

UNIVERSIDADE FEDERAL FLUMINENSE

IAN VILAR BASTOS

**Computação na Borda sobre Aplicações de Baixa
Latência em Redes Sem Fio de Nova Geração**

NITERÓI

2022

UNIVERSIDADE FEDERAL FLUMINENSE

IAN VILAR BASTOS

Computação na Borda sobre Aplicações de Baixa Latência em Redes Sem Fio de Nova Geração

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: Ciência da Computação.

Orientador:
Igor Monteiro Moraes

NITERÓI

2022

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

B327c Bastos, Ian Vilar
Computação na Borda sobre Aplicações de Baixa Latência
em Redes Sem Fio de Nova Geração / Ian Vilar Bastos. - 2022.
108 f.: il.

Orientador: Igor Monteiro Moraes.
Tese (doutorado)-Universidade Federal Fluminense, Instituto
de Computação, Niterói, 2022.

1. Redes Móveis. 2. 5G. 3. Funções Virtuais de Rede. 4.
Aprendizado de Máquina por Reforço. 5. Produção
intelectual. I. Moraes, Igor Monteiro, orientador. II.
Universidade Federal Fluminense. Instituto de Computação.
III. Título.

CDD - XXX

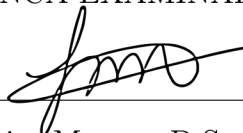
IAN VILAR BASTOS

Computação na Borda sobre Aplicações de Baixa Latência
em Redes Sem Fio de Nova Geração

Tese de Doutorado apresentada ao Programa
de Pós-Graduação em Computação da Uni-
versidade Federal Fluminense como requisito
parcial para a obtenção do Grau de Dou-
tor em Computação. Área de concentração:
Ciência da Computação.

Aprovada em setembro de 2022.

BANCA EXAMINADORA



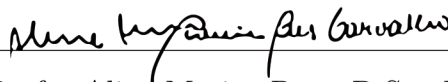
Prof. Igor Monteiro Moraes, D.Sc. - Orientador, UFF



Prof. Luciano Paschoal Gaspar, D.Sc., UFRGS



Prof. Miguel Elias Mitre Campista, D.Sc., UFRJ



Prof. Aline Marins Paes, D.Sc., UFF



Prof. Célio Vinicius Neves de Albuquerque, Ph.D., UFF

Niterói

2022

Ao meu falecido avô Jovino.

Agradecimentos

Primeiramente eu gostaria de agradecer aos meus pais, Vilma e Roberto e aos meus avós, Nevinha, Dulcinéa e Jovino, por todo carinho, incentivo e compreensão ao longo de toda essa jornada.

Aos meus amigos Cássio, Vinicius, Humberto e Murilo que dividi diversas conquistas e dificuldades durante todo esse período. Agradeço a todos os momentos de descontração e palavras de conforto que foram essenciais para me manter motivado e focado no desenvolvimento deste trabalho.

Ao professor Igor, que desde a iniciação científica é responsável por grande parte do meu crescimento profissional e da minha formação acadêmica. Agradeço por todos os conselhos, orientação e principalmente por sua amizade.

A querida Marister, uma mãezona que guarda todos os alunos do laboratório MídiaCom no seu coração enorme. Sempre solícita em ajudar com os problemas de última hora, agradeço por todo o carinho, os almoços maravilhosos que dividiu comigo e a companhia de todas as tardes.

A minhe namorade Hannah, uma pessoa maravilhosa que tive a oportunidade de conhecer e estar junto ao final dessa jornada. Agradeço todo o suporte nos momentos de dificuldade e o enorme companheirismo.

Aos professores Célio e Débora e a equipe do laboratório MídiaCom, por fornecerem um ambiente familiar e propício para o desenvolvimento de novas ideias.

Aos professores Luciano, Miguel, Aline e Célio que gentilmente aceitaram fazer parte da minha banca e contribuir com este trabalho através de suas críticas e revisões.

Aos professores Aline, Mai-Trang e Guy que me forneceram grande auxílio para o desenvolvimento deste trabalho.

A todos os funcionários do Instituto de Computação da Universidade Federal Fluminense pela dedicação no atendimento e serviços prestados.

Por fim, ao CNPq, CAPES, FAPERJ, Proppi/UFF, TBE/ANEEL e CELESC/ANEEL por todos os recursos disponibilizados para que esse trabalho pudesse ser realizado.

Resumo

O consumo de recursos nas redes de telefonia móvel cresce dramaticamente. Aplicações executadas em *laptops*, *smartphones*, *tablets*, *smartwatches* e dispositivos de realidade virtual requerem alta resolução de imagens e vídeos, altas taxas de transferência, baixa latência e conectividade ubíqua. Tais requisitos trazem desafios jamais pensados para as redes móveis, o que torna a quinta geração de telefonia móvel um novo ecossistema ao invés de uma evolução das redes 4G. As redes 5G são uma combinação de tecnologias de infraestrutura de nuvens, funções virtualizadas de rede e modelos de computação de alto desempenho para acomodar o tráfego gerado por bilhões de dispositivos interconectados. Esta tese possui como objetivo estudar técnicas para que as redes 5G ofereçam serviços que cumpram as restrições de aplicações sensíveis ao atraso. Um dos grandes desafios encontra-se na recuperação de conteúdos multimídias na rede de telefonia móvel. Uma vez que o alcance das células nas redes 5G são menores com o uso das pequenas células, a mobilidade dos usuários exige *handoffs* frequentes causando a reconexão do usuário ao servidor de vídeo. Tais reconexões impactam negativamente na qualidade de experiência do usuário devido ao atraso da reconfiguração de rota realizada pelo núcleo das redes de telefonia móvel. Dessa forma, as redes 5G buscam fornecer um ambiente distribuído com recursos de computação em nuvem na borda da rede móvel e oferecer baixas latências com altas larguras de banda próximo aos usuários. Esta tese avalia como a distribuição de *caches* instanciados em ambientes virtualizados próximos às estações base afeta a recuperação dos conteúdos a partir das requisições dos usuários. Além disso, propõe um agente baseado em aprendizado por reforço profundo para capturar a flutuação espaço-temporal da popularidade dos conteúdos e armazená-los nos *caches* antes que as requisições dos UEs sejam recebidas pelas estações base. Os resultados de desempenho demonstram que o agente é capaz de produzir até 65% de acertos em *cache*, assim como reduzir em 61% o atraso de recuperação dos conteúdos requisitados, um fator primordial para reduzir o tempo de inicialização dos vídeos.

Palavras-chave: Redes Móveis, 5G, Baixa Latência, Funções Virtuais de Rede, Aprendizado de Máquina por Reforço.

Abstract

The need for resources in mobile networks has grown dramatically. Applications that run over laptops, smartphones, tablets, smartwatches, and virtual reality devices require high image and video resolutions, high download rates, low latency, and ubiquitous connectivity. Such requirements bring challenges never thought for mobile networks, which makes the fifth generation of mobile networks an entirely new ecosystem. The 5G network will combine cloud infrastructure technologies, virtualized network functions, and high-performance computing models to accommodate the data generated by billions of interconnected mobile devices. This thesis has the goal to study techniques for 5G networks to offer services that meet sensitive application restrictions. One of the main challenges faced by 5G mobile networks is the retrieval of multimedia contents. 5G networks will accommodate several small cells which will impose frequent handoff procedures. Such procedures cause reconnections between the user and the content provider. Frequent reconnections negatively impact in the user quality of experience because of route reconfigurations performed by the mobile network core. 5G mobile networks aim to bring cloud computing capabilities to the edge of mobile networks to offer a low latency with high bandwidth environment closer to the users in a distributed fashion. This thesis evaluates how the distributed caching at the edge of the mobile network impacts the multimedia content retrieval. Notwithstanding, this thesis proposes a deep reinforcement learning agent that captures the contents' spatio-temporal popularity fluctuation and caches the popular contents before the users' requests arrive at the base stations to be satisfied. Performance evaluations show that the proposed agent is able to produce up to 65% of cache hits, and to reduce up to 61% the contents' retrieval delay, a major factor to reduce the videos start time.

Keywords: Mobile Networks, 5G, Low Latency, Virtual Network Functions, Deep Reinforcement Learning.

Lista de Figuras

1.1	Tráfego de dados móveis ao redor do mundo em Exabytes por mês [4]. . . .	2
2.1	As arquiteturas UMTS e GSM.	7
2.2	Arquitetura da rede de acesso a rádio UMTS.	9
2.3	Arquitetura do núcleo das redes UMTS.	11
2.4	Arquitetura da rede de acesso a rádio LTE.	12
2.5	Arquitetura do núcleo da rede LTE.	13
2.6	Arquitetura da NG-RAN.	14
2.7	Interfaces e serviços da gNB.	15
3.1	Arquitetura tradicional de uma estação base em macro-células.	19
3.2	Arquitetura de uma estação base com RRH.	19
3.3	C-RAN com RRH.	20
3.4	Arquitetura MEC de referência.	22
3.5	Servidores MEC dispostos diretamente nas <i>eNodeBs</i>	23
3.6	Servidores MEC dispostos fora das <i>eNodeBs</i>	24
4.1	Topologia com 50 nós utilizada nas simulações e gerada a partir dos parâmetros $\alpha = 0.1$, $\beta = 0.4$ e $L = 70$ pela biblioteca networkx 2.4 da linguagem de programação Python. Os nós em verde representam as <i>eNodeBs</i> , os nós em vermelho representam os servidores MEC, o nó em amarelo representa o nó de controle e os nós cinzas representam os nós encaminhadores. . . .	31
4.2	Distribuição das avaliações dos usuários sobre os diferentes filmes no conjunto de dados da Netflix.	32
4.3	Valores de intermediação para cada servidor MEC encontrados nas Topologias 1 e 2.	34

4.4	Atrasos de transferência das imagens dos <i>cache</i> VNFs ao considerar a soma dos atrasos de transmissão e propagação de cada enlace pertencente ao menor caminho entre o NC e o servidor MEC.	35
4.5	Distribuição da vazão entre os servidores MEC e as <i>eNodeBs</i> nas Topologias 1 e 2.	36
4.6	Resultados de atraso de recuperação e quantidade de <i>cache</i> VNFs instanciados na Topologia 1.	38
4.7	Resultados de atraso de recuperação e quantidade de <i>cache</i> VNFs instanciados na Topologia 2.	39
4.8	Taxa de recuperação média relativa entre os servidores MEC na Topologia 1.	40
4.9	Taxa de recuperação média relativa entre os servidores MEC na Topologia 2.	41
5.1	Rede neural artificial <i>feed-forward</i>	47
5.2	Processo de atualização dos parâmetros da rede neural artificial pelo algoritmo da descida do gradiente estocástico.	49
5.3	Atualização dos parâmetros da rede neural artificial sem o uso de um fator de regulação.	49
5.4	Encadeamento dos neurônios de cada camada da rede neural artificial. . . .	50
5.5	Processo do algoritmo de <i>back-propagation</i> para atualizar os parâmetros entre as camadas da rede neural artificial.	51
5.6	Interação entre agente e ambiente no processo de decisão de Markov. . . .	52
5.7	Interação entre o agente e o ambiente no <i>Q-learning</i> Profundo.	53
6.1	A arquitetura lógica da NG-RAN.	55
6.2	Esquema do <i>Gated Recurrent Unit</i>	59
7.1	Posicionamento dos nós NG-RAN e mobilidade dos UEs.	69
7.2	Características dos enlaces que interconectam os elementos de rede na topologia.	69
7.3	Características das requisições realizadas pelos UEs na simulação.	70
7.4	Evolução das funções de recompensas acumuladas propostas durante o processo de treinamento.	73

7.5	Evolução da <i>Smooth L1 loss</i> para as funções de recompensas acumuladas propostas.	74
7.6	Resultados das métricas ACM e atraso de recuperação médio durante o processo de validação.	75
7.7	Resultados da métrica ACM para as diferentes capacidades de armazenamento do servidor <i>cache</i> e para cada quantidade n de segmentos armazenados através do <i>prefetching</i>	77
7.8	A métrica ACM em função da popularidade do conteúdo.	78
7.9	Métrica RARM para todas as combinações de atraso da Internet e capacidade de armazenamento.	79

Lista de Tabelas

4.1	Notações utilizadas na formulação.	30
4.2	Perfis dos enlaces utilizados na geração da topologia [29].	32
4.3	Parametrização.	33
4.4	Informações das topologias entre o nó de controle e os servidores MEC. . .	34
6.1	Comparação dos trabalhos relacionados em aprendizado por reforço profundo.	67
7.1	Resumo dos parâmetros de simulação.	71
7.2	Resumo dos parâmetros de treinamento.	75

Lista de Abreviaturas e Siglas

3GPP	: <i>Third Generation Partnership Project</i>
5GC	: <i>5G Core Network</i>
ACM	: Acerto em <i>Cache</i> Médio
AMF	: <i>Access and Mobility Function</i>
APs	: <i>Access Points</i>
BBU	: <i>Baseband Processing Unit</i>
BS	: <i>Base Station</i>
C-RAN	: <i>Cloud Radio Access Network</i>
CAPEX	: <i>CAPital EXpenditure</i>
CDMA	: <i>Code Division Multiple Access</i>
CN	: <i>Core Network</i>
CS	: <i>Circuit Switch</i>
DQN	: <i>Deep Q-Network</i>
E-UTRAN	: <i>Evolved UMTS Terrestrial Radio Access Network</i>
eNodeB	: <i>evolved Node B</i>
EPC	: <i>Evolved Packet Core</i>
FCFS	: <i>First Come First Served</i>
FDMA	: <i>Frequency Division Multiple Access</i>
gBN-CU	: <i>gNB-Central Unit</i>
GERAN	: <i>GSM EDGE Radio Access Network</i>

GGSN	: <i>Gateway GPRS Support Node</i>
gNB-DU	: <i>gNB-Distributed Unit</i>
GPRS	: <i>General Packet Radio Services</i>
GRU	: <i>Gated Recurrent Unit</i>
GSM	: <i>Global System for Mobile Communications</i>
HetSNets	: <i>Heterogeneous and Small Cell Networks</i>
HSS	: <i>Home Subscriber Server</i>
IP	: <i>Internet Protocol</i>
LSTM	: <i>Long Short-Term Memory</i>
LTE	: <i>Long Term Evolution</i>
M-GW	: <i>Media Gateway</i>
MAC	: <i>Medium Access Control</i>
MDP	: <i>Markov Decision Process</i>
MEC	: <i>Multi-access Edge Computing</i>
MIMO	: <i>Multiple-Input and Multiple-Output</i>
MME	: <i>Mobility Management Entity</i>
MMS	: <i>Multimedia Messaging Service</i>
MSC	: <i>Mobile Switching Centre</i>
NC	: <i>Nó de Controle</i>
NFV	: <i>Network Function Virtualization</i>
NG-RAN	: <i>New-Generation Radio Access Network</i>
NGN	: <i>Next-Generation Networks</i>
OPEX	: <i>OPerating EXpenditure</i>
P-GW	: <i>Packet Data Network Gateway</i>

PDU	: <i>Protocol Data Unit</i>
PHY	: <i>Physical Layer</i>
PS	: <i>Packet Switched</i>
QoE	: <i>Quality of Experience</i>
RAN	: <i>Radio Access Network</i>
RF	: <i>Radio Functionalities</i>
RLC	: <i>Radio Link Control</i>
RMAR	: <i>Redução Média do Atraso de Recuperação</i>
RNC	: <i>Radio Network Controller</i>
RRC	: <i>Radio Resource Control</i>
RRH	: <i>Remote Radio Head</i>
RRN	: <i>Recurrent Neural Network</i>
RSRP	: <i>Reference Signal Received Power</i>
RSU	: <i>Road-Side Unit</i>
S-GW	: <i>Serving Gateway</i>
SDAP	: <i>Service Data Adaption Protocol</i>
SGD	: <i>Stochastic Gradient Descent</i>
SGSN	: <i>Serving GPRS Support Node</i>
SLA	: <i>Service Level Agreement</i>
SMS	: <i>Short Messaging Service</i>
TDMA	: <i>Time Division Multiple Access</i>
UE	: <i>User Equipment</i>
UMTS	: <i>Universal Mobile Telecommunication System</i>
UPF	: <i>User Plane Function</i>

UTRAN : *UMTS Terrestrial Radio Access Network*

VM : *Virtual Machine*

VNF : *Virtual Network Function*

VRNN : *Variational Recurrent Neural Network*

WNC : *Wireless Network Cloud*

Sumário

1	Introdução	1
1.1	Contexto	2
1.2	Hipóteses	3
1.3	Contribuições	4
1.4	Organização	5
2	As Redes de Telefonia Móvel	7
2.1	<i>Radio Access Network</i> - RAN	8
2.2	<i>Core Network</i> - CN	10
2.3	<i>Long Term Evolution</i> - LTE	10
2.3.1	<i>Evolved UMTS Terrestrial Radio Access Network</i> - E-UTRAN . . .	11
2.3.2	<i>Evolved Packet Core</i> - EPC	12
2.4	<i>New-Generation Radio Access Network</i> - NG-RAN	14
3	Arquiteturas Propostas para as Redes 5G	17
3.1	<i>Cloud Radio Access Network</i> - C-RAN	18
3.2	<i>Multi-access Edge Computing</i> - MEC	20
4	Recuperação de Conteúdos Multimídia na Arquitetura MEC	25
4.1	Proposta de Formulação	26
4.2	Metodologia de Avaliação	29
4.3	Resultados da Formulação	33
4.4	Trabalhos Relacionados	42

