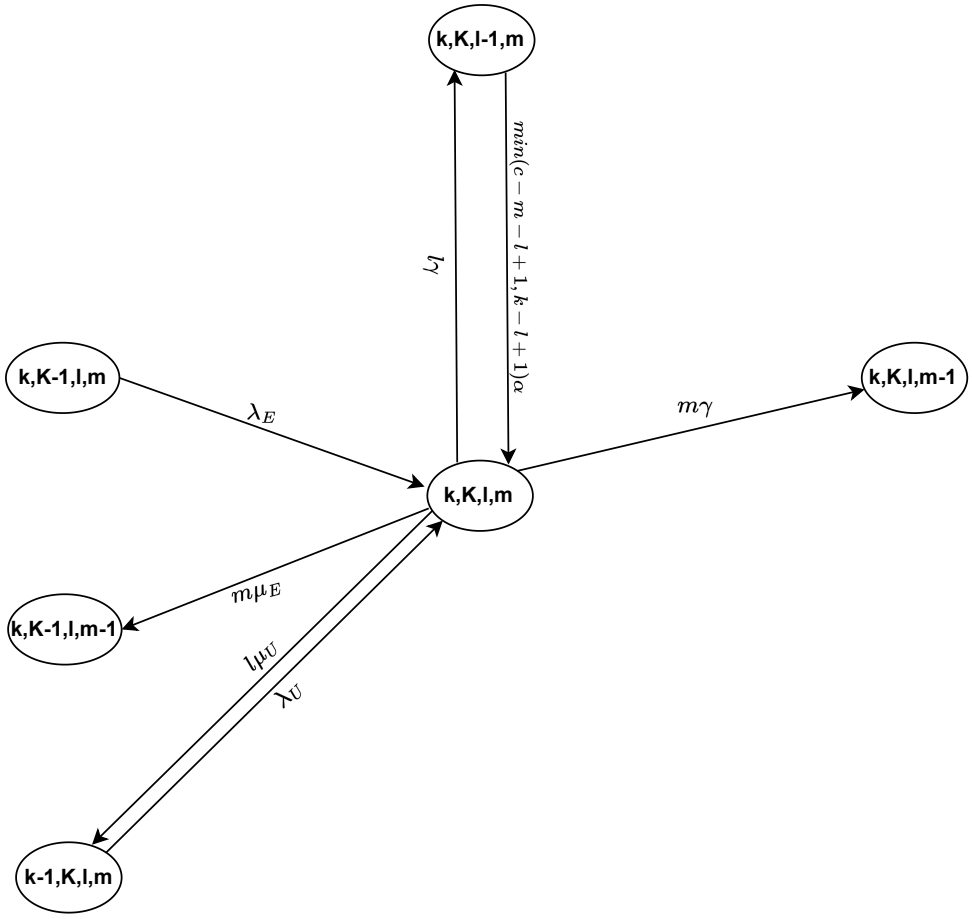
States in which the limit for URLLC and eMBB users has been achieved, with some eMBB and URLLC being served, others waiting in line and no available containers to be activated (see Fig. 74). They are denoted as (k, K, l, m), with $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c = m + l)$ and follow the Eq. 4.7.12.Their related events are listed in Table 70.

Table 70 – Events related to the states $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✗
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✗	✗
URLLC service conclusion	✗	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 74 – States (k, K, l, m), with $i = k$, $j = K$, $0 < l < c$, $0 < m < c$ and $(c = m + l)$



Source: The author (2023)

$$
\begin{aligned}
&[m(\mu_E + \gamma) + l(\mu_U + \gamma)]\pi(k, K, l, m) \\
&= \lambda_U \pi(k-1, K, l, m) + \lambda_E \pi(k, K-1, l, m) + min(c-m-l+1, k+K-m-l+1)\alpha\pi(k, K, l-1, m)
\end{aligned} \tag{4.7.12}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and URLLC being served (see Fig. 75). Eq. 4.7.13 describes these states (i, j, i, j), with $0 < i < c$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $l + m < c$, $j = m$, $i = l$. Table 71 denotes their related events.

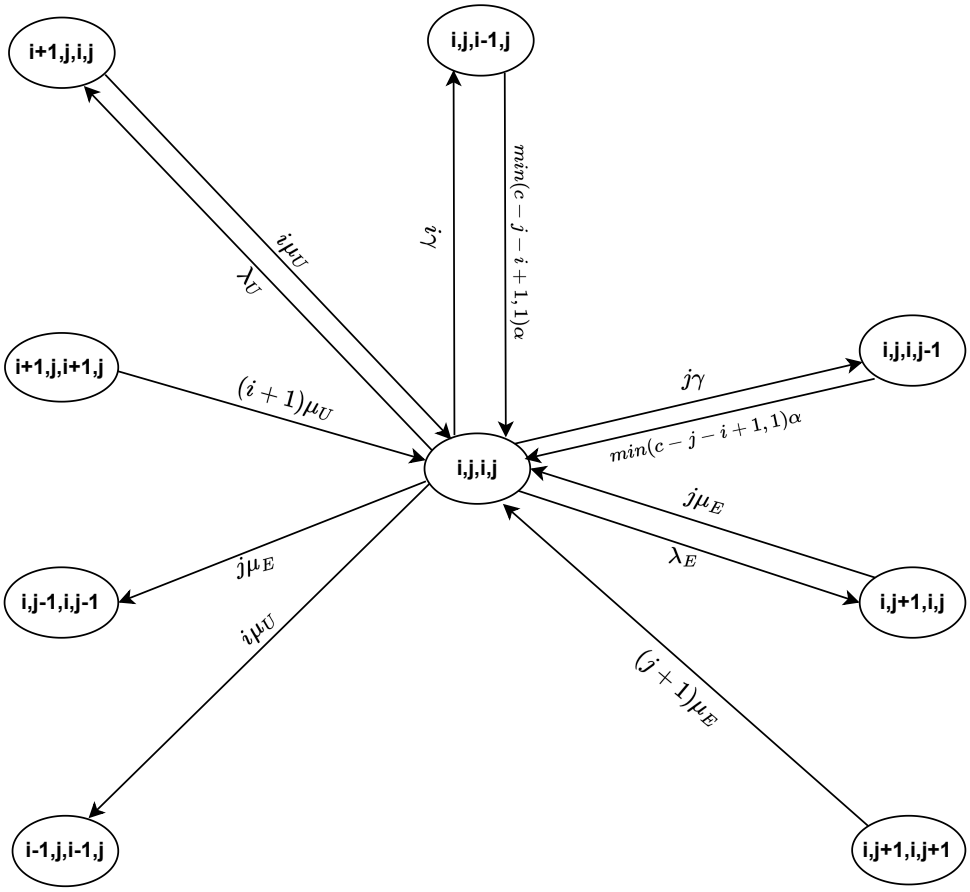
$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + i(\mu_U + \gamma)]\pi(i, j, i, j) \\
&= i\mu_U \pi(i+1, j, i, j) + (i+1)\mu_U \pi(i+1, j, i+1, j) + j\mu_E \pi(i, j+1, i, j) + (j+1)\mu_E \pi(i, j+1, i, j+1) \\
&+ min(c-j-i+1, 1)\alpha\pi(i, j, i-1, j) + min(c-j-i+1, 1)\alpha\pi(i, j, i, j-1)
\end{aligned} \tag{4.7.13}
$$

Table 71 – Events related to the states $0 < i < c$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $l + m < c$, $j = m$, $i = l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 75 – States (i, j, i, j), with $0 < i < c$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $l + m < c$, $j = m$, $i = l$



Source: The author (2023)

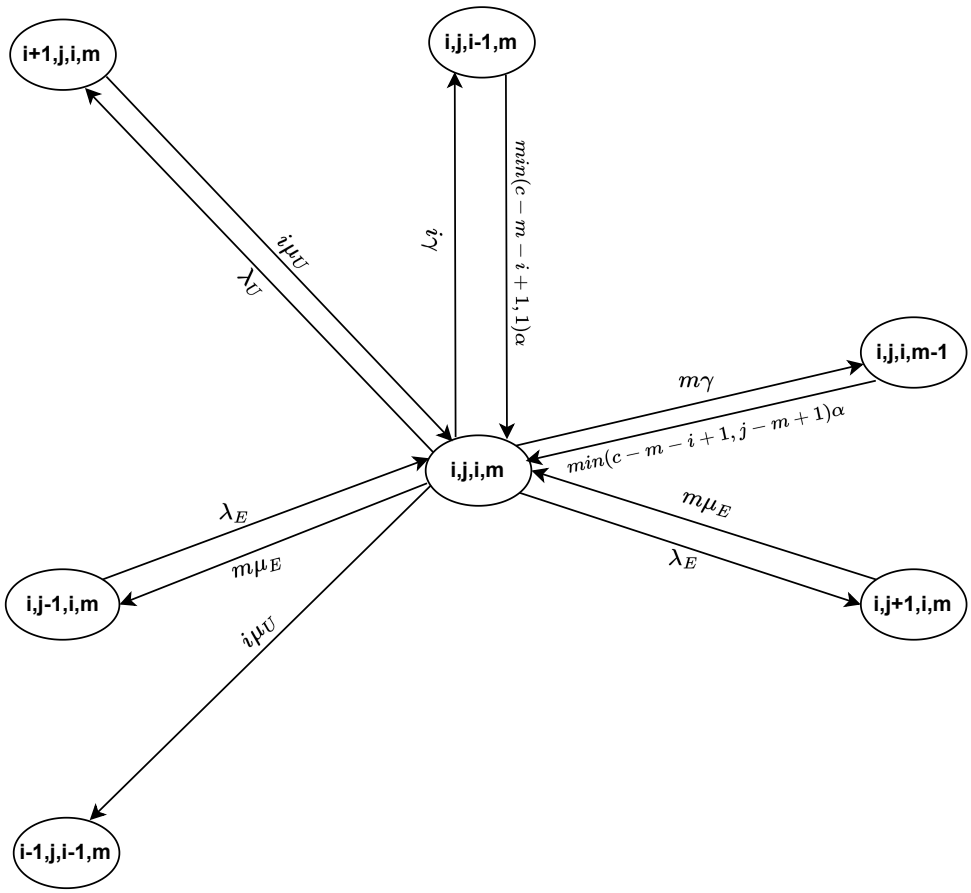
States in which the number of URLLC and eMBB users is lower than their respective limits, with all URLLC and some eMBB being served and no available containers to be activated $(c = m + l)$. In brief, states (i, j, i, m), with $0 < i < c$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c = m + l)$. Fig 76 illustrates the state diagram of these states and Table 72 summarizes their related events. Additionally, their balance equation is given by Eq. 4.7.14.

Table 72 – Events related to the states $0 < i < c$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 76 – States (i, j, i, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i = l$, $j > m$ and $(c = m + l)$



Source: The author (2023)

$$[\lambda_U + \lambda_E + m(\mu_E + \gamma) + i(\mu_U + \gamma)]\pi(i,j,i,m) =$$

$$\lambda_E(i,j{-}1,i,m){+}i\mu_U\pi(i{+}1,j,i,m){+}m\mu_E\pi(i,j{+}1,i,m){+}min(c{-}m{-}i{+}1,1)\alpha\pi(i,j,i{-}1,m)$$

$$+ min(c-m-i+1, j-m+1)\alpha\pi(i,j,i,m-1) \quad (4.7.14)$$

States in which the limit for eMBB users has been achieved and the number of URLLC users is lower than their limit, with all URLLC and some eMBB being served and no available containers to be activated $(c = m+l)$. In summary, states (i, K, i, m), with $0 < i < c$, $j = K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c = m + l)$. The balance equation, state diagram, and related events are given by Eq. 4.7.15, Fig. 77, and Table 73, respectively.

Table 73 – Events related to the states $0 < i < c$, $j = K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c = m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

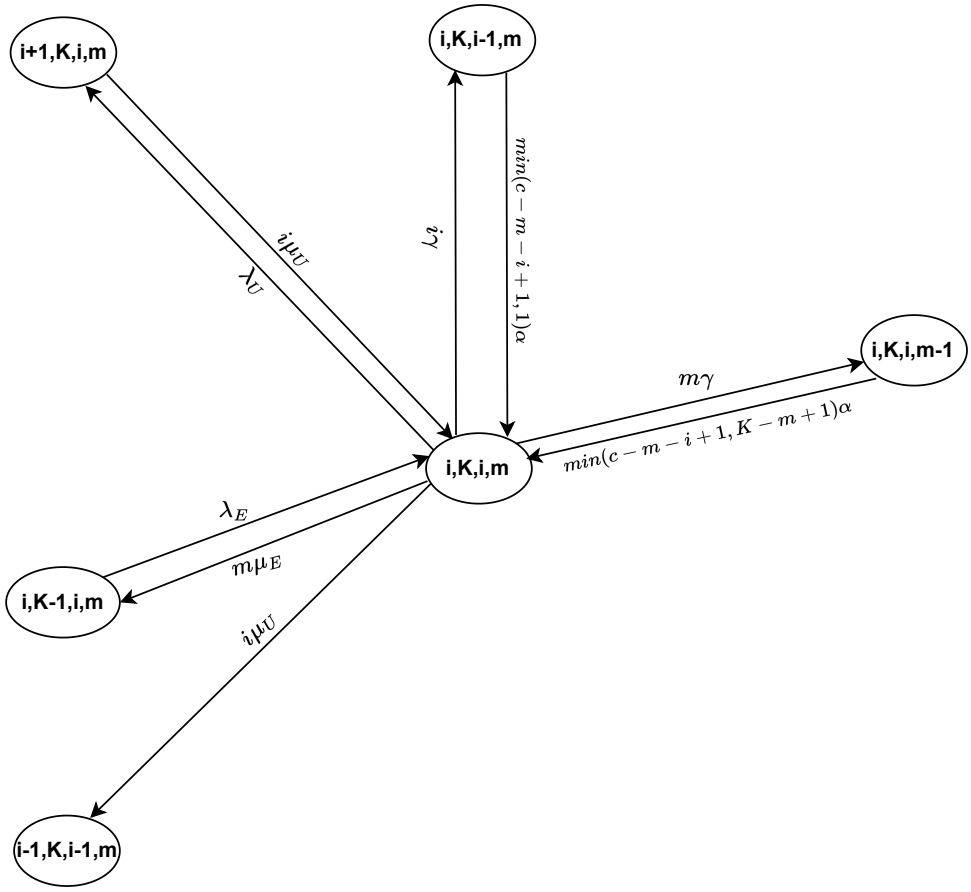
Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + m(\mu_E + \gamma) + i(\mu_U + \gamma)]\pi(i, K, i, m) \\
&= \lambda_E(i, K-1, i, m) + i\mu_U\pi(i+1, K, i, m) + min(c-m-i+1, 1)\alpha\pi(i, K, i-1, m) \\
&\qquad + min(c-m-i+1, K-m+1)\alpha\pi(i, K, i, m-1) \quad (4.7.15)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and some URLLC being served, available containers to be activated $(c > m+l)$ and the number of available containers to be activated is sufficient to process URLLC services in queue $(c-m-l = i-l)$. Fig 78 illustrates the state diagram of these states. These states are denoted as (i, j, l, j), with $0 < i < k$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c > m + l)$ and have balance equation and related events described in Eq. 4.7.16 and Table 74, respectively.

