UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE COMPUTAÇÃO
UFRPE

Caio Bruno Bezerra de Souza

Alocação Dinâmica de Recursos para URLLC em Redes 5G NFV-MEC

Recife

2020

Caio Bruno Bezerra de Souza

Alocação Dinâmica de Recursos para URLLC em Redes 5G NFV-MEC

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Computação

Curso de Bacharelado em Ciência da Computação

Orientador: Danilo Ricardo Barbosa de Araújo

Coorientador: Andson Marreiros Balieiro

Recife

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

S729a Souza, Caio Bruno Bezerra de
Alocação Dinâmica de Recursos para URLLC em Redes 5G NFV-MEC / Caio Bruno Bezerra de Souza. 2020.
56 f. : il.

Orientador: Danilo Ricardo Barbosa de Araujo.
Coorientador: Andson Marreiros Balieiro.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Ciência da Computação, Recife, 2020.

1. Redes 5G. 2. URLLC. 3. NFV. 4. Teoria de Filas. I. Araujo, Danilo Ricardo Barbosa de, orient. II. Balieiro, Andson Marreiros, coorient. III. Título

CDD 004

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Caio Bruno Bezerra de Souza às 09 horas do dia 03 de novembro de 2020, no link https://meet.google.com/tnx-tmvu-kpo, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Alocação Dinâmica de Recursos para URLLC em Redes 5G NFV-MEC**, orientado por Danilo Ricardo Barbosa de Araújo e aprovado pela seguinte banca examinadora:

Prof. Danilo Ricardo Barbosa de Araújo
DC/UFRPE

Prof. Carlos Julian Menezes Araújo
DC/UFRPE

Aos meus pais Jose Souza e Maria Bezerra.

Agradecimentos

Primeiramente gostaria de agradecer a Deus por me proporcionar força e sabedoria durante toda a minha vida.

Sou grato aos meus pais Jose Souza e Maria Bezerra que sempre me incentivaram e apoiaram em todas os momentos da minha vida.

Deixo um agradecimento especial ao meu coorientador Dr. Andson Balieiro que dedicou inúmeras horas para sanar minhas dúvidas mesmo com seu escasso tempo livre.

Agradeço ao meu orientador Dr. Danilo Araújo por aceitar me orientar nesta situação extemporânea.

Também agradeço a meu companheiro de pesquisa Marcos Falcão que sempre me ajudou desde o início deste projeto de pesquisa.

Devo também expressar minha gratidão ao professor Carlos Julian por concordar em avaliar este trabalho.

A todos os meus amigos da UFRPE que compartilharam dos inúmeros desafios que enfrentamos, sempre com o espírito colaborativo.

Por último, quero agradecer também à Universidade Federal Rural de Pernambuco pela elevada qualidade do ensino oferecido.

“Sonhos determinam o que você quer. Ação determina o que você conquista.”
(Aldo Novak)

Resumo

A Quinta Geração de redes móveis (5G) busca suportar uma diversidade de aplicações categorizadas em três tipos: largura de banda móvel melhorada (eMBB, do inglês, *enhanced Mobile Broadband*), comunicação do tipo máquina massiva (mMTC, do inglês, *massive Machine Type Communications*) e comunicação com baixa latência e confiabilidade muito alta (URLLC, do inglês, *Ultra Reliable Low Latency Communications*), em que a última é talvez a mais desafiadora devido as suas restrições de latência fim-a-fim (poucos milissegundos), baixa probabilidade de perda de pacotes e alta disponibilidade da rede, não alcançáveis nas redes móveis atuais. Assim como nas gerações anteriores, grande parte dos esforços de pesquisa tem se concentrado nas redes de acesso por rádio (RAN, do inglês, *Radio Access Network*), sendo o núcleo 5G frequentemente assumido como sendo similar em operação aos dos *datacenters* comuns, embora esteja claro que eles podem não conseguir lidar com os requisitos dos serviços URLLC. O suporte as aplicações URLLC em ambientes de computação de borda multiacesso (MEC, do inglês, *Multi-Access Edge Computing*) utilizando virtualização de funções de rede (NFV, do inglês, *Network Functions Virtualization*) traz desafios, em que diferentes aspectos devem ser considerados. Este trabalho busca analisar o aprovisionamento de recursos para serviços URLLC em redes 5G baseadas em MEC-NFV, considerando o tempo de configuração/inicialização da Função de rede Virtual (VNF, do inglês, *Virtualized Network Function*), a possibilidade de falha durante o atendimento, associados à técnica de pré-inicialização de recursos, determinando os limites de quão minimamente performático o aprovisionamento de recursos do URLLC deve ser. Para isso, foi proposto nesta monografia um modelo analítico baseado em teoria de fila e validado via simulador desenvolvido em Rede de Petri Colorida e o tempo médio de resposta, a probabilidade de bloqueio e o número médio de recursos ativos são analisados sob diferentes taxas de chegada de serviços, taxas de inicialização (*setup*) de recursos, capacidade máxima do sistema, quantidade de recursos (contêineres) e número de contêineres pré-inicializados. A partir disso foi observado que o efeito da taxa de setup menor pode ser mitigado pela préinicialização de contêineres, diminuindo o tempo de espera para atendimento do serviço.

Palavras-chave: Redes 5G. URLLC. NFV. Teoria de Filas.

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), where the latter is perhaps the most challenging due to its end-to-end latency restrictions (few milliseconds), low probability of packet loss and high network availability, which are not reachable in today's mobile networks. As in previous generations, much of the research effort has focused on the Radio Access Network (RAN), with the 5G core often assumed to be similar in operation to that of common data centers, although it is clear that they may not be able to handle the requirements of URLLC services. Support for URLLC applications in Multi-Access Edge Computing (MEC) environments using Network Functions Virtualization (NFV) brings challenges, in which different aspects must be considered. This work seeks to analyze the provisioning of resources for URLLC services in 5G networks based on MEC-NFV, considering the time of configuration / initialization of the Virtualized Network Function (VNF), the possibility of failure during service, associated with the pre-initialization technique resources, determining the limits of how minimally performance the URLLC resource provisioning should be. For this purpose, an analytical model based on queue theory and validated via a simulator developed in a Colored Petri Net was proposed in this monograph and the average response time, the probability of blocking and the average number of active resources are analyzed under different service arrival rates, resource startup rates, maximum system capacity, number of resources (containers) and number of pre-initialized containers. From this it was observed that the effect of the lower setup rate can be mitigated by the pre-initialization of containers, reducing the waiting time for service delivery.

Keywords: 5G networks. URLLC. NFV. Queuing Theory.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IP	Protocolo de Internet
eMBB	Banda Larga Móvel Aprimorada
mMTC	Comunicação Massiva do Tipo Máquina
IoT	Internet das Coisas
URLLC	Comunicação com Confiabilidade Muito Alta e de Latência Muito Baixa
RAN	Redes de Acesso por Rádio
MEC	Computação de Borda Multiacesso
NFV	Virtualização de Funções de Rede
SDN	Rede Definida por Software
VNF	Função de Rede Virtual
ITU	União Internacional de Telecomunicações
3GPP	Third Generation Partnership Project
VLC	Comunicação de Luz Visível
CN	Rede de Núcleo
NAT	Tradução do Endereço da Rede
MME	Entidade de Gestão de Mobilidade
PDN	Rede de Pacotes de Dados
DNS	Sistema de Nomes de Domínio
ESTI	Sociedade Europeia de Imagem Torácica
SLA	Acordo de Nível de Serviço
VM	Máquina Virtual
FMP	Função de Massa de Probabilidade
FDC	Função de Distribuição Cumulativa

FDP	Função de Densidade de Probabilidade
FCFS	Primeiro a Chegar, Primeiro a ser Servido
LCFS	Último a Chegar, Primeiro a ser Servido
SIRO	Serviço Em Ordem Aleatória
RR	Round Robin
PB	Probabilidade de Bloqueio
MS	Número Médio de Serviços no Sistema
nCTNs	Número de Contêineres Ligados no Sistema
RPC	Rede de Petri colorida

Sumário

1 Introdução

As redes móveis vêm evoluindo com o passar dos anos, desde a primeira mudança em que uma rede analógica (Primeira Geração - 1G) evoluiu para digital (Segunda Geração - 2G), com maior eficiência espectral, melhores serviços de dados e *roaming* melhorado. A Terceira Geração (3G) surgiu para prover suporte a serviços multimídia com uma taxa de transferência de dados de pelo menos 2 Mbps, mas com desempenho penalizado por utilizar transferência de pacotes comutada por circuitos (SHARMA, 2013). Este problema foi solucionado na 4ª Geração (4G) com as redes comutadas por pacotes IP, que oferece taxas de dados com pico de 100 Mbps para comunicação de alta mobilidade e 1 Gbps para comunicação de baixa mobilidade (WUTTIDITTACHOTTI; DAENGSI, 2015).

De acordo com previsões, o número de dispositivos móveis conectados deve ultrapassar a quantidade de 30 bilhões até 2020 (ALI, 2018), e isto proporcionará um aumento no tráfego de dados. Levando em consideração este crescente número de dispositivos e o uso de aplicações que demandam uma grande vazão de dados, é requerido que as redes provam maior taxa de transmissão e suporte a uma alta densidade de conexões. Para atender as exigências de transmissão de dados, melhorar as tecnologias atuais não será o bastante, pois o 4G está próximo do seu limite teórico de capacidade de transmissão (WANG et al., 2014).

Neste aspecto, as redes 5G surgem e visam atingir 1000 vezes a capacidade do sistema atual, 10 vezes a eficiência espectral, energética e a taxa de dados, 25 vezes a taxa de transferência média das células, 10 a 100 vezes o número de dispositivos conectados e 5 vezes a redução da latência de ponta a ponta (WANG et al., 2014).

As redes 5G devem suportar três categorias de serviço principais: banda larga móvel aprimorada (eMBB), que oferece conexões de alta velocidade e engloba aplicações como o *streaming* de vídeo de alta resolução; comunicação massiva do tipo máquina (mMTC), capaz de fornecer conectividade a uma grande quantidade de dispositivos inseridos no conceito de Internet das Coisas (IoT) (SHE et al., 2018); e comunicação com confiabilidade muito alta e de latência muito baixa (URLLC), que suporta conexões com requisitos estritos em termos de latência, confiabilidade e disponibilidade (SHE et al., 2018) e permite a utilização de carros autônomos e Internet tátil, por exemplo. Entre as categorias, o URLLC é talvez o mais desafiador devido as suas restrições de latência fim a fim não superior a 1 ms, probabilidade de perda de pacotes de até 10-7 e disponibilidade de rede de 99,999%, não satisfeitas nas redes móveis atuais (LI et al., 2017).

Nas gerações anteriores, grande parte dos esforços de pesquisa são voltados para às redes de acesso por rádio (RAN), mas o desafio da engenharia vai além das atualizações de rádio; requer uma reformulação completa das outras partes da rede (FOUKAS; MARINA; KONTOVASILIS, 2017), por exemplo da rede de núcleo. As preocupações com a RAN são compreensíveis porque o domínio sem fio sofre devido a inúmeros desafios; o núcleo 5G, por outro lado, é frequentemente assumido como sendo similar em operação aos dos *datacenters* comuns, embora agora esteja claro que eles podem não conseguir lidar com os requisitos dos serviços URLLC (JI et al., 2018). Nesse sentido, alterações para além da rede de núcleo são fundamentais para sistemas 5G com o objetivo de reduzir a latência fim a fim, aumentar a confiabilidade e atender aos requisitos das outras categorias de serviços. Desse modo, diferentes tecnologias tais como o uso de MEC (*Multi-Access Edge Computing*), NFV (*Network Functions Virtualization*) e SDN (*Software Defined Networking*) tem sido propostas para compor essa nova geração de redes móveis.

Em termos de URLLC, uma alternativa para reduzir a latência na comunicação com o usuário é posicionar as funções do núcleo da rede (ex. *gateways*), aplicações e serviços para próximo da borda. Neste aspecto, a Virtualização de Função de Rede (NFV, do inglês, *Network Functions Virtualization*) e Computação de Borda Multiacesso (MEC, do inglês, *Multi-Access Edge Computing*) são essenciais para isto ser alcançado. A NFV permite que as funções de rede (e.g. *Network Address Translation, Domain Name System, caching*) sejam virtualizadas e executem em servidores de propósito gerais; já a MEC possibilita que as funções virtualizadas sejam posicionadas na borda da rede.

Entretanto, a transição para MEC pode ser custosa tanto em operação quanto em implantação para o provedor de serviço, pois diferentes dos grandes centros de dados, que concentram os recursos em poucas localizações, os nós MEC são distribuídos em vários pontos e tem menor quantidade de recursos, demandando avaliação de aspectos como espaço físico, energia, rede de acesso, licenças de *hardware* e *software*. Para evitar desperdício de recursos (ex. contêineres) que ficam ativos, mas ociosos, uma solução é realizar a ativação e alocação deles sob demanda, à medida que os serviços chegam (REN et al., 2016). Entretanto, o atraso incorrido pela inicialização dos recursos (ex. transferência e carregamento da imagem da VNF no contêiner) pode causar a violação dos requisitos dos serviços URLLC, que pode ser mitigada mantendo alguns recursos inicializados antecipadamente, baseado em algum critério definido pela operadora (ex. tendência de aumento de demanda). Adicionalmente, falhas de *hardware* ou *software* durante o processamento dos serviços ou o tempo de espera aguardando para processamento devido à limitação de recursos podem ocorrer e causar a violação de tempo de resposta dos serviços URLLC. Assim, o suporte as aplicações URLLC em ambientes MEC-NFV traz desafios, onde diferentes aspectos

devem ser considerados.

1.1 Objetivos

Este trabalho tem como objetivo avaliar o impacto do provisionamento e alocação de recursos para o URLLC em redes 5G, levando em consideração os avanços tecnológicos esperados do NFV-MEC associadas as técnicas de pré-inicialização de recursos, para ajudar a determinar os limites de quão minimamente performático o provisionamento de recursos do URLLC deve ser. Para alcançar este objetivo, os seguintes objetivos específicos foram definidos:

1. Investigar na literatura quais fatores que podem impactar no suporte as aplicações URLLC na rede NFV-MEC.
2. Construir um modelo baseado em teoria de filas para reproduzir um ambiente 5G NFV-MEC composto por uma nuvem de borda e o aprovisionamento de múltiplas VNFs considerando a possibilidade de falha durante o atendimento de serviço (URLLC) e configuração prévia de VNFs (antes das requisições efetivamente serem recebidas) dos recursos de processamento.
3. Analisar o impacto do aprovisionamento de recursos no desempenho das aplicações URLLC em diferentes cenários.

1.2 Organização do trabalho

Este trabalho foi organizado da seguinte maneira: O Capítulo 2 detalha o referencial teórico, que inclui uma apresentação das redes 5G e suas categorias de serviço, as principais tecnologias habilitadoras para a categoria de serviço URLLC, a motivação para o estudo da alocação de recursos para redes 5G NFV-MEC, uma visão geral da teoria das filas e processos de Markov, e uma breve apresentação das Redes de Petri Coloridas. O Sistema 5G URLLC auto escalável considerando falha e Pré-setup é descrito no Capítulo 3. Além disso, é realizada a modelagem deste sistema por meio da teoria das filas e sua validação utilizando redes de petri coloridas. A simulação e os resultados da análise são discutidos no Capítulo 4. Finalmente, o Capítulo 5 conclui esta tese e destaca os trabalhos futuros.

2 Referencial Teórico

Este capítulo apresenta os conceitos fundamentais para o entendimento do trabalho, apresentando as redes 5G e suas categorias de serviço, as principais tecnologias habilitadoras para a categoria de serviço URLLC, uma breve discussão sobre a alocação de recursos para redes 5G NFV-MEC e, finalmente, uma visão geral da teoria das filas e processos de Markov.

2.1 Redes 5G e suas categorias de serviços

A Quinta Geração (5G) de Redes Móveis Sem Fio iniciou a sua operação em alguns países e busca atender demandas que estão além das capacidades dos sistemas atuais, tais como a grande quantidade de dispositivos conectados de aplicações de Internet das Coisas (IoT) e comunicações do tipo dispositivo a dispositivo (ex. automação industrial), o crescimento explosivo de tráfego móvel de alta velocidade (ex. *streaming* de vídeo de definição muito alta e aplicações de realidade virtual) e comunicação com alta restrição de latência e confiabilidade (e.g. telecirurgia e veículos autônomos) (PARVEZ et al., 2018).