5	Aprendizado de Máquina	46
5.1	Redes Neurais Artificiais <i>Feed-Forward</i>	46
5.2	O Método da Descida do Gradiente Estocástico	48
5.3	O Algoritmo <i>Back-Propagation</i>	50
5.4	Aprendizado de Máquina por Reforço	51
5.5	<i>Q-learning</i> Profundo	53
6	Proposta de Agente de Aprendizado de Máquina por Reforço Profundo	54
6.1	Modelo de Sistema	54
6.1.1	Modelo da Rede	55
6.1.2	Modelo de Comunicação	56
6.1.3	Modelo de <i>Caching</i>	57
6.2	Espaço de Estados	58
6.3	Espaço de Ações	61
6.4	Cálculo da Recompensa	62
6.5	Trabalhos Relacionados	63
7	Avaliações do Agente Proposto	68
7.1	Metodologia de Avaliação	68
7.2	Resultados	71
7.2.1	Treinamento	72
7.2.2	Validação	74
7.2.3	Desempenho	76
8	Conclusão	80
8.1	Publicações	82
8.2	Trabalhos Futuros	82

Capítulo 1

Introdução

Em 1980, a primeira geração das redes de telefonia móvel (1G) eram analógicas por natureza e tornou possível a comunicação por voz entre dois usuários através do acesso ao meio pela divisão da frequência (*Frequency Division Multiple Access* - FDMA), ainda que estes usuários estivessem se locomovendo. Dez anos depois, no fim da década de 1980, as redes de telefonia móvel passariam a utilizar tecnologias de acesso ao meio digitais, tais como a divisão em *slots* de tempo (*Time Division Multiple Access* - TDMA), e a divisão por código (*Code Division Multiple Access* - CDMA). Serviços de comunicação através de mensagens de texto (*Short Messaging Service* - SMS) e mensagens multimídias (*Multimedia Messaging Service* - MMS) tornariam-se possíveis devido a maior eficiência espectral proporcionada pelas técnicas de acesso ao meio na segunda geração das redes de telefonia móvel (2G). Posteriormente, ao realizar a transmissão dos dados por comutação de pacotes e não mais por comutação de circuito, a taxa de transmissão passa a ser significativamente maior e conseqüentemente serviços de e-mail e acesso a páginas Web passariam a ocupar as redes de telefonia. No início da década dos anos 2000, o acesso à Internet começaria a se popularizar, trazendo a necessidade de interconectar diversos usuários ao redor do mundo e prover taxas de transmissão ainda maiores. Dessa forma, a terceira geração das redes de telefonia móvel (3G) avançou profundamente em padronizar os serviços independentemente de tecnologia. Tal avanço possibilitou os usuários a utilizarem serviços de conversação por vídeo e grandes transferências de dados. A evolução para a quarta geração (4G) acompanhou a tendência dos serviços que passaram a ser mais utilizados pelos usuários na Internet, os serviços de distribuição de vídeo. Com uma nova arquitetura completamente baseada no *Internet Protocol* (IP), os usuários passam a conseguir utilizar as redes de telefonia móvel para jogar e assistir vídeos de alta qualidade [33, 31, 3, 1, 2].

1.1 Contexto

Assim como em todas as outras gerações, a quinta geração das redes de telefonia móvel está evoluindo para se adequar às aplicações e serviços mais utilizados pelos usuários [32]. A popularização dos dispositivos móveis, assim como o rápido avanço de suas aplicações, traz enormes desafios para que a arquitetura das redes de telefonia móvel se adeque às novas demandas dos usuários móveis. O tráfego de dados móveis ao redor do mundo alcançou 84 Exabytes por mês em 2021, e espera-se que esse número chegue a 368 Exabytes por mês até o final de 2027, como mostra a Figura 1.1. Cada vez mais a comunicação se torna móvel, o número de clientes em redes móveis alcançou o número de 8 bilhões [5]. A Ericsson prevê que até o final de 2025, 45% de todo o tráfego móvel possua sua origem nas redes 5G com a cobertura de 65% da população global [5]. Aplicações como *e-health*, carros autônomos, cidades e casas inteligentes, sensoriamento industrial e ambiental, desafiam as redes de telefonia móvel com maiores capacidades de transmissão, atrasos na comunicação na ordem de poucos milissegundos, e computação de alto desempenho.

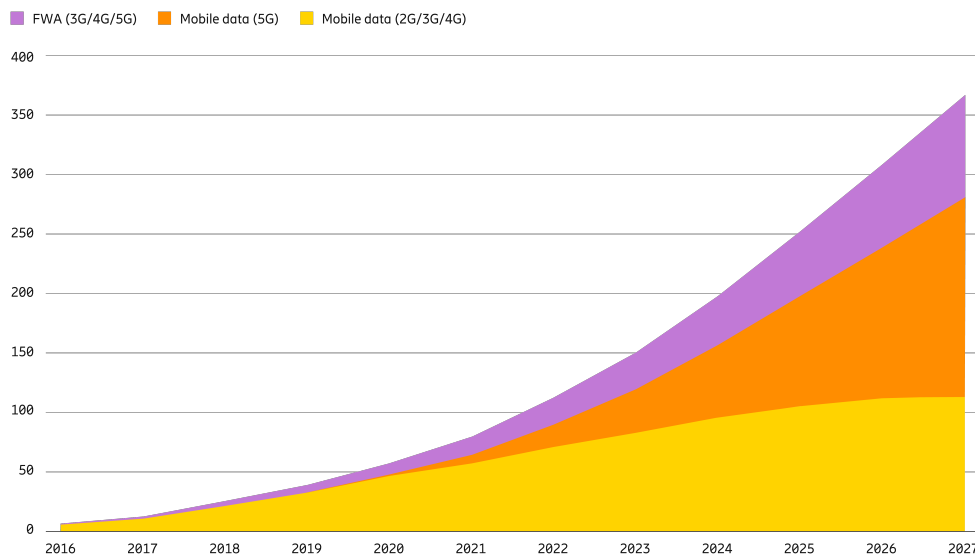


Figura 1.1: Tráfego de dados móveis ao redor do mundo em Exabytes por mês [4].

Na camada física, diversas propostas são pesquisadas para aumentar a eficiência espectral das redes de telefonia móvel. Uma das propostas baseia-se na redução do tamanho das células, ao mesmo tempo em que implanta-se um maior número de células. O aumento no número de células traz outros desafios. A densidade de usuários dentro de uma mesma célula aumenta a interferência intra-célula, aumentando a competição para o acesso ao meio. Além disso, o aumento no número de células amplifica a interferência inter-células. Para aumentar a capacidade espectral dentro da célula, pesquisa-se multiplicar o número de antenas através dos sistemas de Múltiplas Entradas e Múltiplas Saídas (*Multiple-Input*

and Multiple-Output - MIMO), e realizar as transmissões por ondas milimétricas [9]. Outra área de pesquisa para lidar com escassez do espectro licenciado é a implantação de microcélulas [80, 23]. Em algumas áreas urbanas, a implantação de pontos de acesso (*Access Points* - APs) Wi-Fi alcança a densidade de 1000 APs/km² [81]. Dessa forma, utilizar o espectro não-licenciado para descarregar o tráfego do espectro licenciado da macrocélula em pequenas células de redes heterogêneas (*Heterogeneous and Small Cell Networks* - HetSNets) pode aumentar a vazão da rede celular móvel [67].

Fora da camada física, duas arquiteturas são estudadas para superar as limitações de capacidade impostas pela escassez espectral, a *Cloud Radio Access Network* - C-RAN [54, 49], e a *Multi-access Edge Computing* - MEC [8, 38]. A ideia por trás das C-RANs está em centralizar todas as unidades de processamento de banda-base (*Baseband Processing Unit* - BBU) em um ambiente virtualizado e compartilhar os BBUs entre as estações base (*Base Station* - BS) por demanda. Já a arquitetura MEC, utiliza nuvens locais e/ou regionais que estão fisicamente próximas das antenas com o objetivo de complementar a arquitetura C-RAN. Dessa forma, nós que estão na borda da rede celular atuam como um poder computacional adicional para satisfazer requisições de aplicações sensíveis ao atraso, ou para pré-processar dados a serem enviados para a nuvem e economizar a largura de banda dos enlaces [38, 76].

1.2 Hipóteses

Espera-se que o tamanho das células nas redes 5G reduzam, e com o uso das pequenas células, a mobilidade dos usuários exige *handoffs* frequentes causando a reconexão do usuário ao servidor de vídeo. Tais reconexões impactam negativamente na qualidade de experiência do usuário devido ao atraso da reconfiguração de rota realizada pelo núcleo das redes de telefonia móvel. Com o uso de ambientes virtuais próximos às estações base na borda das redes 5G é possível utilizar *caches* atuando como funções de rede virtuais (*Virtual Network Function* - VNF) para reduzir o uso do enlaces pertencentes ao núcleo da rede de telefonia móvel, assim como prover os conteúdos requisitados pelo usuário com menor atraso.

Além disso, diferenciar quais conteúdos são populares e quais não são populares é de fundamental importância como mostra o *Media Cloud*. O *Media Cloud* é uma rede de distribuição de conteúdos virtual e elástica presente na borda da rede que realiza o armazenamento antecipado dos conteúdos populares, e demonstra que seu mecanismo é capaz

de aumentar a qualidade de experiência de seus usuários [41]. Qiu e Cao demonstram que grande parte dos enlaces da infraestrutura das redes de telefonia móvel é ocupada pela transmissão do mesmo subconjunto de conteúdos populares [60]. Dessa forma, espera-se que, ao armazenar os conteúdos mais populares na borda da rede de telefonia móvel, reduza-se o atraso de recuperação dos conteúdos pelos usuários ao atender grande parte das requisições sem encaminhá-las ao provedor do conteúdo. Um estudo realizado pela Intel demonstra que armazenar conteúdos populares na borda da rede de telefonia móvel é capaz de reduzir em até 30% o tempo de espera para iniciar a reprodução de um vídeo, reduzir em até 35% o tráfego de pico nos enlaces da infraestrutura, e aumentar em até 15% a disponibilidade da largura de banda dos rádios [40].

1.3 Contribuições

A primeira contribuição desta tese encontra-se na formulação de um problema de programação linear inteira mista para a obtenção de conteúdos a partir da requisição dos usuários em uma rede de telefonia móvel através de VNFs atuando como *caches* na borda da rede. O primeiro objetivo da formulação proposta encontra-se em estudar como os *cache* VNFs são distribuídos pelos ambientes virtuais locais e/ou regionais disponibilizados pela arquitetura MEC. O segundo objetivo encontra-se em estudar como os conteúdos requisitados são distribuídos entre os *cache* VNFs instanciados pelos ambientes virtuais para atender as requisições dos usuários através da própria infraestrutura da rede de telefonia móvel. A metodologia de avaliação considera um modelo de geração de grafos aleatórios que se aproxima das topologias encontradas nas redes de telefonia móvel. Além disso, as características dos enlaces nas topologias avaliadas, como o atraso de propagação e a largura de banda, são atribuídos através de um perfil real extraídos das redes de telefonia móvel. A popularidade das requisições dos usuários é extraída de um conjunto de dados disponibilizados pela Netflix. Os resultados de simulação mostram que o fator predominante é o atraso de instanciação dos *caches*. Quando o atraso de instanciação aproxima-se do atraso de transmissão dos conteúdos, há uma distribuição mais homogênea dos conteúdos entre os *cache* VNFs. Quando o atraso de instanciação dos *cache* VNFs é muito superior ao atraso de transmissão dos conteúdos, poucos *caches* são instanciados e encontram-se nos ambientes virtuais que possuem maior vazão para a sua instanciação. Nesses casos, cerca de 70% das requisições são atendidas pelos *cache* VNFs instanciados nos ambiente virtuais com maior vazão para instanciação dos *caches*.

A segunda contribuição desta tese encontra-se na proposta de um agente baseado em

aprendizado de máquina por reforço profundo cujo objetivo é armazenar os conteúdos populares em *caches* antes de serem requisitados pelos usuários e satisfazê-los através dos *caches* que se encontram próximos às estações bases. Visto que os resultados da formulação de programação linear inteira mista demonstram que o atraso de instanciação dos *caches* é um fator que deve ser considerado ao aproximar os conteúdos para a borda da rede de telefonia móvel, considera-se que cada estação base está conectada a um ambiente virtual que possui servidor *cache* já instanciado. O agente realiza a estimativa do estado de requisição das estações base e prevê quais conteúdos serão os mais populares naquele determinado momento. Em seguida, seleciona um subconjunto dos conteúdos estimados como os mais populares e os armazena em um servidor *cache* para que a maioria das requisições dos usuários sejam satisfeitas sem a necessidade de serem encaminhadas até o provedor do conteúdo pela Internet. A metodologia de avaliação do agente proposto considera o mesmo modelo de grafo e perfil dos enlaces da metodologia empregada nas avaliações da formulação programação linear inteira mista. Uma vez que o agente emprega uma rede neural artificial para aproximar os estados de requisição estimados para um conjunto de ações, se faz necessária uma etapa de treinamento do modelo do agente e, posteriormente, uma etapa de avaliação dos modelos treinados. Ambas as etapas são analisadas para duas funções de recompensa. Os resultados de desempenho demonstram que agente proposto é capaz de capturar as flutuações espaço-temporais da popularidade dos conteúdos [57] e produzir até 65% de acerto em *cache*, assim como reduzir em 61% o atraso de recuperação dos conteúdos requisitados em relação ao atraso de recuperação pelo provedor do conteúdo.

1.4 Organização

O restante desta tese está organizada da seguinte forma. O Capítulo 2 apresenta a evolução das redes de telefonia móvel, assim como os componentes que as definem. O Capítulo 3 apresenta as novas arquiteturas propostas para a quinta geração das redes de telefonia móvel, as redes 5G. O Capítulo 4 introduz o problema abordado para a recuperação de conteúdos multimídias nas redes 5G através de *cache* VNFs, a formulação proposta para simulação, e discute os resultados obtidos. O Capítulo 5 introduz os conceitos de aprendizado de máquina por reforço e como as redes neurais artificiais são utilizadas para realizar o treinamento do agente. O Capítulo 6 apresenta o agente proposto, incluindo a formulação do modelo de sistema na qual o agente atua, o seu espaço de estados e ações, e as funções de recompensa propostas. O Capítulo 7 apresenta e discute as avaliações do

agente sobre o seu treinamento, a validação do treinamento, e de desempenho na rede de telefonia móvel. O Capítulo 8 apresenta as conclusões da tese, as publicações derivadas, e os trabalhos futuros a serem desenvolvidos.

Capítulo 2

As Redes de Telefonia Móvel

O padrão *Long Term Evolution* (LTE) [19] foi projetado em uma colaboração de diferentes entidades da área de telecomunicações e nomeado como *Third Generation Partnership Project* (3GPP) [58]. O LTE é uma evolução de um sistema mais antigo da 3GPP conhecido como *Universal Mobile Telecommunication System* (UMTS) [66], que já seria uma evolução do *Global System for Mobile Communications* (GSM) [66].

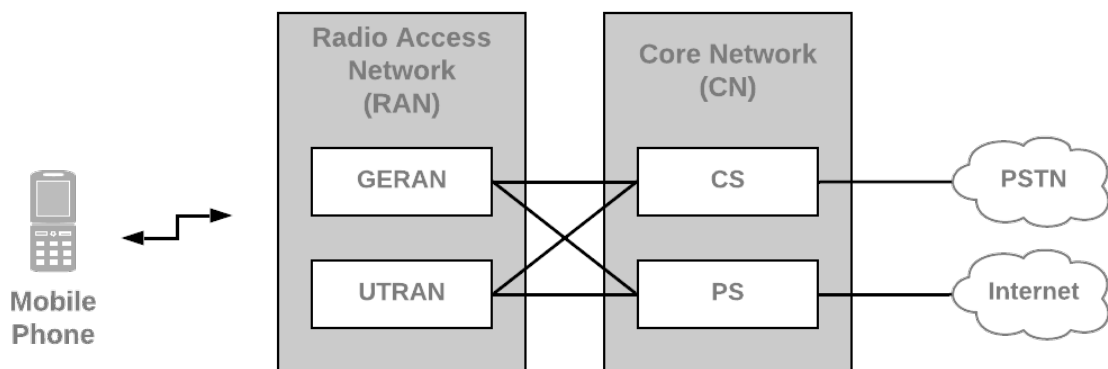


Figura 2.1: As arquiteturas UMTS e GSM.

Tanto o GSM como o UMTS compartilham a mesma arquitetura de rede, ilustrada na Figura 2.1. A arquitetura é basicamente composta por três componentes, o núcleo da rede (*Core Network* - CN), a rede de acesso via rádio (*Radio Access Network* - RAN) e os telefones móveis (*Mobile Phone*). O CN é composto por dois domínios. O domínio chamado *Circuit Switch* (CS) transporta ligações telefônicas através da região geográfica coberta da mesma maneira que ocorre no sistema tradicional de linha fixa. A comunicação é primeiro estabelecida até a rede de comutação telefônica na qual a ligação é comutada para outros operadores de rede. O domínio conhecido como *Packet Switched*

(PS) transporta fluxos de dados, como páginas Web e e-mails, entre os MPs e a Internet.

O transporte realizado em cada domínio não ocorre da mesma maneira. No CS, emprega-se a técnica conhecida como comutação por circuito, na qual uma conexão dedicada de duas vias é estabelecida para cada MP individualmente. Essa conexão dedicada transporta a informação com taxas de transmissão e atrasos constantes. No PS, emprega-se a técnica de comutação por pacote chamada *General Packet Radio Services* (GPRS), na qual um fluxo de dados é segmentado em pacotes e etiquetados com os endereços de origem e destino. Os pacotes são transmitidos entre roteadores, os quais armazenam uma tabela indexando endereços de destino e o seu respectivo vizinho que também conhece o destino. Os roteadores, ao lerem a etiqueta, dão continuidade ao caminho do pacote ao o encaminharem para o vizinho indexado em sua tabela. Os recursos da rede passam a ser compartilhados entre os diferentes MPs, com as transmissões deixando de serem dedicadas e passando a ter taxas de transmissão e atrasos variáveis.