$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c-j-l, i-l)\alpha]\pi(i, j, l, j) \\
&= \lambda_U(i-1, j, l, j) + l\mu_U\pi(i+1, j, l, j) + j\mu_E\pi(i, j+1, l, j) + (j+1)\mu_E\pi(i, j+1, i, j+1) \\
&+(l+1)\gamma(i, j, l+1, j)+min(c-j-l+1, i-l+1)\alpha\pi(i, j, l-1, j)+min(c-i-j+1, 1)\alpha\pi(i, j, l, j-1) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4.7.16)
\end{aligned}
$$

Figure 77 – States (i, K, i, m), with $0 < i < c$, $j = K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c = m + l)$



Source: The author (2023)

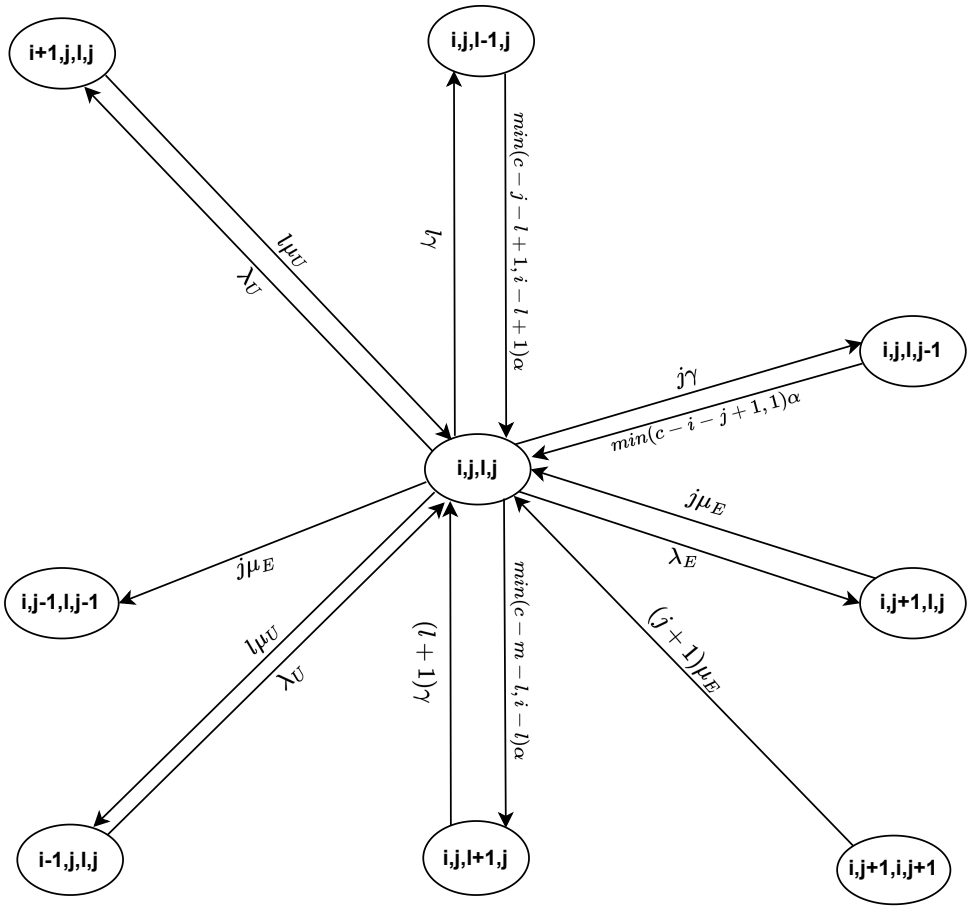
Table 74 – Events related to the states (i, j, l, j), with $0 < i < k$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB and some URLLC being served, available containers to be activated $(c > m + l)$ and the number of available containers to be activated are sufficient to process URLLC services in queue $(c - m - l = i - l)$. Eq. 4.7.17 describes these states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c > m + l)$. The state

Figure 78 – States (i, j, l, j), with $0 < i < k$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$, $(c > m + l)$ and $(c - m - l = i - l)$



Source: The author (2023)

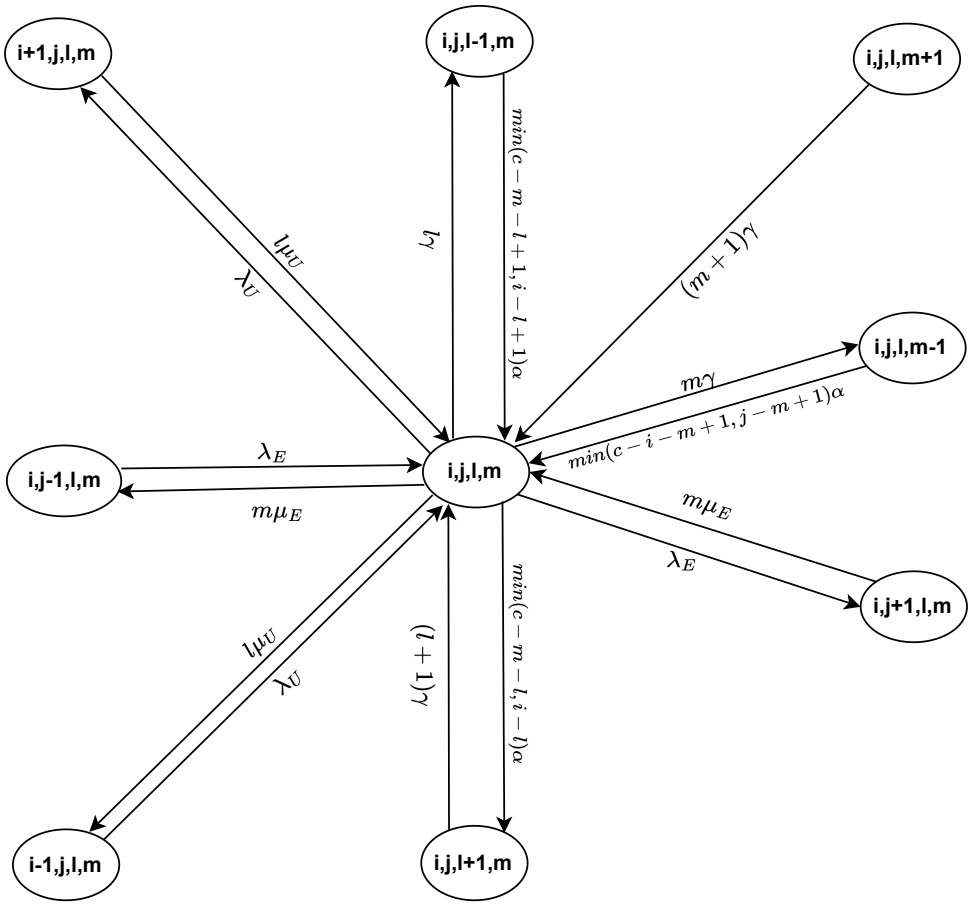
diagram and related events that denote these states are shown in Fig. 79 and Table 75.

Table 75 – Events related to the states $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 79 – States (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j = m$ and $(c > m + l)$



Source: The author (2023)

$$[\lambda_U + \lambda_E + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, i - l)\alpha]\pi(i, j, l, m) =$$
$$\lambda_U \pi(i-1, j, l, m) + \lambda_E \pi(i, j-1, l, m) + m\mu_E \pi(i, j+1, l, m) + l\mu_U \pi(i+1, j, l, m)$$
$$+(l+1)\gamma\pi(i, j, l+1, m)+(m+1)\gamma\pi(i, j, l, m+1)+min(c-m-l+1, i-l+1)\alpha\pi(i, j, l-1, m)$$
$$+ min(c - i - m + 1, j - m + 1)\alpha\pi(i, j, l, m-1) \quad (4.7.17)$$

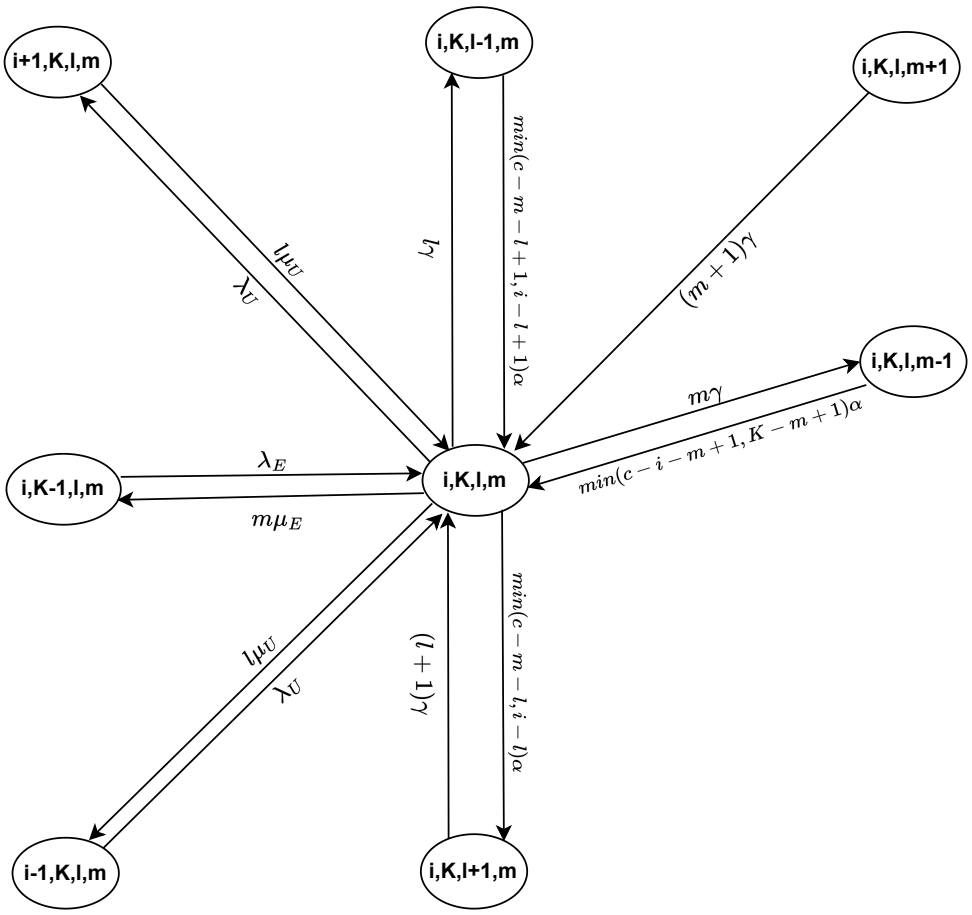
States in which the limit for eMBB users has been achieved and the number of URLLC users is lower than their limit, with some eMBB and URLLC being served, others waiting in line, available containers to be activated and the number of available containers to be activated is sufficient to process URLLC services in queue $(c - m - l = i - l)$, as shown in Fig. 80. In summary, states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c > m + l)$. they follow the Eq. 4.7.18 and their related events are listed in Table 76.

Table 76 – Events related to the states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 80 – States (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$, $(c > m + l)$ and $(c - m - l = i - l)$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - m - l, i - l)\alpha]\pi(i, K, l, m) \\
&= \lambda_U \pi(i-1, K, l, m) + \lambda_E \pi(i, K-1, l, m) + l\mu_U \pi(i+1, K, l, m) + (l+1)\gamma\pi(i, K, l+1, m) \\
&\quad + (m+1)\gamma\pi(i, K, l, m+1) + min(c - m - l + 1, i - l + 1)\alpha\pi(i, K, l-1, m) \\
&\quad + min(c - i - m + 1, K - m + 1)\alpha\pi(i, K, l, m-1) \quad (4.7.18)
\end{aligned}
$$

States in which the limit for eMBB users has been achieved and the number of URLLC users is lower than their limit, with some eMBB and URLLC being served, others waiting in line, available containers to be activated and the number of available containers to be activated is more than sufficient to process URLLC services in queue $(c-m-l > i-l)$. Fig 81 illustrates the state diagram of these states. In summary, states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c > m + l)$, whose balance equation and related events are denoted in Eq. 4.7.19 and Table 77.

Table 77 – Events related to the states (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$ and $(c > m + l)$.

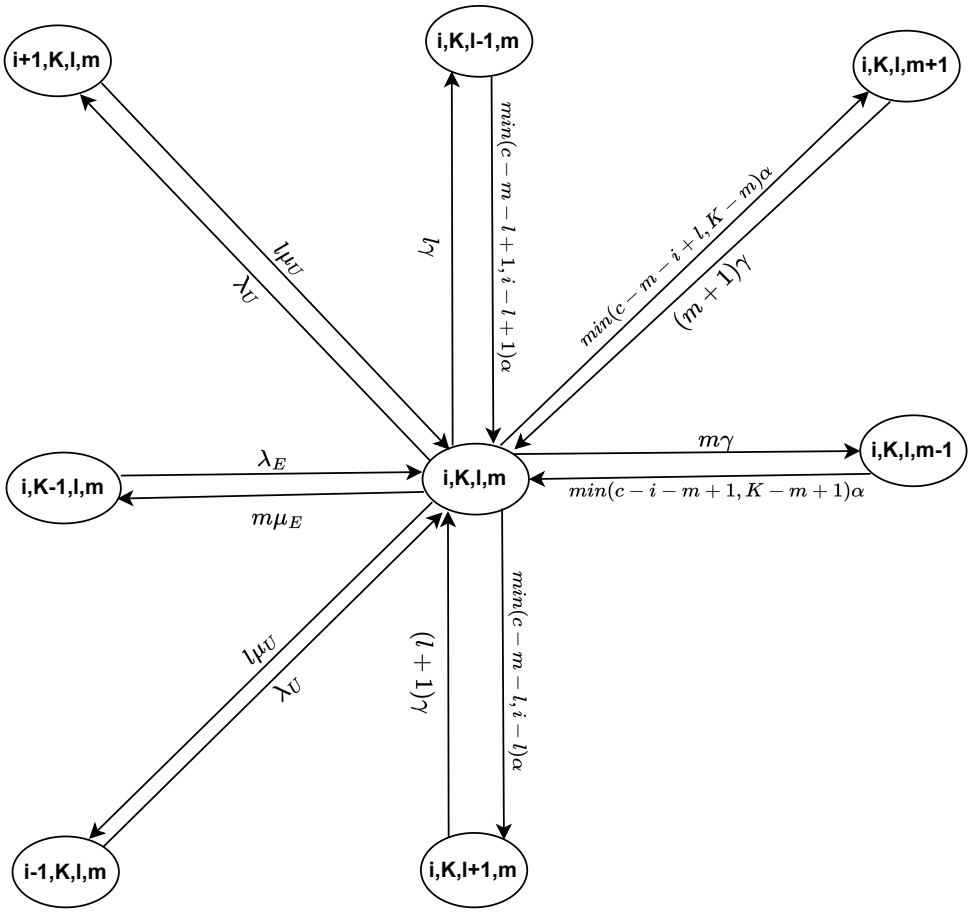
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + m(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c-m-l, i-l)\alpha + min(c-m-i, K-m)\alpha]\pi(i, K, l, m) \\
&= \lambda_U \pi(i-1, K, l, m) + \lambda_E \pi(i, K-1, l, m) + l\mu_U \pi(i+1, K, l, m) \\
&+(l+1)\gamma\pi(i, K, l+1, m) + (m+1)\gamma\pi(i, K, l, m+1) + min(c-m-l+1, i-l+1)\alpha\pi(i, K, l-1, m) \\
&+ min(c-i-m+1, K-m+1)\alpha\pi(i, K, l, m-1) \quad (4.7.19)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all URLLC and some eMBB being served and available containers to be activated $(c > m + l)$. Eq. 4.7.20 describes these states (i, j, i, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c > m + l)$. The state diagram and related events that denote these states are shown in Fig. 82 and Table 78.