Os serviços 5G são categorizados pela União Internacional de Telecomunicações (ITU) e o 3GPP (*Third Generation Partnership Project*) em banda larga móvel melhorada (eMBB, do inglês, *enhanced Mobile Broadband*), comunicação do tipo máquina massiva (mMTC, do inglês, massive *Machine Type Communications*) e comunicação com confiabilidade muito alta e latência muito baixa (URLLC, do inglês, *Ultra Reliable Low Latency Communications*) (POCOVI et al., 2018)(JI et al., 2018), que apresentam diferentes requisitos em termos de latência, *throughput*, densidade de conexão, confiabilidade e consumo energético, por exemplo. A primeira suporta conexões estáveis de alta velocidade (CHEN; CHENG; WANG, 2017) e é relacionada a aplicações que demandam grande largura de banda tais como *streaming* de vídeo de alta resolução e realidade aumentada. A segunda fornece conectividade para um elevado montante dispositivos do tipo máquina, que formam a Internet das Coisas (IoT) (POCOVI et al., 2018). Sensoriamento e monitoramento são exemplos de serviços mMTC, que requerem alta densidade de conexão e eficiência energética (SCHULZ et al., 2017). Já os serviços URLLC abrangem aplicações com requisitos estritos em termos latência, confiabilidade e disponibilidade (SHE et al., 2018). Essa categoria engloba aplicações tais como carros autônomos, internet tátil e controle industrial, cujos requisitos estão além da capacidade das tecnologias de redes sem fio atuais, como por exemplo latência fim a fim de 1 ms, probabilidade de perda de pacotes de 10^{-7} e disponibilidade da rede

de 99,999% (SHE et al., 2018). A latência estrita significa que os dados que não são decodificados no receptor dentro do tempo definido perdem a sua utilidade e podem ser descartados do sistema, resultando em redução de confiabilidade (LI et al., 2017). Já a disponibilidade é dada como a probabilidade de que a confiabilidade e latência demandada pelos usuários sejam satisfeitas pela rede (SHE et al., 2018).

As restrições dos serviços URLLC introduzem vários desafios no projeto das redes 5G. As entidades de padronização, indústria e a academia têm desenvolvido tecnologias e mecanismos para endereçar os desafios, dando maior atenção a rede de acesso via rádio (RAN), tais como a numerologia que permite espaçamento flexível entre subportadoras, diferentes esquemas de codificação e correção de erro, estrutura do frame variável e *mini-slot* para suportar a transmissão de pacotes pequenos, acesso aleatório livre de garantia, uso de ondas milimétricas, comunicação em Terahertz e por luz visível (VLC), diversidade de frequência e interface (GIORDANI et al., 2020) (POCOVI et al., 2018). Entretanto, o atendimento dos requisitos dos serviços URLLC depende dos outros componentes da rede além da RAN, como a rede de núcleo (CN). Em termos de rede núcleo, duas tecnologias são consideradas importantes para compor as redes 5G, a Virtualização de Funções de Rede (NFV) e a e Computação de Borda de Acesso Múltiplo, as quais serão descritas na Seção 2.2.

2.2 As tecnologias NFV e MEC

Os serviços URLLC demandam da rede menor tempo de resposta e ocorrência de falha durante o atendimento do usuário. Assim, a literatura reforça que para evitar atrasos excessivos no *backhaul*, os servidores URLLC devem ser colocados fisicamente mais próximos do usuário final, fenômeno denominado Computação na Borda, com a tecnologia *Multi-access Edge Computing* (MEC) sendo uma das habilitadoras (RUIZ et al., 2018). O MEC é tipicamente caracterizado pelos seguintes atributos: (1) proximidade física entre servidor e usuário, (2) reconhecimento de localização e (3) informação de contexto (PORAMBAGE et al., 2018). Espera-se que os hosts MEC (ou seja, pequenas infraestruturas de computação) sejam implantados nas chamadas bordas da rede, ou seja, colocados a apenas um ou dois saltos de rede longe dos usuários, garantindo suporte a aplicações sensíveis ao atraso, como carros autônomos e automação de fábrica, por exemplo.

O MEC não é apenas razoável para os usuários finais conectados às bordas, mas também para a rede de *backhaul*, uma vez que o descongestionamento do núcleo é provavelmente o efeito colateral.

A NFV é uma iniciativa dirigida pelas operadoras para a criação de funções de rede virtualizadas (VNF), por exemplo *switches*, roteadores, *firewalls* e NATs, usando

máquinas virtuais e/ou contêineres e executando-as em servidores padrões (genéricos) em vez de dispositivos de rede de propósito único, que possuem forte acoplamento entre *hardware* e *softwares*. Duas vantagens relacionadas aos VNFs são a facilidade de realocação de funções de rede em locais diferentes, não exigindo novo *hardware*, além de reduzir custos operacionais e de implantação. O NFV também pode otimizar o escalonamento de tarefas e a alocação de recursos, dada uma quantidade limitada de recursos de computação e *hardware* disponíveis, pois para cada solicitação de serviço que chega, a VNF necessária pode ser diferente.

O uso combinado de NFV e MEC deve permitir um aumento de escalabilidade, pois facilita o escalonamento de recursos sob demanda. A virtualização da função de rede deve suportar a tendência de nuvem móvel, permitindo que funções do núcleo da rede (por exemplo, MME, PDN, DNS, NAT) sejam desacopladas do hardware dedicado (MIJUMBI et al., 2015). De acordo com a ESTI, a MEC pode usar a infraestrutura NFV (NFVI) como a plataforma de virtualização para executar aplicações junto com outras VNFs.

2.3 Estudos em aprovisionamento de recursos em redes 5G baseadas em NFV-MEC

As redes 5G são alvo de uma série de pesquisas que normalmente resumem-se a rede de acesso (RAN) ou que trazem soluções/modelos propostos anteriormente para *datacenters* tradicionais, pois a arquitetura NFV-MEC propõe um movimento de "cloudificação" no sentido da borda. Embora entendamos que esse movimento é natural, uma vez que os requisitos de algumas categorias 5G são muito exigentes quanto a latência e a confiabilidade, há uma lacuna quanto a consideração de fatores que podem impactar no provisionamento de recursos na MEC. Por exemplo, trabalhos anteriores normalmente propõem nuvens que podem ser livres de falhas e/ou tempo de provisionamento, no entanto, dado a sensibilidade do 5G. Até onde foi verificado, os trabalhos anteriores abordaram o provisionamento de recursos em NFV puro ou em NFV-MEC no contexto de *datacenters* tradicionais e/ou usando suposições destes ambientes para 5G, sem considerar que existem subcategorias de serviço que divergem amplamente.

O processo de inicialização de uma instância de VNF é um aspecto crucial nos estudos de desempenho de custos para borda/núcleo da rede 5G, pois durante o processo de instalação a energia é consumida e os recursos são ocupados, mas os serviços não estão sendo atendidos. Além disso, violações do acordo de nível de serviço (SLA, do inglês, Service Level Agreement) podem ocorrer se a VNF demorar muito para iniciar o processamento de fluxos críticos de tráfego. Há vários fatores que podem con-

tribuir para determinar esse atraso, tais como o cabeamento físico até as tecnologias de virtualização. Por exemplo, o tempo de atraso pode ser de 10 minutos ou mais para iniciar uma instância no Microsoft Azure (HILL et al., 2010). Por um lado, alguns trabalhos existentes (HUANG et al., 2016) ainda ignoram o tempo de instalação ou o custo durante o tempo de instalação, tornando a avaliação da alocação / escala de recursos menos confiável. Apesar de alguns autores ignorarem os aspectos relativos ao *setup*, esse fator tem sido abordado em estudos como (VAKILINIA; CHERIET; RAJKUMAR, 2016) (BILAL et al., 2016) (RIGHI et al., 2015) (REN et al., 2016) (REN et al., 2018) em comparação com outros aspectos (por exemplo, taxa de falha). Além disso, nota-se que mesmo trabalhos focados em 5G (REN et al., 2016) (REN et al., 2018) ainda têm dimensões e limites de *datacenters* convencionais (com tráfego web), o que torna a avaliação bastante imprecisa.

Para fornecer diretrizes teóricas às operadoras de telefonia móvel, o contexto e os modelos analíticos devem estar alinhados às tendências tecnológicas atuais. Em particular, no contexto 5G, espera-se que os contêineres baseados em microsserviço preencham essa lacuna deixada pelas VMs comuns. O trabalho (KHAZAEI; BARNA; LITOIU, 2019) adotou essa suposição na modelagem de desempenho regular do *datacenter*, mas os autores não focaram em redes 5G. A modelagem de desempenho considerou três tipos de componentes (Contêineres, Maquinas Virtuais e Maquinas Físicas) escalonáveis em um modelo baseado na teoria de filas. Em resumo, com relação ao alinhamento da tecnologia de virtualização em direção a uma nuvem mais responsiva, acredita-se que este seja um dos trabalhos mais próximos este, pois fornece uma abordagem sistemática para estimar a elasticidade da plataforma de microsserviço.

Os autores em (REN et al., 2016) adotam VMs como recursos a serem provisionados sob demanda. Entretanto, elas possuem um *overhead* de inicialização que inviabiliza os serviços URLLC. Assim, o uso de containers baseados em microsservicos para executar as VNFs pode ser uma alternativa para mitigar esse problema. Já em (XIONG et al., 2019) (HÖSSLER; SIMSEK; FETTWEIS, 2018), soluções voltadas para MEC são apresentadas, entretanto negligenciam o atraso na inicialização dos recursos e considera um sistema livre de falha.

2.4 Estudo sobre teoria de filias e processo Markoviano

Nesta seção destacam-se alguns conceitos sobre processos estocásticos e teoria de filas estudados e que dão base para o entendimento dos modelos analíticos desenvolvidos no projeto.

2.4.1 Variáveis aleatórias

Uma variável aleatória é definida como uma função que demonstra os resultados de um experimento aleatório (BOLCH et al., 2006) e associa um número real a cada elemento do espaço amostral. O valor de uma variável aleatória é definido após a realização do experimento, sendo assim é possível atribuir probabilidades aos valores da variável aleatória. Por exemplo, dada uma caixa com bolas verdes, vermelhas e brancas, remover uma dessas bolas da caixa de forma aleatória pode descrever uma variável aleatória que assume os valores das cores possíveis, em que cada um desses eventos tem uma probabilidade associada de ocorrer. Nesse caso, a variável aleatória assume apenas valores discretos. Existem também experimentos cujo os resultados assumem valores contínuos, como por exemplo, a verificação do tempo entre as falhas de um determinado equipamento. Assim, uma variável aleatória pode ser classificada como contínua ou discreta.

Uma variável aleatória discreta é definida pelo conjunto de valores que pode assumir e pelas probabilidades associadas a esses valores (BOLCH et al., 2006). O conjunto de probabilidades é denominado função de massa de probabilidade (FMP) da variável. Dada uma variável aleatória X que assume apenas valores inteiros não negativos, sua FMP é descrita na Eq. 2.1, que define a probabilidade da variável aleatória X assumir o valor k.

$$p_k = P(X = k),\ para\ k = 0, 1, 2, 3, \ldots \tag{2.1}$$

Para ser considerada uma FMP de uma variável aleatória, uma função deve satisfazer duas condições. A primeira indica que a probabilidade de um valor ocorrer não pode ser negativa. A segunda condição é que a soma de todas as probabilidades deve ser igual a 1. Entre as variáveis aleatórias discretas, se destaca a de Poisson. Essa variável expressa a probabilidade de um evento acontecer k vezes em um determinado intervalo de tempo. Sua FMP é descrita pela Eq. 2.2, em que α é a taxa de ocorrência de eventos. Essa variável tem sido amplamente usada para descrição de eventos como o número de mensagens indesejadas recebidas por dia, o número de requisições feitas a um servidor por hora e o número de clientes atendidos por dia em uma empresa, por exemplo.

$$P(X = k) = \frac{\alpha^k e^{-\alpha}}{k!}. \tag{2.2}$$

Um dos parâmetros mais importantes que podem ser derivados de uma FMP de uma variável aleatória discreta é o valor médio (BOLCH et al., 2006). Esse valor

pode ser obtido pela Eq. 2.3 A média de uma variável aleatória de Poisson é α.

$$\overline{X} = E[X] = \Sigma k P(X = k). \tag{2.3}$$

Uma variável aleatória é contínua se puder assumir todos os valores no intervalo [a, b]. Ela é descrita por sua função de distribuição cumulativa (FDC), que é expressa pela Eq. 2.4. Essa equação define a probabilidade de a variável aleatória X assumir valores menores ou iguais a x.

$$F_x(x) = P(X \leq x). \tag{2.4}$$

Variáveis aleatórias continuas também podem ser descritas por sua função de densidade de probabilidade (FDP), que representa a inclinação da FDC. A FDP pode ser obtida através da derivada do FDC, como mostrado na Eq. 2.5

$$f_x(x) = \frac{dF_x(x)}{dx}. \tag{2.5}$$

A FDP possui algumas propriedades análogas a FMP, descritas nas Eqs. 2.6-2.10. A Eq. 2.8 mostra como calcular a probabilidade de uma variável aleatória X assumir valores entre x_1 e x_2. A Eq. 2.9 indica que a probabilidade de uma variável aleatória contínua assumir um valor específico (ponto) é igual a 0. A última propriedade (Eq. 2.10) descreve como calcular a probabilidade de X seja superior a um determinado valor x_3.

$$f_x(x) \geq 0 \; para \; todo \; k. \tag{2.6}$$

$$\int f_x(x)dx = 1. \tag{2.7}$$

$$P(x_1 \leq X \leq x_2) = \int_{x_1}^{x_2} f_x(x)dx. \tag{2.8}$$

$$P(X = k) = \int_{x}^{x} f_x(x)dx = 0. \tag{2.9}$$

$$P(X > x_3) = \int_{x_3}^{\infty} f_x(x)dx. \tag{2.10}$$

De forma similar as variáveis aleatórias discretas, o valor médio de uma variável aleatória contínua também é importante, e pode ser obtido usando a FDP, como mostrado na Eq. 2.11.

$$\overline{X} = E[X] = \int_{-\infty}^{\infty} x f_x(x) dx. \tag{2.11}$$

A distribuição exponencial é um tipo de distribuição de variável aleatória contínua bastante empregada em teoria de filas para descrever o tempo entre chegadas e o tempo de serviço dos clientes (BOLCH et al., 2006). As Eq. 2.12 e Eq. 2.13 descrevem as FDC e FDP de uma varável aleatória exponencial com parâmetro λ, que é o inverso da média, conforme Eq. 2.14.

$$F_x(x) = \begin{cases} 1 - e^{-\lambda x}, & 0 \leq x \leq \infty \\ 0, & \text{caso contrário.} \end{cases} \tag{2.12}$$

$$f_x(x) = \lambda e^{-\lambda x}. \tag{2.13}$$

$$\overline{X} = \frac{1}{\lambda}. \tag{2.14}$$

Um aspecto importante da distribuição exponencial é sua propriedade sem memória. Ou seja, o tempo até o próximo evento independe do tempo decorrido desde o evento anterior (JAIN, 1992) e é descrito na Eq. 2.15.

$$P[X \leq t + u | x > u] = 1 - e^{-\lambda t} = P[X \leq t]. \tag{2.15}$$

A relação entre as variáveis aleatórias exponenciais e Poisson é bastante importante. Quando os intervalos entre chegadas são exponencialmente distribuídos e os tempos entre chegadas sucessivos são independentes com a média $\overline{X}$, o número de eventos de chegada em um intervalo fixo (com duração t) é determinado por uma distribuição de Poisson com o parâmetro (taxa de chegada) $\alpha = t\frac{1}{\overline{X}}$.

2.4.2 Teoria das filas

Em diversas situações é comum se utilizar filas de espera para regular o acesso a diferentes tipos de serviço ou recurso, como em estacionamentos para pagamento do *ticket*, na receita federal para ressarcimento do imposto e em hospitais para atendimento médico, por exemplo. A teoria de filas é um ramo da teoria da probabilidade aplicada que lida com a quantificação dessas situações (COOPER, 1981). Ela é aplicada

na análise de desempenho de sistemas em diferentes áreas, como redes de comunicação, sistemas de computadores, linhas de produção e assim por diante. Embora o termo teoria de filas seja frequentemente usado para descrever a teoria matemática da espera nas filas, ela também se aplica a modelos baseados em sistemas em que não há formação de filas. Um exemplo geral de sistema de enfileiramento pode ser descrito quando os clientes passam a usar um serviço fornecido por um ou mais servidores. Caso todos os servidores estejam ocupados, o cliente deve executar um fluxo predefinido, como esperar pelo serviço ou desistir. Caso contrário, o cliente será atendido e deixará o sistema após o atendimento. Sistemas como esses podem ser definidos em termos das seguintes características (JAIN, 1992):

- Processo de chegada: descreve a sequência de chegadas de clientes para o serviço. Geralmente, o processo de chegada de clientes é estocástico. Portanto, é necessário conhecer a distribuição de probabilidade que descreve os tempos entre chegadas sucessivas de clientes (GROSS, 2008). O processo de chegada mais comum é o Poisson, o que significa que o tempo entre chegadas são independentes e identicamente distribuídos por uma distribuição exponencial;
- Distribuição do tempo de serviço: especifica em quanto tempo é realizado o atendimento ao cliente. É usual supor que os tempos de serviço sejam variáveis aleatórias independentes e identicamente distribuídas. É comum descrever os tempos de serviço nos sistemas de filas através da distribuição exponencial;
- Número de servidores: especifica quantos servidores são considerados no sistema para atender os clientes;
- Capacidade do sistema: define o número máximo de clientes simultâneos no sistema, que pode ser limitado pela quantidade de recursos ou para evitar um tempo de espera maior que o aceitável, por exemplo. Nesta capacidade são considerados clientes que estão aguardando atendimento e os que já estão em atendimento. Na maioria dos sistemas, a capacidade do sistema é finita. Mas existem sistemas em que ela é muito grande e para facilitar sua análise, uma capacidade infinita é assumida (JAIN, 1992);
- Tamanho da população: é o número total de clientes em potencial que podem acessar o sistema. Em muitos casos reais, é finito. Mas há casos em que é considerado infinito;
- Disciplina de serviço: define a ordem em que os clientes são atendidos. A mais comum é o Primeiro a Chegar, Primeiro a ser Servido (FCFS), onde os clientes são atendidos por ordem de chegada. Existem também outras disciplinas, como Último a Chegar, Primeiro a ser Servido (LCFS), Serviço Em Ordem Aleatória

(SIRO), *Round Robin* (RR) e Prioridades estáticas, por exemplo. Este último seleciona os clientes para serem atendidos com base nas prioridades que são permanentemente atribuídas a eles. Além disso, uma disciplina de preempção pode ser usada em conjunto com LCFS ou as prioridades estáticas. Essa disciplina interrompe e antecipa o cliente que está sendo atendido no momento, se houver um cliente na fila com maior prioridade (BOLCH et al., 2006).