2.1 *Radio Access Network* - RAN

A Figura 2.2 ilustra a arquitetura da rede de acesso a rádio UMTS. A RAN gerencia a comunicação via rádio entre o CN e os MPs. Na GSM a rede de acesso via rádio era chamada de *GSM EDGE Radio Access Network* (GERAN) e na UMTS chamada de *UMTS Terrestrial Radio Access Network* (UTRAN). No UMTS, o componente mais importante dentro da RAN é a estação base, denominada de *Node B*. Cada *Node B* possui um ou mais conjuntos de antenas, as quais são utilizadas para estabelecer a comunicação com os MPs em um ou mais setores. Nesse sentido, o conceito de célula pode possuir dois significados distintos. Em alguns países, célula e setor são um só, enquanto que em outros países a célula pode significar um conjunto de setores controlados por uma única estação base [66]. As *Node Bs* são agrupadas através do *Radio Network Controller* (RNC). O RNC tem como objetivos passar as informações de voz e os pacotes de dados entre a *Node B* e o CN e controlar a comunicação via rádio através de mensagens de sinalização que são transparentes para o MP.

As células são limitadas pelo alcance máximo que as estações base conseguem escutar os seus transmissores. Sua capacidade de transmissão também é limitada, vista que a banda disponível é compartilhada entre todos os MPs que pertencem à célula. Tais limitações levam a criação de diferentes tipos de células. A macrocélula provê ampla cobertura com o alcance de alguns quilômetros. A microcélula possui a cobertura de

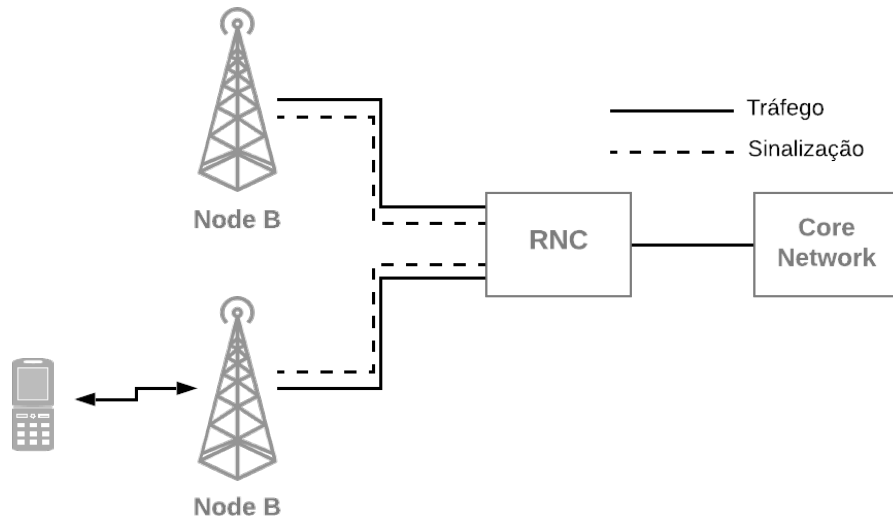


Figura 2.2: Arquitetura da rede de acesso a rádio UMTS.

algumas centenas de metros e costuma ser implantada em áreas urbanas de grande densidade. A pico-célula costuma ser implantada em grandes ambientes internos, como um shopping ou grandes ambientes empresariais e possui coberturas de algumas dezenas de metros. A femto-célula costuma ser implantada em ambientes internos pequenos, como casas e escritórios, com poucos metros de cobertura. Quando os MPs se locomovem de uma região da rede celular para outra, sua comunicação com a célula na qual está conectado é interrompida e transferida para outra célula. Essa mudança de células pode ser realizada através de duas técnicas, o *handover* ou o *cell reselection*.

O *cell reselection* é utilizado quando o MP está no estado ocioso. O seu objetivo está em maximizar a bateria do MP e minimizar a carga de sinalização na rede celular. A técnica é realizada com o MP acordando em cada ciclo descontínuo de recepção, no qual o MP monitora a frequência na qual está alocado por mensagens de rastreamento (*paging messages*). Essas mensagens são utilizadas pelo MP para medir a potência do sinal de referência (*Reference Signal Received Power* - RSRP) da sua estação base servidora. Se o RSRP estiver acima de um limiar, o MP continua associado a sua estação base servidora e não realiza a medição do RSRP de estações base vizinhas. Dessa forma, o MP minimiza o número de medidas a serem realizadas no período no qual está acordado e conseqüentemente reduz a utilização da bateria. Se o RSRP estiver abaixo do limiar, o MP começa a realizar medições das estações base vizinhas e troca para a estação base melhor classificada.

O *handover* é utilizado quando o MP está no estado conectado. Diferentemente do

procedimento realizado no *cell reselection*, no estado conectado o MP pode estar transmitindo a uma taxa alta. Visto que o MP está ativo, o procedimento de *handover* é iniciado pela estação base ao enviar uma mensagem de reconfiguração indicando a lista de medidas desejadas, a identificação das medidas e o momento no qual o MP deve reportar as medidas para a estação base. Ao receber o relatório de medidas do MP, a estação base servidora decide a nova estação base do MP e requisita à nova estação base para assumir o controle do MP. A nova estação base examina o pedido e o aceita ou não dependendo dos seus recursos disponíveis. Ao aceitar, a nova estação base transmite uma mensagem de reconfiguração para a estação base servidora indicando a configuração para que o MP consiga se comunicar. A estação base servidora encaminha a mensagem de reconfiguração ao MP, ao mesmo tempo em que envia uma mensagem de transferência de custódia para a nova estação base com os pacotes recebidos do MP que estão fora de ordem, assim como as mensagens transmitidas ao MP que ainda não foram reconhecidas. O MP se reconfigura ao receber a mensagem da estação base servidora e inicia sua comunicação com a nova estação base.

2.2 Core Network - CN

A arquitetura do CN, ilustrada pela Figura 2.3, é composta essencialmente do (i) *Media Gateway* (M-GW), (ii) *Mobile Switching Centre Server* (MSC), (iii) *Gateway GPRS Support Node* (GGSN), (iv) *Serving GPRS Support Node* (SGSN), (v) *Home Subscriber Server* (HSS). No CS, o M-GW é o responsável por rotear as ligações telefônicas de uma região da rede celular para outra, enquanto que o MSC gerencia as mensagens de sinalização de configuração das ligações, assim como das desconexões. No PS, o GGSN realiza a comunicação com o mundo exterior ao da rede celular, enquanto que o SGSN realiza o roteamento dos fluxos de dados entre as estações base e o GGSN. O SGSN também é responsável por gerenciar a sinalização das mensagens de configuração e desconexão dos fluxos de dados. O HSS é a base de dados central que contém a informação sobre todos os assinantes da rede celular e é compartilhado entre os dois domínios.

2.3 Long Term Evolution - LTE

O padrão *Long Term Evolution* é a evolução do acesso à rádio e da infraestrutura conectada ao acesso à rádio. O acesso à rádio baseia-se na evolução do UTRAN, chamado E-UTRAN, que atualiza as tecnologias da camada física. Já a infraestrutura é a evolução

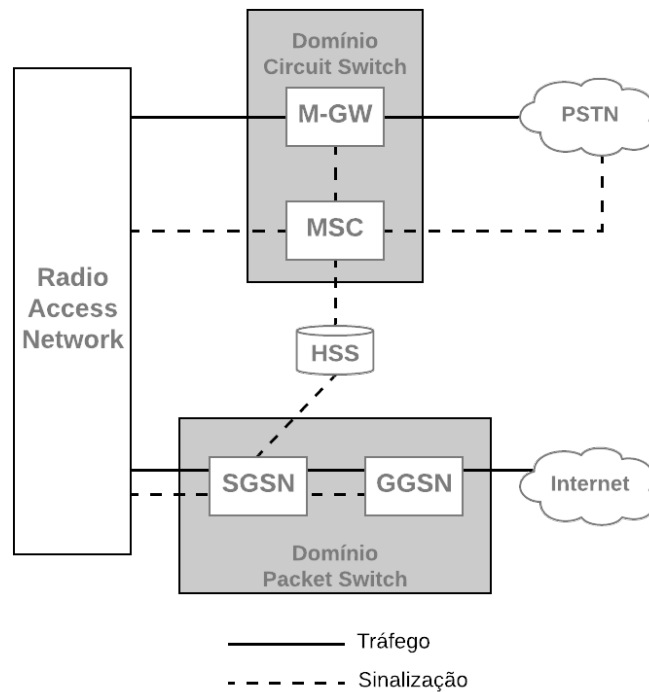


Figura 2.3: Arquitetura do núcleo das redes UMTS.

da arquitetura LTE, cujos componentes podem ser resumidos em (i) equipamento de usuário (*User Equipment* – UE), (ii) rede de acesso a rádio (*Evolved UMTS Terrestrial Radio Access Network* - E-UTRAN) e (iii) núcleo da rede (*Evolved Packet Core* - EPC). Dessa forma, a rede LTE é composta pelo seu núcleo (EPC) e pela sua rede de acesso (E-UTRAN).

2.3.1 *Evolved UMTS Terrestrial Radio Access Network* - E-UTRAN

A Figura 2.4 ilustra a arquitetura do *Evolved UMTS Terrestrial Radio Access Network*. O E-UTRAN realiza os serviços de comunicação de rádio entre o UE e o EPC através de um único componente, o *evolved Node B* (eNodeB). O eNodeB é a estação base responsável por gerenciar a comunicação de um UE dentro de uma ou mais células. Um UE comunica-se somente com uma única eNodeB por vez, e essa eNodeB é chamada de eNodeB servidora.

A eNodeB possui duas principais responsabilidades. A primeira responsabilidade está em transmitir sinais de rádio para os UEs através da banda reservada para o seu canal de descida (*downlink*), e receber sinais de rádio dos UEs através da banda reservada para o canal de subida (*uplink*), utilizando as funções de processamento de sinais analógicos

e digitais presentes nas interfaces de rádio LTE. A segunda responsabilidade está em controlar operações de baixo nível, como o envio de mensagens de sinalização de *handover* das comunicações em andamento.

As eNodeBs podem estar tanto conectadas ao EPC como também podem estar diretamente conectadas a outras eNodeBs. A interface de comunicação entre eNodeBs é utilizada unicamente com o propósito de agilizar a sinalização de *handover* e encaminhar os pacotes remanescentes ao UE em sua nova célula após sua locomoção. A conexão entre eNodeBs é opcional no padrão LTE, visto que todo o processo de *handover* pode ser realizado via o EPC.

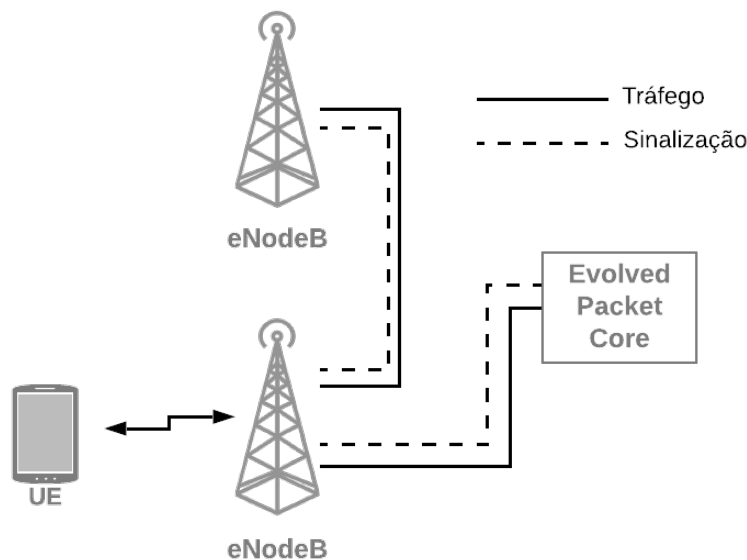


Figura 2.4: Arquitetura da rede de acesso a rádio LTE.

2.3.2 Evolved Packet Core - EPC

A arquitetura do EPC está ilustrada na Figura 2.5 e o seu papel é controlar o estado dos UEs e de estabelecer os fluxos de datagramas IP respeitando a qualidade de serviço oferecida entre o UE e o roteador de saída para as redes externas. Para realizar essa tarefa, o EPC possui diversos componentes lógicos, cada um com os seus respectivos serviços [19]:

- **Packet Data Network Gateway (P-GW):** É o nó que atua como ponto de contato com o mundo exterior. Cada P-GW provê endereçamento IP aos UEs e estabelece suas conexão com as redes externas, como a Internet. É considerado a âncora final do sistema celular.

- **Serving Gateway (S-GW):** É o nó que atua como roteador entre as eNodeBs e o P-GW dentro do EPC. Uma típica rede de telefonia móvel é composta por diversos S-GWs, cada um cobrindo uma região geográfica distinta. Cada UE é atribuído à um único S-GW, que pode ser modificado caso o UE se locomova longe o suficiente.
- **Mobility Management Entity (MME):** É o principal nó de controle dentro do EPC. Sua função essencial é gerenciar a sinalização entre os UEs e o EPC, ao realizar a autenticação do UE e gerenciar sua mobilidade ao requisitar e liberar os recursos utilizados no S-GW e no eNodeB. Além disso, o MME gerencia o perfil de assinaturas dos UEs e seus serviços de conectividade. Assim como o S-GW, os MMEs são associados a regiões geográficas e cada UE é atribuído à um único MME, que pode ser modificado caso o UE se locomova longe o suficiente.
- **Home Subscriber Server (HSS):** É a base de dados central que contém as informações de todos os assinantes da rede de telefonia móvel. O HSS possui registrado todos os P-GW aos quais o UE pode se conectar e em qual o UE está atualmente conectado. Além disso, o HSS é responsável por validar a autenticação do UE que inicia o processo de conexão com a rede.

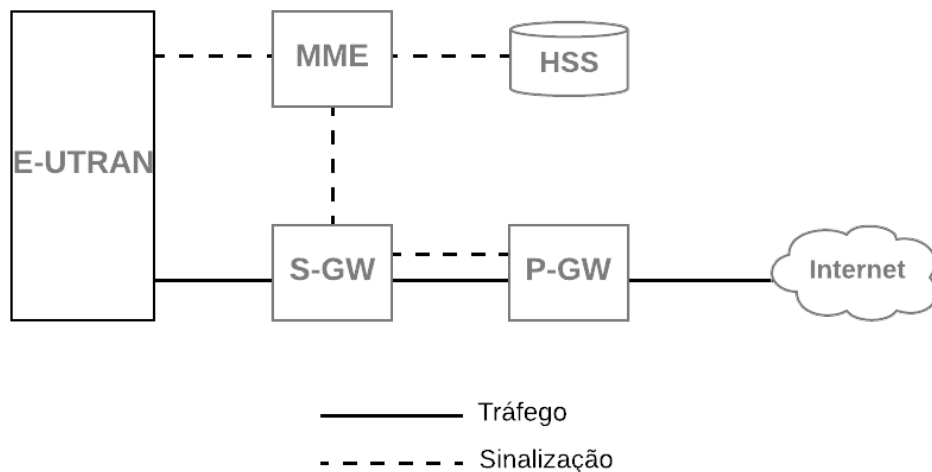


Figura 2.5: Arquitetura do núcleo da rede LTE.

2.4 New-Generation Radio Access Network - NG-RAN

A arquitetura que compõe as redes de telefonia móvel LTE é composta por componentes lógicos, como as *eNodeBs*, o S-GW, o P-GW, o MME, e o HSS. Cada componente é geralmente composto de hardware e software proprietários, acoplados em um único dispositivo. Já a arquitetura das redes de telefonia móvel chamadas de *Next-Generation Networks* (NGN) possui como objetivo permitir a “softwarização” dos componentes lógicos. A “softwarização” define que as funções da rede, incluindo os protocolos e os serviços, possuam o software implementado de forma independente do hardware que o executa [14]. Além disso, na arquitetura da rede de acesso das redes NGN, denominada *New-Generation Radio Access Network* (NG-RAN), os nós NG-RAN são lógicos. Um nó NG-RAN pode ser uma estação base 5G, chamada gNB, assim como pode ser uma estação base de nova geração 4G, chamada ng-eNB [48]. A Figura 2.6 ilustra a visão geral da arquitetura NG-RAN.

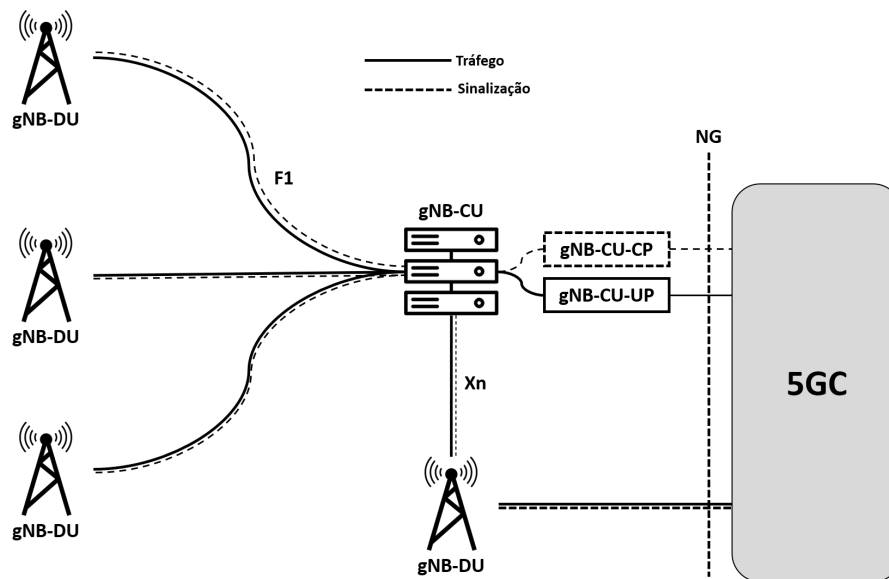


Figura 2.6: Arquitetura da NG-RAN.