Figure 81 – States (i, K, l, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i > l$, $(c > m + l)$ and $(c - m - l > i - l)$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + m(\mu_E + \gamma) + i(\mu_U + \gamma) + min(c - m - i, j - m)\alpha]\pi(i, j, i, m) \\
&= \lambda_E(i, j-1, i, m) + i\mu_U\pi(i+1, j, i, m) + (i+1)\mu_U\pi(i+1, j, i+1, m) + m\mu_E\pi(i, j+1, i, m) \\
&+ min(c - m - i + 1, 1)\alpha\pi(i, j, i-1, m) + min(c - m - i + 1, j - m + 1)\alpha\pi(i, j, i, m-1) \\
&+ (m+1)\gamma\pi(i, j, i, m+1) \quad (4.7.20)
\end{aligned}
$$

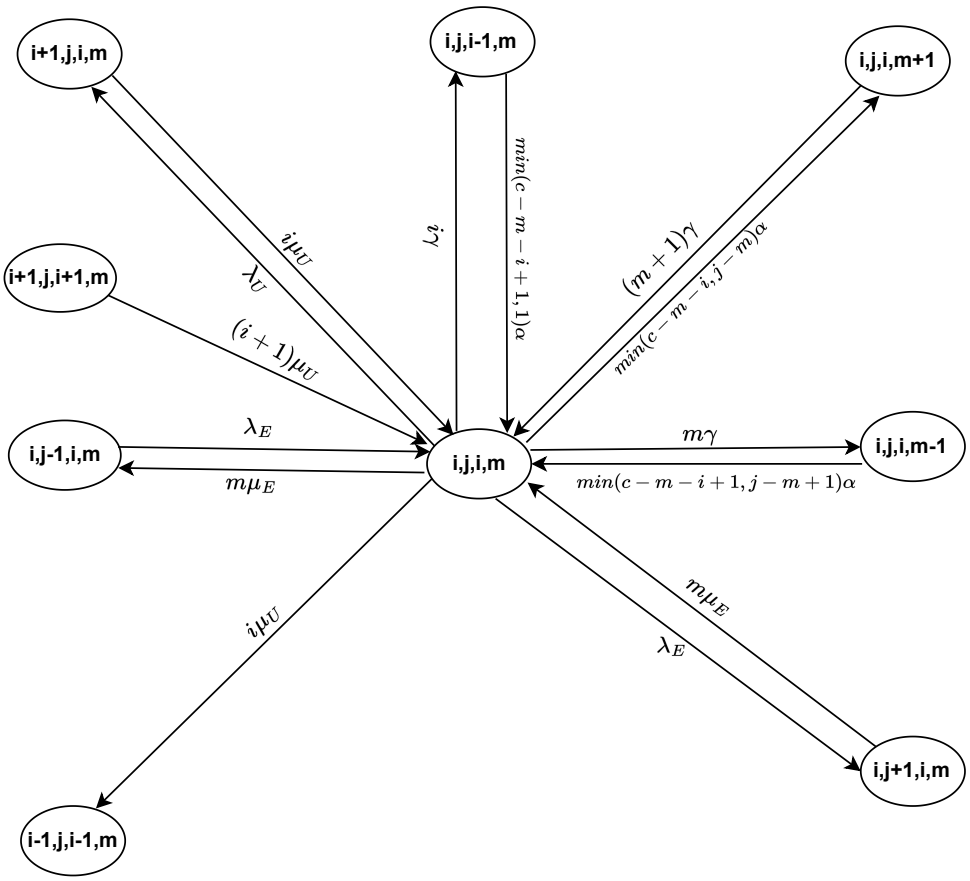
States in which the limit for eMBB users has been achieved and the number of URLLC users is lower than their limit, with some eMBB and all URLLC being served, available containers to be activated $(c > m + l)$. In summary, states (i, K, i, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c > m + l)$. The balance equation, state diagram, and related events are given by Eq. 4.7.21, Fig. 83, and Table 79, respectively.

Table 78 – Events related to the states $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 82 – States (i, j, i, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c > m + l)$



Source: The author (2023)

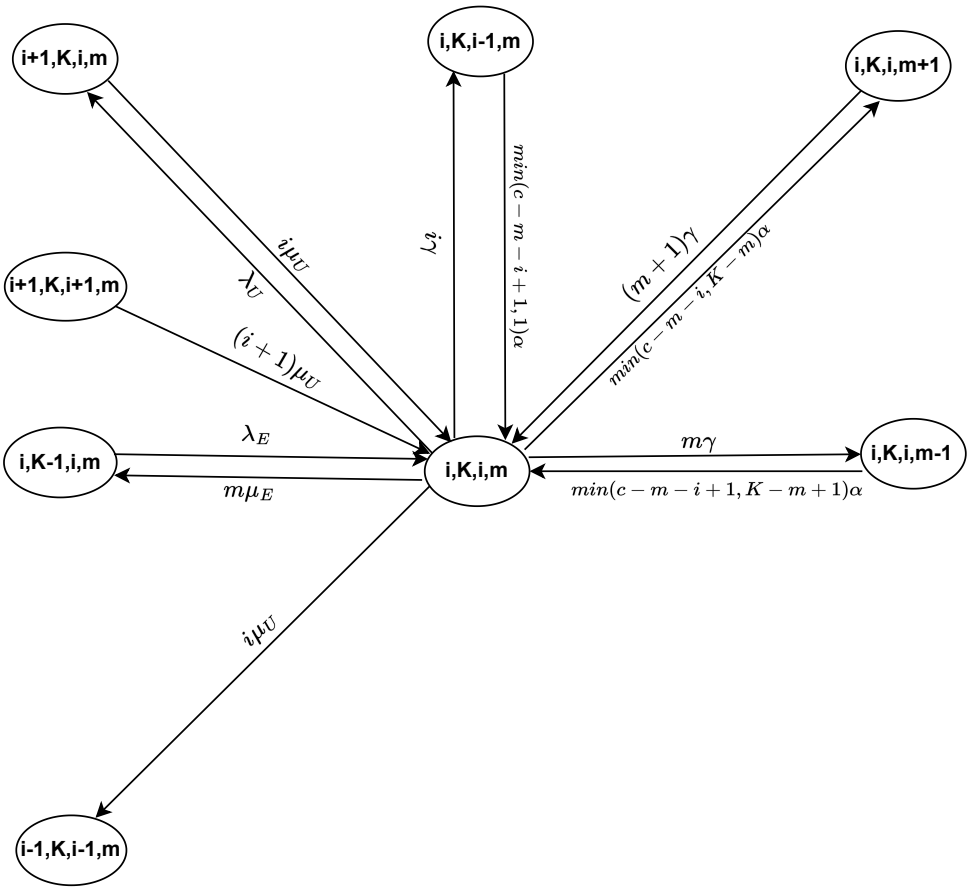
$$[\lambda_U + m(\mu_E + \gamma) + i(\mu_U + \gamma) + min(c - m - i, K - m)\alpha]\pi(i, K, i, m)$$
$$= \lambda_E(i, K-1, i, m) + i\mu_U\pi(i+1, K, i, m) + (i+1)\mu_U\pi(i+1, K, i+1, m) + min(c-m-i+1, 1)$$
$$\alpha\pi(i, K, i-1, m) + min(c-m-i+1, K-m+1)\alpha\pi(i, K, i, m-1) + (m+1)\gamma\pi(i, K, i, m+1) \quad (4.7.21)$$

Table 79 – Events related to the states $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $j > m$, $i = l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✓	✗
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✗	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 83 – States (i, K, i, m), with $0 < i < k$, $j = K$, $0 < l < c$, $0 < m < c$, $i = l$, $j > m$ and $(c > m + l)$



Source: The author (2023)

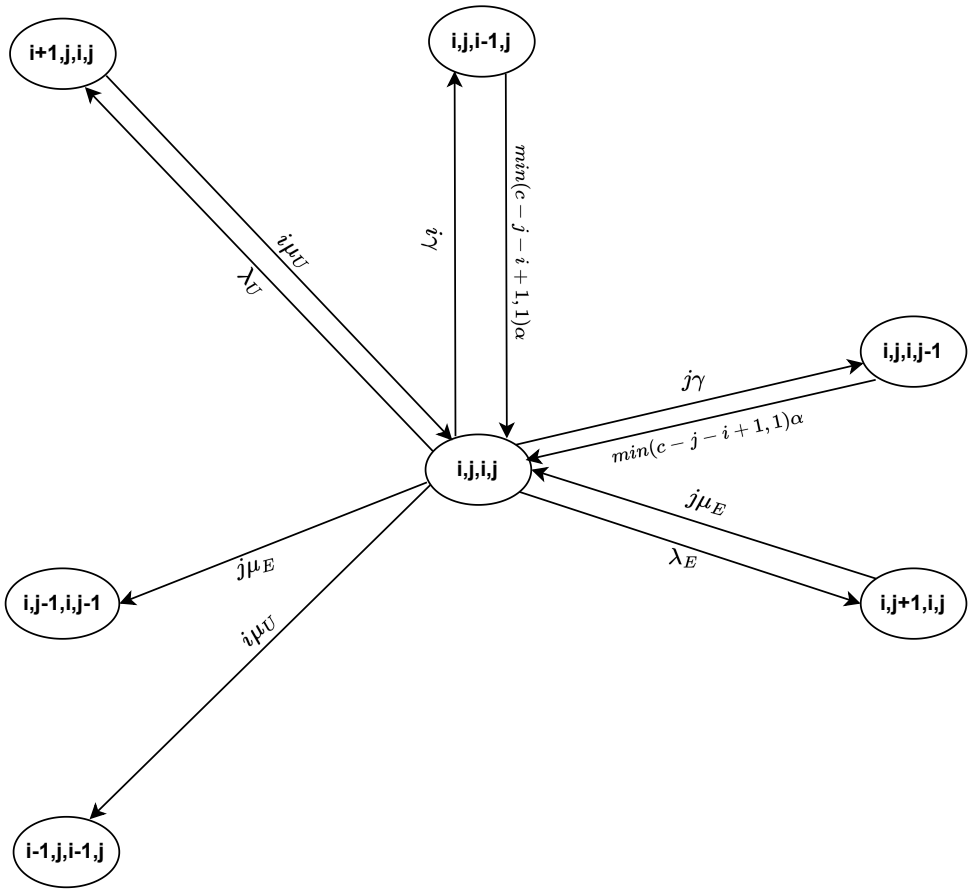
States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and URLLC being served, no available containers to be activated $(c = m + l)$. In summary, state (i, j, i, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $l + m = c$, $j = m$, $i = l$ as in Table 80. The state diagram and balance equation that denote these states are shown in Fig. 84 and Eq. 4.7.22.

Table 80 – Events related to the states $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $l + m = c$, $j = m$, $i = l$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✗	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✗
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✗	✓

Source: The author (2023)

Figure 84 – States (i, j, i, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $l + m = c$, $j = m$, $i = l$



Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + i(\mu_U + \gamma)]\pi(i,j,i,j) \\
&\quad = i\mu_U\pi(i+1,j,i,j) + j\mu_E\pi(i,j+1,i,j) + min(c-ji+1,1)\alpha\pi(i,j,i-1,j) \\
&\qquad + min(c-j-i+1,1)\alpha\pi(i,j,i,j-1) \quad (4.7.22)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with some eMBB and some URLLC being served, available containers to be activated $(c > m + l)$ and the number of available containers to be activated are more than sufficient to process URLLC services in queue $(c - m - l > i - l)$ (see Fig. 85). In summary, states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c > m + l)$. They follow the Eq. 4.7.23 and their related events are listed in Table 81.

Table 81 – Events related to the states (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c > m + l)$.

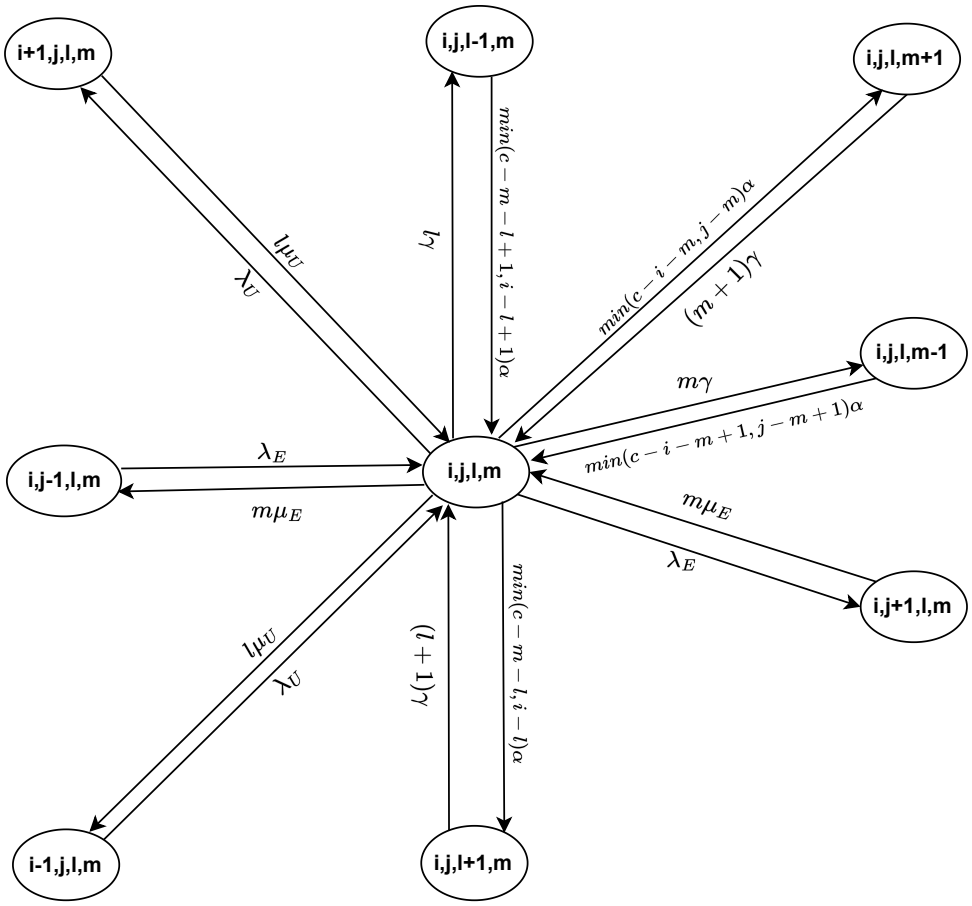
Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✓	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✓
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✓	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

$$
\begin{aligned}
&[\lambda_U+\lambda_E+m(\mu_E+\gamma)+l(\mu_U+\gamma)+min(c-m-l,i-l)\alpha+min(c-i-m,j-m)\alpha]\pi(i,j,l,m) \\
&= \lambda_U\pi(i-1,j,l,m) + \lambda_E\pi(i,j-1,l,m) + m\mu_E\pi(i,j+1,l,m) + l\mu_U\pi(i+1,j,l,m) \\
&+(l+1)\gamma\pi(i,j,l+1,m)+(m+1)\gamma\pi(i,j,l,m+1)+min(c-m-l+1,i-l+1)\alpha\pi(i,j,l-1,m) \\
&+ min(c-i-m+1,j-m+1)\alpha\pi(i,j,l,m-1) \quad (4.7.23)
\end{aligned}
$$

States in which the number of URLLC and eMBB users is lower than their respective limits, with all eMBB and some URLLC being served, available containers to be activated $(c > m+l)$ and the number of available containers to be activated is sufficient to process URLLC services in queue $(c - m - l > i - l)$. Eq. 4.7.24 describes these states (i, j, l, j), with $0 < i < k$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c > m + l)$. The state diagram and related events that denote these states are shown in Fig. 86 and Table 82.