Para especificar um sistema de filas, é preciso definir essas seis características. Uma notação bastante empregada para isso é a de Kendall que usa símbolos e barras como A/S/m/N/K/SD para descrever o sistema de fila, onde A indica a distribuição de tempo entre chegadas, S é a distribuição do tempo de serviço, m é o número de servidores, N é a capacidade do sistema, K é o tamanho da população e SD é a disciplina de serviço. Para indicar que os horários entre chegadas e os tempos de serviço são exponenciais distribuídos, utiliza-se a letra M nesta notação, por exemplo. As letras associadas a outras distribuições podem ser encontradas em (JAIN, 1992). Além disso, acrônimos como FCFS, SIRO e LCFS, podem ser adotados para indicar a disciplina de serviço considerada em um sistema de filas.

Quando a capacidade do sistema é infinita, o tamanho da população é infinito ou a disciplina de serviço é FCFS, uma notação simplificada pode ser usada. Para esses casos, o símbolo que descreve a característica considerada infinita ou a disciplina FCFS é omitido na notação. Portanto, um sistema em fila M/M/1/∞/∞/FCFS pode ser representado como M/M/1, por exemplo.

2.4.2.1 Tipos de filas

Existem diversos tipos de modelos de filas na literatura. Serão apresentados dois que podem auxiliar na compreensão dos modelos de filas desenvolvidos no nesta monografia. Outros tipos de filas podem ser encontrados em (GROSS, 2008).

Uma fila M/M/1 é amplamente usada para modelar sistemas que possuem um único servidor. Nesta fila, os tempos entre chegadas e os tempos de serviço são exponencialmente distribuídos, sem limitação para o tamanho populacional e capacidade do sistema, e a disciplina de serviço adotada é FCFS (JAIN, 1992). Para analisar esse tipo de fila é importante destacar dois parâmetros, a taxa média de chegada de clientes λ e a taxa média de serviço μ. Uma fila M/M/1 é um processo de nascimento-morte (GROSS, 2008), em que as chegadas de clientes podem ser consideradas como 'nascimentos' para o sistema, partidas como 'mortes' e as taxas λ e μ são fixas. A Fig. 1 ilustra o diagrama de estados deste tipo de fila, onde cada estado i modela a existência de i usuários no sistema.

Considerando a condição de estabilidade $\lambda < \mu$, isto é, o número de usuários

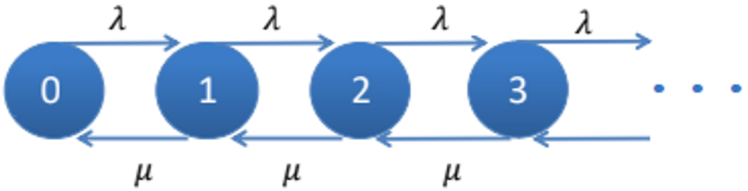

Figura 1 – Diagrama de transição de estado para uma fila M/M/1.

no sistema não crescerá indefinidamente, a probabilidade dos estados do sistema em estado estacionário (probabilidade de n clientes no sistema) pode ser obtidos por meio da resolução das equações de balanço de fluxo (fluxo de entrada igual fluxo de saída) e a condição de contorno $\Sigma p_i = 1$ (GROSS, 2008). Portanto, as seguintes equações (Eq. 2.16) podem ser obtidas no diagrama ilustrado na Fig. 1.

$$\begin{cases} (\lambda + \mu)p_n = \mu p_{n+1} + \lambda p_{n-1} \\ \lambda p_0 = \mu p_1. \end{cases} \tag{2.16}$$

2.4.2.2 Filas M/M/m/N

A fila M/M/m/N é um modelo com taxa média de chegada de clientes igual a λ por unidade de tempo, m servidores idênticos disponíveis no sistema, com taxa de serviço igual a μ clientes por unidade de tempo. Nesse sistema, se houver pelo menos um servidor ocioso, o cliente que chega é atendido imediatamente. Caso contrário, o cliente aguardará na fila para ser atendido. Como no sistema de filas anterior, na fila M/M/m/N, o tamanho da população não é limitado, e o estado do sistema é determinado pelo número de clientes no sistema (JAIN, 1991). Além disso, essa fila pode ser modelada como um processo de nascimento-morte, com $\lambda_n = \lambda$ para $n \geq 0$ e $\mu_n = n\mu$ (quando $0 \leq n \leq m-1$) onde n é o número de usuários no sistema. O número de clientes permitidos no sistema a qualquer momento é limitado a N (GROSS et al., 2008). Portanto, depois que o sistema estiver cheio, todas as chegadas de novos clientes serão bloqueadas. No diagrama de transição de estados, a taxa de chegada λ é 0 quando $n \geq N$, como mostrado na Fig. 2.

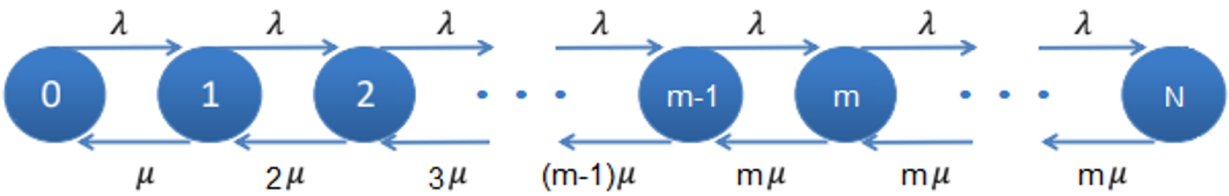

Figura 2 – Diagrama de transição de estado para uma fila M/M/m/N.

A probabilidade de ter n clientes no sistema (em regime estacionário) é dada pela Eq. 2.17, onde $\rho = \frac{\lambda}{m\mu}$ denota a intensidade do tráfego no sistema (JAIN, 1992).

$$p_n = \begin{cases} (1 + \frac{(1-\rho^{(N-m+1)})(m\rho^m)}{m!(1-\rho)} + \sum_{n=m}^{N} \frac{m\rho^n}{n})^{-1}, & n = 0 \; e \; \rho \neq 1 \\ \frac{m\rho^n}{n!} p_0, & 1 \leq n \leq m-1 \\ \frac{\rho^n m^m}{m!} p_0, & m \leq n \leq N. \end{cases} \tag{2.17}$$

2.4.2.3 Disciplinas prioritárias

Nos sistemas anteriores, foi utilizada a disciplina de serviço FCFS, em que os clientes são atendidos por ordem de chegada. Porém, existem sistemas de enfileiramento nos quais as prioridades são associadas aos clientes, e aqueles com maior prioridade são selecionados para serviço, independentemente do tempo de admissão no sistema(BOLCH et al., 2006).

Os sistemas de enfileiramento com prioridade podem ser classificados como preemptivos ou não preemptivos. Nos preemptivos, o cliente com a maior prioridade é atendido imediatamente, mesmo que outro com menor prioridade já esteja em atendimento. Em sistemas não preemptivos, não há interrupção, e o cliente de maior prioridade simplesmente é colocado ao início da fila para esperar o atendimento.

Além disso, em sistemas preemptivos, há outras decisões a serem tomadas. Como por exemplo se os clientes interrompidos durante a preempção podem retomar seus serviços ou se devem ser retirados do sistema. E caso os clientes possam retomar seu serviço, se o serviço continua a partir do ponto de preempção ou se é reiniciado.

2.4.3 Redes de Petri Coloridas

As Redes de Petri Coloridas (RPCs) fornecem uma estrutura adequada para modelar e simular sistemas concorrentes e distribuídos e outros sistemas de grande escala com comunicação assíncrona e síncrona. RPCs combinam os recursos de uma linguagem de programação de alto nível com a representação gráfica de redes de Petri básicas. As RPCs utilizam redes de Petri básicas para modelar características de comunicação, simultaneidade e sincronização, enquanto a linguagem de programação fornece os primitivos para a manipulação de valores de dados e a caracterização de tipos de dados. Os RPCs podem ser usados para modelagem de vários domínios de aplicação, incluindo protocolos de comunicação, por exemplo. Um modelo RPC é uma demonstração executável de um sistema, que inclui os estados do sistema e os eventos para alterar o estado do sistema para explorar o comportamento do sistema (SHAHIDINEJAD; GHOBAEI-ARANI; ESMAEILI, 2019).

Os elementos dos RPCs são lugares, transições, arcos e inscrições. Os lugares, transições e arcos são representados por elipses, retângulos e setas, respectivamente. Os locais representam o status do sistema, que pode conter vários conjuntos de *tokens* (conjunto de cores). O conjunto de *tokens* nos locais especifica o estado do sistema. As transições representam possíveis eventos que causam uma mudança no estado pela manipulação de *tokens* usando as regras de disparo. A distribuição dos *tokens* nos locais representa a marcação do sistema que determina se condições definidas foram atendidas para disparar uma transição. Os arcos indicam a relação entre lugares e transições e determinam como um estado muda quando um evento ocorre. As inscrições de arcos como funções utilizadas para especificar o número de *tokens* transferidos entre estados (SHAHIDINEJAD; GHOBAEI-ARANI; ESMAEILI, 2019).

Uma RPC é declarada como $RPC = (P, T, A, \Sigma, V, C, G, E, I)$ (SHAHIDINEJAD; GHOBAEI-ARANI; ESMAEILI, 2019), onde:

- P é um conjunto finito de lugares (P = $\{p_1, p_2, ..., p_n\}$).
- T é um conjunto finito de transições (T = $\{t_1, t_2, ..., t_m\}$), tal que $P \cap T = \varnothing$
- $A \subseteq P \times T \cup T \times P$ é um conjunto finito de arcos direcionados.
- Σ é um conjunto finito de conjuntos de cores não vazios direcionados.
- V é um conjunto finito de variáveis, de modo que o $Tipo[v] \in \Sigma$ para todas as variáveis $v \in V$
- $C : P \to \Sigma$ é uma função de conjunto de cores que atribui um conjunto de cores a cada local.
- $G : T \to EXPR_V$ é uma função de guarda que atribui uma guarda a cada transição t tal que $Tipo[G(t)] = Bool$.
- $E : A \to EXPR_V$ é uma função de expressão de arco que atribui uma expressão de arco a cada arco a tal que $Tipo[E(a)] = C(p)_{MS}$, onde p é o local conectado ao arco a.
- $I : P \to EXPR_\varnothing$ é uma função de inicialização que atribui uma expressão de inicialização a cada lugar p de forma que $Type[I(p)] = C(p)_{MS}$.

3 Sistema 5G URLLC auto escalável considerando falha e Pré-setup

Este capítulo apresenta um modelo analítico que permite mensurar os limites de uma rede 5G URLLC baseada em NFV-MEC, com VNFs implementadas através de contêineres. Estes, por sua vez, são escalados sob demanda. Para isso, o tempo de provisionamento da VNF (inicialização dos contêineres), a possibilidade de falha durante o atendimento e a configuração prévia dos recursos são considerados conforme a Fig. 3.

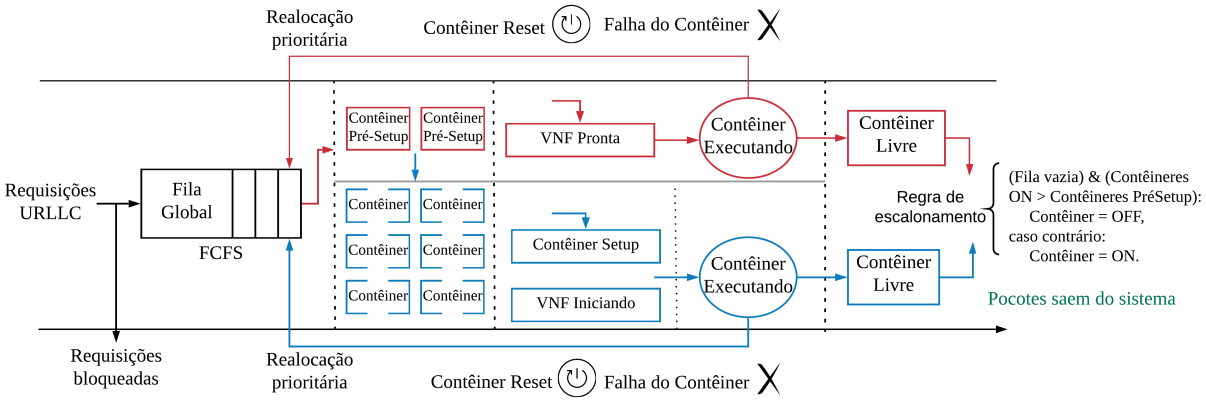

Figura 3 – Sistema URLLC NFV-MEC.

O sistema é composto de c contêineres, dos quais n são ligados previamente ($n < c$), e possui um limite máximo de k serviços URLLC simultâneos. A cada chegada de um novo serviço, caso o limite k não tenha sido excedido, a solicitação é admitida no sistema e um contêiner é alocado para o atendimento da demanda (caso haja container ativo e disponível) e um outro contêiner é ligado (caso haja algum desligado) em seguida visando manter o número de recursos ativos e disponíveis no sistema iguais a n. Caso todos os contêineres já estejam ocupados, os serviços serão colocados em um buffer finito de tamanho q, com $q = k-c$. Durante o atendimento do serviço, o contêiner está suscetível a ocorrência de falhas. Nesse caso, ele é reiniciado e o serviço é realocado para um contêiner disponível. Caso não haja disponibilidade, o serviço é recolocado no buffer tendo maior prioridade no atendimento em relação aos novos serviços. Em ambos os casos, o processamento do serviço é reiniciado.

A Figura 4 apresenta os detalhes da operação de um exemplo com $n = 1$, $c = 3$ e $k = 3$. O primeiro evento em t1, é uma solicitação de serviço regular, alocada para atendimento no recurso ligado antecipadamente (CTNR1), enquanto paralelamente é

ligado outro contêiner, visando manter o sistema com um recurso disponível caso outra solicitação chegue, essa configuração requer um período de espera até que o recurso esteja pronto em t2. Em t3, chega uma nova solicitação de serviço, alocada para o CTNR2 que já está disponível, enquanto o CTNR3 é ligado previamente, ficando disponível para atendimento apenas em t4. Em t5, uma falha durante o atendimento força o CNTR2 a reiniciar e mover o serviço atual para outro recurso disponível (CTNR3). Em t6, o CTNR2 volta a operação e permanece ligado aguardando que outro serviço chegue para atendimento no sistema. Após terminar o processamento das requisições em t7 e t8, os recursos (CTNR 1 e CTNR3) são desativados, pois o número de contêineres livres ligados previamente já foi atingido com o CTNR2.

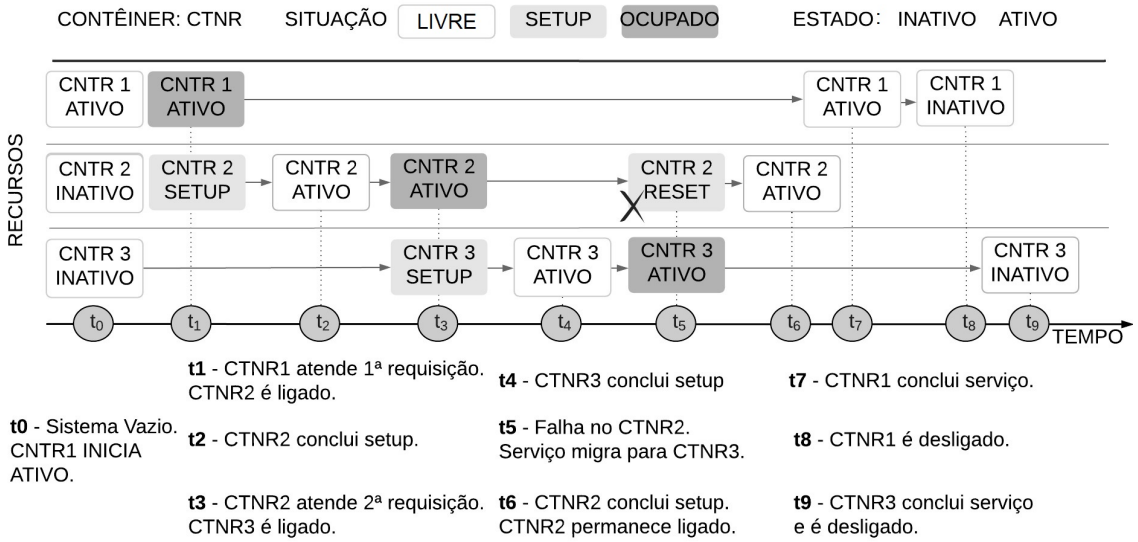

Figura 4 – Exemplo de operação para n = 1, c = 3 e k = 3.