O nó lógico NG-RAN se comunica com o núcleo da rede de telefonia móvel 5G, chamado de 5G *Core Network* (5GC), através da interface lógica *New Generation* (NG). Através da interface NG, as redes de telefonia móvel 5G são capazes de conectar os UEs com outras redes externas através de um serviço chamado sessão do protocolo da unidade de dados (*Protocol Data Unit* - PDU). Uma sessão PDU é composta por uma sequência de túneis NG estabelecidos pelos protocolos UDP/IP da arquitetura TCP/IP com o 5GC e por uma ou mais portadoras de rádio estabelecidas entre o nó NG-RAN e os UEs [17].

A comunicação entre os nós lógicos NG-RAN é realizada através da interface Xn [6] para facilitar as operações de mobilidade e as funções de gerenciamento, como a de coordenação de interferência entre as células.

A arquitetura NG-RAN também incorpora o conceito de arquitetura distribuída, visto que as gNBs são divididas em duas unidades lógicas, a unidade gNB-central (gNB-*Central Unit* - gNB-CU) e a unidade gNB-distribuída (gNB-*Distributed Unit* - gNB-DU). Dessa forma, a gNB-CU é responsável pelo controle de uma ou mais gNB-DUs através da interface F1 e é dividida em duas partes. As funções do plano de controle são gerenciadas pelo gNB-CU-CP, enquanto que as funções do plano de dados são gerenciadas pelo gNB-CU-UP, seguindo o paradigma das redes definidas por software. A Figura 2.7 ilustra as interfaces e os serviços presentes nas gNBs.

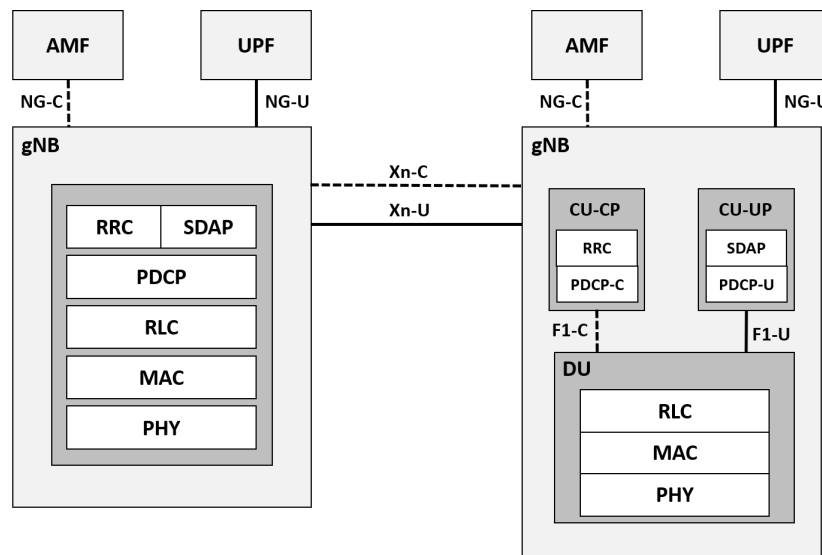


Figura 2.7: Interfaces e serviços da gNB.

Dentre os serviços que encontram-se na gNB-CU, o controle dos recursos de rádio (*Radio Resource Control* - RRC) pertence ao plano de controle. O RRC é o protocolo responsável por sinalizar o estabelecimento e a liberação de conexões; difundir informações do estado da rede, estabelecer, reconfigurar, e liberar as portadoras de rádio; realizar os procedimentos de mobilidade; realizar os controles de potência. Já o serviço oferecido pelo protocolo de adaptação do serviço de dados (*Service Data Adaption Protocol* - SDAP) pertence ao plano de dados e realiza o mapeamento dos pacotes e a portadora de rádio na qual os pacotes são transmitidos com o objetivo de marcá-los apropriadamente e garantir a qualidade de serviço contratado. O serviço oferecido pelo protocolo de convergência dos pacotes de dados (*Packet Data Convergence Protocol* - PDCP) encontra-se tanto no plano de dados como no plano de controle do gNB-CU, e tem como objetivo oferecer serviços

relacionados à segurança e compressão dos cabeçalhos dos pacotes.

A gNB-DU oferece serviços voltados para a gerência de recursos relacionados ao meio físico. O controle dos enlaces de rádio (*Radio Link Control* - RLC) garante a entrega confiável dos fluxos de dados e gerencia a segmentação dos dados. O serviço de controle de acesso ao meio (*Medium Access Control* - MAC) realiza o escalonamento dos fluxos de dados entre os UEs e o gNB. A camada física (*Physical Layer* - PHY) é responsável pelas funções de processamento dos sinais analógicos e digitais.

Pelo núcleo da rede 5G, a função de acesso e mobilidade (*Access and Mobility Function* - AMF) encontrada no 5GC é responsável por comunicar-se com o gNB-CU-CP através da interface NG e controlar as funções relacionadas à mobilidade de todo o núcleo da rede, isto é, considerando todos os nós NG-RAN. Além disso, realiza a autenticação dos UEs na rede de telefonia móvel. A AMF contém parte das funcionalidades presentes no componente MME da arquitetura LTE.

A função do plano de usuário (*User Plane Function* - UPF) encontrada no 5GC é responsável por comunicar-se com o gNB-CU-UP e gerenciar as sessões PDU dos UEs com as redes externas e realizar o encaminhamento dos pacotes. Realiza procedimentos de ancoragem para que ocorram os *handovers*, além de garantir as políticas de tráfego e de qualidade de serviço. Dessa forma, podem haver múltiplos UPFs associados a um único UE. A UPF contém parte das funcionalidades encontradas nos componentes S-GW e P-GW da arquitetura LTE.

Capítulo 3

Arquiteturas Propostas para as Redes 5G

A eficiência espectral no padrão LTE se aproxima do limite descoberto por Shannon [55]. As maneiras mais estudadas para aumentar a capacidade das redes de telefonia móvel está em aumentar o número de células, criar estruturas complexas e heterogêneas de pequenas células [39] ou através de técnicas para multiplicar o número de antenas simultaneamente utilizadas por múltiplos usuários na mesma frequência [30, 36]. O problema com ambas as propostas está no aumento do nível de interferência inter-células e intra-células, além do aumento no custo de implantação.

O custo de implantação das redes de telefonia móvel inclui as despesas de capital (*CAPital EXpenditure* - CAPEX) e as despesas de operação (*OPerating EXpenditure* - OPEX). As despesas relacionadas ao CAPEX são associadas à construção e planejamento da rede de telefonia móvel, como os equipamentos de rádio-frequência, equipamentos de banda base, licenças de software, aluguéis de linhas de transmissão e mão de obra. As despesas relacionadas ao OPEX cobrem as necessidades de operação, manutenção e atualização da rede de telefonia móvel. Dessa forma, ambos os custos de CAPEX e OPEX aumentam conforme novas estações base são implantadas. As estações base são os equipamentos mais caros da infraestrutura de uma rede de telefonia móvel, ao mesmo tempo que são os componentes de maior consumo energético, chegando a consumir 72% de toda a energia necessária [54].

Uma forma de reduzir os custos nas redes de telefonia móvel, e simplificar sua operação e manutenção, está em transformar os serviços oferecidos na infraestrutura em serviços de computação em nuvem [38, 50]. Duas arquiteturas propostas para as redes 5G exploram os conceitos de computação em nuvem para alcançar tal objetivo, a *Cloud Radio Access Network* (C-RAN) [15] e a *Multi-access Edge Computing* (MEC) [51]. A C-RAN propõe

o desacoplamento do processamento de banda-base das funcionalidades de rádio. As funcionalidades de rádio (*Radio Functionalities - RF*) são mantidas nas estações base, enquanto que o processamento de banda-base (BBU) é agrupado e virtualizado, com as estações base consumindo os BBUs por demanda. A MEC propõe a introdução de plataformas virtualizadas na arquitetura das redes de telefonia móvel com o objetivo de executar aplicações móveis na sua borda. O objetivo é aproximar os serviços da computação em nuvem aos usuários da rede de telefonia móvel e fornecer respostas com menores latências às aplicações, assim como prover funcionalidades de pré-processamento e computações cientes de contexto [52].

O restante do capítulo descreve em detalhes os conceitos técnicos da arquitetura C-RAN [15] na Seção 3.1, e da arquitetura MEC [51] na Seção 3.2. A arquitetura C-RAN é assumida nesta tese como responsável pela gerência dos recursos nas estações base em função da quantidade de usuários que está associado a ela. Enquanto que a arquitetura MEC é responsável por manter ambientes virtuais através de servidores próximos às estações base nos quais funções de redes virtuais podem ser instanciados para satisfazer as requisições de aplicações multimídia sem que seja necessário acessar uma rede externa, como a Internet.

3.1 *Cloud Radio Access Network - C-RAN*

A C-RAN é definida como uma arquitetura para a rede de acesso nas redes de telefonia móvel na qual os recursos de banda-base são agrupados, compartilhados entre as *eNodeBs* e servidos por demanda. As principais funções de uma *eNodeB* podem ser resumidas nas funcionalidades de rádio (RF) e no processamento de banda-base (BBU). Na arquitetura tradicional das redes de telefonia móvel, ambos RF e BBU são integrados na *eNodeB*. O módulo da antena é geralmente localizado próximo ao módulo do rádio, conectados por cabos coaxiais. Essa arquitetura é comumente encontrada nas redes 1G e 2G e está ilustrada na Figura 3.1.

Uma nova arquitetura de estação base é introduzida nas redes 3G baseada na separação da unidade de rádio (*Remote Radio Head - RRH*) da unidade de processamento de banda-base (BBU). O RRH passa a ser responsável por realizar o processamento digital, conversão do sinal analógico para o digital, amplificar a potência e os filtros do sinal [42]. O BBU e o RRH passam a estar conectados por um enlace de fibra óptica, o que torna possível separá-los à uma distância de até 40 km. Dessa forma, o BBU pode ser colocado

em locais estratégicos para que sejam realizados sua manutenção e atualização, ilustrado pela Figura 3.2.

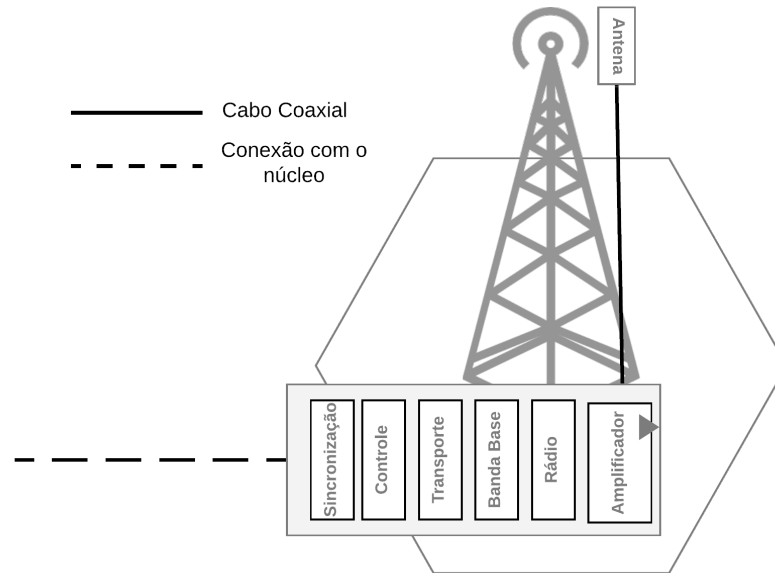


Figura 3.1: Arquitetura tradicional de uma estação base em macro-células.

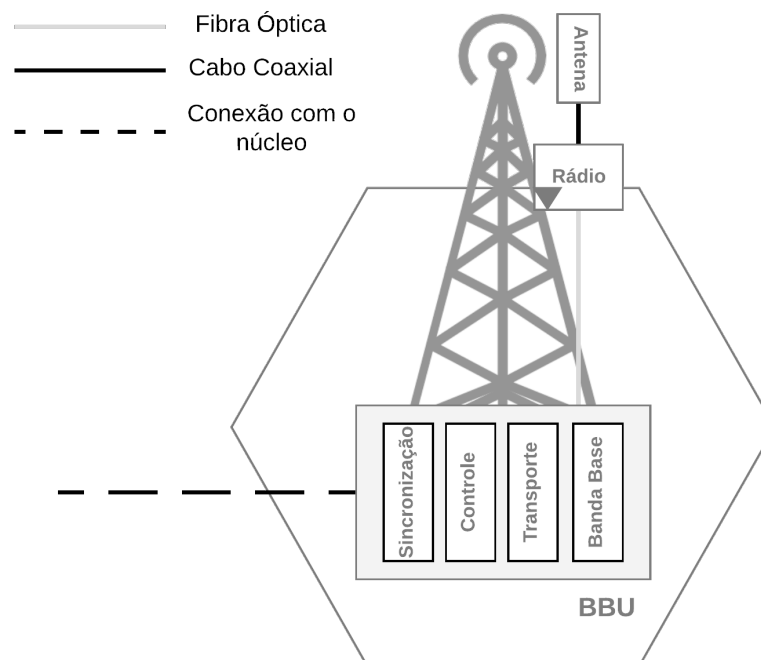


Figura 3.2: Arquitetura de uma estação base com RRH.

Na C-RAN, os BBUs são centralizados em uma entidade virtualizada denominada *BBU Pool*. O *BBU Pool* é um *cluster* de virtualização no qual os BBUs são compartilhados

por demanda entre as estações base para realizar o processamento de banda-base. O conceito das C-RANs foi introduzido pela IBM [49] como *Wireless Network Cloud* com o objetivo de propor um sistema de comunicação sem fio distribuído [84]. A proposta projeta uma arquitetura de rede de telefonia móvel na qual um usuário é capaz de se comunicar com uma densa quantidade de antenas distribuídas cujos sinais são processados em centros de processamento distribuídos. A Figura 3.3 ilustra a arquitetura de uma C-RAN.

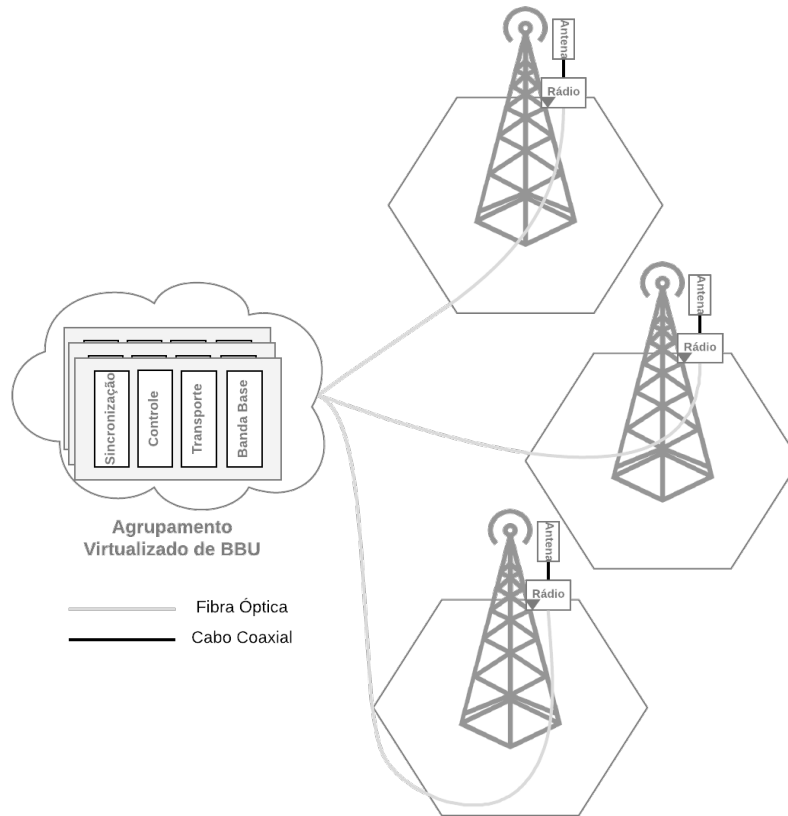


Figura 3.3: C-RAN com RRH.

3.2 Multi-access Edge Computing - MEC

O primeiro conceito a trazer um nível de computação na borda da rede celular foram os *cloudlets* [64]. A ideia por trás dos *cloudlets* está em instalar dispositivos com alto poder computacional em localizações estratégicas para oferecer processamento e armazenamento aos UEs. O *cloudlet* atua como um *hotspot* Wi-Fi, mas ao invés de oferecer conectividade com a Internet, oferece serviços de nuvem. Um conceito mais abrangente sobre computação na borda é introduzido pela Cisco, chamado de computação na névoa (*Fog Computing*), com o objetivo de possibilitar o processamento de uma densa quan-

tidade de UEs conectados na borda da rede de telefonia móvel [13]. A computação na névoa é um dos pilares para a Internet das Coisas (*Internet of Things*) [10], visto que o conceito cria e aproxima dos UEs uma rede de computação em nuvem distribuída que os interconecta. O problema da computação na névoa está na ausência de uma integração com a arquitetura da rede de telefonia móvel.