Figure 85 – States (i, j, l, m), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $j > m$, $i > l$ and $(c > m + l)$



Source: The author (2023)

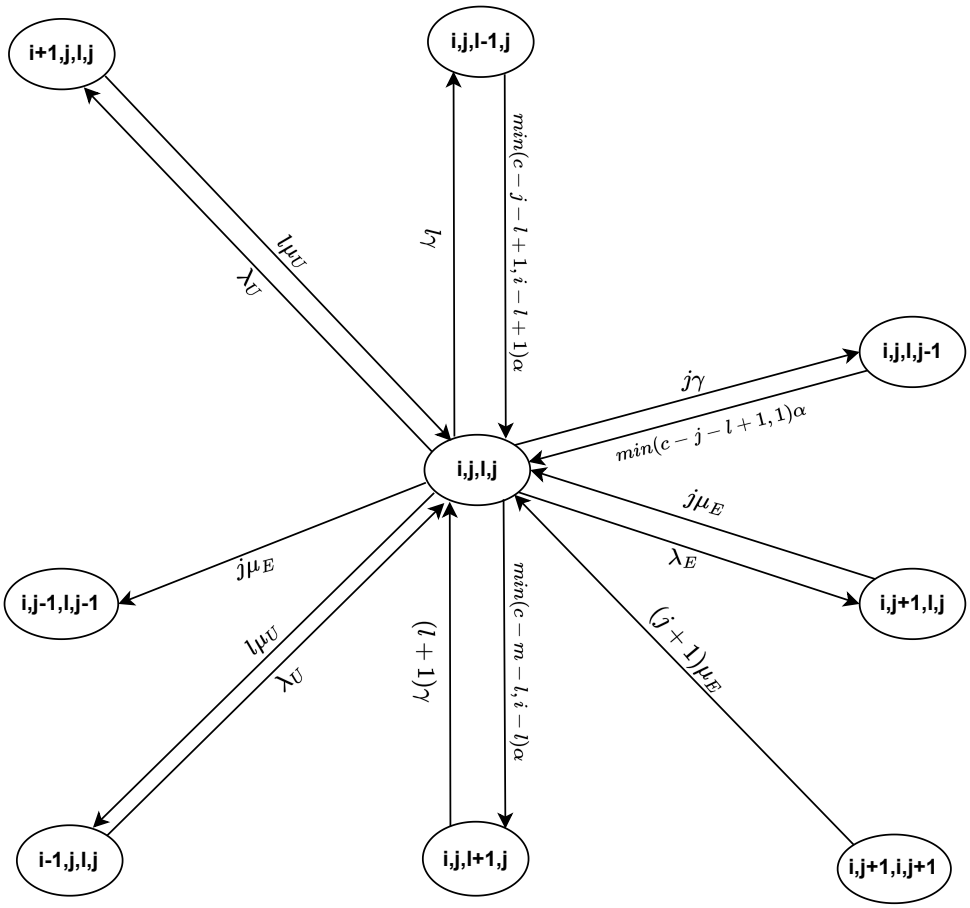
$$
\begin{aligned}
&[\lambda_U + \lambda_E + j(\mu_E + \gamma) + l(\mu_U + \gamma) + min(c - j - l, i - l)\alpha]\pi(i, j, l, j) \\
&= \lambda_U \pi(i1, j, l, j) + l\mu_U \pi(i + 1, j, l, j) + j\mu_E \pi(i, j + 1, l, j) + (j + 1)\mu_E \pi(i, j + 1, l, j + 1) \\
&\quad + (l + 1)\gamma\pi(i, j, l + 1, j) + min(c - j - l + 1, i - l + 1)\alpha\pi(i, j, l - 1, j) \\
&\quad + min(c - l - j + 1, 1)\alpha\pi(i, j, l, j - 1) \quad (4.7.24)
\end{aligned}
$$

Table 82 – Events related to the states $0 < i < k$, $0 < j < c$, $0 < l < c$, $0 < m < c$, $j = m$, $i > l$ and $(c > m + l)$.

Events	Flow Direction	
	IN	OUT
URLLC user arrival	✓	✓
eMBB user arrival	✗	✓
Container initialization for URLLC service	✓	✓
Container initialization for eMBB service	✓	✗
URLLC service conclusion	✓	✓
eMBB service conclusion	✓	✓
Container Failure - eMBB service	✗	✓
Container Failure - URLLC service	✓	✓

Source: The author (2023)

Figure 86 – States (i, j, l, j), with $0 < i < k$, $0 < j < K$, $0 < l < c$, $0 < m < c$, $i > l$, $j = m$ and $(c > m+l)$



Source: The author (2023)

4.8 PERFORMANCE METRICS

In this section, we consider the steady-state analysis of the CTMC under study, followed by the derivation of two performance metrics for each user type (eMBB and URLLC), namely

Availability (A) and Mean Response Time (T) and also the mean Power Consumption for the system (PC).

4.8.1 Availability (A)

The adoption of MEC and NFV environment in proximity to User Equipment (UE) has been widely acknowledged for its potential to reduce latency and enhance reliability. However, the limited resources of edge nodes impose constraints on their service capacity, which is typically known as system availability. Consequently, when the maximum capacity is reached, two primary alternatives emerge: forwarding the flow to a neighboring MEC node or redirecting it to the central cloud [Sarrigiannis et al. 2020]. These alternatives involve establishing a new route comprising multiple intermediate hops, which can introduce significant uncertainty concerning latency and reliability. As a result, it becomes essential to analyze the availability of edge nodes. Nevertheless, its significance is particularly more pronounced in the context of URLLC services compared to eMBB, i.e., while MEC availability remains important for eMBB applications, they primarily focus on delivering high data rates, rather than on the stringent latency and reliability requirements found in the URLLC category.

In our model, the MEC availability refers to the system's ability to offer the minimum amount of functional and accessible VNFs or buffer positions. In addition, due to the service prioritization, the MEC node availability is segmented in terms of each service category, i.e., URLLC (A_U) and eMBB (A_E) respectively, being described in Equations 4.8.1 and 4.8.2, which are obtained by summing the probabilities of all states except those representing full capacity for each type of service.

$$A_U = 1 - \sum_{j=0}^{K} \sum_{l=0}^{c} \sum_{m=0}^{min(c-l,j)} \pi_{k,j,l,m} \tag{4.8.1}$$

$$A_E = 1 - \sum_{i=0}^{k} \sum_{m=0}^{c} \sum_{l=0}^{min(c-m,i)} \pi_{i,K,l,m} \tag{4.8.2}$$

4.8.2 Response Time (T)

Taking into consideration the previous discussion on Availability, response time assumes a crucial role in URLLC applications, while also maintaining relevance for eMBB applications. Recognizing that the significance may vary depending on the service category, we have chosen to analyze them separately, as denoted by Equations 4.8.5 and 4.8.6. We define the Response Time for each category as the interval between the service arrival (on the edge node) and its conclusion, which includes any setup/restart times if these events are triggered. Then, the Response Time is obtained by calculating the mean number of online services in the system for each category as in Equations 4.8.3 and 4.8.4 and by dividing them by the accepted service rate.

$$\overline{U}_U = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} i\pi_{i,j,l,m} \tag{4.8.3}$$

$$\overline{U}_E = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} j\pi_{i,j,l,m} \tag{4.8.4}$$

$$T_U = \frac{\overline{U}_U}{\lambda_U A_U}. \tag{4.8.5}$$

$$T_E = \frac{\overline{U}_E}{\lambda_E A_E}. \tag{4.8.6}$$

4.8.3 Power Consumption (PC)

The computational power consumption is an important component of the operational costs and must be considered by the service provider for resource planning to address cost-performance trade-off. In this model, the mean power consumption ($\overline{PC}$) is formed from the combination of the mean number of virtual resources and energy consumption constants for each operating state: Setup and Busy. The power consumption (in Watts) of a single container in setup state is denoted as P^{CT}_{setup} while in the busy state is P^{CT}_{busy}. It is important to note that this metric is calculated for the combined set of service categories.

The mean number of containers $\overline{CT}$ in each state (Busy and Setup) is described in Eqs. (4.8.7) and (4.8.8) and are detailed in the next few lines. Eq. (4.8.7) captures the mean amount

of containers in the busy state by iterating over each system state service load and varying the combination of the number of each container type from 0 until the number of services from a particular category or the maximum resources available in the system. Moreover, Eq. (4.8.8) calculates the mean number of containerized VNFs in setup by iterating over states where the number of online services is greater than the total number of active resources for each service category. Finally, the total mean power consumption ($\overline{PC}$) is given by Eq. (4.8.9).

$$\overline{CT}_{busy} = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} (l+m)\pi_{i,j,l,m} \tag{4.8.7}$$

$$\overline{CT}_{setup} = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} min((c-l-m),(i+j-l-m))\pi_{i,j,l,m} \tag{4.8.8}$$

$$\overline{PC} = P_{setup}^{CT} \overline{CT}_{setup} + P_{busy}^{CT} \ \overline{CT}_{busy} \tag{4.8.9}$$

4.9 CHAPTER SUMMARY

This chapter discussed a CTMC-based analytical representation for a single node NFV-MEC, considering containers for processing URLLC and eMBB requests, along with events like container setup, failures, and repair times. The chapter was structured to detail the mathematical model and its conditions for different system states, facilitating comprehension. The chapter defined system states, equations for availability, response time, and power consumption, which will be used in Chapter 5 to analyze the system performance in the context of URLLC and eMBB applications.

5 MODEL VALIDATION AND RESULT ANALYSIS

The analytical results were validated against extensive discrete-event simulations (Figs. 87-111), where the lines denote the analytical and the markers represent simulation results. With regards to the main parameters, we have followed a subset of the 3GPP Release 16 (TR 38.824) [3GPP 2020]. With the exception of the first scenario (Section 5.1), which evaluates the impact of each user type on each other by adopting multiple eMBB request rates (λ_E), each subsequent scenario simultaneously assesses the influence of a pair of parameters: (Section 5.2) container setup rates (α) and failure rates (γ), which aims to demonstrate the impact of hardware and software improvements in order to reduce the time in which network functions are made available to attend to system services, and at the same time to highlight the impacts of using less reliable components to provide the service; (Section 5.3) URLLC service rate (μ_U) and eMBB service rate (μ_E), with the objective to illustrate how enhancements in service request process speed, achieved through the utilization of advanced processing units and optimized algorithms, can positively impact the system's overall functionality; (Section 5.4) total number of containers (C) and the buffer size for eMBB users (K), which demonstrates how augmenting the parallel processing capacity of the system affects both its cost and the quality of service. Concomitantly, it also considers the implications of increasing the system's capacity to admit a higher number of eMBB services; and (Section 5.5) total number of containers (C) and the buffer size for URLLC users (k). This section shares a similar objective to the previous one but focuses on the impact of expanding the system's capacity to accommodate URLLC service requests. In all scenarios, the URLLC service arrivals (λ_U) ranged from 2.5 to 25 requests/ms in order to analyze the system performance under different URLLC loads. Unless stated otherwise, the baseline values for failure (γ) and setup rates (α) were set to 0.001 and 1 unit/ms, respectively, in accordance with [Kaur et al. 2017]. For the power consumption of each container in different operation states, we adopted the values from the network-intensive experiment in [Morabito 2015], which are summarized in Table 83. The remaining parameters can be found in Table 84.

A simulation model was adopted to validate the analytical one. In this kind of simulator, the interactions between requests and attendance are implemented so that the same performance metrics can be obtained and compared to those from the analytical model. A discrete-event simulation replicates the dynamics of a multifaceted network system, predicated upon an

Table 83 – Power Consumption Values

Parameter	Value
Idle Container Energy Consumption (P_{idle}^{CT})	0 W
Setup Container Energy Consumption (P_{setup}^{CT})	8 W
Busy Container Energy Consumption (P_{busy}^{CT})	23 W

Source: Morabito (2015)

Table 84 – Experiment Sets

Section	Varying Parameters	λ_E	α	γ	μ_U	μ_E	C	K	k
5.1	λ_E	5,10,15,20,25,30	1	10^{-3}	2	2	10	20	20
5.2	α, γ	10	1,2,4	10^{-2}, 10^{-3}	2	2	10	20	20
5.3	μ_U, μ_E	10	1	10^{-3}	1,2,4	1,2	10	20	20
5.4	C, K	10	1	10^{-3}	2	2	4,8,12	16,24	20
5.5	C, k	10	1	10^{-3}	2	2	4,8,12	20	16,24

Source: The author (2023)

ordered sequence of discrete events, with each event happening at a particular time instant, which may cause a state change [Tako and Robinson 2009]. Furthermore, the performance metrics are not produced via analytical inference from probability distributions. Instead, they are calculated as arithmetic means from diverse simulation runs.

The adopted discrete event simulator is based on colored Petri nets and was developed using the simulation mode of the CPN tools tool [Ratzer et al. 2003]. The simulator implements the functionalities of request arrival, system access control, service queueing, service prioritization, failure during service processing, and automatic container scaling for both service types. The occurrence times of events are defined during the simulation and can follow probability distributions. The default time scale used in the simulator is one microsecond, but other scales can be employed.

The simulator comprises three modules: Service Arrival, Container Management, and Service Attendance and Service Failure. User admission facilitates immediate container initialization for service provisioning, contingent on available resources. Container initiation follows a given pattern (e.g. exponentially distributed), and initialized containers enter a buffer for prompt response to requests. Failures occur if a generated failure time is shorter than a service's completion time, leading to task re-queuing and container reset. When service completion time is shorter, the container returns to the buffer for further tasks or potential shutdown, following system-defined priorities.

Next sections (5.1 - 5.5) display average results in which the analytical results were plotted

in lines and the simulations[1] were plotted in symbols. For every point calculated using the analytical model, 10 simulation instances, comprising 27000000 simulation steps and 2200000 services attended each, were conducted. The Bootstrap method [Singh and Xie 2008] was employed, with both resample size and the number of (re)samplings set at 30 and 1000, respectively. This was done considering a 95% confidence level. Bars were omitted due to the negligible difference between upper and lower bounds and to prevent overcrowding of the graphs.

5.1 EFFECTS OF VARYING THE EMBB ARRIVAL RATE (λ_E)

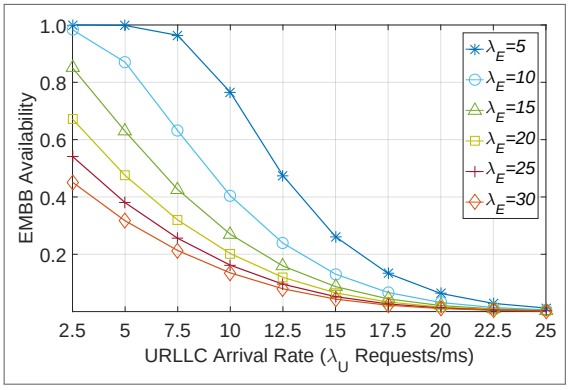
This scenario evaluates the impacts of varying the eMBB service request arrival rate, from 5 up to 30 arrivals/ms, resulting in six curves. These represent different eMBB loads, where the blue curves (light and dark) correspond to small loads (5 and 10, respectively), green and yellow to medium loads (15 and 20, respectively), and red and orange to higher loads (25 and 30, respectively).

Regarding the Availability of both Enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low Latency Communications (URLLC) users, Figs. 87-88 depict strictly decreasing curves. Notably, the Availability for eMBB users (Fig. 87) displays a greater disparity among the configurations, whereas the results for URLLC users (Fig. 88) exhibit overlapping patterns. This observation aligns with expectations, given that the URLLC service category is accorded higher priority over eMBB, rendering the eMBB arrival rate (λ_E) inconsequential for URLLC Availability. Conversely, in Fig. 87, eMBB users contend for unoccupied containers, i.e., those not utilized by either eMBB or URLLC users. As the curves represent varying eMBB user loads, the overall eMBB Availability fluctuates, with higher values corresponding to curves indicating lower eMBB arrival rates (e.g., $\lambda_E = 5$ and $\lambda_E = 10$). Consequently, the curves in Fig. 87 exhibit a more pronounced decline compared to those in Fig. 88, as the former is influenced by both eMBB and URLLC arrival rates while the latter is solely influenced by the URLLC arrival rate. Moreover, it is noteworthy that the eMBB user Availability (Fig. 87) converges to zero at $\lambda_U = 22.5$, whereas the URLLC Availability (Fig. 88) remains above 80% at the same point. This finding appears reasonable for the majority of future service categories, but it is considered suboptimal for URLLC applications.