3.1 Modelo Analítico

O modelo proposto adota uma fila M/M/c/k/ com tempo de inicialização (*setup time*), falha de contêineres em atendimento, n contêineres pré-inicializados e disciplina de serviço FCFS regular para a categoria de serviço URLLC. Os estados do modelo são representados pelo par $(i\ e\ j)$, com $i, j \in \mathbb{N}$, onde i denota o número de contêineres ligados e j o número de serviços no sistema. A chegada de serviços obedece um processo de Poisson com taxa λ. O atendimento dos serviços é realizado pelos c contêineres idênticos disponíveis no sistema, com tempo de serviço exponencialmente distribuído com taxa μ. Similarmente, o tempo entre falhas e o tempo de inicialização dos contêineres seguem distribuições exponenciais com taxas γ e α, respectivamente. A Fig. 5 apresenta o diagrama de espaço de estados do modelo, em que as transições para a direita (esquerda) indicam a chegada de novos serviços (o termino de um serviço sem desligar um contêiner). As transições verticais para baixo (cima) indicam a

inicialização de um novo contêiner (a falha do contêiner durante o atendimento). Por fim, as transições na diagonal representam o fim do serviço com o desligamento de um contêiner ocioso conforme a política de pré-configuração utilizada.

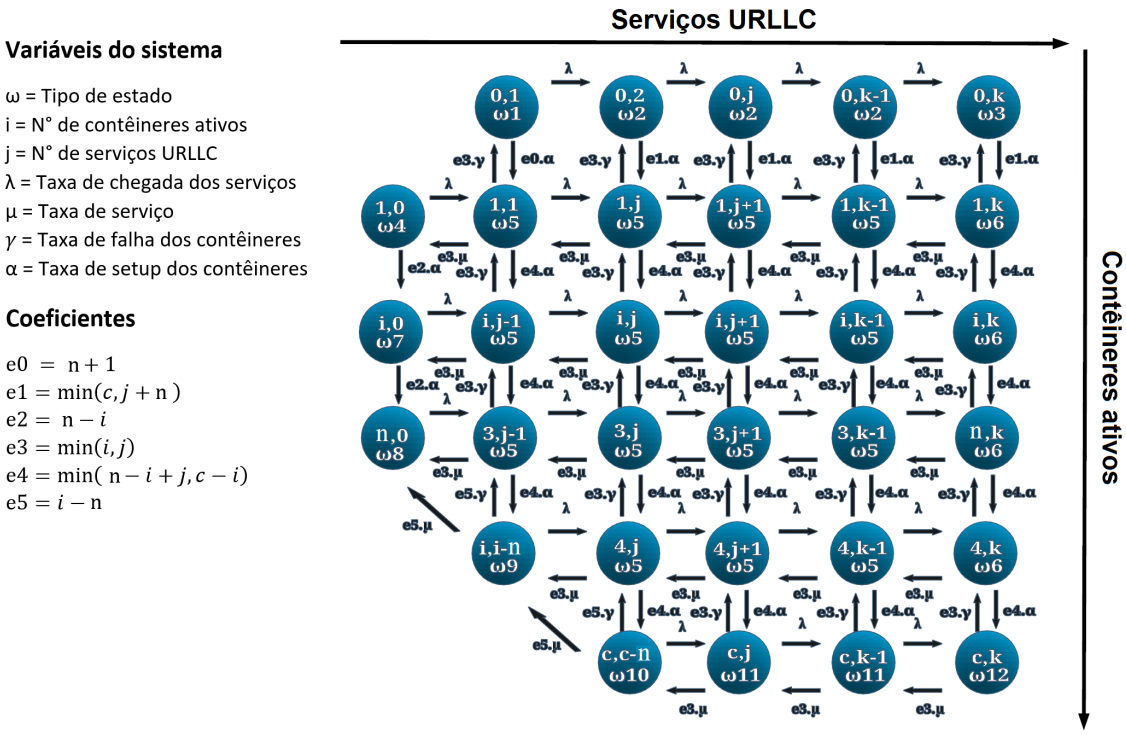

Figura 5 – Diagrama de transição de estado.

O espaço de estado possível é dado por $\Omega = \{(i,j|0 < i+j, 0 \leq i \leq c, 0 \leq j \leq k,$ desde que $i-n \leq j, c \leq k, c \geq 1, k \geq c$ e $n \geq 1\}$. Para derivar métricas de desempenho do sistema, a probabilidade de estados com o sistema em regime estacionário $\pi(i,j)$ precisa ser encontrada, o que pode ser feito através da resolução do sistema linear formado pela 3.1 condição de normalização e (3.2-3.14) as equações de balanço de fluxo.

$$\Sigma_{(i,j)\in\Omega}\pi(i,j) = 1. \tag{3.1}$$

Para melhor entendimento, o espaço de estado foi dividido em 12 subconjuntos que englobam as possíveis situações que o sistema pode apresentam. Assim, os estados do tipo $\omega_s \in \Omega$ com S de 1 a 12, possuem equações de fluxo similares.

O estado do sistema em que não há contêineres ligados, pois todos os contêineres ligados previamente falharam, e existe apenas um serviço a ser atendido é do tipo ω_1, $(i = 0, j = 1)$, e tem equação de balanço dado por 3.2. Já os estados do tipo ω_2 $(i = 0, 1 < j < k)$ modelam a formação de fila no sistema com mais de um serviço a ser atendido e nenhum contêiner ligado. Esse, esse tipo de estado possui equação de fluxo 3.3. Seguindo, o estado da extrema direita ω_3, em que $i = 0$ e $j = k$, representa

a situação em que o sistema atingiu o limite de serviços que podem ser admitidos com nenhum contêiner ligado, para esse estado a equação de balanço 3.4 é utilizada.

$$(\lambda + (n+1)\alpha)\pi_{0,1} = (\gamma\pi)_{1,1}. \tag{3.2}$$

$$(\lambda + (min\ (c, j+n)\alpha))\pi_{0,j} = \lambda\pi_{0,j-1} + \gamma\pi_{1,j}. \tag{3.3}$$

$$(min\ (c, k+n)\alpha)\pi_{0,k} = \lambda\pi_{0,k-1} + \gamma\pi_{1,k}. \tag{3.4}$$

O estado do tipo ω_4 $(i = 1, j = 0)$ denota a situação que o sistema possui apenas um contêiner ligado previamente e não existe nenhum serviço para atendimento. Este estado somente é possível quando $(n > 1)$ e é expresso pela equação de balanço 3.5. Os estados ω_5 denotam alguns containers ativos e serviços URLLC em atendimento, ou seja, $0 < i < c$ e $(mx.(1, i–n)) < j < k$. Eles são os estados internos do modelo e seguem a equação 3.6. Já os estados do tipo ω_6 representam o sistema cheio e com o número de contêineres ligados menor que c $(0 < i < c, j = k)$. Nessas condições, novas admissões de serviço não são possíveis. A equação 3.7 descreve esses estados. De forma análoga ao estado ω_4, os estados ω_7 representam a recuperação do sistema após sucessivas falhas, com a ligação previa de contêineres (para $n > 1$) e sem serviços admitidos ($0 < i < n, j = 0$), para esse tipo de estado se aplica a equação 3.8.

$$(\lambda + (n-i)\alpha)\pi_{1,0} = \mu\pi_{1,1}. \tag{3.5}$$

$$\begin{aligned}(\lambda + (min(i,j)\mu) + (min\ (n-i+j, c-i)\alpha) + (min\ (i,j)\gamma)\pi_{i,j} = \lambda\pi_{i,j-1} \\ +(min\ (i, j+1))\mu\pi_{i,j+1} + min\ (n-i+1+j, c-i+1)\alpha\pi_{i-1,j} + (min\ (i+1, j)\gamma)\pi_{i+1,j}.\end{aligned} \tag{3.6}$$

$$\begin{aligned}((min\ (i,k)\mu) + (min\ (n-i+k, c-i)\alpha) + (min\ (i,k)\gamma)\pi_{i,k} = \lambda\pi_{i,k-1} + \\ (min\ (n-i+1+k, c-i+1)\alpha)\pi_{i-1,k} + (min\ (i+1, k)\gamma)\pi_{i+1,k}.\end{aligned} \tag{3.7}$$

$$(\lambda + (n-i)\alpha)\pi_{i,0} = \mu\pi_{i,1} + ((n-i+1)\alpha)\pi_{i-1,0}. \tag{3.8}$$

O estado ω_8 representa o estado inicial do sistema com n contêineres ligados previamente e nenhum serviço no sistema ($i = n, j = 0, n > 1$). Para ele se aplica a

equação 3.9. Quando $n = 1$, o estado ω_8 assume a equação 3.10. Os estados da diagonal ω_9 modelam situações em que há serviços em atendimento e com n contêineres ligados e disponíveis para atendimento ($n < i < c, j = i–n,$ com $c–n > 1$). Quando o sistema está nesse tipo de estado, a conclusão de um serviço conduz ao desligamento de um contêiner, mantendo apenas n contêineres ligados previamente. Esse tipo de estado é expresso pela equação 3.11.

$$\lambda\pi_{n,0} = \mu\pi_{n,1} + \mu\pi_{n+1,1} + \alpha\pi_{n-1,0}. \tag{3.9}$$

$$\lambda\pi_{n,0} = \mu\pi_{n,1} + \mu\pi_{n+1,1}. \tag{3.10}$$

$$\begin{aligned}(\lambda + (i–n)\mu + (i–n)\gamma)\pi_{i,i-n} = (i–n + min\ (n,1)\mu)\pi_{i,i-n+1} \\ +((i–n+1)\mu)\pi_{i+1,i-n+1} + \alpha\pi_{i-1,i-n}.\end{aligned} \tag{3.11}$$

Similar aos estados do tipo ω_9, o estado ω_{10} modela o sistema quando há serviços em atendimento, c contêineres ligados com n livres para atendimento ($i = c, j = c–n$). Assim, caso novos serviços sejam admitidos, eles serão imediatamente atendidos pelos n contêineres ligados antecipadamente e enfileirados assim que esse número se esgotar, sem que um contêiner seja ligado. Para este estado se aplica a equação 3.12. Já a situação em que o sistema apresenta todos os contêineres ligados ($i = c$), com novos serviços sendo atendidos pelos contêineres ligados previamente ($c - n < j \leq c$) ou enfileirados, mas não alcançando o limite máximo do sistema ($c < j < k$), é descrita pelos os estados do tipo ω_{11} ($i = c, (c–n) < j < k, n > 1$), que possuem a equação de balanço 3.13. Por fim, o estado ω_{12} retrata o sistema na sua capacidade total de serviços atingida, com todos os contêineres processando serviços. Este estado segue a equação 3.14.

$$\begin{aligned}(\lambda + (c–n)\mu + (c–n)\gamma)\pi_{c,c-n} = ((c–n + min\ (n,1))\mu)\pi_{c,c-n+1} \\ +\alpha\pi_{c-1,c-n}.\end{aligned} \tag{3.12}$$

$$\begin{aligned}(\lambda + min\ (c,j)\mu + (min\ (c,j)\gamma))\pi_{c,j} = \lambda\pi_{c,j-1} \\ +(min\ (c,j+1)\mu)\pi_{c,j+1} + \alpha\pi_{c-1,j}.\end{aligned} \tag{3.13}$$

$$(min\ (c,k)\mu + min\ (c,k)\gamma)\pi_{c,k} = \lambda\pi_{c,k-1} + \alpha\pi_{c-1,k}. \tag{3.14}$$

Calculada a probabilidade de estados com o sistema em regime estacionário $\pi(i,j)$, é possível obter a probabilidade de bloqueio PB utilizando a equação 3.15, e o número médio de serviços no sistema MS aplicando a equação 3.16. O tempo de resposta médio do sistema MRT pode ser obtido utilizando as métricas PB e MS na equação 3.17. Uma métrica muito importante do ponto de vista energético é o número de contêineres ligados no sistema nCTNs, essa métrica pode ser obtida a partir da equação 3.18.

$$PB = \sum_{i=0}^{c} \pi(i,k). \tag{3.15}$$

$$MS = \sum_{i=0}^{n} \sum_{j=1}^{k} \pi(i,j)j + \sum_{i=n+1}^{c} \sum_{j=i-n}^{k} \pi(i,j)j. \tag{3.16}$$

$$MRT = \frac{MS}{\lambda(1-PB)}. \tag{3.17}$$

$$nCTNs = \sum_{i=1}^{n} \sum_{j=0}^{k} \pi(i,j)i + \sum_{i=n+1}^{c} \sum_{j=i-n}^{k} \pi(i,j)i. \tag{3.18}$$

3.2 Modelo de simulação

Simulação de eventos discretos baseada em Rede de Petri colorida foi utilizada na validação do modelo (DORDA, 2010). Para isso, o modo de simulação da ferramenta CPN tools foi empregado. O simulador implementa as funcionalidades de chegada de requisição, controle de acesso ao sistema, fila de atendimento, falha durante o processamento do serviço, escalonamento automático e pré-inicialização de contêineres. Os tempos de ocorrência dos eventos são definidos durante a simulação e podem seguir distribuições de probabilidade ou algum outro critério definido pelo usuário. A escala de tempo padrão adotada no simulador é de um microssegundo, mas outras são possíveis de serem utilizadas. A Figura 6 ilustra o simulador representado em RPC através do *software* CPN Tools versão 4.0.1 (TOOLS, 2018). Ele possui 22 lugares, 16 transições, 58 arcos e 12 cores. Para facilitar o entendimento, o modelo do simulador foi dividido em três partes: Chegada de Usuários (Serviços), Gerência dos Contêineres e Atendimento e Falha do Serviço. Essas partes são descritas nas sessões seguintes.

3.2.1 Chegada de Usuários (Serviços)

A Figura 7 ilustra a parte do simulador responsável por modelar o processo de chegada das requisições (usuários) no sistema. Primeiramente, uma requisição é

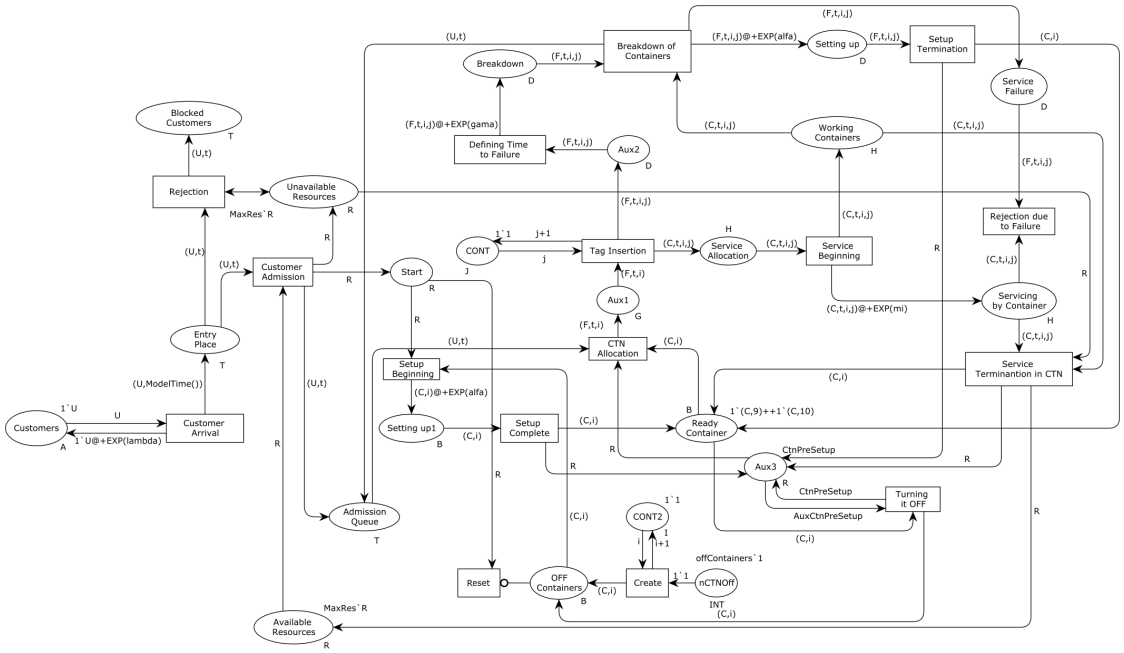

Figura 6 – Simulador em RPC.

gerada através do disparo da transição ***Customer Arrival***, que posiciona um *token* **U** (que representa o usuário/requisição) no lugar ***Entry Place***. Essa transição é habilitada quando há algum *token* **U** no lugar ***Customers***, consumindo um deles no seu disparo. Quando isso acontece, além do ***Entry Place***, essa transição insere um *token* **U** no lugar ***Customers***. Entretanto, este último só recebe o *token* após um intervalo de tempo exponencialmente distribuído com uma dada média $1/\lambda$ configurada pelo usuário do simulador. Essa retroalimentação do lugar ***Customers*** permite que a chegada de usuários (serviços) para atendimento ocorra continuamente durante a simulação seguindo os intervalos definidos. Além da exponencial, outras distribuições e formas de definição dos intervalos entre as chegadas são suportadas pelo simulador, tais como o uso de intervalo fixo, por exemplo. É interessante ressaltar que o lugar ***Customers*** é inicializado com um *token* **U**. Para que seja possível calcular o tempo de resposta das requisições/serviços, faz necessário registrar o instante em que os usuários (serviços) chegam ao sistema. Assim, ao chegar ao lugar Entry Place, o *token* U é associado com o instante dessa ocorrência, obtido através da função ***ModelTime*()**.