O conceito que introduz uma arquitetura que integra a computação na borda com a rede de telefonia móvel é proposto pelo Instituto de Telecomunicações Europeu de Padronização (*European Telecommunications Standards Institute* - ETSI), chamado de *Multi-access Edge Computing* [38]. O principal objetivo da arquitetura MEC é prover uma integração sem retalhos entre as funcionalidades da computação em nuvem na rede de telefonia móvel. A padronização da MEC ainda está em seus primeiros passos e sua terminologia para apresentar as especificações conceituais, arquiteturais e de seus elementos funcionais em rascunho [24]. A arquitetura de referência é composta por elementos funcionais e pontos de referência que permitem a interação entre os elementos, os quais não necessariamente são representados por nós físicos [25]. Elementos funcionais são encorajados a serem executados via software sobre uma infraestrutura virtualizada. A infraestrutura de virtualização pode ser interpretada como um centro de dados no qual máquinas virtuais (*Virtual Machines* - VMs) representam os elementos funcionais. Assume-se que algumas propostas arquiteturais do ETSI NFV serão reutilizados na arquitetura MEC final, visto que a ideia por trás das redes com funções virtuais (*Network Function Virtualization* - NFV) é virtualizar todos os nós que representam uma função de rede [26].

A Figura 3.4 ilustra a arquitetura de referência. As aplicações podem iniciar sua comunicação diretamente do UE ou a partir de um portal de serviços disponibilizado por terceiros. Tanto o UE como o portal de serviços interagem com o sistema MEC através do gerenciamento no nível do sistema. O gerenciamento no nível do sistema inclui um **Proxy de Gerência do Ciclo das Aplicações**, o qual é responsável por mediar as requisições, como a sua fase de iniciação, finalização, e realocação das aplicações para o **Sistema Operacional de Suporte**. O **Sistema Operacional de Suporte** é responsável por admitir as aplicações e as encaminhar para o **Orquestrador**. O **Orquestrador** é o núcleo do gerenciamento no nível do sistema, é o elemento funcional que possui a visão geral sobre todos os recursos dos serviços MEC, como processamento, armazenamento, e utilização dos enlaces de rede. Dessa forma, o **Orquestrador** aloca os recursos virtuais na arquitetura MEC para as aplicações em fase de iniciação ao considerar seus requisitos. Além disso, o **Orquestrador** também é responsável por ampliar ou reduzir a quantidade de recursos utilizada por uma aplicação em execução.

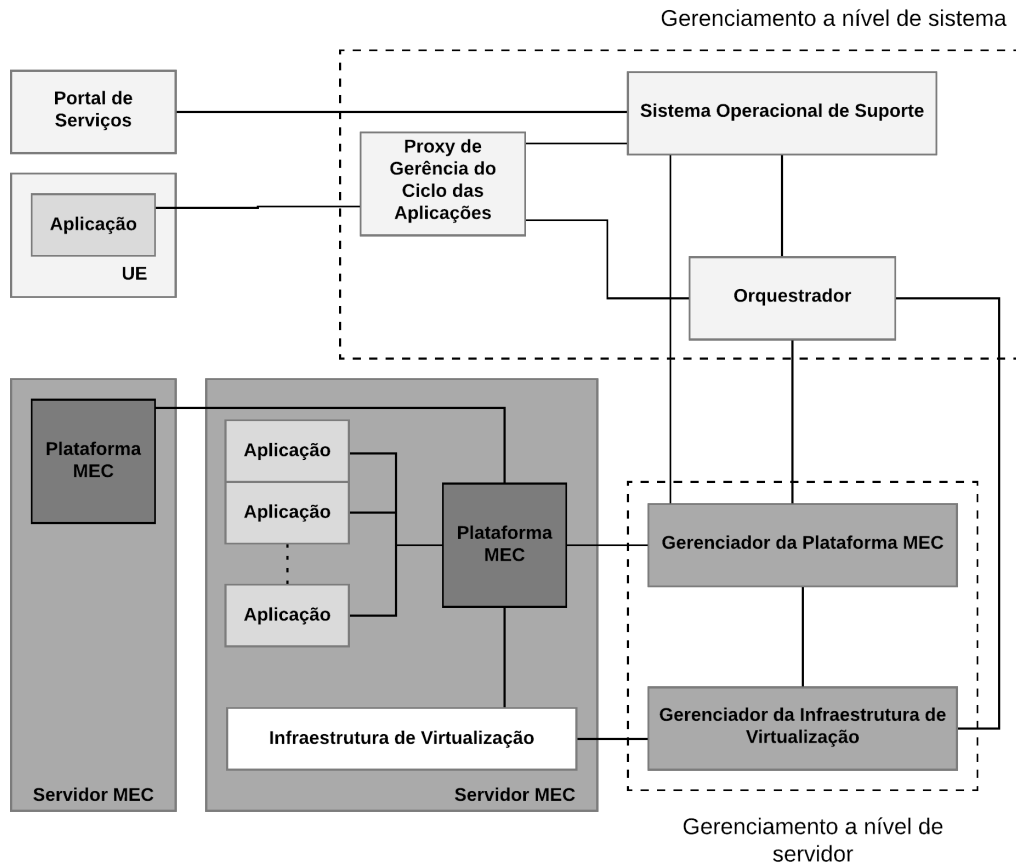
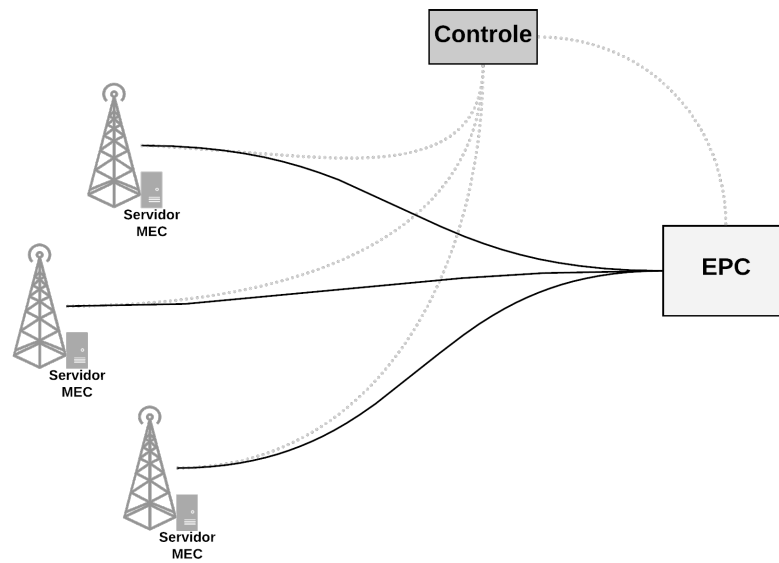
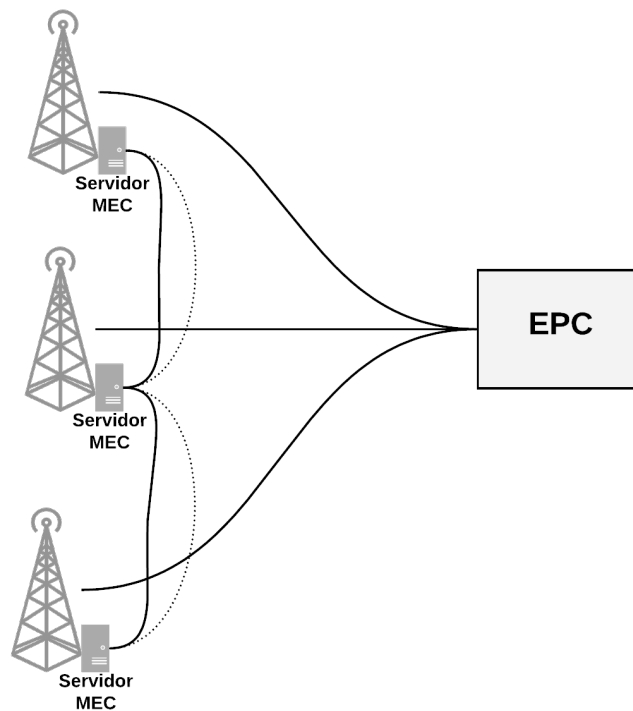


Figura 3.4: Arquitetura MEC de referência.

O gerenciamento no nível do sistema é interconectado ao gerenciamento no nível do servidor, o qual compõe o **Gerenciador da Plataforma MEC** e o **Gerenciador da Infraestrutura de Virtualização**. O **Gerenciador da Plataforma MEC** é responsável por receber as requisições do gerenciamento no nível do sistema e gerenciar o ciclo de vida das aplicações, suas regras e serviços autorizados e regras de tráfego. O **Gerenciador da Infraestrutura de Virtualização** é responsável pela alocação, gerenciamento, e liberação dos recursos requisitados fornecidos pela **Infraestrutura de Virtualização** localizada nos servidores MEC. Os servidores MEC representam os recursos virtualizados na arquitetura e hospedam as aplicações em execução através de VMs. A estrutura de implantação da arquitetura de referência não é especificada, o que possibilita múltiplas disposições dentro da rede de telefonia móvel. Uma opção de implantação é dispor os servidores MEC diretamente nas *eNodeBs* com a entidade responsável pela sinalização de controle separado, como ilustra a Figura 3.5(a). A Figura 3.5(b) ilustra a opção das *eNodeBs* se comunicarem diretamente para transferência de dados e sinalização de controle.



(a) Servidores MEC dispostos nas *eNodeBs* com entidade para gerenciar a sinalização de controle.

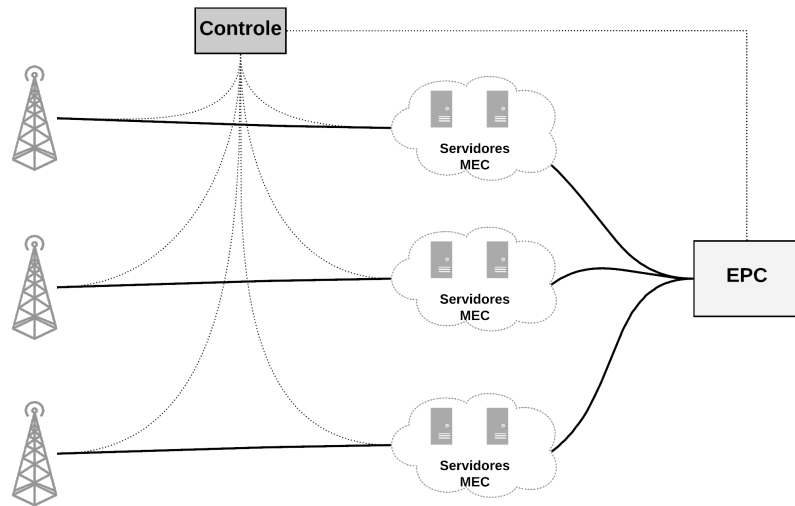


(b) Servidores MEC dispostos nas *eNodeBs* com fluxo de dados e de sinalização de controle diretos.

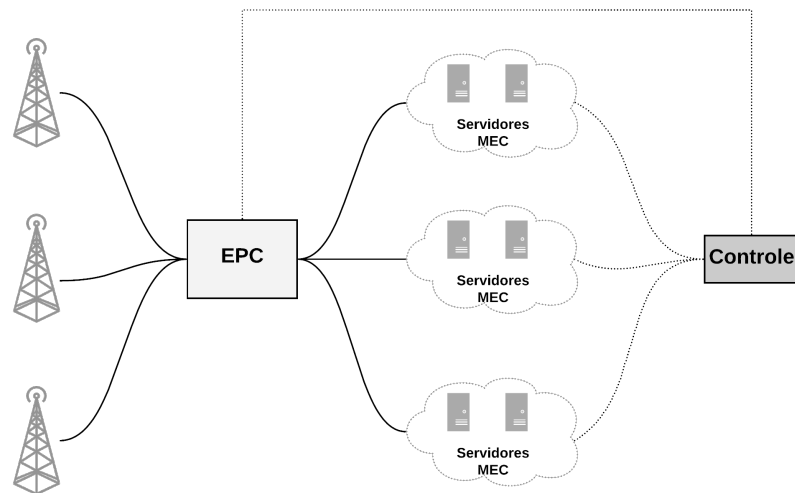
Figura 3.5: Servidores MEC dispostos diretamente nas *eNodeBs*.

Outra opção é dispor os servidores MEC fora das *eNodeBs*. Dessa forma, há opções de manter os servidores MEC mais próximos das *eNodeBs* ou mais afastados e a escolha do

posicionamento em relação à proximidade das *eNodeBs* depende de alguns fatores. Para oferecer baixas latência às aplicações de tempo real, o ideal é manter os servidores MEC mais próximos das *eNodeBs*, como mostra a Figura 3.6(a). Por outro lado, para acomodar um grande número de servidores MEC, e que permita sua escalabilidade e facilidade de manutenção, a configuração ideal ideal é a de mantê-los mais próximos do EPC, como mostra a Figura 3.6(b).



(a) Servidores MEC dispostos nas *eNodeBs* com entidade para gerenciar a sinalização de controle.



(b) Servidores MEC dispostos nas *eNodeBs* com fluxo de dados e de sinalização de controle diretos.

Figura 3.6: Servidores MEC dispostos fora das *eNodeBs*.

Capítulo 4

Recuperação de Conteúdos Multimídia na Arquitetura MEC

O objetivo desta tese é avaliar e propor políticas de recuperação de conteúdos multimídia a partir das requisições dos usuários através da infraestrutura das redes 5G. Atualmente, grande parte das mídias transferidas pela Internet é recuperada através dos protocolos *Hypertext Transmission Protocol* (HTTP) e *Transmission Control Protocol* (TCP) [35]. A capacidade da transmissão fim-a-fim varia em poucos segundos devido às condições do canal de rádio, operações de *handover* e do uso interno dos enlaces na rede de telefonia móvel. Como resultado dessas variações, o TCP não consegue estimar precisamente a capacidade de transmissão fim-a-fim e, logo, não só a qualidade de experiência (*Quality of Experience* - QoE) é afetada como os recursos da rede são sub-utilizados.

Muitas aplicações sensíveis ao atraso, como as de distribuição de vídeo, as baseadas em computação visual e inteligência artificial demandam respostas quase em tempo-real. Portanto, implantar serviços e recursos próximos aos usuários móveis pode preservar os requisitos de aplicações sensíveis ao atraso. O paradigma da arquitetura MEC utiliza nuvens locais e/ou regionais fisicamente mais próximas das *eNodeBs* para complementar a arquitetura C-RAN. Na arquitetura MEC, nós de borda atuam como um poder computacional adicional para satisfazer de uma forma mais imediata as requisições dos usuários ou para pré-processá-las antes de enviá-las para a nuvem centralizada, economizando a banda dos enlaces que conectam internamente a rede de telefonia móvel [38].

Neste capítulo, formula-se como um problema de programação inteira linear mista a recuperação de mídias através da arquitetura MEC, na qual as mídias são armazenadas em funções de rede virtuais atuando como *caches*. As topologias do *backhaul* das redes de telefonia móvel são próximas dos grafos gerados a partir do modelo do grafo de Waxmann.

Portanto, as topologias de rede consideradas nas avaliações são geradas aleatoriamente através do modelo de geração de grafos aleatórios proposto por Waxmann [75] e os enlaces das topologias seguem perfis reais das redes de telefonia móvel. Os resultados de simulação evidenciam um compromisso entre o custo de recuperação das mídias e o custo associado à instanciação dos *caches* em redes de grandes diâmetros. Além disso, as *eNodeBs* capazes de instanciar os *caches* com baixos atrasos de instanciação são responsáveis por até 70% das requisições satisfeitas. As principais contribuições desse capítulo são:

- uma formulação baseada em programação linear que aloca de forma ótima as instâncias de *cache* VNF nos ambientes virtuais presentes nos servidores MEC das *eNodeBs*, e mapeia as mídias a serem armazenadas nos *caches* instanciados;
- o custo de instanciação ao considerar um nó de controle e diferentes custos de instanciação para técnicas de virtualização distintas;
- resultados que indicam diferentes soluções ótimas dependendo do diâmetro da rede e da técnica de virtualização adotada.

O capítulo está organizado da seguinte forma. A Seção 4.2 descreve a metodologia utilizada nas avaliações apresentadas neste capítulo. A Seção 4.1 apresenta a formulação de programação linear proposta e os resultados obtidos através das simulações da formulação são apresentados na Seção 4.3.

4.1 Proposta de Formulação

Nesta seção, apresenta-se a formulação para a obtenção de mídias pelos UEs ao recuperá-las de VNFs atuando como *caches* em um problema de programação linear inteira mista. O cenário considera que há ambiente virtuais disponíveis nos servidores MEC e as mídias requisitadas pelos UEs podem ser armazenadas nesses ambientes virtuais através da instanciação de VNFs que atuam como *caches*. Um nó de controle (NC) na rede de telefonia móvel é responsável pela instanciação dos *cache* VNFs e pela decisão de qual *cache* VNF armazena a mídia requisitada. Considera-se o atraso de recuperação como a soma dos atrasos de transmissão e propagação dos enlaces que compõem o *backhaul* da infraestrutura da rede de telefonia móvel. Tanto a metodologia como os resultados apresentados não consideram o atraso de propagação do meio sem fio e o atraso de transmissão entre os UEs e as *eNodeBs*. O problema formulado pode ser interpretado como

(i) a atribuição dos *cache* VNFs aos ambiente virtualizados associados às *eNodeBs*, (ii) a atribuição das mídias requisitadas nos *cache* VNFs instanciados para satisfazer uma ou mais requisições dos UEs. A formulação proposta é capaz de demonstrar como os parâmetros da rede de telefonia móvel influenciam no padrão de distribuição das mídias entre os *cache* VNFs ao assumir que a imagem da VNF deve ser transferida a partir do nó central para ser instanciada.