Regarding the analysis of the Response Time (Figs. 89-90), significant disparities can be

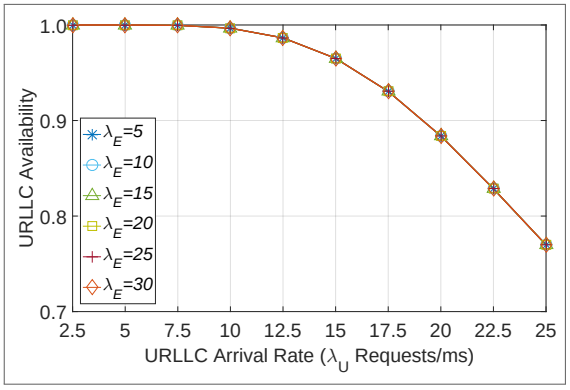
[1] The simulator is available in the repository https://github.com/CaioWalker/URLLC-eMBB-MEC-Simulator

Figure 87 – Availability eMBB



Source: The author (2023)

Figure 88 – Availability URLLC

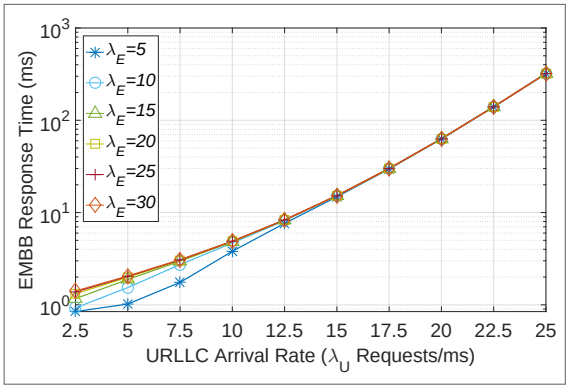


Source: The author (2023)

observed, starting with the employed scale. In Fig. 89, the Response Time for eMBB users exhibits a wide range of values spanning from 1 ms up to 300 ms. In contrast, Fig. 90 depicts a considerably narrower range, with the Response Time for URLLC users ranging from 0.8 ms to 0.94 ms, these values indicate that across all load scenarios assessed in this system configuration, the latency requirements for delivering all URLLC services listed in Table 1 are consistently met. Despite these distinctions, the curves in both figures exhibit substantial overlap across the majority of the evaluated points, ultimately converging to the same final value. However, the key distinction lies in their respective behaviors. In Fig. 89, the curves demonstrate a monotonically increasing trend, while Fig. 90 displays a sudden drop in the Response Time for URLLC users until $\lambda_U = 10$. Beyond this, all curves resume an upward trajectory, converging to 0.89 ms at $\lambda_U = 25$, which is lower than the initial value of approximately 0.94 ms at $\lambda_U = 2.5$. This unexpected behavior can be attributed to the container setup delay, during which requests await the completion of container loading. Consequently, all curves experience a decrease in Response Time from $\lambda_U = 2.5$ to $\lambda_U = 10$, followed by a steady increase. However, the Response Time values do not reach the same levels as at $\lambda_U = 2.5$, as all containers have already been initialized. Additionally, in Fig. 90, slight variations in the results are observed between $\lambda_U = 2.5$ and $\lambda_U = 7.5$, attributed to the presence of eMBB users. These users also contribute to the (re)initialization of containers when an eMBB request is completed and immediately followed by a URLLC request, triggering a new container initialization process, which explains the small differences among the curves in this interval.

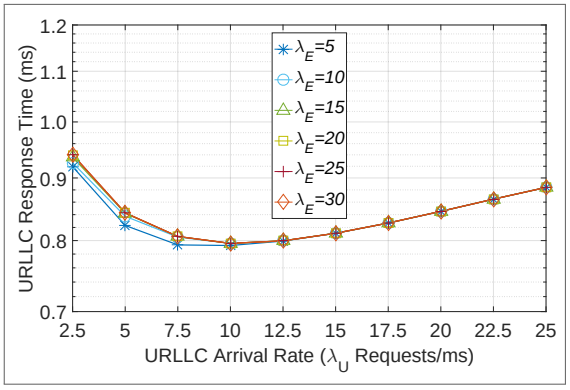
The last performance metric for this scenario is the Energy Consumption (Fig. 91), which exhibited two different behaviors from $\lambda_U = 2.5$ to $\lambda_U = 10$: an increasing trend for part of

Figure 89 – Response Time eMBB



Source: The author (2023)

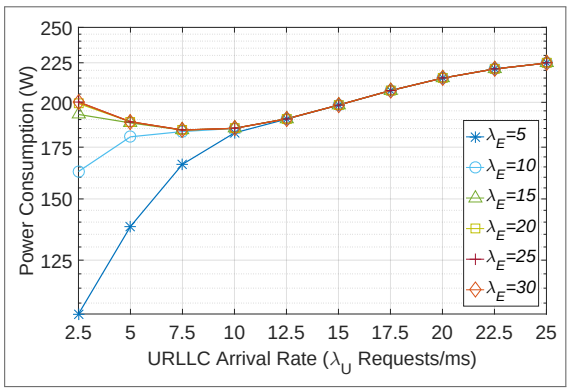
Figure 90 – Response Time URLLC



Source: The author (2023)

the configurations ($\lambda_E = 5$ and $\lambda_E = 10$) and a decreasing trend for the remaining curves. This is due to the summation of the arrival rates of both user types, i.e., when the sum of the arrival rates is lower than the total processing capacity of the system's containers, the curves tend to increase since the idle containers are being activated to meet newly arrived requests. Conversely, when these rates exceed the processing capacity of the system, a slight decreasing trend can be observed in the curves. This is attributed to the re-initialization of containers to prioritize URLLC requests. During container re-initialization, the containers spend more time in setup mode, which uses less energy compared to a processing state, thus resulting in lower energy consumption. The curves tend to converge as the arrival rate of URLLC requests increases, causing fewer eMBB requests to be served and subsequently reducing the number of container re-initializations for different service types. As the containers are no longer being reinitialized, they spend more time in the processing state, leading to a new increase in overall energy consumption.

Figure 91 – Power Consumption



Source: The author (2023)

5.2 EFFECTS OF VARYING THE CONTAINER SETUP RATE (α) AND SERVICE FAILURE RATE (γ)

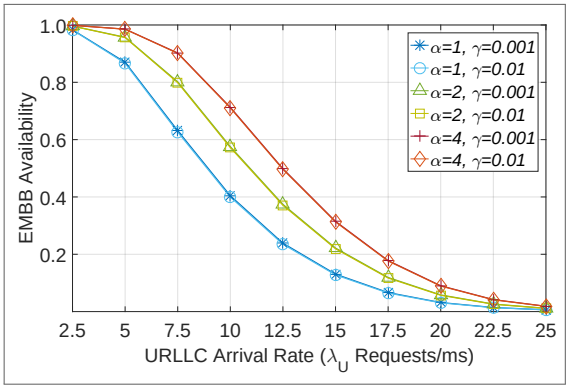
In this scenario, it is conducted an investigation on the impact of varying the container setup rate (α) in combination with changes in the service failure rate (γ) in the context of future mobile communications, particularly for URLLC and eMBB applications.

The availability of eMBB services, as shown in Figure 92, exhibited significant variations among the curves with different setup rates ($\alpha = 1$, $\alpha = 2$, and $\alpha = 4$), while overlapping with configurations having the same setup rate but different failure rates. Notably, the absolute differences in availability reached up to 30% for $\lambda_U = 10$ when comparing the $\alpha = 1$ (light and dark blue) and $\alpha = 4$ (red and orange) configurations. Higher container setup rates were observed to result in increased availability and reduced user waiting times in the buffer. Interestingly, the experiment revealed that even when the service failure rate was increased by a factor of ten, it did not significantly impact the system availability for eMBB users, which can be attributed to the buffer's capacity to accommodate failed service requests. Moreover, consistent with the previous scenario, the availability for eMBB applications diminished rapidly across all tested configurations, in contrast to the URLLC availability shown in Figure 93, which experienced a comparatively smaller impact due to its higher priority.

Regarding the availability for URLLC users (Figure 93), it was observed that the container setup rate (α) had a relatively minor impact on the availability curves compared to the eMBB case. Specifically, the differences in availability among the curves with different α values were limited to approximately 2% at $\lambda_U = 15$, when comparing the $\alpha = 1$ (light and dark blue) and $\alpha = 4$ (red and orange) configurations. As for the impact of different failure rates, a more pronounced difference was noted when compared to the eMBB case in Figure 92, where overlapping occurred. For the URLLC, container failures produced a slight difference among the curves with the same α, making it possible to distinguish between, for instance, the light and dark blue curves. In other words, the URLLC is significantly more sensitive to the failure rate than the eMBB.

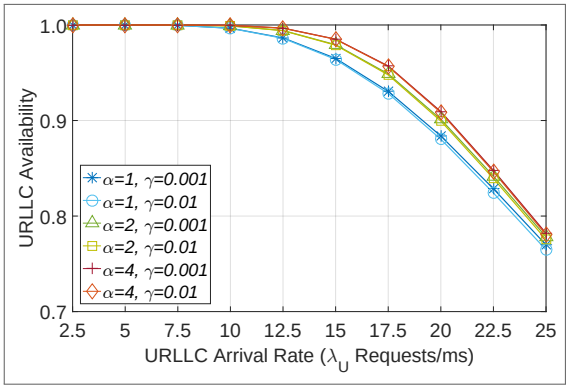
When examining the eMBB Response Time depicted in Fig. 94, it becomes apparent that a higher container setup rate leads to a reduced response time, as expected. Initially, since there is little competition for resources between eMBB and URLLC users, the difference between the evaluated configurations is of a few milliseconds. However, as the URLLC request arrival rate intensifies, this disparity becomes more pronounced. The increasing URLLC arrival

Figure 92 – Availability eMBB



Source: The author (2023)

Figure 93 – Availability URLLC



Source: The author (2023)

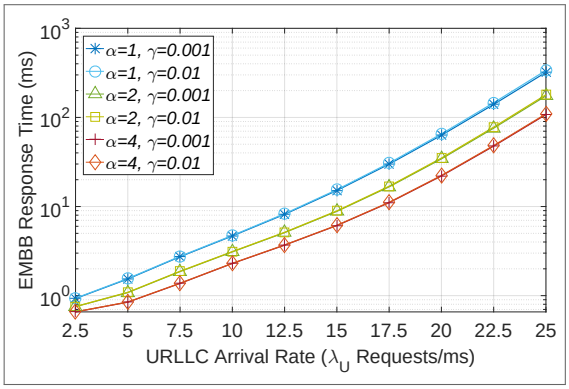
rate creates a higher demand for resources, and since it has a higher priority, the eMBB requests are interrupted, either restarting service in another container or waiting in the buffer for available resources, causing the eMBB response time to be more affected. In such cases, only system configurations with a α of 4 have the capacity to handle high-resolution video streaming services, which demand a latency of under 20 milliseconds [Sugito et al. 2020] when λ_U reaches 20 arrivals per millisecond, a response time about 60% less than the configuration with a α of 1. It was also noticeable that the failure rate had little impact in this experiment, which explains the pair of overlapped curves with the same values of α.

With regards to the Response Time of URLLC users (Fig. 95), the container setup rate has a more pronounced impact compared to the previous scenario in Fig. 89, where the only varying parameter was λ_E. This is particularly evident at the initial stages of the curves when containers are predominantly powered off or allocated to the eMBB users. During this period, the low arrival rate of URLLC services translates to shorter waiting times for a container to become available, reducing the overall response time. However, as the URLLC service arrival rate increases, this disparity diminishes, ultimately converging towards the end of the curves when the majority of containers are occupied by URLLC services.

Furthermore, it is noteworthy that a higher failure rate leads to an increase in the response time, since the failure occurrence becomes more frequent, especially for higher λ_U values, impacting the service time due to the need for container resets. However, similarly to the Availability in Fig. 93, this remains relatively insignificant compared to the differences caused by altering the setup rate. This results in a more distinguishable difference among the pair of curves that were overlapping (e.g., light and dark blue). Finally, as the curves approach the system's capacity, a greater number of containers remain active to accommodate the

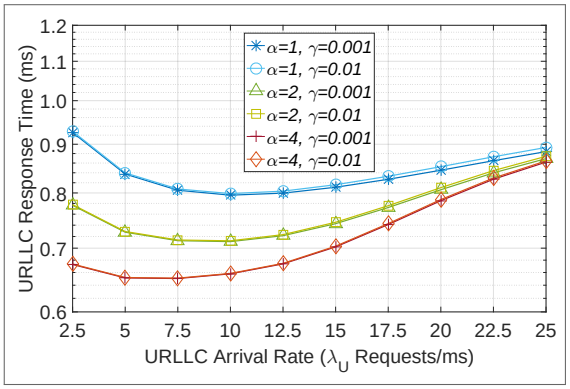
incoming service requests, resulting in a temporary decline in the response time. Nevertheless, as resource competition intensifies within the URLLC service category, the response time gradually escalates once again and all curves tend to converge around 0.9 ms. At this point, all system configurations remain capable of providing service to robotic and telepresence systems, which require a latency of 1 ms [Siddiqui et al. 2023].

Figure 94 – Response Time eMBB



Source: The author (2023)

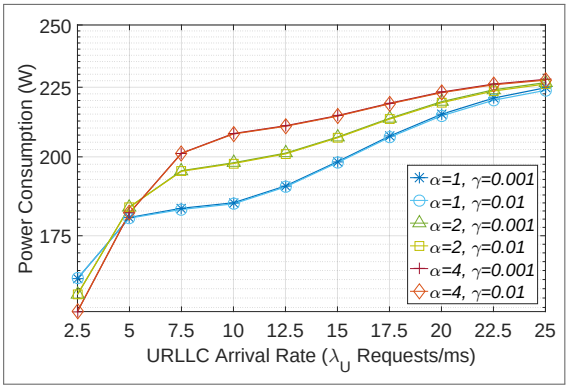
Figure 95 – Response Time URLLC



Source: The author (2023)

Regarding energy consumption (Figure 96), higher container setup rates, such as the green/yellow ($\alpha = 2$) and red/orange ($\alpha = 4$) curves, lead to greater energy consumption. This can be attributed to the fact that with higher setup rates, less time is spent in the setup phase, making containers more frequently available. Since the processing phase requires more power compared to the setup phase, the total energy consumption monotonically increases, converging around $\lambda_U = 25$ to 225 W. In other words, while higher container setup rates enhance both availability (Figures 92-93) and response time (Figures 94-95), they also contribute to higher energy consumption. Additionally, although the impact was small, it is worth noting that curves depicting higher service failure rates exhibit lower energy consumption when comparing the pair of curves with the same α (e.g., light and dark blue lines). This is due to the increased number of container resets for failed requests, leading to a higher proportion of containers in the setup state.

Figure 96 – Power Consumption



Source: The author (2023)

5.3 EFFECTS OF VARYING THE URLLC SERVICE RATE (μ_U) AND THE EMBB SERVICE RATE (μ_E)

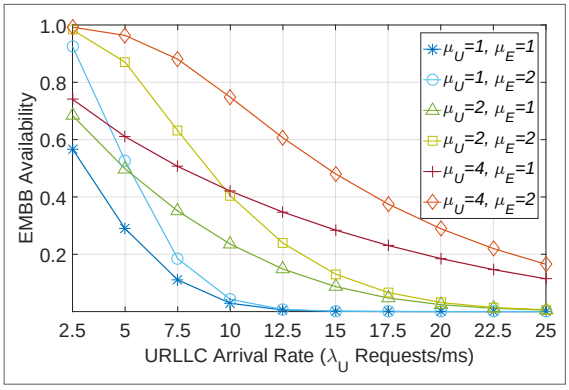
This study aims to assess the influence of different service rates on each user type, specifically the URLLC service rate (μ_U) and the eMBB service rate (μ_E).

Fig. 97 illustrates that a higher eMBB service rate leads to increased availability for this service category, particularly in the leftmost region of the graph. For configurations with the same μ_U values, the curve with $\mu_E = 2$ exhibits higher availability compared to those with $\mu_E = 1$. For example, at $\lambda_U = 7.5$, the configuration with ($\mu_U = 2$, $\mu_E = 1$) demonstrates an availability of 38%, while its counterpart ($\mu_U = 2$, $\mu_E = 2$) exhibits 62%, representing a significant difference of 24%. However, this effect diminishes as the URLLC arrival rate increases, resulting in convergence at the rightmost part of the graph. Moreover, a higher URLLC service rate implies less time spent by these requests monopolizing the resources, leading to greater availability. This explains why configurations with $\mu_U = 1$ and $\mu_U = 4$ are shifted to the left and right, respectively, compared to the adopted baseline ($\mu_U = 2$).