O controle de admissão do sistema é realizado de acordo com a quantidade de *tokens* do tipo **R** (que representa os recursos) presentes nos lugares ***Unavailable Resources*** e ***Available Resources***. Esta quantidade indica a quantidade de recursos que estão indisponíveis (já ocupados) e disponíveis (para alocação), respectivamente. Caso haja ***MaxRes*** *tokens* em ***Unavailable Resources***, isso implica que não há mais espaço no sistema para novos usuários. Assim, um *token* em ***Entry Place*** é movido para ***Blocked Customers*** através do disparo da transição ***Rejection***, representando que o usuário foi bloqueado, não admitido no sistema. A variável ***MaxRes*** representa

a quantidade máxima de usuários que o sistema suporta simultaneamente (seja em atendimento ou em fila para atendimento). Por outro lado, quando há algum *token* em ***Available Resources*** e em ***Entry Place***, a transição ***Customer Admission*** é disparada, indicando a admissão do usuário no sistema. Ela consome um *token* de cada um desses lugares e posiciona um *token* de requisição/usuário **(U,t)** em ***Admission Queue*** e um de recurso **R** em ***Unavailable Resources*** e ***Start***. Este último pode ocasionar a inicialização de um contêiner, tratada no módulo Gerencia dos contêineres.

Os lugares ***Unavailable Resources*** e ***Available Resources*** são complementares no controle de admissão, ou seja, a soma dos *tokens* presentes neles é sempre igual à ***MaxRes***. A admissão de um usuário/requisição retira um *token* de recurso de ***Available Resources*** e adiciona um em ***Unavailable Resources***. Ao passo que o termino de serviço realiza o oposto. Os lugares ***Available Resources*** e ***Unavailable Resources*** são inicializados com ***MaxRes*** e nenhum token, respectivamente.

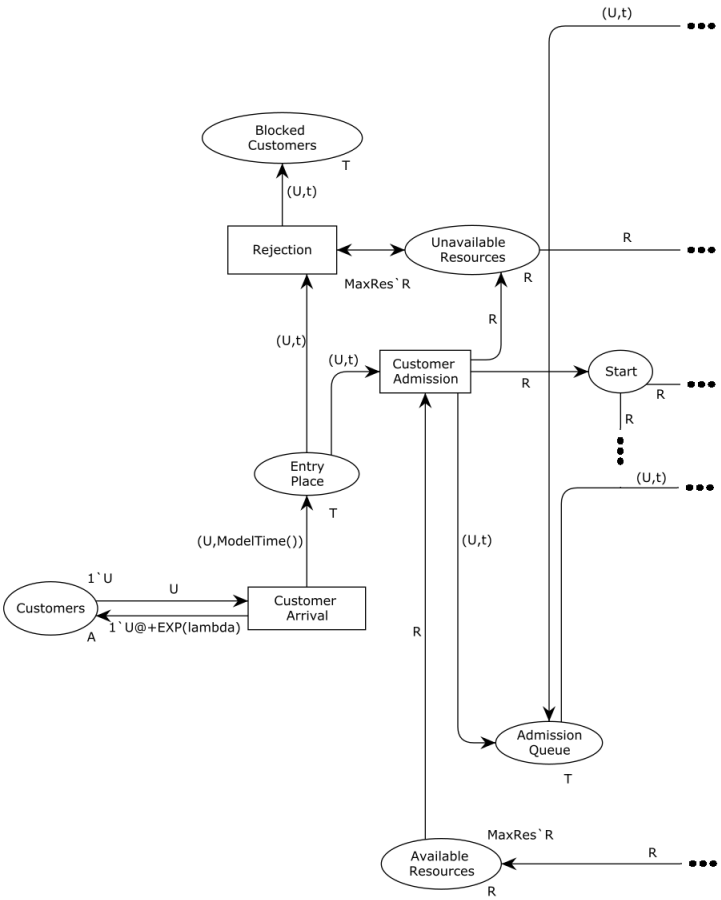

Figura 7 – Módulo de chegada das requisições.

3.2.2 Gerência dos Contêineres

A admissão do usuário (disparo da transição ***Customer Admission***) insere um

token de recurso em **Start**, que possibilitará a inicialização imediata de um novo contêiner para suprir o contêiner que atenderá o usuário admitido, mantendo o número de contêineres pré-inicializados de acordo com o configurado para o sistema. Caso não haja contêineres desligados, ou seja, *tokens* em ***OFF Conteiners***, a transição ***Reset*** será disparada e consumirá os *tokens* presentes em Start para que eles não se acumulem e gerem inconsistências no simulador. Dado que existem contêineres desligados, *tokens* em ***OFF Conteiners***, a inicialização de um deles é realizada, através do disparo da transição ***Setup Beginning***, que insere um *token* **(C,i)** (que representa o contêiner i) no lugar ***Setting up1***, após um tempo exponencialmente distribuído com média $1/\alpha$.

A criação dos contêineres desligados é realizada utilizando a transição ***Create*** e *tokens* dos lugares ***CONT2*** e ***nCTNOff***. Os lugares ***CONT2*** e ***nCTNOff*** são inicializados com um e ***offContainers*** *tokens* com valor igual a **'1**, respectivamente, sendo ***offContainers*** o número total de contêineres do sistema subtraído da quantidade que devem ficar ligados previamente. O lugar ***nCTNOff*** não é realimentado após se esgotar. O disparo da transição ***Create*** consome um *token* de ***nCTNOff*** e ***CONT2*** e associa um identificador **i** ao *token* **(C,i)**, que será inserido no lugar ***OFFContainers***. Este último lugar modela o *pool* de containers disponíveis para inicialização. Além disso, a transição ***Create*** insere um novo *token* no lugar ***CONT2***, cujo valor é o incremento em uma unidade do anterior, ou seja, **i+1**. Deste modo, ***CONT2*** e ***Create*** codificam um contador.

Com um *token* (contêiner) no lugar ***Setting up1***, a transição ***Setup Complete*** é habilitada e o seu disparo insere um *token* de contêiner **(C,i)** no lugar ***Ready Container***. Este, por sua vez, desempenha o papel de *buffer* de contêineres disponíveis para atendimento imediato de requisições. O lugar ***Ready Container*** é inicializado com quantidade de *tokens* que representa número de contêineres que devem ficar ligados previamente. Sendo um *buffer* de containers já inicializados e disponíveis para atendimento, quando o término de um atendimento acontece, o container correspondente pode ser posicionado neste *buffer*. Desta forma, **R*eady Container*** também recebe *tokens* gerados pelas transições ***Service Termination in CTN***, que libera um contêiner após o atendimento, e ***Setup Termination***, que reinicia um contêiner após uma falha.

O desligamento de um contêiner só é possível quando o número de contêineres ligados e disponíveis para atendimento ultrapassa o número de contêineres configurados para ficarem ligados previamente, denotado aqui como **n**. Para codificar essa política no simulador, o lugar **Aux3** foi criado. Ele recebe e tem *tokens* consumidos pelas mesmas transições que o lugar ***Ready Container***, exceto um arco da transição ***Turning it OFF***. Com isso os lugares ***Aux3*** e ***Ready Container*** sempre possuem a mesma quantidade de *tokens*. A transição ***Turning it OFF*** consome um *token* de contêiner **(C,i)** do lugar ***Ready Container*** e ***n+1*** *tokens* do lugar ***Aux3*** através da va-

riável ***AuxCtnPreSetup***. Esta transição insere um *token* de contêiner **(C,i)** no lugar ***offContainers*** e retroalimenta o lugar ***Aux3*** com ***n*** *tokens*, desligando assim apenas os contêineres excedentes.

Com os contêineres prontos para atendimento, representado pelos *tokens* em ***Ready Container***, as requisições de serviço que aguardam no *buffer* de admissão, modelado através do lugar ***Admission Queue***, são associadas a um contêiner disponível por meio de seu ID. Essa alocação é representada pelo disparo da transição ***CTN Allocation***, que consome um *token* de cada um desses dois lugares (e do ***Aux3*** também) e insere um *token* de alocação **(F,t,i)** no lugar ***Aux1***.

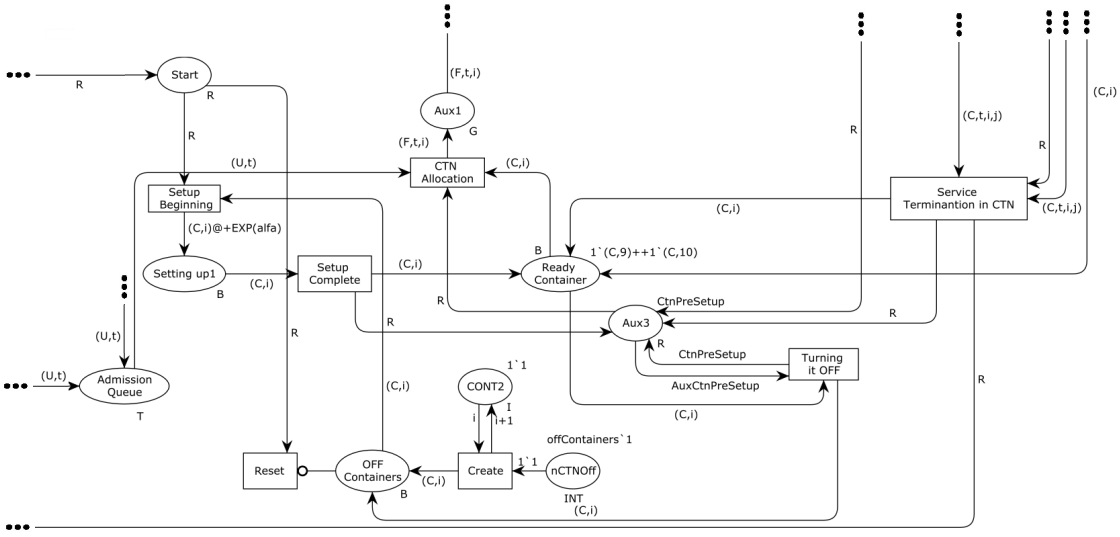

Figura 8 – Módulo de Gerência dos Contêineres.

3.2.3 Atendimento e falha de Serviço

A Figura 9 ilustra a parte do simulador responsável por modelar o atendimento e a eventual ocorrência de falha durante o atendimento. Após a chegada do *token* de alocação **(F,t,i)** no lugar ***Aux1***, a transição ***Tag Insertion*** é habilitada, essa transição adiciona uma marcação de falha **j** ao *token* de alocação. O valor **j** é dado pelo valor do *token* que a transição consome do lugar ***CONT***, utilizado para implementar um contador. Essa marcação de falha (identificador **j**) permite saber durante qual atendimento de requisição/serviço o container falhou. Um container pode atender diversos serviços ao longo da simulação. Após a inserção da marcação de falha, o *token* gerado **(F,t,i,j)** é posicionado em ***Aux2*** e um *token* de alocação com a marcação de falha **(C,t,i,j)** é inserido em ***Service Allocation***. Com o *token* em ***Aux2***, a transição ***Defining Time to Failure*** é disparada, consumindo um *token* de ***Aux2*** e adicionando um em ***Breakdown*** após um tempo exponencialmente distribuído com média $1/\gamma$, definida pelo usuário do simulador.

Paralelamente a inserção do *token* no lugar **Aux2**, há a inserção do *token* **(C,t,i,j)** no lugar ***Service Allocation***, o que habilita a transição ***Service Beginning***. Esta representa o início do atendimento do serviço, posicionando um *token* **(C,t,i,j)** no lugar ***Working Containers*** e outro em ***Servicing by Container*** após um tempo com distribuição exponencial e média $1/\gamma$. O lugar ***Working Containers*** representa o *pool* de contêineres que estão processando serviços (requisições) no momento e, consequentemente, podem apresentar falhas. Desta forma, os *tokens* posicionados nele podem habilitar tanto a transição que representa a falha em container (***Breakdown of Containers***) quanto a que indica o término de atendimento com sucesso no container (***Service Termination in CTN***). O que determinará qual das transições será habilitada será o instante em que os *tokens* ficarão disponíveis nos lugares ***Breakdown*** e ***Servicing by Container***, que são exponencialmente distribuídos e modelam os tempos de falha e serviço, respectivamente.

Caso o *token* esteja disponível no lugar ***Servicing by Container*** primeiro, a transição ***Service Termination in CTN*** é disparada, indicando o término do atendimento da requisição, consumindo *tokens* **(C,t,i,j)** dos lugares ***Servicing by Container*** e ***Working Containers*** e adicionando *tokens* de recurso **R** em ***Aux3***, ***Available Resources*** e ***Ready Containers***. Caso contrário, a transição ***Breakdown of Containers*** é disparada, consumindo os *tokens* de alocação **(C,t,i,j)** e falha **(F,t,i,j)** dos lugares ***Working Containers*** e ***Breakdown***, respectivamente, enviando de volta a requisição para a fila de atendimento através da inserção de *token* em ***Admission Queue*** e inserindo um *token* de falha **(F,t,i,j)** em ***Service Failure*** (de forma imediata) e em ***Setting up*** após um tempo com distribuição exponencial e média $1/\alpha$, que denota o processo de reinicialização do container. Deste modo, o lugar ***Setting up*** modela o *pool* de contêineres que falharam durante o atendimento e foram reinicializados. Com o *token* **(F,t,i,j)** disponível em ***Setting up***, a transição ***Setup Termination*** é disparada, denotando que o processo de reinicialização do contêiner finalizou e ele está pronto para realizar processamento. Ela insere um *token* **(C,i)** no lugar ***Ready Container***.

3.2.4 Métricas de Desempenho

Como no modelo analítico, métricas de desempenho e indicadores tais como número médio de contêineres processando serviço, número de usuário admitidos no sistema, número médio de usuários no sistema, tempo médio de resposta e probabilidade de bloqueio de usuário são possíveis de serem obtidas através do simulador. Para prover tais indicadores de desempenho foram utilizadas funções de monitoramento que verificaram o número médio de *tokens* presentes nos lugares durante a simulação, a quantidade de ocorrência de transições, funções de capturas de tempo e coletores de dados. Tais funções são disponíveis ou implementadas no CPN tools.

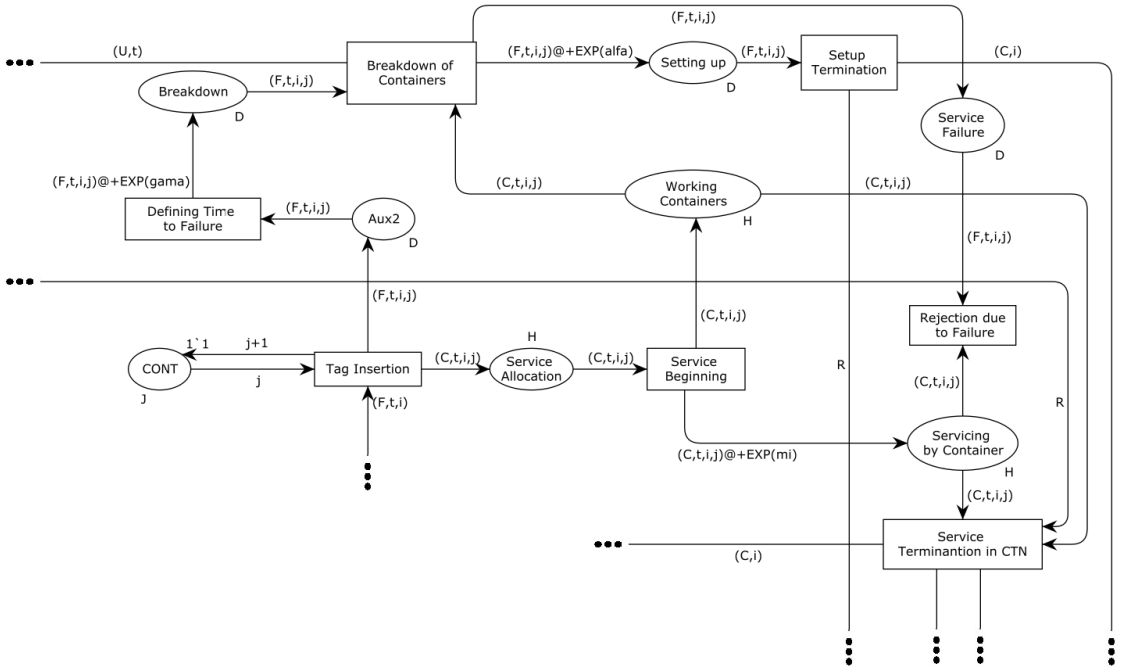

Figura 9 – Módulo de Atendimento e Falha de Serviços.