A rede é modelada como um grafo não-direcionado $G(N, E)$, no qual N é o conjunto de todos os nós pertencentes à rede e E é o conjunto de enlaces que interconectam esses nós. O conjunto $B \subseteq N$ representa as *eNodeBs* e $M \subseteq N$ representa os servidores MEC. A função $D_{i,j}$, tal que $i \in B$ e $j \in M$, representa a soma do atraso de propagação e do atraso de transmissão para que a *eNodeB* i recupere a mídia do servidor MEC j . A formulação possui três variáveis de solução. A variável t^j indica a quantidade de *cache* VNFs hospedados por um servidor MEC j . A obtenção da mídia ν pertencente a distribuição de requisições λ_j presente na *eNodeB* i através do *cache* VNF instanciado no servidor MEC j é representada pela variável binária $X_{i,j}^\nu$. Além disso, a variável binária Y_i^ν representa que a *eNodeB* i recupera a mídia ν através da Internet. Com o objetivo de representar o esforço associado em transferir a imagem do *cache* VNF a partir do NC para um servidor MEC $j \in M$, assume-se um atraso de transferência referente à Equação 4.1. O atraso de transferência p^j é equivalente ao atraso de propagação total entre o NC e o servidor MEC j somado ao atraso de transmissão da imagem do *cache* VNF até o servidor MEC j . O atraso de propagação total é calculado pelo somatório do atraso de propagação entre cada par de nós pertencentes ao menor caminho do NC ao servidor MEC j . Já o atraso de transmissão é calculado pela razão entre o tamanho da imagem e o menor valor de largura de banda dentre todos os enlaces pertencentes ao menor caminho entre o NC e o servidor MEC j . A capacidade de armazenamento de cada *cache* VNF instanciado é representado por θ .

$$p^j = \sum_{(x,y) \in SP(NC,j)} T_{x,y}^{\text{Propagação}} + \frac{\text{tam_imagem}}{\min(BW_{NC,j})}. \quad (4.1)$$

A formulação desenvolvida é apresentada a seguir e todas as notações utilizadas nesta formulação estão resumidas na Tabela 4.1.

$$\text{minimizar } \sum_{i \in B} \sum_{\nu \in \lambda_i} Y_i^\nu D_{i,Int.} + \sum_{i \in B} \sum_{\nu \in \lambda_i} \sum_{j \in M} X_{i,j}^\nu D_{i,j} + \sum_{j \in M} t^j p^j \quad (4.2)$$

sujeito a

$$CPU_{\text{Cache VNF}}.t^j \leq CPU_j, \forall j \in M \quad (4.3)$$

$$RAM_{\text{Cache VNF}}.t^j \leq RAM_j, \forall j \in M \quad (4.4)$$

$$\sum_{i \in B} \sum_{\nu \in \lambda_i} X_{i,j}^\nu \leq W.t^j, \forall j \in M, W \geq |N||M| \quad (4.5)$$

$$\sum_{j \in M} X_{i,j}^\nu \leq W. \sum_{i \in B} Y_i^\nu, \forall i \in B, \forall \nu \in \lambda_i, W \geq |N||M| \quad (4.6)$$

$$\sum_{j \in M} X_{i,j}^\nu + Y_i^\nu = 1, \forall i \in B, \forall \nu \in \lambda_i \quad (4.7)$$

$$\sum_{i \in B} \sum_{\nu \in \lambda_i} \mu_\nu X_{i,j}^\nu \leq \theta.t^j, \forall j \in M \quad (4.8)$$

A função objetivo da formulação apresentada é definida na Equação 4.2. A função objetivo define que o objetivo está em minimizar o atraso de recuperação da mídia ν requisitada pela *eNodeB* i . A mídia ν pode ser recuperada pela Internet através das combinações exploradas pela primeira parcela $\sum_{i \in B} \sum_{\nu \in \lambda_i} Y_i^\nu D_{i,Int.}$. A mídia também pode ser recuperada através de um *cache* VNF que tenha sido instanciado em um servidor MEC j e que possua a mídia armazenada em virtude de uma requisição anterior através das combinações exploradas a partir da segunda parcela $\sum_{i \in B} \sum_{\nu \in \lambda_i} \sum_{j \in M} X_{i,j}^\nu D_{i,j}$. A função objetivo também define que a quantidade de *cache* VNFs instanciados na rede deve ser minimizada, visto que o atraso de transferência da imagem do *cache* VNF até um servidor MEC não é desprezível. Essa condição é imposta pela terceira parcela da função objetivo $\sum_{j \in M} t^j p^j$.

As Equações 4.3 e 4.4 definem que, para um *cache* VNF ser instanciado, o servidor MEC j deve possuir recursos de CPU e memória RAM disponíveis. A Equação 4.3 define que a quantidade de núcleos de CPU disponíveis no servidor MEC j , CPU_j , deve ser superior a quantidade de núcleos utilizada pela instância de *cache* VNF, $CPU_{\text{Cache VNF}}$. Já a Equação 4.4 define que a quantidade de memória RAM disponível no servidor MEC j , RAM_j , deve ser superior à quantidade de memória RAM utilizada pela instância de *cache* VNF, $RAM_{\text{Cache VNF}}$.

A Equação 4.5 define que toda recuperação de mídia feita por uma *eNodeB* através de um servidor MEC só pode ser realizada se houver ao menos um *cache* VNF instanciado entre os servidores MEC. A Equação 4.6 define que toda recuperação de mídia feita por uma *eNodeB* através de um servidor MEC só pode ser realizada se ao menos uma cópia da mídia foi anteriormente recuperada pela Internet por alguma *eNodeB*. A constante W presente nas Equações 4.5 e 4.6 é um método utilizado para garantir que inequação seja satisfeita independentemente do valor produzido pelo somatório $\sum_{j \in M} X_{i,j}^\nu$. Considere o caso no qual $\sum_{j \in M} X_{i,j}^\nu$ é maior que t^j para a Equação 4.5, uma vez que múltiplas *eNodeBs* requisitam a mesma mídia ν e há somente e somente há um único *cache* VNF instanciado. Nesse caso, a inequação deve se manter, visto que o objetivo é que haja pelo menos um *cache* VNF instanciado para que as múltiplas requisições possam ser atendidas sem que seja necessário recuperar as mídias a partir da Internet. O mesmo pode ocorrer para a Equação 4.6, visto que múltiplas requisições para a mesma mídia ν podem ser atendidas desde que a mídia ν tenha sido recuperada ao menos uma vez pela Internet anteriormente. Ao atribuir uma valor a W que seja superior à soma de todos os nós presentes na rede, as desigualdades são garantidas.

A Equação 4.7 impõe ao modelo que toda mídia requisitada por uma *eNodeB* seja atendida, ou através da Internet, ou através de um servidor MEC, mas nunca pelos dois ao mesmo tempo. Por último, a Equação 4.8 define que a restrição de armazenamento deva ser respeitada, isto é, a quantidade de mídias armazenadas em um servidor MEC deve ser inferior a capacidade de um *cache* VNF multiplicado pela quantidade de *cache* VNFs instanciados no servidor MEC.

A Seção 4.2 apresenta a metodologia empregada nas avaliações a partir das diferentes parametrizações aplicadas à formulação proposta. A Seção 4.3 apresenta os resultados obtidos através das simulações realizadas sobre a formulação completa, assim como são geradas as topologias consideradas nas avaliações e os parâmetros utilizados em cada avaliação.

4.2 Metodologia de Avaliação

Para avaliar como as mídias se distribuem na rede, assim como são instanciados os *cache* VNFs, em relação à capacidade de armazenamento e do custo de configuração de VNFs, implementa-se um simulador na linguagem de programação Python. O simulador gera as topologias utilizadas, as requisições dos UEs e o modelo de otimização. As

Tabela 4.1: Notações utilizadas na formulação.

Notação	Descrição	Tipo
N	Nós da rede	Conjunto
E	Enlaces que interconectam os nós da rede	Conjunto
B	Nós da rede que representam as <i>eNodeBs</i>	Conjunto
M	Nós da rede que representam os servidores MEC	Conjunto
φ	Mídias	Conjunto
λ_i	Distribuição das requisições na <i>eNodeB i</i>	Conjunto
μ_ν	Tamanho de cada mídia ν em bytes	Parâmetro
θ	Capacidade de cada <i>cache</i> VNF instanciado em bytes	Parâmetro
p^j	Atraso associado à transferência da imagem no NC ao servidor MEC j	Parâmetro
t^j	Quantidade de <i>cache</i> VNFs instanciadas no servidor MEC j	Variável
$X_{i,j}^\nu$	Recuperação da mídia ν pela <i>eNodeB i</i> a partir do <i>cache</i> VNF j	Variável
Y_i^ν	Recuperação da mídia ν pela <i>eNodeB i</i> a partir da Internet	Variável

duas topologias geradas para a avaliação derivam-se da aproximação de características encontradas nas topologias de redes de telefonia móvel. Topologias do *backhaul* das redes de telefonia móvel aproximam-se do modelo encontrado no grafo de Waxmann [75]. Os grafos de Waxmann são produzidos de acordo com o modelo Erdoes-Renyi com a inclusão do posicionamento dos vértices no plano e a probabilidade de interconexão dos vértices é baseada na distância que os separa no plano.

$$P(\{u, v\}) = \beta e^{\frac{-d(u,v)}{\alpha L}} \quad (4.9)$$

A Equação 4.9 representa a probabilidade de dois vértices, u e v , possuírem uma aresta os conectando. A probabilidade considera a distância entre os dois vértices, $d(u, v)$, a distância euclidiana máxima possível entre dois nós, $L = \max(d(u, v)) \forall u, v \in N$, e os parâmetros $\alpha, \beta \in (0, 1]$. Valores mais altos do parâmetro β definem uma maior densidade de arestas entre os vértices e valores mais altos do parâmetro α definem que arestas entre nós mais distantes estarão mais presentes no grafo em relação às arestas de menores distâncias. A Figura 4.1 ilustra as duas topologias geradas com 50 nós na rede com a mesma parametrização encontrada em Fonseca *et al.* [27], no qual $\alpha = 0.1$, $\beta = 0.4$, e $L = 70$. Dentre os 50 nós da rede, 10 são *eNodeBs*, representadas pelos vértices em verde, 4 são servidores MEC, representados pelos vértices em vermelho, 1 nó é o NC, representado pelo vértice em amarelo, e os demais são nós encaminhadores, representados

pelos vértices em cinza.

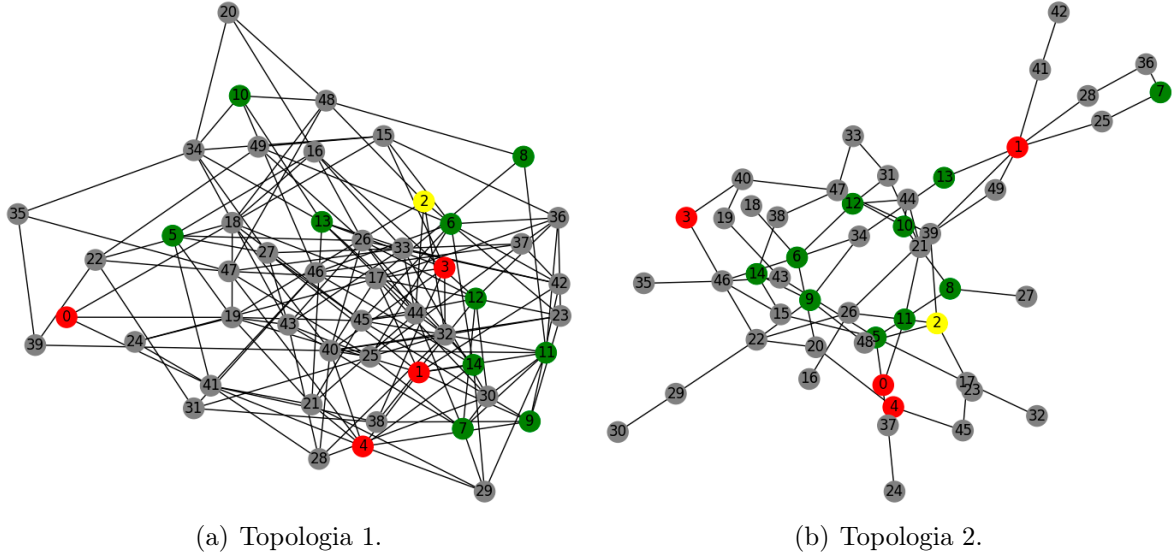


Figura 4.1: Topologia com 50 nós utilizada nas simulações e gerada a partir dos parâmetros $\alpha = 0.1$, $\beta = 0.4$ e $L = 70$ pela biblioteca `networkx 2.4` da linguagem de programação Python. Os nós em verde representam as *eNodeBs*, os nós em vermelho representam os servidores MEC, o nó em amarelo representa o nó de controle e os nós cinzas representam os nós encaminhadores.

Cada aresta do grafo representa um enlace da rede e possui como atributos um valor de largura de banda e um valor de atraso de propagação. Tais valores são atribuídos de acordo com os perfis descritos na Tabela 4.2 através da distância entre cada par de vértices do grafo gerado. O valor de distância presente na Tabela 4.2 que mais se aproxima da distância entre os vértices é atribuído à aresta que os conecta. Os parâmetros de simulação estão resumidos na Tabela 4.1. As 10 *eNodeBs* presentes na topologia requisitam as mídias aleatoriamente. A quantidade de mídias requisitada por cada *eNodeB* está fixada em 50 unidades. O conjunto de 50 mídias é sorteado seguindo a distribuição apresentada na Figura 4.2 e esse conjunto é atribuído a cada *eNodeB* ao passar por um filtro que substitui cada uma das 50 mídias com 50% de probabilidade. O filtro é utilizado nas simulações em virtude da quantidade de mídias requisitadas ser baixa para evidenciar a popularidade das mídias. Valores maiores de mídias a serem requisitadas e/ou número mais altos de *eNodeBs* aumentam o número de combinações a serem analisadas pelo *solver* e consumindo mais memória RAM que a máquina na qual as simulações são executadas possui. O tamanho das mídias segue uma distribuição Gamma com os parâmetros $\alpha = 1,8$ e $\beta = 5441,93$, que produzem um tamanho médio de 9809,52 KB, mediana de 8480,33 KB, e uma covariância de 0,93 KB [7].

As requisições por conteúdos que chegam às *eNodeBs* consideram um conjunto de

Tabela 4.2: Perfis dos enlaces utilizados na geração da topologia [29].

Tecnologia	Largura de Banda (Gbps)	Atraso de Propagação (μ s)	Distância (km)
mmWave (60-80 GHz)	0.9, 1.25, 1.5, 2, 3, 4, 8	1-20	0.3-6
μ Wave (6-60 GHz)	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.25, 1.5, 2	1-100	0.3-30
Cobre (1000/10G/40G BASE-T)	1, 10, 40	0.05-0.5, 0.275, 0.15	0.001-0.1, 0.055, 0.03
Fibra SMF @ 1310 nm (1000, 10G, 40G, 100GBASE-EX, LR, LR-4)	1, 10, 40, 100	1-200, 50, 50, 50	0.2-40, 10, 10, 10
Fibra SMF @ 1550 nm (1000, 10G, 40G, 100GBASE-ZX, ER, ER-4)	1, 10, 40, 100	1-350, 200, 200, 200	0.2-70, 40, 40, 40
TbE (Em desenvolvimento)	200, 400	1-50	0.2-10

dados disponibilizado pela Netflix [12] como modelo. O conjunto de dados, como mostra a Figura 4.2, é composto por 100 milhões de avaliações em uma escala de 1 a 5, representadas pelo conjunto A , que estão distribuídas sobre aproximadamente 18 mil filmes, representados pelo conjunto φ . Assume-se que as mídias possuem diferentes popularidades, neste caso a popularidade de cada filme do conjunto de dados, e que a probabilidade de uma mídia ser requisitada é proporcional ao número de avaliações que o filme representando a mídia recebeu pelos usuários no conjunto de dados. A probabilidade de um filme $\nu \in \varphi$ com identificação i , ν_i , ser selecionado, é igual a quantidade de avaliações $\alpha \in A$ recebidas pelo filme ν_i , α_{ν_i} , sobre a soma das avaliações recebidas por todos os filmes do conjunto de dados, $\sum_{\nu \in \varphi} \alpha_{\nu}$, como mostra a Equação 4.10. Os parâmetros das simulações estão apresentados na Tabela 4.3 e as simulações são executadas em uma máquina com 16 GB de memória RAM, sistema operacional Ubuntu 16.04.6 LTS, utilizando um interpretador Python 3.5 e o *solver* IBM ILOG CPLEX Optimization Studio 12.8.0.

$$P(\nu) = \frac{\alpha_{\nu}}{\sum_{\nu \in \varphi} \alpha_{\nu}}. \quad (4.10)$$

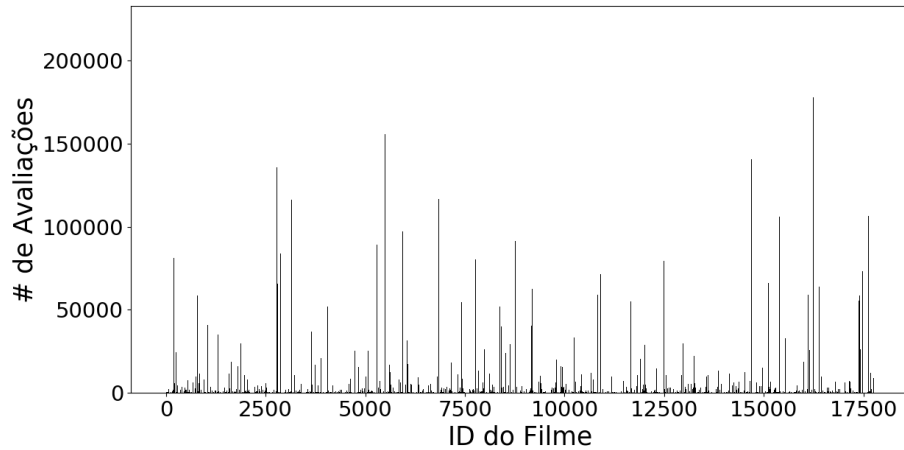


Figura 4.2: Distribuição das avaliações dos usuários sobre os diferentes filmes no conjunto de dados da Netflix.