From the perspective of URLLC user availability (Fig. 98), it is observed that the eMBB service rate (μ_E) has an insignificant impact on this performance metric, resulting in overlapping curves. Conversely, higher URLLC service rates ($\mu_U = 2$ and $\mu_U = 4$) lead to greater availability as the requests are serviced more rapidly. For instance, at $\lambda_U = 20$, configurations with $\mu_U = 1$ (light and dark blue) exhibit an availability of approximately 50%, while those with $\mu_U = 2$ (green and yellow) achieve around 88%, i.e., a substantial difference of 48%.

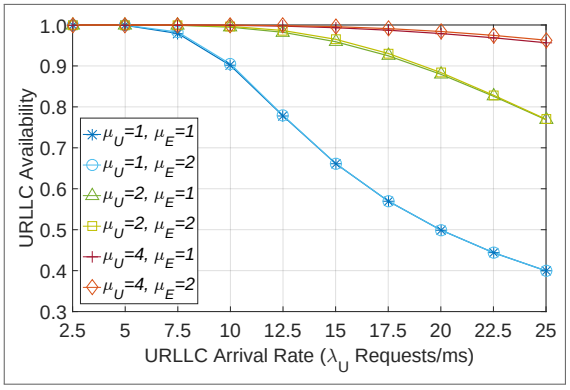
Regarding the eMBB response time in Fig. 99, the experiment demonstrates that a higher service rate for this category, represented by configurations where $\mu_E = 2$ (light blue, yellow,

Figure 97 – Availability eMBB



Source: The author (2023)

Figure 98 – Availability URLLC



Source: The author (2023)

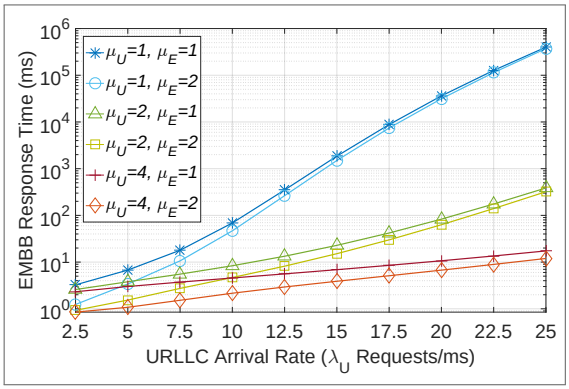
and orange lines), results in shorter response times compared to their respective counterparts with $\mu_E = 1$ (dark blue, green, and red lines). However, the performance difference between the two curves with $\mu_U = 1$ (light and dark blue) and the two curves with $\mu_U = 2$ (green and yellow) is minimal. Notably, the performance difference becomes more pronounced for configurations with $\mu_U = 4$ (red and orange lines). These configurations consistently maintain the eMBB response time below 100 ms throughout the experiment, a threshold considered crucial for multiple eMBB applications such as the FWA service.

Fig. 100 further reveals that a higher service rate for eMBB users, represented by configurations with $\mu_E = 2$ (light blue, yellow, and orange lines), also leads to shorter URLLC response times compared to their respective counterparts with $\mu_E = 1$ (dark blue, green, and red lines). This is attributed to eMBB requests spending less time occupying containers, which are then reinitialized to handle incoming URLLC requests. However, in most cases, this difference is below 0.1 ms and may not be significant even for URLLC applications. Conversely, an increase in the URLLC service rate ($\mu_U = 1$, $\mu_U = 2$, and $\mu_U = 4$) results in shorter response times for this service category, with a more substantial impact. For example, at $\lambda_U = 10$, the orange curve ($\mu_U = 4$, $\mu_E = 2$) shows a response time of approximately 0.5 ms, whereas the yellow curve ($\mu_U = 2$, $\mu_E = 2$) exhibits 0.8 ms. This 0.3 ms difference is significant for URLLC applications, as some require a response time of 1.2 ms or less, while others, such as Robotics and Telepresence, demand at most only 1 ms [Siddiqui et al. 2023].

In configurations where $\mu_U = 1$ (light and dark blue lines), an interesting behavior is observed in Fig. 100. As the URLLC request arrival rate approaches the system's processing capacity, a decrease in the response time for this service category is observed. This is attributed to URLLC containers spending more time active and less time in the setup state, thereby

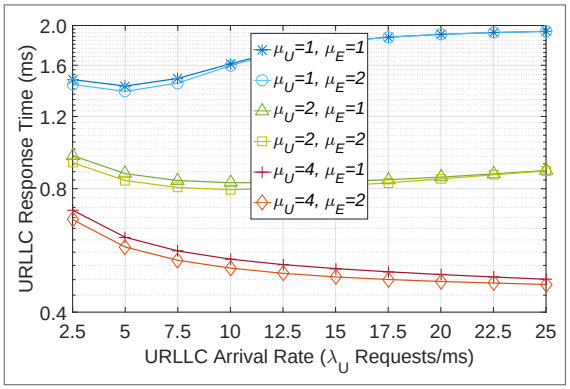
reducing the impact of this component. However, shortly thereafter, there is an increase in the response time due to competition for processing resources within the same service category, resulting from a larger number of URLLC requests waiting in the buffer. This behavior is also present in configurations with $\mu_U = 2$ and $\mu_U = 4$, but for larger $\lambda_U > 25$ values, which are not represented in this figure.

Figure 99 – Response Time eMBB



Source: The author (2023)

Figure 100 – Response Time URLLC

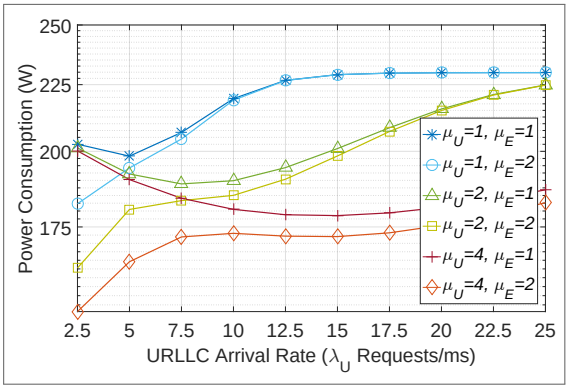


Source: The author (2023)

In terms of energy consumption (Fig. 101), once again, a higher service rate for eMBB users leads to lower energy consumption, especially in the leftmost region of the figure, corresponding to low URLLC loads, i.e., when the system is predominantly occupied by eMBB requests. This observation aligns with our earlier analysis on availability (Figs. 97-98), where configurations with $\mu_E = 2$ (light blue, yellow, and orange lines) outperform their respective counterparts with $\mu_E = 1$ (dark blue, green, and orange lines). In other words, higher availability corresponds to lower energy consumption. Consequently, the configuration order is inverted in Figure 101, with the red and orange lines representing the most energy-efficient configurations.

Furthermore, when considering the three different configurations with $\mu_U = 1$, $\mu_U = 2$, and $\mu_U = 4$, significant differences of up to 40 W were observed. For instance, at $\lambda_U = 10$, the configuration with $\mu_U = 4$ and $\mu_E = 2$ (orange line) exhibits a consumption of approximately 175 W, while the configuration with $\mu_U = 2$ and $\mu_E = 2$ (yellow line) consumes around 215 W. This finding is particularly relevant as the experiment maintained the same amount of resources (containers) for all curves, varying only the service rates. In subsequent experiments, different resource and buffer amounts will be analyzed.

Figure 101 – Power Consumption



Source: The author (2023)

5.4 EFFECTS OF VARYING THE NUMBER OF CONTAINERS (C) AND EMBB BUFFER SIZE (K)

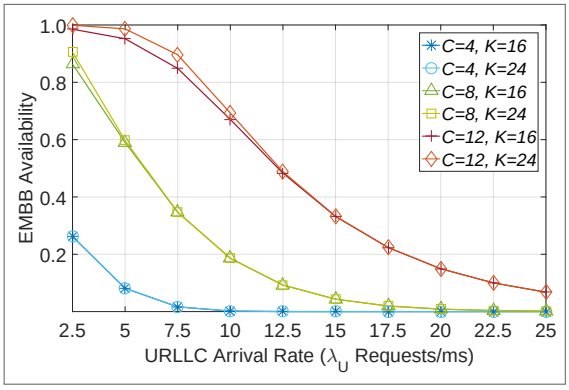
This scenario evaluates the impact of varying the number of containers (C) concomitantly with the the buffer size for eMBB users (K).

In both Figs. 102-103, it is noticeable that the number of containers has a significant impact on the availability for both user service classes, showing higher availability for environments with a greater number of containers, represented by the configurations where $C = 12$ (red and orange lines), followed by $C = 8$ (green and yellow). For instance, in Fig. 102 at $\lambda_U = 10$, the availability for the configurations with $C = 8$ is approximately 20% whereas for the configurations with $C = 12$ is around 69%, i.e., a gap of almost 49%. On the other hand, the tested buffer alternatives had little impact on the eMBB availability, indicating that it would require much larger values than the adopted ones ($K = 16$ and $K = 24$). However, this is not feasible since the buffer will also impact the response time, which will be further evaluated.

As for the system's URLLC availability (Fig. 103), the analysis follows the same pattern for the eMBB, i.e., the container number drastically impacts the availability whereas the eMBB buffer sizes had barely no effect, resulting in overlapping pair of curves: light/dark blue, green/yellow, and red/orange.

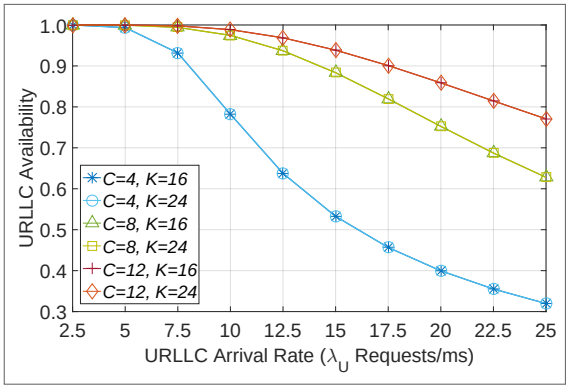
In Fig. 104, a larger buffer size for the eMBB service category also results in an increase in the proportional response time. This is due to the number of service requests ahead of each newly admitted eMBB request, which has to wait in queue. On the other hand, a greater number of available containers also implies a shorter queue time, reducing the contribution of this component to the response time. Once again, it can be observed that undersizing the

Figure 102 – Availability eMBB



Source: The author (2023)

Figure 103 – Availability URLLC



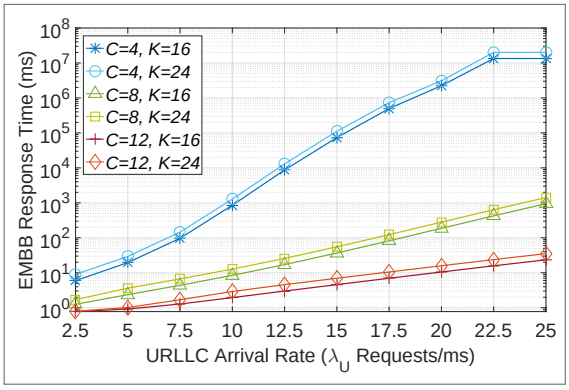
Source: The author (2023)

number of containers can render the service unfeasible for lower-priority users, resulting in large response times, e.g., for configurations where $C = 4$ (light and dark blue lines), these particular configurations are suited for the Smart Office service, which requires a maximum latency of 10 ms [Stallings 2021], only when $\lambda_U = 2.5$. In contrast, the remaining configurations under evaluation can accommodate this application with a λ_U as high as 12.5.

Similar to the URLLC availability in Fig. 102, varying the eMBB buffer size has also little impact on the URLLC response time in Fig. 105. In other words, the response time is solely impacted by the variation in the number of containers. Only system configurations with $C = 8$ and $C = 12$ are capable of serving Robotics services, because even for $\lambda_U = 2.5$, which is the smallest evaluated in the experiment, configurations with $C = 4$ presented a response time greater than 1 ms. Despite this, the configurations with $C = 4$ presented a response time of less than 2 ms for all evaluated λ_U, proving to be capable of serving the Smart Transportation Systems service that allows latencies between 10 and 100 ms [Siddiqui et al. 2023]. A particularity can be found on the leftmost part of this figure, where the curves with $C = 8$ (green and yellow lines) and $C = 12$ (red and orange lines) first decrease the response time, and, in the case of $C = 8$ it rises again, reaching the same initial value at $\lambda_U = 25$. This is likely due to the container setup time, which is either serving eMBB requests or powered off, considering the low URLLC demand from $\lambda_U = 2.5$ until $\lambda_U = 10$. On the other hand, as the URLLC arrival rate increases, a decrease in the response time of service requests can be observed. This occurs because more containers are available for service, reducing the waiting time in relation to the container setup delay.

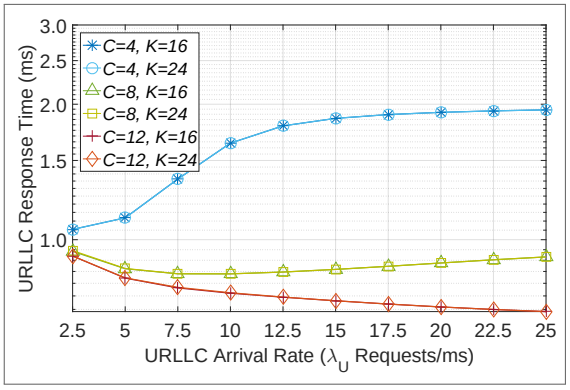
As opposed to the response time, a higher amount of containers inevitably implies a higher energy consumption (Fig. 106). The energy consumption is not exactly proportional to the

Figure 104 – Response Time eMBB



Source: The author (2023)

Figure 105 – Response Time URLLC

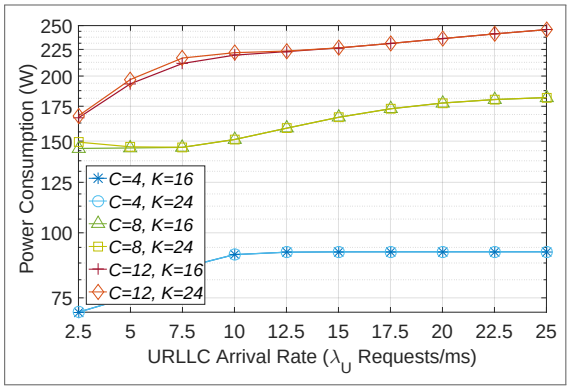


Source: The author (2023)

increase in the number of containers, if we observe the point where the $\lambda_U = 10$, we can observe that between the blue and green curves, the number of containers doubles from 4 to 8, but the same does not occur with the energy consumption that increases by approximately 70%. This occurs because the number of containers being processed also depends on the workload that arrives in the system, that is, the energy consumption would only double together with the number of containers if the demand for system service was sufficient to occupy all the containers available in the two configurations of the system.

However, there is very little difference in the energy consumption comparing each pair of configurations with the same container amounts, i.e., different buffer sizes. A larger eMBB buffer results only in slightly higher energy consumption because more users tend to wait in the queue. This prevents the container from being powered off and restarted, resulting in less time in setup and more time in processing, consuming more energy.

Figure 106 – Power Consumption



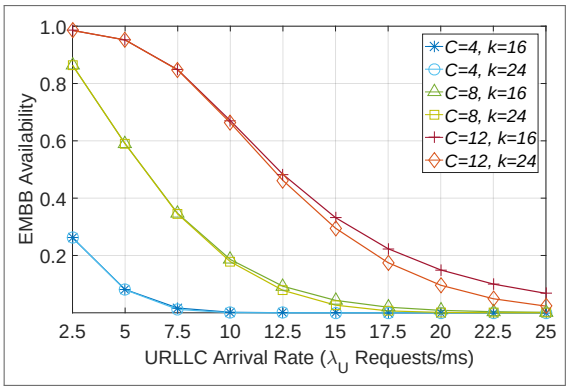
Source: The author (2023)

5.5 EFFECTS OF VARYING THE NUMBER OF CONTAINERS (C) AND THE URLLC BUFFER SIZE (K)

This section aims to evaluate the impact of the number of containers (C) along with the URLLC buffer size (k).