Para a obtenção da probabilidade de bloqueio (PB), adotou-se a razão entre o número de ocorrências da transição ***Rejection*** e da transição ***Customer Arrival***, denotada na Eq. 3.19.

$$PB = \frac{Rejection}{Costomer\ Arrival} \tag{3.19}$$

A métrica Tempo de Resposta Médio (MRT) foi obtida a partir das médias dos resultados das subtrações entre o instante de ocorrência da transição **Service Terminantion in CTN** e a chegada de cada *token* de requisição correspondente, conforme denotado na Eq. 3.20.

$$PB = ModelTime() - t \tag{3.20}$$

Por fim, a métrica número médio de contêineres ativos (processando serviço ou apenas ligado), indicada por nCTNs, foi obtida através da soma do número médio de *tokens* dos lugares ***Ready Container*** e ***Working Containers***.

3.2.5 Descrição formal da RPC

Nesta seção, é formalmente descrita a RPC que descreve o simulador da Seção 3.2, ela apresenta os seguintes elementos:

$$P = \left\{ \begin{array}{l} Customers,\ Blocked\ Customers,\ Entry\ Place,\ Unavailable\ Resources, \\ Available\ Resources,\ Admission\ Queue,\ Start,\ Setting\ up1,\ CONT, \\ Breakdown,\ OFF\ Containers,\ CONT2,\ Aux1,\ Aux2,\ nCTNOff,\ Ready \\ Container,\ Service\ Allocation,\ Aux3,\ Working\ Containers,\ Setting\ up, \\ Servicing\ by\ Container,\ Service\ Failure \end{array} \right\}$$

$$T = \left\{ \begin{array}{l} Customer\ Arrival,\ Rejection,\ Customer\ Admission,\ Setup\ Beginning, \\ Reset,\ Setup\ Complete,\ Defining\ Time\ to\ Failure,\ Create,\ CTN \\ Allocation,\ Tag\ Insertion,\ Breakdown\ of\ Containers,\ Turning\ it\ OFF, \\ Service\ Beginning,\ Setup\ Termination,\ Rejection\ due\ to\ Failure,\ Service \\ Terminantion\ in\ CTN \end{array} \right\}$$

$$\Sigma = \{I,\ J,\ R,\ TOA,\ C,\ F,\ G,\ A,\ T,\ B,\ D,\ H\}$$

$$V = \{i, j, t\}$$

$$G(t) = \{true\ para\ todo\ t \in T\}$$

$$I(p) = \left\{ \begin{array}{ll} 1`U & se\ p \in \{Customers\}, \\ MaxRes`R & se\ p \in \{Available\ Resources\}, \\ 1`1 & se\ p \in \{CONT, CONT2\}, \\ offContainers`1 & se\ p \in \{nCTNOff\} \end{array} \right\}$$

$$C(p) = \left\{ \begin{array}{l} R\ se\ p \in \{Unavailable\ Resources,\ Start,\ Aux3, Available\ Resources\}, \\ T\ se\ p \in \{Blocked\ Customers,\ Entry\ Place,\ Admission\ Queue\}, \\ A\ se\ p \in \{Customers\}, \\ B\ se\ p \in \{Setting\ up1,\ Ready\ Container,\ OFF\ Containers\}, \\ I\ se\ p \in \{CONT2,\ nCTNOff\},\ G\ se\ p \in \{Aux1\}, \\ J\ se\ p \in \{CONT\}, \\ D\ se\ p \in \{Aux2,\ Breakdown,\ Setting\ up,\ Service\ Failure\} \\ H\ se\ p \in \left\{ \begin{array}{l} Service\ Allocation,\ Working\ Containers, \\ Servicing\ by\ Container \end{array} \right\} \end{array} \right\}$$

$$
E(a) = \begin{cases}
(C,i)\ se\ a \in \begin{Bmatrix} (Ready\ Container \rightarrow CTN\ Allocation), (Ready\ Container \rightarrow Turning\ it\ OFF), (Service\ Terminantion\ in\ CTN \rightarrow Ready\ Container), \\ (Setting\ up1 \rightarrow Setup\ Complete), (Setup\ Complete \rightarrow Ready\ Container), (Setup\ Termination \rightarrow Ready\ Container), (Turning\ it\ OFF \rightarrow OFF\ Containers) \end{Bmatrix}, \\
(C,i)@ + EXP(alfa)\ se\ a \in \{(Setup\ Beginning \rightarrow Setting\ up1)\}, \\
(C,t,i,j)\ se\ a \in \begin{Bmatrix} (Service\ Allocation \rightarrow Service\ Beginning), (Service\ Beginning \rightarrow Working\ Containers), (Servicing\ by\ Container \rightarrow Rejection\ due\ to\ Failure), (Servicing\ by\ Container \rightarrow Service\ Terminantion\ in\ CTN), (Tag\ Insertion \rightarrow Service\ Allocation), (Working\ Containers \rightarrow Breakdown\ of\ Containers), (Working\ Containers \rightarrow Service\ Terminantion\ in\ CTN) \end{Bmatrix}, \\
(C,t,i,j)@ + EXP(mi)\ se\ a \in \left\{\begin{pmatrix} Service\ Beginning \rightarrow Servicing \\ by\ Container \end{pmatrix}\right\}, \\
(F,t,i)\ se\ a \in \{(Aux1 \rightarrow Tag\ Insertion), (CTN\ Allocation \rightarrow Aux1)\}, \\
(F,t,i,j)\ se\ a \in \begin{Bmatrix} (Aux2 \rightarrow Defining\ Time\ to\ Failure), (Breakdown \rightarrow Breakdown\ of\ Containers), (Breakdown\ of\ Containers \rightarrow Service\ Failure), (Service\ Failure \rightarrow Rejection\ due\ to\ Failure), (Setting\ up \rightarrow Setup\ Termination), (Tag\ Insertion \rightarrow Aux2) \end{Bmatrix}, \\
(F,t,i,j)@ + EXP(alfa)\ se\ a \in \left\{\begin{pmatrix} Breakdown\ of\ Containers \rightarrow \\ Setting\ up \end{pmatrix}\right\}, \\
(F,t,i,j)@ + EXP(gama)\ se\ a \in \left\{\begin{pmatrix} Defining\ Time\ to\ Failure \rightarrow \\ Breakdown \end{pmatrix}\right\}, \\
(U, ModelTime())\ se\ a \in \{(Customer\ Arrival \rightarrow Entry\ Place)\}, \\
(U,t)\ se\ a \in \begin{Bmatrix} (Admission\ Queue \rightarrow CTN\ Allocation), (Breakdown\ of\ Containers \rightarrow Admission\ Queue), (Customer\ Admission \rightarrow Admission\ Queue), (Entry\ Place \rightarrow Customer\ Admission), (Entry\ Place \rightarrow Rejection), \\ (Rejection \rightarrow Blocked\ Customers) \end{Bmatrix}, \\
1\ se\ a \in \{(nCTNOff \rightarrow Create)\}, \\
1U@+EXP(lambda)\ se\ a \in \{(Customer\ Arrival \rightarrow Customers)\}, \\
AuxCtnPreSetup\ se\ a \in \{(Aux3 \rightarrow Turning\ it\ OFF)\}, \\
CtnPreSetup\ se\ a \in \{(Turning\ it\ OFF \rightarrow Aux3)\}, \\
i\ se\ a \in \{(CONT2 \rightarrow Create)\}, \\
i+1\ se\ a \in \{(Create \rightarrow CONT2)\}, \\
j\ se\ a \in \{(CONT \rightarrow Tag\ Insertion)\}, \\
j+1\ se\ a \in \{(Tag\ Insertion \rightarrow CONT)\}, \\
MaxRes`R\ se\ a \in \begin{Bmatrix} (Rejection \rightarrow Unavailable\ Resources), (Unavailable\ Resources \rightarrow Rejection) \end{Bmatrix}, \\
\varnothing se\ a \in \{(OFF\ Containers \rightarrow Reset)\}, \\
R\ se\ a \in \begin{Bmatrix} (Aux3 \rightarrow CTN\ Allocation), (Available\ Resources \rightarrow Customer\ Admission), (Customer\ Admission \rightarrow Start), (Customer\ Admission \rightarrow Unavailable\ Resources), (Service\ Terminantion\ in\ CTN \rightarrow Aux3), (Service\ Terminantion\ in\ CTN \rightarrow Available\ Resources), (Setup\ Complete \rightarrow Aux3), \\ (Setup\ Termination \rightarrow Aux3), (Start \rightarrow Reset), (Start \rightarrow Setup\ Beginning), (Unavailable\ Resources \rightarrow Service\ Terminantion\ in\ CTN) \end{Bmatrix}, \\
U\ se\ a \in \{(Customers \rightarrow Customer\ Arrival)\}
\end{cases}
$$

$$A = \{(Customers \rightarrow Customer\ Arrival), (Customer\ Arrival \rightarrow Customers), (Customer\ Arrival \rightarrow Entry\ Place), (Entry\ Place \rightarrow Rejection), (Entry\ Place \rightarrow Customer\ Admission), (Rejection \rightarrow Blocked\ Customers), (Rejection \rightarrow Unavailable\ Resources), (Unavailable\ Resources \rightarrow Rejection), (Customer\ Admission \rightarrow Unavailable\ Resources), (Customer\ Admission \rightarrow Start), (Customer\ Admission \rightarrow Admission\ Queue), (Unavailable\ Resources \rightarrow Service\ Terminantion\ in\ CTN \rightarrow Available\ Resources), (Admission\ Queue \rightarrow CTN\ Allocation), (Breakdown\ of\ Containers \rightarrow Admission\ Queue), (Start \rightarrow Setup\ Beginning), (Start \rightarrow Reset), (OFF\ Containers \rightarrow Reset), (Setup\ Beginning \rightarrow Setting\ up1), (Setting\ up1 \rightarrow Setup\ Complete), (OFF\ Containers \rightarrow Setup\ Beginning), (CONT \rightarrow Tag\ Insertion), (Tag\ Insertion \rightarrow CONT), (Setup\ Complete \rightarrow Ready\ Container), (Setup\ Complete \rightarrow Aux3), (CTN\ Allocation \rightarrow Aux1), (Aux1 \rightarrow Tag\ Insertion), (Tag\ Insertion \rightarrow Service\ Allocation), (Service\ Allocation \rightarrow Service\ Beginning), (Tag\ Insertion \rightarrow Aux2), (Aux2 \rightarrow Defining\ Time\ to\ Failure), (Defining\ Time\ to\ Failure \rightarrow Breakdown), (Breakdown \rightarrow Breakdown\ of\ Containers), (CONT2 \rightarrow Create), (Create \rightarrow CONT2), (nCTNOff \rightarrow Create), (Create \rightarrow OFF\ Containers), (Turning\ it\ OFF \rightarrow OFF\ Containers), (Turning\ it\ OFF \rightarrow Aux3), (Aux3 \rightarrow Turning\ it\ OFF), (Ready\ Container \rightarrow Turning\ it\ OFF), (Ready\ Container \rightarrow CTN\ Allocation), (Aux3 \rightarrow CTN\ Allocation), (Service\ Terminantion\ in\ CTN \rightarrow Aux3), (Setup\ Termination \rightarrow Aux3), (Service\ Terminantion\ in\ CTN \rightarrow Ready\ Container), (Setup\ Termination \rightarrow Ready\ Container), (Breakdown\ of\ Containers \rightarrow Service\ Failure), (Service\ Failure \rightarrow Rejection\ due\ to\ Failure), (Breakdown\ of\ Containers \rightarrow Setting\ up), (Setting\ up \rightarrow Setup\ Termination), (Working\ Containers \rightarrow Breakdown\ of\ Containers), (Working\ Containers \rightarrow Service\ Terminantion\ in\ CTN), (Service\ Beginning \rightarrow Working\ Containers), (Service\ Beginning \rightarrow Servicing\ by\ Container), (Servicing\ by\ Container \rightarrow Rejection\ due\ to\ Failure), (Servicing\ by\ Container \rightarrow Service\ Terminantion\ in\ CTN)\}$$

4 Resultados

Para analisar o impacto do escalonamento com pré-inicialização de recursos para os serviços URLLC, configurou-se os modelos (analítico e de simulação) de modo a representar um pequeno nó MEC. Do ponto de vista da aplicação, os modelos baseados em fila para URLLC são mal caracterizados na literatura, pois pouco se sabe sobre o tráfego real dos serviços URRLC e as taxas de falha e configuração do contêiner podem variar amplamente devido a diferenças de *hardware* e *software* que provém tal recurso. Entretanto, buscou-se contornar tais aspectos utilizando insights recentes sobre as taxas de serviço e chegada dispostos em (3GPP, 2020), que descreve um tempo de serviço da rede núcleo de até 1 ms (1 serviço/ms) e chegadas de serviço com até a 20 solicitações/ms. Desde modo, adotou-se a o mesmo valor para a taxa de serviço e a mesma escala para a taxa de chegada em conjunto com uma taxa de falha igual a (γ) 0,001 e de setup (α) de acordo com (NGMN, 2019). Variações na taxa de chegada (λ de 5 a 30), número de contêineres pré-inicializados (n de 1 a 4), número total de contêineres (c de 5 a 20), taxa de configuração de contêineres (α de 1 a 4) e capacidade máxima do nó MEC (k de 10 e 25) foram realizadas para analisar seus efeitos na probabilidade de bloqueio do serviço (PB), número médio de contêineres ligados (nCTNs) e tempo médio de resposta (MRT). Caso não seja especificado o contrário, os seguintes parâmetros são definidos como valores padrões para o modelo analítico e a simulação: $n = 2$, $c = 10$, $k = 15$, $\mu = 1$, $\alpha = 1$, $\gamma = 0,001$, $\lambda = [5, 30]$. No modelo de simulação foram executados aproximadamente 30 milhões de passos. A Tabela 1 sumariza os parâmetros adotados, organizando-os em cenários, onde em cada um, além da taxa de chegada, outro parâmetro é variado.

Parâmetro	Cenário A	Cenário B	Cenário C	Cenário D
CTNs pre-inicializados (n)	2	2	[1, 4]	2
CTNs no sistema (c)	10	[5, 20]	10	10
Max. de serviços URLLC (k)	15	25	15	[10, 25]
Taxa de serviço URLLC (μ)	1	1	1	1
Taxa de setup dos CTNs (α)	[1, 4]	1	1	1
Taxa de falha dos CTNs (γ)	0,001	0,001	0,001	0,001
Taxa de chegada URLLC (λ)	[5, 30]	[5, 30]	[5, 30]	[5, 30]

Tabela 1 – Parâmetros de configuração.

As Figuras 10-21 ilustram resultados obtidos através dos modelos analítico e de simulação em termos de probabilidade de bloqueio do serviço, número médio de contêineres ligados e tempo médio de resposta, onde as linhas e os pontos denotam os resultados analíticos e de simulação, respectivamente. Cada ponto do resultado

analítico é o valor médio dos resultados de 15 instâncias de simulação, com 10000 chegadas de serviço cada. As seções a seguir apresentam os impactos de λ, α, c, n e k nas métricas PB, nCTNs e MRT.

4.1 Impactos da taxa de chegada, λ

As Figuras 10, 13, 16 e 19 apresentam o impacto de λ na probabilidade de bloqueio dos usuários (PB). Nelas, é possível observar que, inicialmente, a PB é próxima de 0 e cresce gradualmente a medida que λ aumenta. Este comportamento é esperado e resultado da crescente ocupação dos contêineres do sistema, acarretando em uma maior quantidade de serviços aguardando processamento. O aumento no tempo de espera faz com que o limite de serviços no sistema seja atingido mais rapidamente.

O impacto de λ no nCTNs é mostrado nas Figuras 11, 14, 17 e 20. Em geral, o comportamento das curvas pode ser dividido em duas fases, subida e estabilização. Na primeira, o número de contêineres do sistema é suficiente para atender a demanda de serviços, sendo ligados gradualmente conforme o aumento da carga de chegada de usuário, λ. Já na fase de estabilização, com a demanda muito elevada, os contêineres permanecem predominantemente ligados para atendê-la, atingindo o limite de contêineres ligados simultaneamente.

As Figuras 12, 15, 18 e 21 apresentam o impacto de λ no MRT. O comportamento das curvas pode ser dividido em duas fases: subida e estabilização. No início da subida, MRT assume valores próximos de $\frac{1}{\mu}$, mas à medida que λ aumenta, o tempo de configuração de novos contêineres para atendimento de novos serviços impacta diretamente MRT. A fase de estabilização da curva é atingida quando todos os contêineres já estão ligados e atendendo aos serviços de forma quase ininterrupta.

4.2 Impactos da taxa de configuração de contêineres (Cenário A)

A Figura 10 apresenta os resultados de PB sob diferentes valores de taxa de configuração de contêineres α. Quanto maior o valor de α, mais rápido os contêineres são configurados, inicializados para processamento. Como se observa, as curvas são próximas entre si, com exceção do ponto em que a taxa de chegada (λ) se aproxima do número total de contêineres no nó MEC (a saber 10), mostrando uma PB ligeiramente maior quando os contêineres demoram mais a ficar operantes. Essa similaridade de desempenho pode indicar que o efeito da taxa de setup menor pode ser mitigado pela pré-inicialização de contêineres, que neste cenário é igual a 2. Quando a demanda é leve (em torno de 5 ou 6), a pré-inicialização consegue manter a probabilidade de bloqueio em níveis similares as configurações com menor tempo de setup

do recurso, pois eles conseguem ficar disponíveis (ativos) próximos ao tempo que os usuários chegam. À medida que a demanda aumenta, em torno de λ igual a 8 ou 10, a chegada de usuários se torna mais frequente, gerando ainda a pré-inicialização. Entretanto, o usuário tende a aguardar o setup do contêiner finalizar. Quando a demanda de serviços URRLC (λ) é muito alta, as diferentes taxas de setups não impactam na PB, pois nessa situação os recursos (contêineres) ficam predominantemente ativos, processando serviços ou pré-inicializados, de modo que será pouco provável que eles reinicializem devido a término de serviço. A reinicialização será predominantemente ocasionada por falhas nos contêineres durante o processamento dos serviços.