Tabela 4.3: Parametrização.

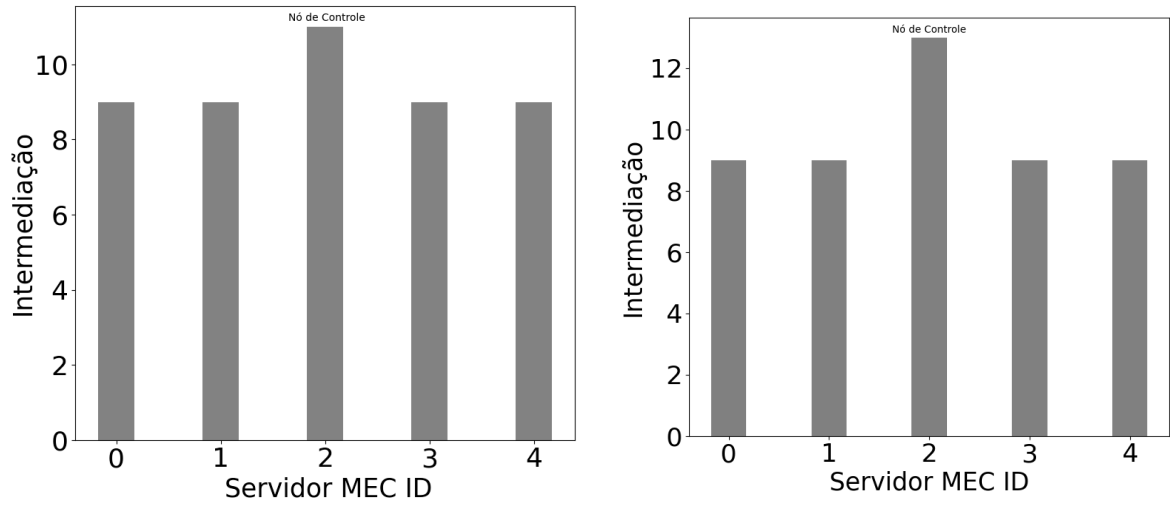
Parâmetro	Valor
# de <i>eNodeBs</i> requisitantes	10 <i>eNodeBs</i>
# de mídia requisitadas por <i>eNodeB</i>	50 mídias
Tamanho de <i>cache</i>	[50, 100] MB, passo em 10 MB
Tamanho das imagens nos <i>cache</i> VNFs	[50, 500] MB, passo em 50 MB
Núcleos de CPU por servidor MEC	8 núcleos
Memória RAM por servidor MEC	16 GB
Consumo de CPU por <i>cache</i> VNF	1 núcleo
Consumo de memória RAM por <i>cache</i> VNF	2 GB

A formulação é derivada do problema de atribuição generalizado (*Generalized Assignment Problem* - GAP). O problema tem como objetivo alocar n tarefas em l máquinas com a restrição de que cada tarefa $j = 1, \dots, n$ deve ser atribuída exatamente a uma única máquina. Cada máquina m possui espaço para acomodar um determinado número de tarefas e cada tarefa ocupa um certo espaço da máquina ao ser atribuído a ela. O processo de atribuir uma tarefa l a uma máquina m incorre em um custo de atribuição. O objetivo do problema está em encontrar a atribuição de todas as tarefas com o menor custo de atribuição possível. Como o GAP é um problema NP-difícil e a formulação deste trabalho é derivada de sua formulação, a complexidade computacional da formulação apresentada também é não-polinomial e cresce com as combinações formadas pelo número de *eNodeBs* na rede, com o número de *eNodeBs* requisitantes e o número de mídias requisitadas.

4.3 Resultados da Formulação

O primeiro passo realizado antes de iniciar as simulações é selecionar o nó da rede responsável por transferir as imagens dos *cache* VNFs até os servidores MEC, o NC. Utiliza-se como critério para a seleção do NC o do nó que possui o maior valor de intermediação para a transferência da imagem, como mostra a Figura 4.3. O nó com maior valor de intermediação é o que está presente no maior número de menores caminhos entre todos os pares de servidores MEC da rede, característica importante pela redundância de caminhos, visto que apresenta maior probabilidade de possuir um caminho com alta vazão e baixo atraso de propagação. O nó selecionado como NC em ambas as topologias é o servidor MEC de identificação 2, visto que possui o maior valor de intermediação, e portanto, o servidor MEC 2 não disponibiliza recursos para a instanciação de *cache* VNFs nas Topologias 1 e 2. A Figura 4.4 apresenta o atraso de transferência das imagens de *cache* VNFs entre o NC e os servidores MEC calculados a partir da Equação 4.1 para

cada tamanho de imagem utilizado nas avaliações. A Tabela 4.4 apresenta as informações topológicas entre o NC e os servidores MEC para cada topologia, incluindo a distância euclidiana, atraso de propagação, número de saltos e vazão. A Figura 4.5 apresenta a distribuição da vazão entre cada servidor MEC e as 10 *eNodeBs* presentes nas topologias. A Figura 4.5(a) possui um valor próximo a 100 Gbps que dificulta a visualização do eixo y e a comparação entre os servidores MEC. Dessa forma, há um gráfico adicional que mostra a distribuição sem o valor próximo a 100 Gbps do servidor MEC 1 indicado pela seta em vermelho.



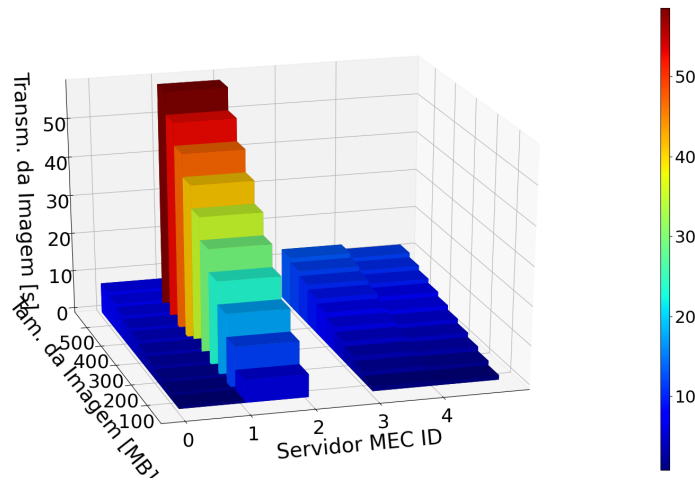
(a) Valor de intermediação para cada nó selecionado como servidor MEC na Topologia 1 considerando somente pares de servidores MEC. (b) Valor de intermediação para cada nó selecionado como servidor MEC na Topologia 2 considerando somente pares de servidores MEC.

Figura 4.3: Valores de intermediação para cada servidor MEC encontrados nas Topologias 1 e 2.

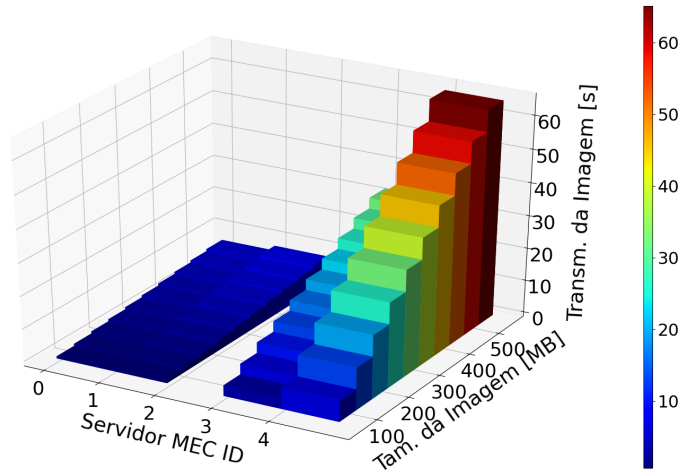
Tabela 4.4: Informações das topologias entre o nó de controle e os servidores MEC.

Servidor MEC	Distância (km)	Atraso de Prop. (μ s)	Saltos	Vazão (Gbps)
Topologia 1				
0	0,82	285	2	1
1	0,32	197	3	0,2
3	0,58	198	3	0,9
4	0,96	410	3	1
Topologia 2				
0	0,38	124	2	1.25
1	0,38	314	2	1
3	0,72	451	5	0,6
4	0,61	327	5	0,3

A Figura 4.6 apresenta o tempo médio de recuperação das mídias, assim como o número médio de *cache* VNFs instanciados para cada combinação de capacidade de *cache* e tamanho de imagem relacionados à Topologia 1. Observa-se que pela Figura 4.6(a)



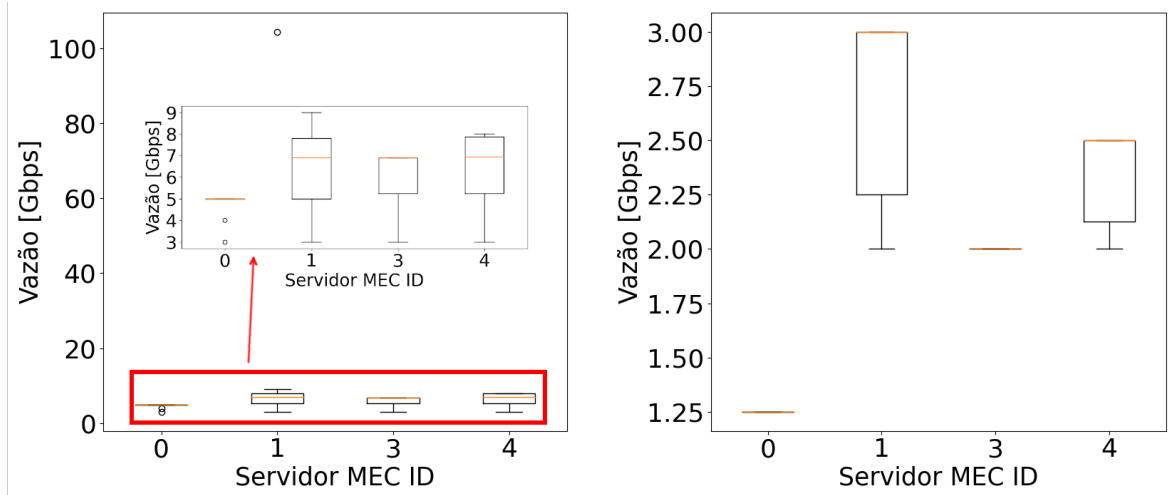
(a) Atraso de transferência entre o NC e os servidores MEC na Topologia 1.



(b) Atraso de transferência entre o NC e os servidores MEC na Topologia 2.

Figura 4.4: Atrasos de transferência das imagens dos *cache* VNFs ao considerar a soma dos atrasos de transmissão e propagação de cada enlace pertencente ao menor caminho entre o NC e o servidor MEC.

que os maiores valores de atraso de recuperação estão associados a menor capacidade de *cache* combinado ao maior tamanho de imagem a ser transferida pelo NC aos servidores MEC. Visto que transferir imagens grandes é custoso para os enlaces da rede, espera-se que poucos *cache* VNFs sejam instanciados na rede, como mostra a Figura 4.6(b). Além disso, *cache* VNFs com capacidades de 50 MB são capazes de armazenar poucas mídias, uma vez que o tamanho médio das mídias requisitadas é de aproximadamente 9 MB. Já os menores valores de atraso de recuperação estão diagonalmente opostos, quando as capacidades dos *cache* VNFs estão no maior valor avaliado e o tamanho das imagens está no menor valor avaliado. A Figura 4.7 mostra os mesmos resultados relacionados à Topologia 2 e observa-se que os resultados obtidos na Topologia 2 seguem as mesmas



(a) Distribuição da vazão entre cada servidor MEC e as *eNodeBs* na Topologia 1.

(b) Distribuição da vazão entre cada servidor MEC e as *eNodeBs* na Topologia 2.

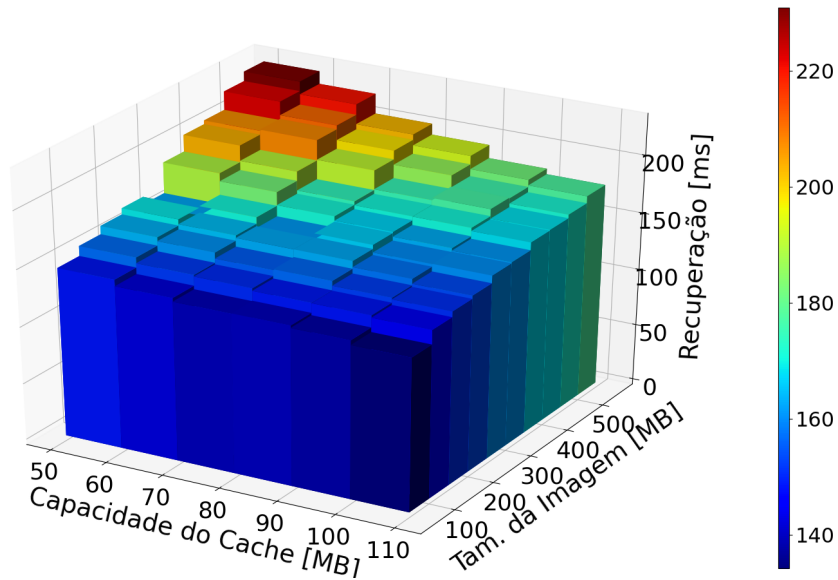
Figura 4.5: Distribuição da vazão entre os servidores MEC e as *eNodeBs* nas Topologias 1 e 2.

características encontradas na Topologia 1.

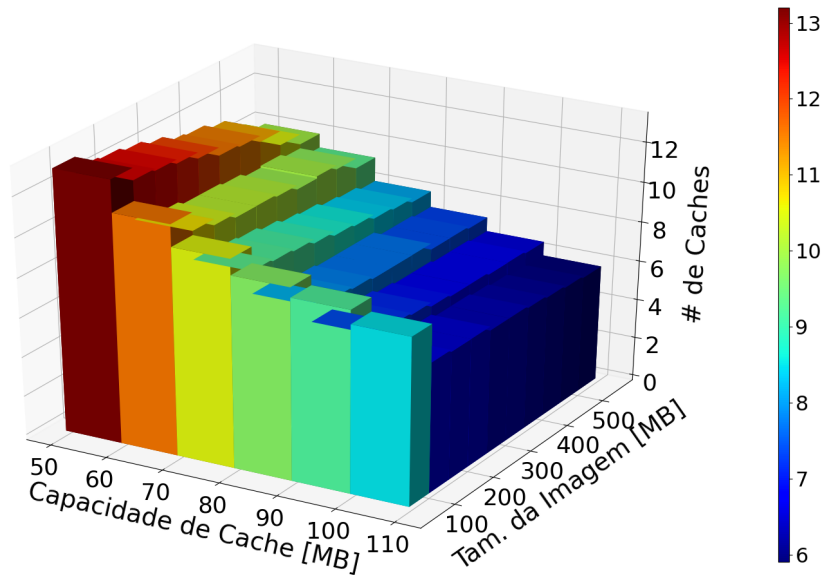
Apesar dos resultados similares encontrados no atraso de recuperação e na quantidade de *cache* VNFs instanciados, a distribuição dos *cache* VNFs nos servidores MEC é diferente nas duas topologias, como mostram as Figuras 4.8 e 4.9. Observa-se que na Topologia 1, o Servidor MEC 4 satisfaz cerca de 70% das requisições para quase todas as combinações de tamanho da imagem do *cache* VNF e capacidade da instância do *cache* VNF. Como mostra a Figura 4.4(a), o Servidor MEC 4 é o que apresenta o menor atraso de transferência das imagens a partir do NC. Como o tamanho das imagens dos *cache* VNFs são significativamente maiores que o tamanho médio das mídias requisitadas, se torna mais vantajoso transferir as imagens para o Servidor MEC que não mantém o processo de armazenar as mídias em espera por um longo período de tempo. Já o Servidor MEC 0 e o Servidor MEC 3 apresentam as distribuições de recuperação similares, exceto para os casos nos quais a capacidade da instância de *cache* VNF é pequena e o tamanho das imagens a serem transferidas é grande. Quando o tamanho das imagens dos *cache* VNFs é grande, o atraso de transferência das imagens e o custo para transferi-las na rede é alto. Mais especificamente, quando o tamanho da imagem do *cache* VNF é de 500 MB, e capacidade de armazenamento da instância do *cache* VNF é de 50 MB, o número de instâncias de *cache* VNFs é pequeno, o que torna o papel da vazão entre o Servidor MEC e as *eNodeBs* muito importante. A Figura 4.5(a) mostra que o Servidor MEC 0 provê quase que a mesma vazão para todas *eNodeBs* na Topologia 1. Como consequência do alto custo de instanciamento dos *cache* VNFs, a Figura 4.8(a) mostra que o Servidor MEC

0 é responsável por satisfazer cerca de 70% das requisições para a combinação restrita de imagens de *cache* VNF de 500 MB a serem transferidas com a instância do *cache* VNF capaz de armazenar somente 50 MB.

Diferentemente da Topologia 1 na qual o Servidor MEC 4 apresenta melhores resultados para quase todas as combinações de tamanho de imagem do *cache* VNF e capacidade das instâncias de *cache* VNF, a Topologia 2 apresenta uma taxa de recuperação quase que igualmente distribuída entre o Servidor MEC 0 e o Servidor MEC 1. A Figura 4.4(b) mostra que ambos Servidor MEC 0 e Servidor MEC 1 possuem atrasos de transferência das imagens de *cache* VNF muito próximos independentemente do tamanho das imagens. Entretanto, a Figura 4.5(b) mostra que o Servidor MEC 1 possui uma distribuição da vazão entre as *eNodeBs* mais alta em relação ao Servidor MEC 0, o que explica a taxa de recuperação um pouco maior. Já o atraso de transferência das imagens de *cache* VNF entre o NC e o Servidor MEC 3 é significativamente mais alto comparado aos atrasos apresentados para o Servidor MEC 0 e o Servidor MEC 1. A Figura 4.9(c) mostra que como consequência do atraso de transferência maior das imagens, o Servidor MEC 3 somente apresenta taxas de recuperação significativas para tamanhos menores das imagens de *cache* VNF, como os tamanhos de 100 MB e 200 MB. Para tamanhos de imagens menores, o Servidor MEC 3 apresenta uma distribuição de vazão mais alta comparada à distribuição de vazão do Servidor MEC 0, o que explica as taxas de recuperação significativas para tamanhos de imagens de *cache* VNF menores.