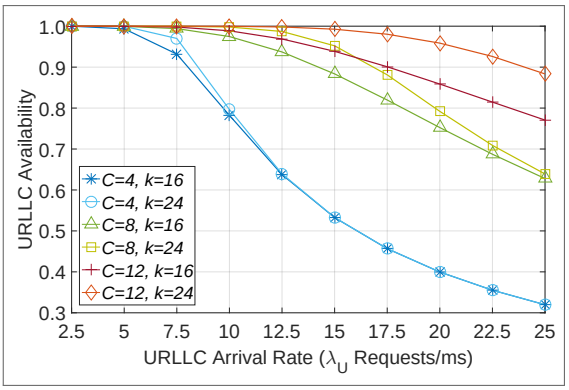
With regards to the Availability (Figs. 107-108) we have very similar observations as those conducted in the previous scenario. However, in Fig. 107 it is noticeable that there is an inversion in the order of the curves (from the most to the least available), which is clearly shown by comparing the curves with $C = 12$ (red and orange). In this case, the red curve, which has fewer URLLC buffer positions ($k = 16$) presents a greater eMBB availability than the orange ($k = 24$). This happens because as more URLLC requests are stored, there is a guarantee that they will be serviced instead of dropped, as it happens with the curve with fewer URLLC buffer positions. Thus, the overall URLLC load increases, pressuring down the eMBB availability. Conversely, in Fig. 108, the orange curve ($k = 24$) displays a greater availability than the red one ($k = 16$), which was expected since the evaluated metric is the URLLC Availability, i.e., a larger URLLC buffer size enhances the URLLC availability, such that when the sum of the arrival rates for both service categories approaches the total processing capacity, a larger buffer size implies greater availability.

Figure 107 – Availability eMBB



Source: The author (2023)

Figure 108 – Availability URLLC



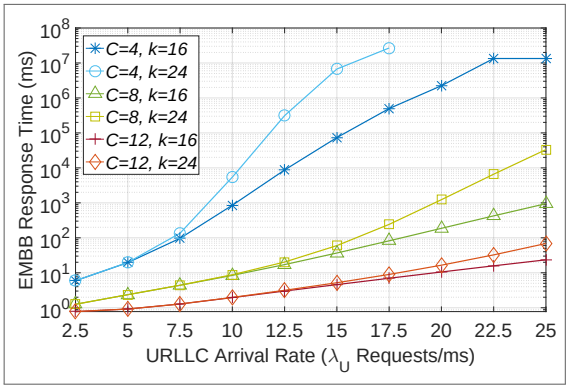
Source: The author (2023)

Regarding the response times depicted in Figs. 109-110, it is evident that the larger URLLC buffer size exerts a significant negative impact on both the eMBB and URLLC response times. However, this impact can be alleviated by increasing the total number of containers, as presented in the curves for both figures. At the leftmost part of Fig. 109 ($\lambda_U = 2.5$), the eMBB response time remains below 10 ms for all tested configurations, albeit with varying growth

rates. For instance, the curves corresponding to $C = 4$ exhibit exponential growth, while those associated with $C = 12$ display linear increase. Consequently, higher container quantities result in improved eMBB response times, particularly under higher URLLC loads approaching system capacity. In this scenario, it becomes evident that system configurations with $C = 8$ and $C = 12$ can effectively fulfill the demands of Virtual and Augmented Reality services, which necessitate a latency of up to 8 milliseconds [Raca et al. 2020], for λ_U as high as 10, whereas configurations with $C = 4$ can only accommodate these services for lambda values equal to 2.5. Conversely, larger URLLC buffers lead to degraded eMBB response times, as evidenced by the curves with $k = 24$ presenting higher response times compared to their respective counterparts with $k = 16$. Notably, a distinct characteristic observed in this experiment is that starting from $\lambda_U = 17.5$ and beyond, the light blue curve (representing $C = 4$ and $k = 24$) assumes unfeasible values (too high magnitude). This occurrence is likely attributed to the intensified pressure from URLLC arrivals coupled with the adoption of large buffer size, resulting in an excessively large eMBB response time.

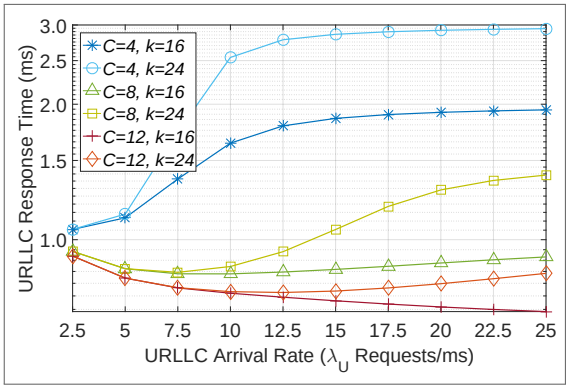
In Fig. 110, the range of possible URLLC response time values is considerably lower than that of the previous experiment, owing to the higher priority accorded to URLLC requests. Nevertheless, there is considerable variation in the behavior of each curve. Some curves exhibit strictly ascending behavior, while others display both descending and ascending phases. Furthermore, one curve exhibits a strictly descending pattern. Nonetheless, the order of curves in terms of URLLC response time remains consistent with the previous experiment (Fig. 109). It is worth noting that, for a larger interval of λ_U, the curves are expected to exhibit similar behavior with minor shifts. Regarding the light and dark blue curves, it can be inferred that the system capacity is swiftly reached, resulting in higher URLLC response times as the buffer becomes more heavily utilized. Nonetheless, even in these cases, the URLLC response time remains at an acceptable level of 3 ms, which is highly suitable for the majority of URLLC applications that typically require response times ranging from up to 10 ms, such as Factory Automation [Siddiqui et al. 2023]. As for the strictly descending curve (in red), it is likely that the URLLC response time decreases because new URLLC arrivals are promptly processed by containers that were previously in the setup mode, thereby bypassing the setup delay. Additionally, the smaller buffer size ($k = 16$) leads to fewer requests in the waiting queue, thereby contributing to a lower overall URLLC response time compared to configurations with larger buffer sizes, such as $k = 24$ (represented by the orange line).

Figure 109 – Response Time eMBB



Source: The author (2023)

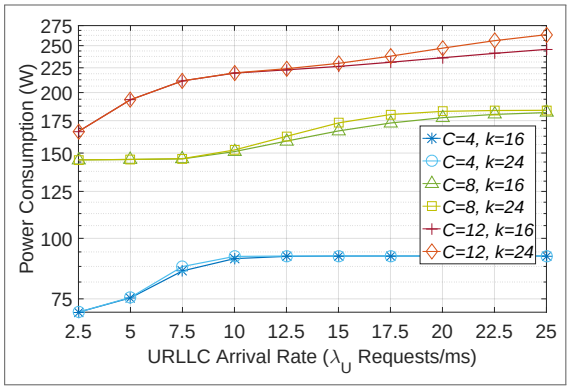
Figure 110 – Response Time URLLC



Source: The author (2023)

Regarding the energy consumption illustrated in Fig. 111 within this scenario, a similar observation can be made compared to the previous experiment. It is evident that an increased number of containers leads to higher energy consumption, aligning with our expectations. Moreover, for the majority of the evaluation frame, the size of the URLLC buffer exhibits minimal influence on this particular performance metric. This is evident from the overlapping pair of curves, particularly noticeable for low URLLC arrival rates. This outcome was anticipated since the buffered requests do not consume resources while in the queue. Thus, in most cases, the buffer size does not significantly impact the energy consumption. However, a slight increase in energy consumption is observed when the system approaches full capacity and utilizes more buffer positions. This phenomenon occurs due to the containers spending a greater amount of time in a processing state, resulting in reduced periods of being powered off or undergoing restart procedures.

Figure 111 – Power Consumption



Source: The author (2023)

5.6 CHAPTER SUMMARY

In this chapter, extensive discrete-event simulations were conducted to compare analytical results with simulation outcomes. The scenarios examined the effects of parameters such as the eMBB arrival rate, container setup rates, failure rates, URLLC and eMBB service rates, number of containers, and buffer sizes for eMBB and URLLC users. The findings revealed that higher eMBB arrival rates led to decreased availability and increased response times, while URLLC availability remained relatively stable. Container setup rates and failure rates impacted both availability and response times, with higher setup rates improving availability and reducing response times. Moreover, the URLLC service rates had a greater influence on URLLC availability, while eMBB service rates affected eMBB availability. In the experiments concerning the total number of containers, we discussed how they had a significant impact on the availability and response times, with more containers improving both metrics. On the other hand, the buffer sizes had a minor effect on availability but influenced response times, with larger eMBB buffer sizes increasing response times and larger URLLC buffer sizes decreasing response times. Lastly, the power consumption increased with the number of containers but was minimally affected by the multiple buffer sizes that were tested. These findings provide valuable insights for optimizing system configurations for future mobile communication networks, which, for instance, can be used to tune the MEC node computational dimensions or to determine to what extent must a given parameter improve in order to successfully allow the coexistance between eMBB and URLLC considering the MEC-NFV architecture, thus offering a systematic framework for mobile operators to establish a reference point for managing operational costs while maintaining a manageable set of performance metrics.

6 CONCLUSION AND FUTURE WORKS

This chapter concludes this dissertation by offering some considerations, showing its main value as a contribution to studies in the field, and proposing future studies.

6.1 FINAL CONSIDERATIONS

This work has addressed the combination of MEC, NFV, and dynamic virtual resource allocation within the context of coexisting 5G service categories: URLLC and eMBB. It designed a model to evaluate how requests are managed by the virtualization resources of a single MEC node, with a primary focus on meeting the requirements of URLLC services. We proposed a CTMC-based model to characterize dynamic virtual resource allocation and incorporated five performance metrics, which are relevant not only for URLLC and eMBB services (e.g., availability and response time) but also for service providers (e.g., power consumption). To make the model more practical, we integrated factors like resource failures, service prioritization, and setup (repair) times into the formulation, since they can incur significant impacts on the 5G applications' requirements. This model enables an understanding of how the 5G network core behaves in serving different service categories by applying service prioritization to efficiently share processing resources. Key findings indicate that higher eMBB arrival rates decrease availability and increase response times, while URLLC availability remains stable. Container setup rates and failure rates substantially affect both availability and response times, with higher setup rates enhancing availability and reducing response times. URLLC service rates primarily influence URLLC availability, whereas eMBB service rates affect eMBB availability. The number of containers emerges as a significant factor, enhancing both availability and response times, while buffer sizes mainly impact response times. Power consumption increases with the number of containers but is minimally affected by variations in buffer size. We anticipate that our work will stimulate further research in the MEC-NFV domain, providing valuable insights for the design of MEC-NFV architecture, business models, and mechanisms to address the resource allocation under different communication constraints.

6.2 CONTRIBUTIONS

The main contributions of this work can be summarized as follows:

- Description and classification on the main works in the field of MEC-NFV resource allocation focusing on mathematical models.
- Description of the main benefits and drawbacks related to the virtualization layer elements that compose the MEC-NFV environment.
- Development of a MEC-NFV node model incorporating dynamic scaling capabilities and service prioritization to accommodate two distinct 5G service categories. Additionally, the formulation of essential performance metrics closely tied to URLLC and eMBB services.
- Evaluations of the impact of varying the sizing of different parameters of a MEC-NFV node on metrics such as average response time, energy consumption and service availability were analyzed.

6.2.1 Publication List

This section describes the author's list of published papers during my final years as an undergraduate and in this master program (Table 85).

Table 85 – Publication List

Reference	Source	Title
[Falcão et al. 2023]	2023 EuCNC 6G Summit	Dynamic Resource Allocation for URLLC in UAV-Enabled Multi-Access Edge Computing
[Falcao et al. 2022]	Journal of Supercomputing	An analytical framework for URLLC in hybrid MEC environments
[Souza et al. 2021]	IEEE Latin America Transac.	Modelling and Analysis of 5G Networks Based on MEC-NFV for URLLC Services
[Balieiro et al. 2021]	IEEE LATINCOM	A Fuzzy-Genetic Approach for 5G/6G Opportunistic Slicing

Source: The author (2022)

Furthermore, stemming directly from this dissertation, three papers are envisioned for submission. The first will briefly introduce the designed model and showcase partial results, aimed at a conference. The second will offer a comprehensive description of the model and detailed analysis, targeting a journal publication. The final paper will focus on the designed CPN-based simulator, presenting new findings, also intended for a journal submission

6.3 FUTURE WORKS

This section is segmented into three groups of future works related to the content of this thesis. The first is related to improvements on the proposed model. The second is concerned with optimization schemes that could use the proposed model and the last group contains other research approaches towards the same problem (e.g., Testbeds).

6.3.1 Related Mathematical Models

6.3.1.1 Reducing model computational complexity

Typically, addressing Markov chains with Ω states using a straightforward algorithm incurs a computational cost of $O(\Omega)^4$, which can pose difficulties in achieving quick solutions. Another challenge is streamlining specific steps for solving the proposed analytical model's linear system by introducing reasonable approximations that maintain a low computational overhead.

6.3.1.2 Derivation of Cumulative Distribution Function (CDF) for the Response time

Due to strict requirements for response times in URLLC applications, it is important to assess the potential risks associated with not meeting these time constraints. To address this challenge, the development of mathematical equations that calculate reliability as a function of the probability that a specified response time will be met by applying Cumulative Distribution Functions emerges as a pertinent and valuable approach to improve this model.

6.3.1.3 Worst Case and Bound-based Models

Recent applications of mathematical techniques for characterizing extreme events and establishing precise limits have been recently used in the strategic planning of wireless networks with a focus on achieving low latency and high reliability. Among these methodologies, extreme value theory, Meta distribution analysis, and network calculus stand out as vital contributors. These approaches depart from traditional reliance on averages and offer an interesting way for more accurate assessments, particularly in terms of worst-case latency values, through bound analysis. Consequently, a new challenge lies in comparing the findings of this study with those

derived from one or more of these alternative methodologies.

6.3.2 Resource Allocation Problem Formulation and Solutions

Utilizing the model presented in this work, we conducted an in-depth analysis of how various system parameters impact performance metrics across different service categories. While our experiments initially necessitated full knowledge of all input parameters, we recognize the potential for optimizing a subset of these parameters. Specifically to address the dimensioning optimization challenge by focusing on the MEC node's resource component and its impacts on the performance metrics of the services. To address this challenge effectively, we can propose the utilization of a heuristic optimization technique designed for solving this multi-objective problem.

6.3.3 Testbed and Simulation

Small-scale network experiments in the field of MEC-NFV are relatively infrequent, likely attributed to the evolving landscape of open-source tools [Zhao et al. 2021]. Nonetheless, we hold the belief that these tools will become more accessible in the foreseeable future. This would enable the comparison of analytical outcomes with real-world testbeds and simulations, particularly with established frameworks known to the community, including OpenAirInterface, Open5GS, and simulation tools such as OMNeT++, NS-3, and CloudSim.

REFERENCES

3GPP. System architecture for the 5g system (5gs). *White Paper*, 2020.

3GPP. *5G; Architecture for enabling Edge Applications(3GPP TS 23.558 version 17.3.0 Release 17))*. [S.l.], 2022.

3GPP. *5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 17.4.0 Release 17)*. [S.l.], 2022.

ABDELHADI, M.; SOROUR, S.; ELSAWY, H.; ELSAYED, S. A.; HASSANEIN, H. Parallel computing at the extreme edge: Spatiotemporal analysis. In: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. [S.l.: s.n.], 2022. p. 5692–5698.