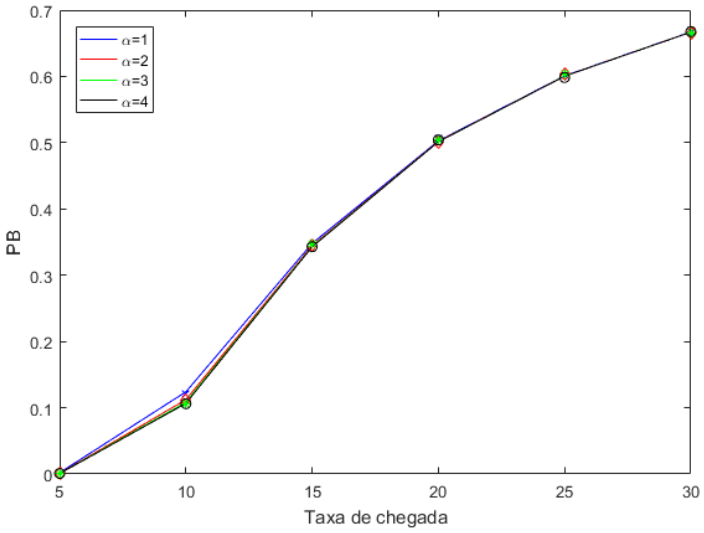

Figura 10 – Impactos de α na probabilidade de bloqueio (PB).

A Figura 11 representa os impactos de α no número de contêineres ativos (nCTNs). Observa-se que quanto maior o valor de α, maior é a inclinação das curvas durante a fase anterior a saturação do sistema, pois menor é o tempo que os contêineres levam para ficar prontos para atendimento. Quando a taxa de chegada é alta (valor de λ), a taxa de *setup* não exerce influência no número de contêineres ativos, pois todos os recursos ficam predominantemente ativos para satisfazer a demanda, reinicializando apenas em caso de falha.

A Figura 12 mostra o impacto de α no MRT. Similar a Figura 11 a influência do valor α é observada na inclinação das curvas. Um valor de α maior diminui o tempo que os contêineres levam para ficar prontos, com isso os serviços URLLC começam a ser processados (menor tempo de espera na fila) mais rapidamente e, consequentemente, experimentam um menor MRT. Esse comportamento se observa quando a taxa de chegada varia até por volta de 15, quando o processo de inicialização de container acontece com maior frequência. Para taxas de chegadas maiores, não se observa influência dos tempos de setups no MRT, pois os contêineres ficam ativos a maior parte do tempo. Nestes casos, quando um container finaliza o processamento de um

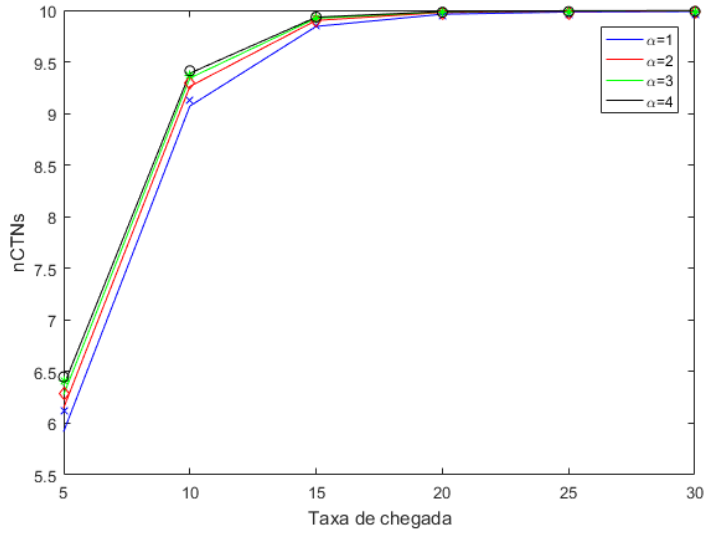

Figura 11 – Impactos de α no número médio de contêineres ligados (nCTNs).

serviço URLLC, ele de imediato inicia o processamento de outro, não ocorrendo ou demandando a sua reinicialização.

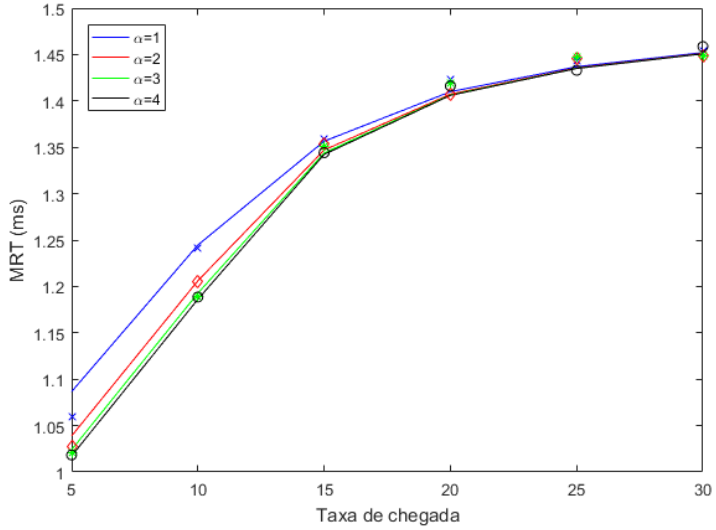

Figura 12 – Impactos de α no tempo médio de resposta (MRT).

4.3 Impactos do número de contêineres (Cenário B)

Na Figura 13 são avaliadas as probabilidades de bloqueio quando nós MEC com quantidade de recursos (contêineres, c) diferentes são considerados. As quatro curvas aumentam de acordo com λ. Quando λ aumenta, um c maior significa que mais contêineres podem ser usados para lidar com as crescentes solicitações de serviço, diminuindo o número de serviços aguardando processamento na fila. Portanto, PB diminui de acordo com o aumento de c no sistema. Para taxas de chegada (λ) mais baixas

(ex. 5 requisições/ms) e considerando um nó MEC com 5 contêineres (c) a taxa de bloqueio de serviços URLLC é de 4.4%. O incremento de c em 100% acarreta em uma pequena redução (4.4%) na PB, resultando numa relação custo-ganho pouco atrativa para o provedor de serviço. Em contrapartida, esse mesmo incremento resulta em uma redução de 45% na PB quando se tem o dobro de demanda (taxa de chegada igual a 10 requisições/ms). A redução na PB proporcionada pelo incremento de 5 contêineres diminui conforme λ aumenta. Quando a taxa de chegada atinge 30 requisições/ms, o incremento de 5 contêineres ao sistema, reduz a PB em aproximadamente 16%. Os eventos de bloqueio de solicitação podem impactar significativamente aplicações URLLC, pois quando um bloqueio ocorre, as alternativas naturais são encaminhar as solicitações bloqueadas para um nó vizinho ou nuvem central (SARRIGIANNIS et al., 2019), que traz incertezas quanto aos níveis de qualidade de serviço a ser alcançado pelas aplicações e podendo ambos incorrer em violações de requisitos dos serviços URLLC (ex. latência e confiabilidade). Logo o operador deve levar em consideração uma configuração especifica de contêineres disponíveis no nó MEC de acordo com a demanda de usuário tendo em mente a relação custo-ganho.

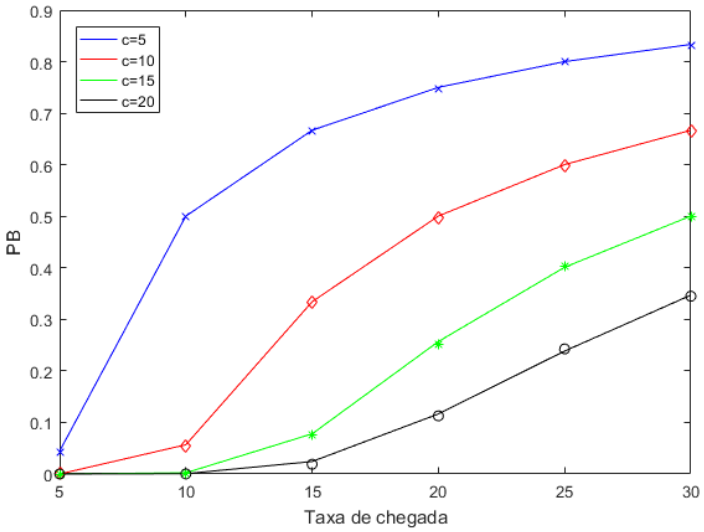

Figura 13 – Impactos de c na probabilidade de bloqueio (PB).

Os resultados do nCTNs quando nós MEC com diferentes quantidades de contêineres são analisados na Figura 14. Sob diferentes taxas de chegadas de usuários. Nota-se que para todos os valores de c, as curvas crescem a medida que a taxa de chegadas de usuários aumenta e então atingem um limite após o valor numérico de λ ultrapassar a quantidade de contêineres do nó, c. Um c maior proporciona uma maior faixa de número de contêiner ativos, que refletirá a demanda serviços URLLC que chega no nó MEC.

Já em termos de MRT, a Figura 15 ilustra o impacto do número de contêineres no nó MEC sob diferentes cargas de serviços URLLC. Nota-se que quanto maior o

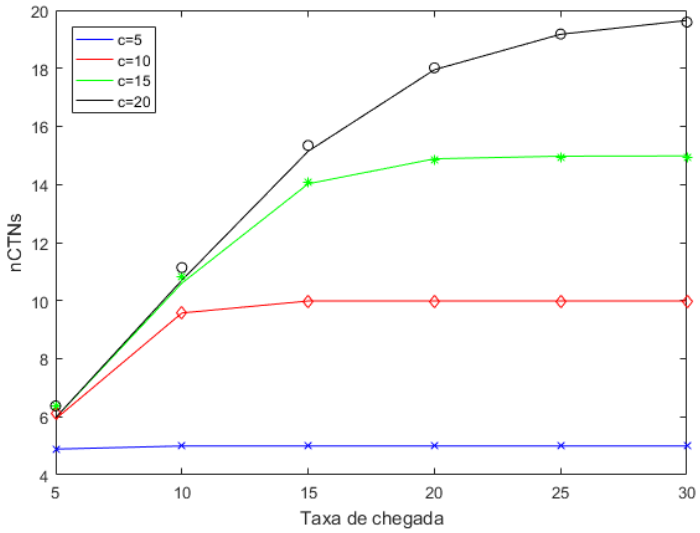

Figura 14 – Impactos de c no número médio de contêineres ligados (nCTNs).

valor de c, maior a quantidade de serviços URLLC podem ser processados em paralelo, diminuindo a quantidade de requisições na fila para atendimento e implicando em um menor tempo de resposta. O tempo médio de resposta (MRT) tende a se estabilizar a medida que a taxa de chegada (λ) se aproxima da capacidade do sistema (k), pois MRT será computado apenas pelo tempo de processamento da requisição no contêiner somado do tempo máximo de espera na fila. Considerando a aplicação de automação industrial, cuja a restrição de latência é de 2 ms (3GPP Versão 16 (Rel-16)), nota-se que a configuração com 5 contêineres não atende a este requisito para nenhum valor de taxa de chegada. Dobrando a quantidade de contêineres ($c = 10$), o tempo de resposta consegue ficar abaixo do estipulado quando a demanda de serviços é baixa (λ até 10). Ao passo que adotando as outras configurações ($c = 15$ e $c = 20$) consegue-se suportar a aplicação para todos os valores de carga de serviços analisados.

A princípio poderia se escolher a configuração com maior número de contêineres para compor o nó MEC e atender o serviço URLLC. Entretanto, essa escolha poderia incorrer em um maior custo de operação ou ociosidade de recurso principalmente em momentos de baixa carga. Desta forma, analisando de uma perspectiva custo e satisfação do requisito da aplicação, para demanda baixa, o operador poderia selecionar a configuração com $c = 10$ e para demandas maiores a configuração com $c = 15$ poderia ser empregada.

4.4 Impactos do número de contêineres pré-inicializados (Cenário C)

Em relação a PB, na Figura 16 são apresentados os resultados quando dife-

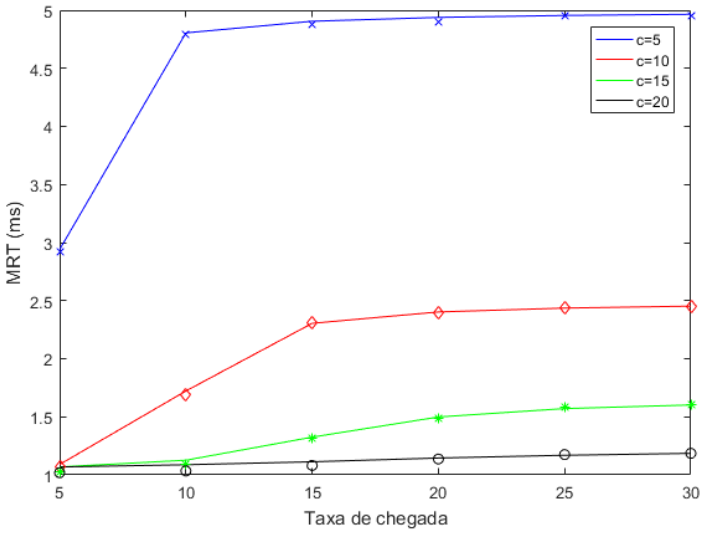

Figura 15 – Impactos de c no tempo médio de resposta (MRT).

rentes números de contêineres pré-inicializados são considerados e para diferentes valores de taxa de chegada. Observa-se que n não tem impacto na PB para taxas de chegadas muito baixas (quando λ não é suficiente para formar fila) ou muito altas (quando o número de contêineres ativos se aproxima ou é igual a quantidade total do nó MEC, c), ou seja, no início e no final das curvas. A diferença de desempenho se percebe quando a taxa de chegada de serviços URRLC fica em torno de c e k. Um maior n leva a uma PB menor, pois com mais contêineres prontos para atendimento, o processamento dos serviços pode iniciar mais rapidamente e a ocupação da fila tende a diminuir, possibilitando a admissão de novos serviços.

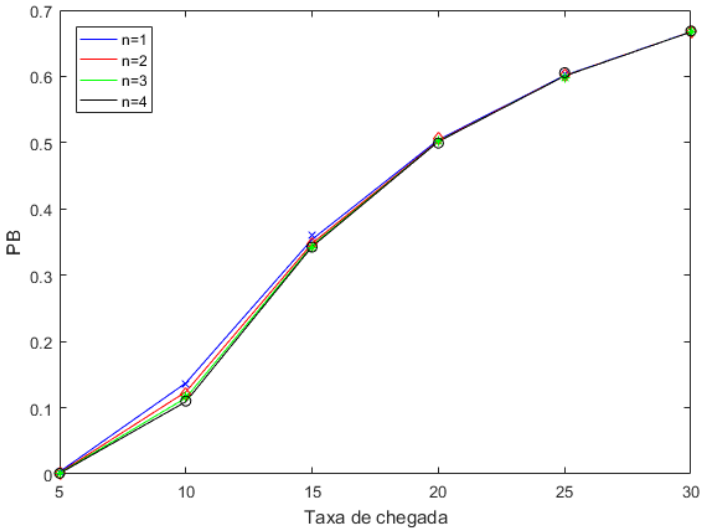

Figura 16 – Impactos de n na probabilidade de bloqueio (PB).

As Figuras 17 e 18 ilustram os impactos de n em nCTNs e MRT, respectivamente. Como já era esperado, na Figura 17, n mostra maior impacto em nCTNs no

ponto inicial, a medida que λ aumenta os valores de nCTNs tendem a convergir para o limite c. Um n maior acarreta em uma saturação de nCTNs mais rápida em relação a λ.

Em contrapartida, na Figura 18 os contêineres ligados previamente impactam diretamente em MRT, diminuindo o tempo de resposta para maiores valores de n no início das curvas, pois novos serviços terão de aguardar menos por um contêiner pronto para atendimento ou até mesmo serem atendidos de imediato. Um nó MEC configurado para manter 2 contêineres ligados previamente tem uma redução de 6% no tempo de resposta para taxa de chegada (λ) igual a 5 em relação a um nó com apenas 1 contêiner ligado previamente. A medida que λ aumenta essa redução em MRT tende a diminuir, pois os tempos entre as chegadas passam a ser muito menores em relação ao tempo de ligação dos contêineres ativados com a sua chegada. Por exemplo, quando λ dobra (10) o ganho alcançado com $n = 2$ é de 3,87% em relação ao $n = 1$.