ALI, R.; ZIKRIA, Y. B.; BASHIR, A. K.; GARG, S.; KIM, H. S. Urllc for 5g and beyond: Requirements, enabling incumbent technologies and network intelligence. *IEEE Access*, v. 9, p. 67064–67095, 2021.

ANTEVSKI, K.; BERNARDOS, C. J.; COMINARDI, L.; de la Oliva, A.; MOURAD, A. On the integration of nfv and mec technologies: architecture analysis and benefits for edge robotics. *Computer Networks*, v. 175, p. 107274, 2020. ISSN 1389-1286.

BABA, H.; HIRAI, S.; NAKAMURA, T.; KANEMARU, S.; TAKAHASHI, K.; OMOTO, T.; AKIYAMA, S.; HIRABARU, S. End-to-end 5g network slice resource management and orchestration architecture. In: *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. [S.l.: s.n.], 2022. p. 269–271.

BAIRAGI, A. K.; MUNIR, M. S.; ALSENWI, M.; TRAN, N. H.; ALSHAMRANI, S. S.; MASUD, M.; HAN, Z.; HONG, C. S. Coexistence mechanism between embb and urllc in 5g wireless networks. *IEEE Transactions on Communications*, v. 69, n. 3, p. 1736–1749, 2021.

BALIEIRO, A.; FALCAO, M.; SOUZA, C.; DIAS, K.; ALVES, E. A fuzzy-genetic approach for 5g/6g opportunistic slicing. *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, p. 1–6, 2021.

BLANCO, B.; FAJARDO, J. O.; GIANNOULAKIS, I.; KAFETZAKIS, E.; PENG, S.; PéREZ-ROMERO, J.; TRAJKOVSKA, I.; KHODASHENAS, P. S.; GORATTI, L.; PAOLINO, M.; SFAKIANAKIS, E. Technology pillars in the architecture of future 5g mobile networks: Nfv, mec and sdn. *Computer Standards Interfaces*, v. 54, 01 2017.

BOLCH, G.; GREINER, S.; MEER, H.; TRIVEDI, K. Queueing networks and markov chains. *Wiley New York*, 2006.

CHAKRABORTY, P.; CORICI, M. A comparative study for time series forecasting within software 5g networks. In: *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. [S.l.: s.n.], 2020. p. 1–7.

CONȚU, C.; CIOBANU, A.; BORCOCI, E.; VOCHIN, M.-C.; LI, F. Y. An automation platform for slice creation using open source mano. In: *2022 14th International Conference on Communications (COMM)*. [S.l.: s.n.], 2022. p. 1–6.

COOPER, R. Introduction to queueing theory. *Elsevier North Holland*, v. 2, 1981.

DU, K.; WANG, L.; WEN, X.; LIU, Y.; NIU, H.; HUANG, S. Ml-sld: A message-level stateless design for cloud-native 5g core network. *Digital Communications and Networks*, v. 9, n. 3, p. 743–756, 2023. ISSN 2352-8648.

EMARA, M.; ELSAWY, H.; FILIPPOU, M. C.; BAUCH, G. Spatiotemporal dependable task execution services in mec-enabled wireless systems. *IEEE Wireless Communications Letters*, v. 10, n. 2, p. 211–215, 2021.

ERICSSON. *Fixed wireless access on a massive scale with 5G*. 2016. Available at: <https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/fixed-wireless-access-on-a-massive-scale-with-5g>.

ERICSSON. *Ericsson Mobility Report*. [S.l.], 2023.

ETSI. *Multi-access Edge Computing (MEC);Framework and Reference Architecture;ETSI GS MEC 003 V3.1.1)*. [S.l.], 2022.

FALCAO, M.; SOUZA, C.; BALIEIRO, A.; DIAS, K. An analytical framework for urllc in hybrid mec environments. *The Journal of Supercomputing*, v. 78, 02 2022.

FALCãO, M.; SOUZA, C.; BALIEIRO, A.; DIAS, K. Dynamic resource allocation for urllc in uav-enabled multi-access edge computing. In: *2023 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*. [S.l.: s.n.], 2023. p. 293–298.

FENG, D.; SHE, C.; YING, K.; LAI, L.; HOU, Z.; QUEK, T. Q. S.; LI, Y.; VUCETIC, B. Toward ultrareliable low-latency communications: Typical scenarios, possible solutions, and open issues. *IEEE Vehicular Technology Magazine*, v. 14, n. 2, p. 94–102, 2019.

FILIPPOU, M. C.; SABELLA, D.; EMARA, M.; PRABHAKARAN, S.; SHI, Y.; BIAN, B.; RAO, A. Multi-access edge computing: A comparative analysis of 5g system deployments and service consumption locality variants. *IEEE Communications Standards Magazine*, v. 4, n. 2, p. 32–39, 2020.

GHOSH, A.; MAEDER, A.; BAKER, M.; CHANDRAMOULI, D. 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. *IEEE Access*, v. 7, p. 127639–127651, 2019.

GROSS, D.; SHORTLE, J.; THOMPSON, J.; HARRIS, C. Fundamentals of queueing theory. *John Wiley Sons, Inc*, v. 4, 2008.

HUANG, H.; MIAO, W.; MIN, G.; TIAN, J.; ALAMRI, A. Nfv and blockchain enabled 5g for ultra-reliable and low-latency communications in industry: Architecture and performance evaluation. *IEEE Transactions on Industrial Informatics*, v. 17, n. 8, p. 5595–5604, 2021.

JAIN, R. The art of computer systems performance analysis: Techniques for experimental design, measurement. *Wiley New York*, 1991.

KALOXYLOS, A. A survey and an analysis of network slicing in 5g networks. *IEEE Communications Standards Magazine*, v. 2, n. 1, p. 60–65, 2018.

KAUR, K.; DHAND, T.; KUMAR, N.; ZEADALLY, S. Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers. *IEEE Wireless Communications*, v. 24, n. 3, p. 48–56, 2017.

KEKKI, S.; FEATHERSTONE, W. Mec in 5g networks. *ETSI White Paper*, n. 28, p. 1–28, 2018.

KEMENY, J. Finite markov chains. *van Nostrand Company*, 1960.

KHAN, A. Key characteristics of a container orchestration platform to enable a modern application. *IEEE Cloud Computing*, v. 4, n. 5, p. 42–48, 2017.

KIM, Y.; PARK, S. Calculation method of spectrum requirement for imt-2020 embb and urllc with puncturing based on m/g/1 priority queuing model. *IEEE Access*, v. 8, p. 25027–25040, 2020.

KOONAMPILLI, J. B. N.; VUTUKURU, M.; SIVALINGAM, K. M.; BALASUBRAMANIAN, A.; VINODH, R. V.; SESHASAYEE, S.; GOKHALE, K.; KASHYAP, D. V.; KAMATH, R. R. Demonstration of 5g core software system in india's indigenous 5g test bed. In: *2021 International Conference on COMmunication Systems NETworkS (COMSNETS)*. [S.l.: s.n.], 2021. p. 101–103.

LENTISCO, C. M.; BELLIDO, L.; CáRDENAS, A.; MOYANO, R. F.; FERNáNDEZ, D. Design of a 5g multimedia broadcast application function supporting adaptive error recovery. *IEEE Transactions on Multimedia*, v. 25, p. 378–388, 2023.

LI, C.; CAI, Q.; ZHANG, C.; MA, B.; LUO, Y. Computation offloading and service allocation in mobile edge computing. *The Journal of Supercomputing*, v. 77, p. 1–30, 12 2021.

LI, W.; JIN, S. Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity. *The Journal of Supercomputing*, v. 77, n. 8, 2021.

LIU, T.; FANG, L.; ZHU, Y.; TONG, W.; YANG, Y. A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing. *IEEE Transactions on Mobile Computing*, v. 21, n. 8, p. 2687–2700, 2022.

MA, S.; CHEN, X.; LI, Z.; CHEN, Y. Performance evaluation of urllc in 5g based on stochastic network calculus. *Mobile Networks and Applications*, v. 26, 06 2021.

MAHDI, M. N.; AHMAD, A. R.; QASSIM, Q. S.; NATIQ, H.; SUBHI, M. A.; MAHMOUD, M. From 5g to 6g technology: Meets energy, internet-of-things and machine learning: A survey. *Applied Sciences*, v. 11, n. 17, 2021. ISSN 2076-3417.

MEHMETI, F.; PORTA, T. F. L. Modeling and analysis of mmtc traffic in 5g base stations. In: *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*. [S.l.: s.n.], 2022. p. 652–660.

MIJUMBI, R.; SERRAT, J.; GORRICHO, J.-L.; BOUTEN, N.; TURCK, F. D.; BOUTABA, R. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, v. 18, n. 1, p. 236–262, 2016.

MORABITO, R. Power consumption of virtualization technologies: An empirical investigation. *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, p. 522–527, 2015.

NORRIS, J. Markov chains. *Cambridge University Press*, 1998.

PANA, V.; BABALOLA, O. 5g radio access networks: A survey. *Array*, v. 14, p. 100170, 04 2022.

PéREZ, D. E.; LóPEZ, O. L. A. Robust downlink multi-antenna beamforming with heterogenous csi: Enabling embb and urllc coexistence. *IEEE Transactions on Wireless Communications*, v. 22, n. 6, p. 4146–4157, 2023.

RACA, D.; LEAHY, D.; SREENAN, C. J.; QUINLAN, J. J. Beyond throughput, the next generation: A 5g dataset with channel and context metrics. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. [S.l.]: Association for Computing Machinery, 2020. (MMSys '20), p. 303–308.

RATZER, A. V.; WELLS, L.; LASSEN, H. M.; LAURSEN, M.; QVORTRUP, J. F.; STISSING, M. S.; WESTERGAARD, M.; CHRISTENSEN, S.; JENSEN, K. Cpn tools for editing, simulating, and analysing coloured petri nets. In: AALST, W. M. P. van der; BEST, E. (Ed.). *Applications and Theory of Petri Nets 2003*. [S.l.]: Springer Berlin Heidelberg, 2003. p. 450–462. ISBN 978-3-540-44919-5.

RESEARCH; MARKET. *Multi-access Edge Computing Market Size, Share Trends Analysis Report By Solution (Hardware, Software, Services), By End-use (IT Telecom Smart Buildings, Datacenters, Energy Utilities), By Region, And Segment Forecasts, 2023 - 2030*. [S.l.], 2023.

RODRIGUEZ, V. Q.; GUILLEMIN, F. Automating the deployment of 5g network slices using onap. In: *2019 10th International Conference on Networks of the Future (NoF)*. [S.l.: s.n.], 2019. p. 32–39.

SANTOYO-GONZALEZ, A.; CERVELLO-PASTOR, C. Edge nodes infrastructure placement parameters for 5g networks. *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, p. 1–6, 2018.

SARRIGIANNIS, I.; RAMANTAS, K.; KARTSAKLI, E.; MEKIKIS, P.-V.; ANTONOPOULOS, A.; VERIKOUKIS, C. Online vnf lifecycle management in an mec-enabled 5g iot architecture. *IEEE Internet of Things Journal*, v. 7, n. 5, p. 4183–4194, 2020.

SCOTECE, D.; NOOR, A.; FOSCHINI, L.; CORRADI, A. 5g-kube: Complex telco core infrastructure deployment made low-cost. *IEEE Communications Magazine*, p. 1–7, 2023.

SETAYESH, M.; BAHRAMI, S. Resource slicing for embb and urllc services in radio access network using hierarchical deep learning. In: . [S.l.: s.n.], 2022. v. 21, n. 11.

SIDDIQUI, M. U. A.; ABUMARSHOUD, H.; BARIAH, L.; MUHAIDAT, S.; IMRAN, M. A.; MOHJAZI, L. Urllc in beyond 5g and 6g networks: An interference management perspective. *IEEE Access*, v. 11, p. 54639–54663, 2023.

SINGH, K.; XIE, M. *Bootstrap: A Statistical Method*. 2008. Available at: <http://www.stat.rutgers.edu/home/mxie/rcpapers/bootstrap.pdf>.

SOHAIB, R. M.; ONIRETI, O.; SAMBO, Y.; SWASH, R.; ANSARI, S.; IMRAN, M. A. Intelligent resource management for embb and urllc in 5g and beyond wireless networks. *IEEE Access*, p. 1–1, 2023.

SOUZA, C.; FALCAO, M.; BALIEIRO, A.; DIAS, K. Modelling and analysis of 5g networks based on mec-nfv for urllc services. *IEEE Latin America Transactions*, v. 19, n. 10, p. 1745–1753, 2021.

SOUZA, C.; FALCAO, M.; BALIEIRO, A.; DIAS, K. Modelling and analysis of 5g networks based on mec-nfv for urllc services. *IEEE Latin America Transactions*, v. 19, n. 10, p. 1745–1753, 2021.

SPRECHER NURIT AND, E. O.; KUURE PEKKA, N. A.; SOLOWAY, A.; HARMAND, A.; CHITTURI, S.; REZNIK, A.; LI ALICE, C. C. et al. Harmonizing standards for edge computing - a synergized architecture leveraging etsi isg mec and 3gpp specifications. *ETSI White Paper*, n. 36, p. 1–13, 2020.

STALLINGS, W. *5G Wireless: A Comprehensive Introduction*. Addison-Wesley, 2021. ISBN 9780136767145. Available at: <https://books.google.fi/books?id=-V4OzgEACAAJ>.

SUGITO, Y.; IWASAKI, S.; CHIDA, K.; IGUCHI, K.; KANDA, K.; LEI, X.; MIYOSHI, H.; KAZUI, K. Video bit-rate requirements for 8k 120-hz hevc/h.265 temporal scalable coding: experimental study based on 8k subjective evaluations. *APSIPA Transactions on Signal and Information Processing*, Cambridge University Press, v. 9, p. e5, 2020.

TAKO, A.; ROBINSON, S. Comparing discrete-event simulation and system dynamics: Users' perceptions. *JORS*, v. 60, p. 296–312, 03 2009.

TALEB, T.; CORICI, M.; PARADA, C.; JAMAKOVIC, A.; RUFFINO, S.; KARAGIANNIS, G.; MAGEDANZ, T. Ease: Epc as a service to ease mobile core network deployment over cloud. *IEEE Network*, v. 29, n. 2, p. 78–88, 2015.

TANG, Q.; ERMIS, O.; NGUYEN, C. D.; OLIVEIRA, A. D.; HIRTZIG, A. A systematic analysis of 5g networks with a focus on 5g core security. *IEEE Access*, v. 10, p. 18298–18319, 2022.

TONG, Z.; ZHANG, T.; ZHU, Y.; HUANG, R. Communication and computation resource allocation for end-to-end slicing in mobile networks. *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, p. 1286–1291, 2020.

WEISSBERGER, A. *Multi-access Edge Computing (MEC) Market, Applications and ETSI MEC Standard*. 2021. Available at: <https://techblog.comsoc.org/2021/12/15/multi-access-edge-computing-mec-market-applications-and-technology-part-i/>.

XUE, P.; JIANG, Z. Secrouting: Secure routing for network functions virtualization (nfv) technology. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 69, n. 3, p. 1727–1731, 2022.

YOUSAF, F. Z.; SCIANCALEPORE, V.; LIEBSCH, M.; COSTA-PEREZ, X. Manoaas: A multi-tenant nfv mano for 5g network slices. *IEEE Communications Magazine*, v. 57, n. 5, p. 103–109, 2019.

YU, Y. Mobile edge computing towards 5g: Vision, recent progress, and open challenges. *China Communications*, v. 13, n. Supplement2, p. 89–99, 2016.

ZHANG, T.; QIU, H.; LINGUAGLOSSA, L.; CERRONI, W.; GIACCONE, P. Nfv platforms: Taxonomy, design choices and future challenges. *IEEE Transactions on Network and Service Management*, v. 18, n. 1, p. 30–48, 2021.

ZHAO, L.; ZHOU, G.; ZHENG, G.; I., C.-L.; YOU, X.; HANZO, L. Open-source multi-access edge computing for 6g: Opportunities and challenges. *IEEE Access*, PP, 11 2021.