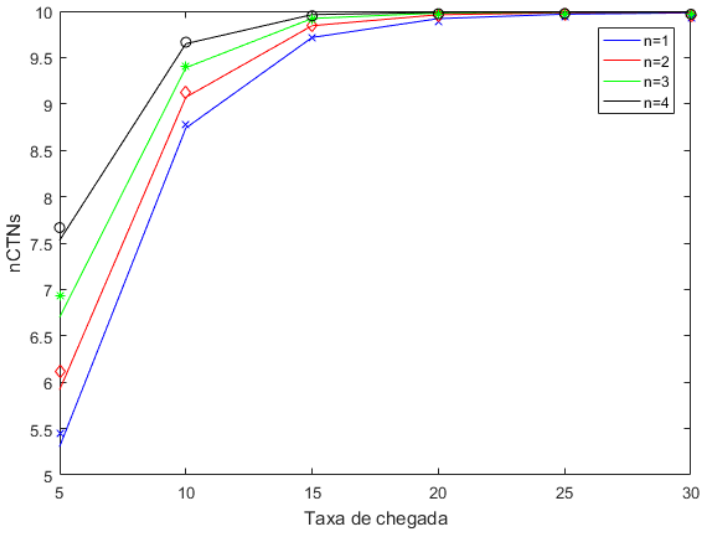

Figura 17 – Impactos de n no número médio de contêineres ligados (nCTNs).

4.5 Impactos da capacidade do sistema (Cenário D)

Diferentes configurações da capacidade do sistema (k) são avaliadas em relação a PB na Figura 19. Inicialmente a PB é próxima de 0, pois os contêineres do sistema são capazes de lidar com a fila formada com uma taxa de chegada de serviços baixa. A medida que λ se aproxima do número de contêineres no sistema (c), o impacto de k é evidenciado, pois, um k maior aumenta o tamanho da fila para novos serviços diminuindo sensivelmente a PB. Quando λ ultrapassa c, nota-se que um k elevado não garante PB baixa, pois com valores de λ mais altos, a admissão no sistema é definida através da vazão dos c contêineres disponíveis, fazendo com que o limite de usuários simultâneos no sistema (k) seja atingido com frequência.

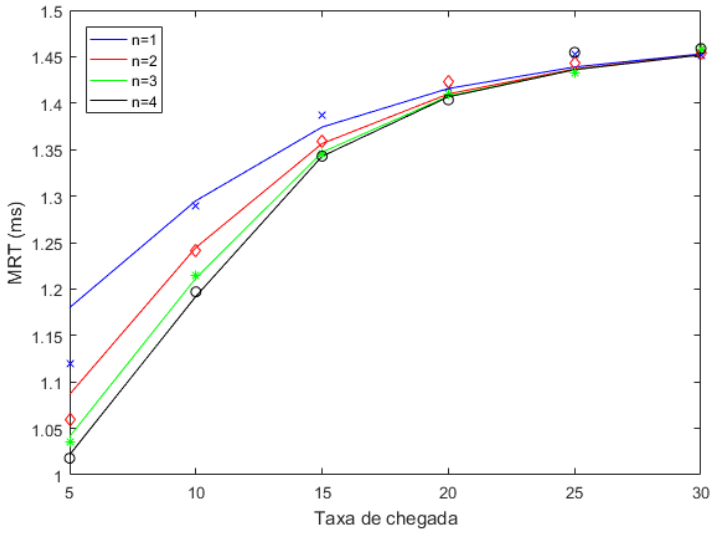

Figura 18 – Impactos de n no tempo médio de resposta (MRT).

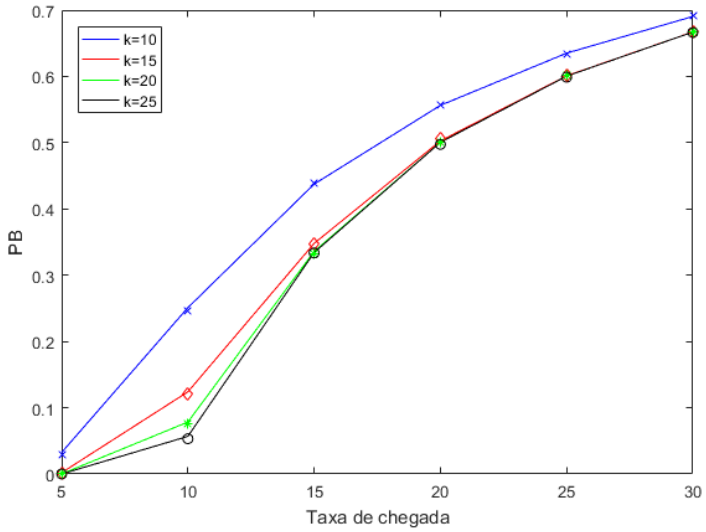

Figura 19 – Impactos de k na probabilidade de bloqueio (PB).

Na Figura 20 são representados os impactos de k em nCTNs. As curvas mostram que a medida que λ aumenta, um k maior possibilita que mais serviços possam aguardar o processamento dos contêineres, fazendo com que nCTNs atinja o limite (c) mais rapidamente em decorrência do uso quase ininterrupto dos contêineres, que, por conta da alta demanda, raramente são desligados após o termino do processamento de um serviço. No ponto em que λ é igual a c, o aumento de 50% em relação ao cenário que permite 10 usuários simultâneos no sistema ($k = 10$), acarreta em um aumento de aproximadamente 1,2 contêineres ligados no sistema (nCTNs).

O gráfico na Figura 21 mostra os impactos de k em MRT. Foi observado que apesar de um maior k diminuir a PB, MRT aumenta drasticamente com a quantidade de serviços aguardando para serem processados, ou seja, o atraso na fila, que é uma

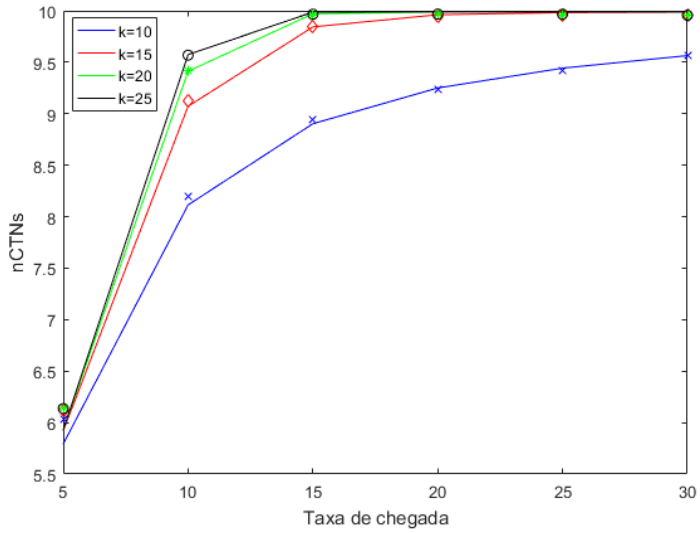

Figura 20 – Impactos de k no número médio de contêineres ligados (nCTNs).

componente do tempo de resposta, acaba aumentando a sua contribuição, podendo acarretar em violações de SLA, inviabilizando o serviço.

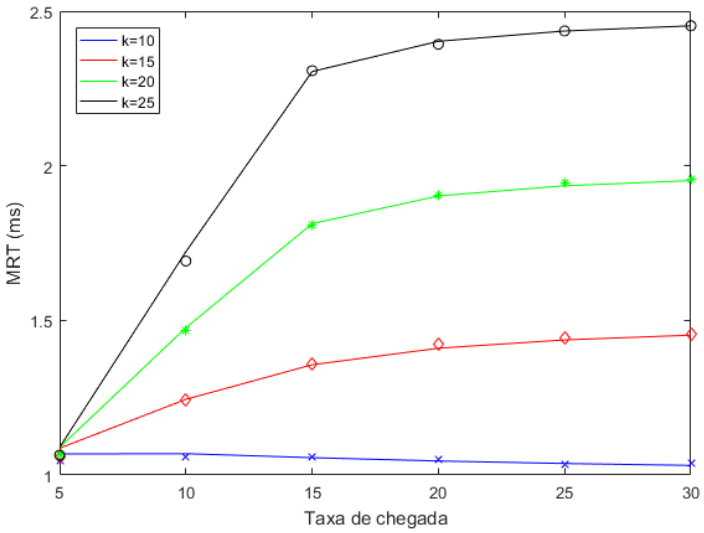

Figura 21 – Impactos de k no tempo médio de resposta (MRT).

5 Conclusão

As tecnologias NFV e MEC são consideradas para compor a rede de núcleo 5G e prover suporte ao atendimento das aplicações URLLC, pois permitem com que funções de rede e aplicações possam ser executadas mais próximo do usuário final, reduzindo a latência fim-a-fim. Entretanto, a utilização destas tecnologias traz desafios no aprovisionamento de recursos para os serviços URLLC.

Neste trabalho analisou-se o aprovisionamento de recursos para serviços URLLC em redes 5G baseadas em MEC-NFV através de um modelo baseado em teoria de fila, onde aspectos como o tempo de inicialização de VNF, a possibilidade de falha durante o atendimento e a pré-inicialização de recursos foram considerados. Avaliações de diferentes cenários foram conduzidas e métricas como o tempo médio de resposta, número médio de contêineres ativos e a probabilidade de bloqueio de serviço foram analisadas.

Foi observado que o efeito da taxa de *setup* menor pode ser mitigado pela pré-inicialização de contêineres, diminuindo o tempo de espera para atendimento do serviço. Apesar do aumento da capacidade de admissão de serviços do sistema diminuir a probabilidade de bloqueio, este incremento aumenta drasticamente o tempo de resposta dos serviços, pois a capacidade de atendimento será limitada pelos contêineres disponíveis no sistema. Com isso o operador deve levar em consideração uma configuração especifica de contêineres disponíveis no nó MEC de acordo com a demanda de serviço tendo em mente a relação custo-ganho.

Em um cenário realista, este modelo poderia ajudar o provedor de serviço a dimensionar um nó MEC para atendimento de serviços URLLC, visando alcançar os níveis de qualidade de serviço requeridos pelas aplicações com um custo de operação minimizado. Para isso devem ser levados em consideração a carga de serviço imposta ao ambiente e a capacidade de atendimento dos recursos utilizados.

Como trabalhos futuros, apontam-se a inserção de novas características no modelo de filas, de modo que ele consiga representar cenários de diferentes tipos de serviços (ex. eMBB e URLLC). Além disso, técnicas como a análise de reserva de recurso e priorização de serviço podem ser inseridas no modelo para avaliar seus impactos no atendimento dos serviços. Outro ponto de estudo que pode ser explorado é a adoção de ambientes experimentais para a validação dos modelos analíticos e análise de técnicas voltadas para a alocação de recursos para serviços URLLC.

Referências

3GPP. *Release description; Release 16*. 2020. Citado na página 43.

ALI, K. *Modeling, Analysis, and Design of 5G Networks using Stochastic Geometry*. Tese (Doutorado), 2018. Citado na página 13.

BILAL, A. et al. Dynamic cloud resource scheduling in virtualized 5g mobile systems. In: IEEE. *2016 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2016. p. 1–6. Citado na página 19.

BOLCH, G. et al. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. [S.l.]: John Wiley & Sons, 2006. Citado 4 vezes nas páginas 20, 22, 24 e 26.

CHEN, Y.-J.; CHENG, L.-Y.; WANG, L.-C. Prioritized resource reservation for reducing random access delay in 5g urllc. In: IEEE. *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. [S.l.], 2017. p. 1–5. Citado na página 16.

COOPER, R. B. *Introduction to Queueing Theory, Second Edi*. [S.l.]: North Holland Inc., NY, USA, 1981. Citado na página 22.

DORDA, M. Modelling and simulation of unreliable m/m/n/n queueing system. *Perner's Contacts*, v. 5, n. 4, p. 37–45, 2010. Citado na página 33.

FOUKAS, X.; MARINA, M. K.; KONTOVASILIS, K. Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture. In: *Proceedings of the 23rd annual international conference on mobile computing and networking*. [S.l.: s.n.], 2017. p. 127–140. Citado na página 14.

GIORDANI, M. et al. Toward 6g networks: Use cases and technologies. *IEEE Communications Magazine*, IEEE, v. 58, n. 3, p. 55–61, 2020. Citado na página 17.

GROSS, D. *Fundamentals of queueing theory*. [S.l.]: John Wiley & Sons, 2008. Citado 3 vezes nas páginas 23, 24 e 25.

HILL, Z. et al. Early observations on the performance of windows azure. In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. [S.l.: s.n.], 2010. p. 367–376. Citado na página 19.

HÖSSLER, T.; SIMSEK, M.; FETTWEIS, G. P. Mission reliability for urllc in wireless networks. *IEEE Communications Letters*, IEEE, v. 22, n. 11, p. 2350–2353, 2018. Citado na página 19.

HUANG, G. et al. Auto scaling virtual machines for web applications with queueing theory. In: IEEE. *2016 3rd International Conference on Systems and Informatics (ICSAI)*. [S.l.], 2016. p. 433–438. Citado na página 19.

JAIN, R. The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modeling. by raj jain. new york: John wiley & sons, 1991. pp. 720. us $52.00 (hardcover). *International Journal of Legal Information*, Cambridge University Press, v. 20, n. 1, p. 63–64, 1992. Citado 4 vezes nas páginas 22, 23, 24 e 26.

JI, H. et al. Ultra-reliable and low-latency communications in 5g downlink: Physical layer aspects. *IEEE Wireless Communications*, IEEE, v. 25, n. 3, p. 124–130, 2018. Citado 2 vezes nas páginas 14 e 16.

KHAZAEI, H.; BARNA, C.; LITOIU, M. Performance modeling of microservice platforms considering the dynamics of the underlying cloud infrastructure. *arXiv preprint arXiv:1902.03387*, 2019. Citado na página 19.

LI, C.-P. et al. 5g ultra-reliable and low-latency systems design. In: IEEE. *2017 European Conference on Networks and Communications (EuCNC)*. [S.l.], 2017. p. 1–5. Citado 2 vezes nas páginas 13 e 17.

MIJUMBI, R. et al. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications surveys & tutorials*, IEEE, v. 18, n. 1, p. 236–262, 2015. Citado na página 18.

NGMN. *5G Extreme Requirements: End-to-End Considerations*. 2019. Citado na página 43.

PARVEZ, I. et al. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 3098–3130, 2018. Citado na página 16.

POCOVI, G. et al. Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements. *IEEE Network*, IEEE, v. 32, n. 2, p. 8–15, 2018. Citado 2 vezes nas páginas 16 e 17.

PORAMBAGE, P. et al. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 2961–2991, 2018. Citado na página 17.

REN, Y. et al. Dynamic auto scaling algorithm (dasa) for 5g mobile networks. In: IEEE. *2016 IEEE global communications conference (GLOBECOM)*. [S.l.], 2016. p. 1–6. Citado 2 vezes nas páginas 14 e 19.

REN, Y. et al. Asa: Adaptive vnf scaling algorithm for 5g mobile networks. In: IEEE. *2018 IEEE 7th international conference on cloud networking (CloudNet)*. [S.l.], 2018. p. 1–4. Citado na página 19.

RIGHI, R. da R. et al. Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *IEEE Transactions on Cloud Computing*, IEEE, v. 4, n. 1, p. 6–19, 2015. Citado na página 19.

RUIZ, L. et al. A genetic algorithm for vnf provisioning in nfv-enabled cloud/mec ran architectures. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 8, n. 12, p. 2614, 2018. Citado na página 17.

SARRIGIANNIS, I. et al. Online vnf lifecycle management in an mec-enabled 5g iot architecture. *IEEE Internet of Things Journal*, IEEE, v. 7, n. 5, p. 4183–4194, 2019. Citado na página 47.

SCHULZ, P. et al. Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine*, IEEE, v. 55, n. 2, p. 70–78, 2017. Citado na página 16.

SHAHIDINEJAD, A.; GHOBAEI-ARANI, M.; ESMAEILI, L. An elastic controller using colored petri nets in cloud computing environment. *Cluster Computing*, Springer, p. 1–27, 2019. Citado 2 vezes nas páginas 26 e 27.

SHARMA, P. Evolution of mobile wireless communication networks-1g to 5g as well as future prospective of next generation communication network. *International Journal of Computer Science and Mobile Computing*, v. 2, n. 8, p. 47–53, 2013. Citado na página 13.

SHE, C. et al. Improving network availability of ultra-reliable and low-latency communications with multi-connectivity. *IEEE Transactions on Communications*, IEEE, v. 66, n. 11, p. 5482–5496, 2018. Citado 3 vezes nas páginas 13, 16 e 17.

TOOLS, C. *CPN Tools; A tool for editing, simulating, and analyzing Colored Petri nets*. 2018. Citado na página 33.

VAKILINIA, S.; CHERIET, M.; RAJKUMAR, J. Dynamic resource allocation of smart home workloads in the cloud. In: IEEE. *2016 12th International Conference on Network and Service Management (CNSM)*. [S.l.], 2016. p. 367–370. Citado na página 19.

WANG, C.-X. et al. Cellular architecture and key technologies for 5g wireless communication networks. *IEEE communications magazine*, IEEE, v. 52, n. 2, p. 122–130, 2014. Citado na página 13.

WUTTIDITTACHOTTI, P.; DAENGSI, T. Qoe of social network applications: A study of voip quality from skype vs line over 3g and 4g. In: IEEE. *2015 Seventh International Conference on Ubiquitous and Future Networks*. [S.l.], 2015. p. 462–464. Citado na página 13.

XIONG, K. et al. Dynamic resource provisioning and resource customization for mixed traffics in virtualized radio access network. *IEEE Access*, IEEE, v. 7, p. 115440–115453, 2019. Citado na página 19.