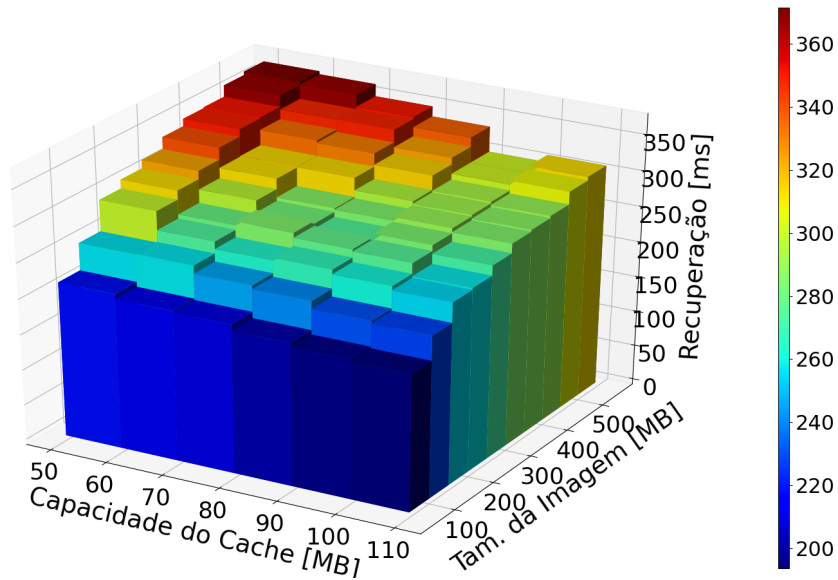


(a) Atraso de recuperação das mídias para cada combinação de capacidade de *cache* e tamanho de imagem.

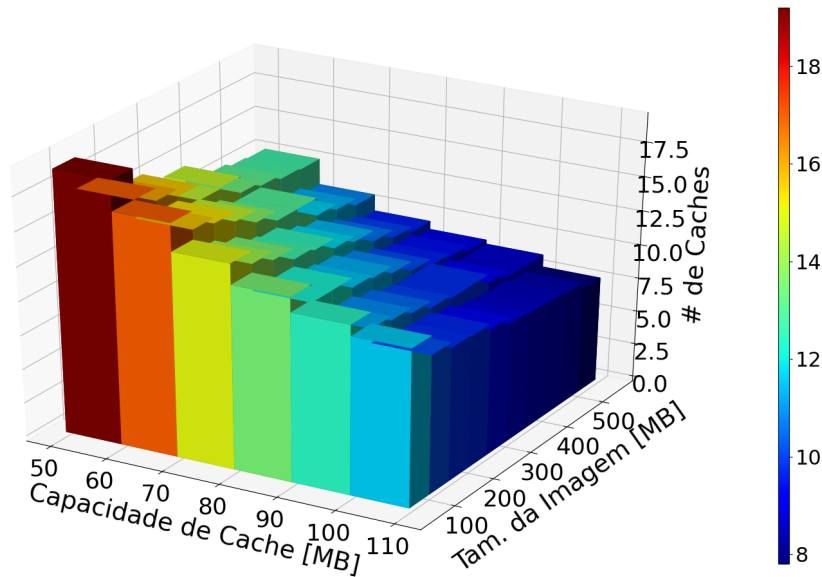


(b) Quantidade de *cache* VNFs instanciados para cada combinação de capacidade de *cache* e tamanho de imagem.

Figura 4.6: Resultados de atraso de recuperação e quantidade de *cache* VNFs instanciados na Topologia 1.

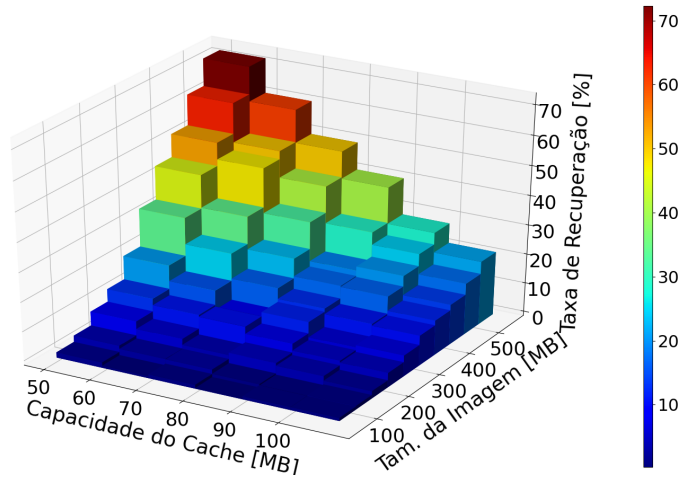


(a) Atraso de recuperação das mídias para cada combinação de capacidade de *cache* e tamanho de imagem.

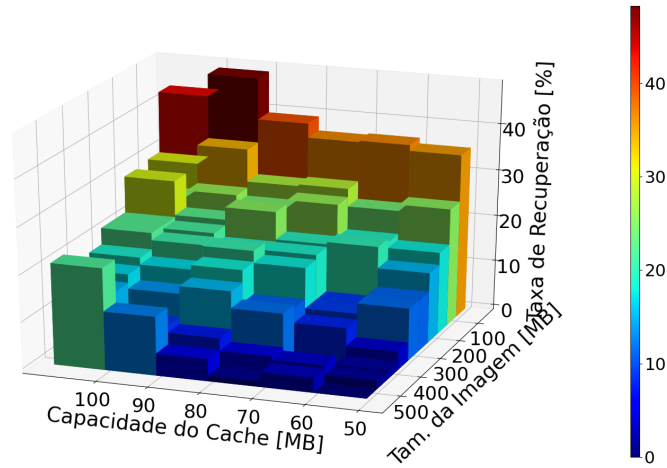


(b) Quantidade de *cache* VNFs instanciados para cada combinação de capacidade de *cache* e tamanho de imagem.

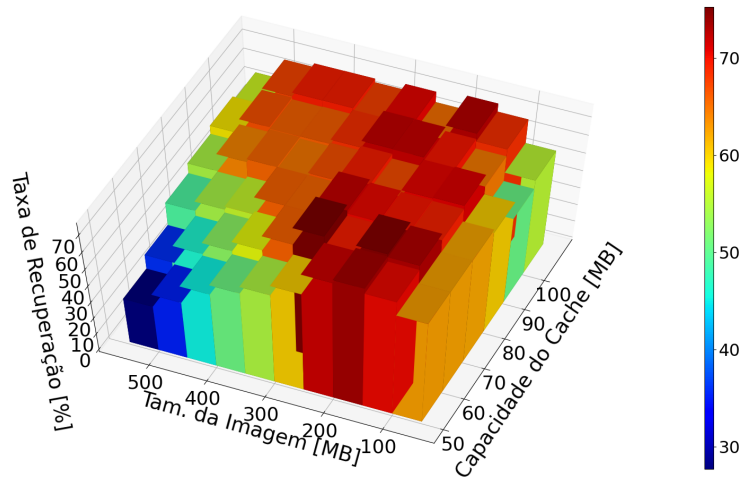
Figura 4.7: Resultados de atraso de recuperação e quantidade de *cache* VNFs instanciados na Topologia 2.



(a) Porcentagem de mídias recuperadas através do Servidor MEC 0 sobre o total de mídias recuperados por *cache* VNFs.

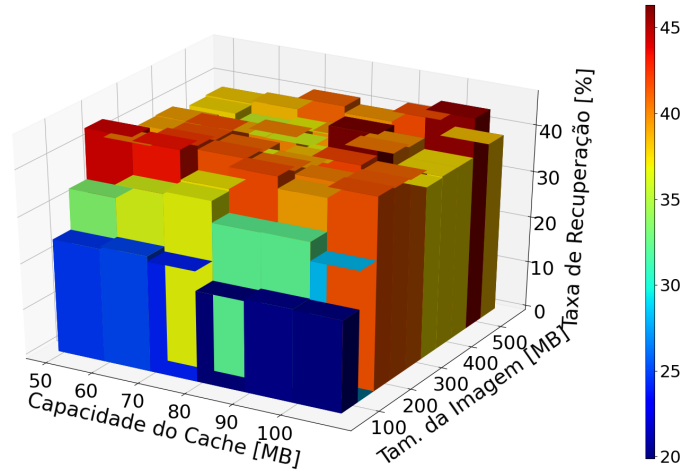


(b) Porcentagem de mídias recuperadas através do Servidor MEC 3 sobre o total de mídias recuperados por *cache* VNFs.

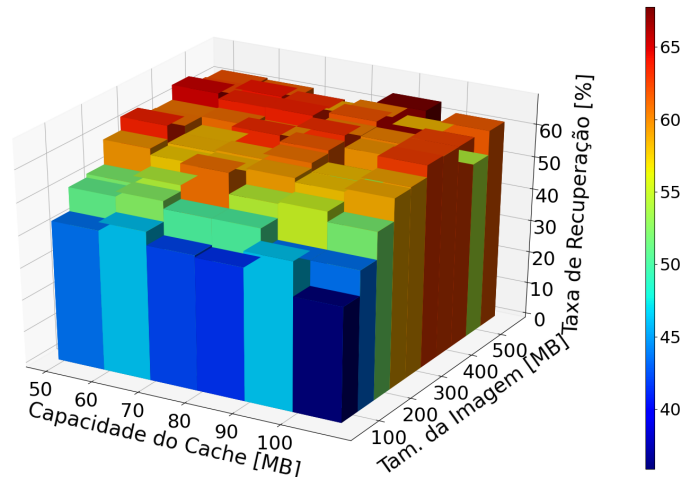


(c) Porcentagem de mídias recuperadas através do Servidor MEC 4 sobre o total de mídias recuperados por *cache* VNFs.

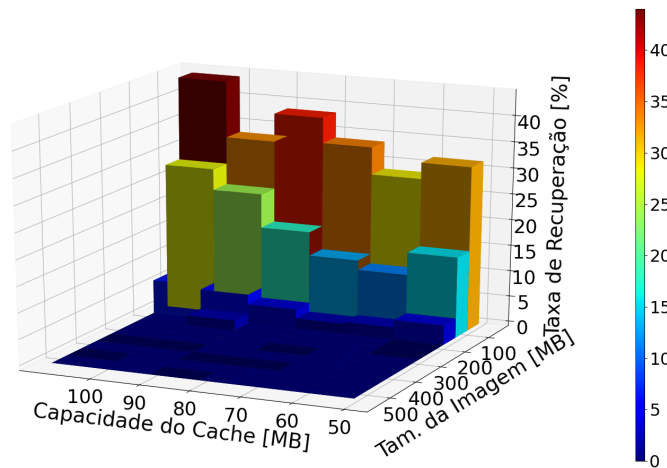
Figura 4.8: Taxa de recuperação média relativa entre os servidores MEC na Topologia 1.



(a) Porcentagem de mídias recuperadas através do Servidor MEC 0 sobre o total de mídias recuperados por *cache* VNFs.



(b) Porcentagem de mídias recuperadas através do Servidor MEC 1 sobre o total de mídias recuperados por *cache* VNFs.



(c) Porcentagem de mídias recuperadas através do Servidor MEC 3 sobre o total de mídias recuperados por *cache* VNFs.

Figura 4.9: Taxa de recuperação média relativa entre os servidores MEC na Topologia 2.

4.4 Trabalhos Relacionados

Baştuğ *et al.* [11] discutem que as RANs devem mudar seu paradigma reativo para um paradigma proativo. Os autores argumentam que é mais viável somente tomar uma decisão após a chegada de fluxos de dados na RAN com o aumento exacerbado de UEs, *streaming* de vídeo, aplicações web e das redes sociais. As redes celulares devem proativamente explorar as informações de contexto dos usuários para antecipar futuras requisições. Afirma-se no trabalho que usuários móveis são previsíveis até um certo grau e a rede celular pode pré-armazenar conteúdos que um conjunto de usuários possua alta probabilidade de requisitá-los. Como a capacidade de armazenamento de usuários móveis está cada vez maior, os autores defendem que os conteúdos sejam proativamente armazenados nos próprios UEs para serem acessados localmente, sem a necessidade de se comunicar com a RAN.

Paschos *et al.* [57] argumentam que a infraestrutura celular evoluiu para acompanhar o crescimento de dispositivos móveis que estão acessando a Internet. No entanto, a explosão de RANs e de UEs tornam os enlaces de *backhaul* congestionados caso nenhuma outra solução seja proposta para reduzir sua carga. Nesse sentido, armazenar conteúdos na borda da rede celular possui o potencial em aliviar a infraestrutura cabeada das redes celulares. Entretanto, técnicas de *caching* normalmente empregadas em redes de distribuição de conteúdos não são apropriadas para as redes celulares, visto que o número de conteúdos alcançáveis é ordens de magnitude maior e sua popularidade varia bruscamente com o passar do tempo. Políticas de retirada de *cache* baseadas em modelos de popularidade estáticos, como a *Least Recently Used* (LRU), não produzem ganhos satisfatórios. As políticas devem se encaixar no período que os usuários acessam os conteúdos como pulsos e a amplitude desses pulsos serem vistas como popularidades instantâneas para produzir melhores estimativas de popularidade e, portanto, de *caching* [44]. Os autores também afirmam que algoritmos de aprendizado de máquina e *big data* serão primordiais para rastrear as variações de popularidade.

Li *et al.* [47, 46] propõem o uso de *caching* como serviço (*Caching as a Service* - CaaS) nas redes celulares. Com a tendência em virtualizar estações base em conjuntos de BBUs, assim como os elementos do EPC, as redes 5G devem incorporar o *caching* como parte do ambiente virtual para aprimorar a entrega de serviços multimídia. Os autores discutem os diferentes prós e contras relacionados ao local de implantação dos recursos de *caching*. Recursos físicos mais próximos das eNBs são naturalmente mais escassos, o que torna mais difícil produzir um grande número de acertos em *cache*. Ao mover o

ambiente virtual mais adentro da EPC os recursos virtuais das *cache* VMs aumentarão, assim como os acertos em *cache*. Por outro lado, a complexidade de roteamento dentro da EPC aumentará, assim como a carga nos enlaces de *backhaul*. Além disso, o controlador centralizado do CaaS terá um papel fundamental com as *cache* VMs livres para serem instanciadas em qualquer ambiente da rede celular com o objetivo de atender os requisitos de QoS e dos usuários móveis. O controlador será responsável por aumentar e diminuir a escala de *cache* VMs instanciadas, na decisão do local de instanciação das *cache* VMs, em configurar rotas eficientes e em pré-armazenar conteúdos baseando-se nas aplicações utilizadas pelo usuários móveis.

Moon *et al.* [56] propõem um sistema de virtualização de *cache* baseado em NFV e em um modelo hierárquico de *caching*. Os autores projetam o plano de dados e o plano de controle do sistema. O plano de dados é responsável por receber as requisições dos usuários móveis através de um protocolo conhecido como *Common Public Radio Interface* (CPRI). O roteador de borda verifica se o conteúdo requisitado já está armazenado em um *cache* de borda ao analisar as informações do cabeçalho HTTP através da técnica de inspeção profunda de pacotes (*Deep Packet Inspection* - DPI). A análise do fluxo recebido é utilizada para consultar a tabela de encaminhamento (*Forwarding Information Base* - FIB). Caso o conteúdo requisitado não esteja armazenado em um *cache* virtual na borda da rede, o roteador de borda encaminha o fluxo para o S-GW através do protocolo de tunelamento (*General Packet Radio Service Tunneling Protocol* - GTP). O S-GW consulta sua FIB para determinar se o conteúdo requisitado está armazenado no EPG. Se não for o caso, a requisição é encaminhada para o P-GW para ser recuperado pela Internet. O conteúdo requisitado segue o caminho inverso dentro da rede celular para ser entregue ao UE e o orquestrador de *caches* virtuais (*Virtual Cache Orchestrator* - vCO) decide o *cache* virtual no qual o conteúdo será armazenado. No plano de controle, o vCO é responsável em calcular o roteamento ótimo e em obter informações sobre os *caches* virtuais.

Yala *et al.* [78] propõem a formulação de um problema de otimização multi-objetivo para a alocação de funções de rede virtuais com o objetivo de minimizar a latência para os usuários e maximizar a disponibilidade do serviço. Os autores argumentam que em um ambiente MEC, os serviços pode ser alocados para a borda da rede e prover baixas latências e podem ser alocados para infraestruturas tradicionais de nuvem e prover alta disponibilidade. A latência é definida no trabalho como o atraso para que o usuário consiga acessar o serviço. O serviço é representado como uma instância de VNF que é hospedada em uma única máquina física e a latência é o atraso na comunicação do

usuário até a máquina física. Como modelo de disponibilidade, os autores consideram que ambas as instâncias de VNF, e as máquinas físicas, possuem probabilidade de falha. Recursos físicos na borda da rede de telefonia móvel são escassos quando comparados ao recursos disponíveis na infraestrutura da nuvem, dessa forma as instâncias de VNF que são hospedadas na borda da rede de telefonia móvel possuem maiores chances de falha comparado com as instâncias de VNF que são hospedadas na infraestrutura da nuvem. O modelo de custo considera a capacidade de CPU necessária para a alocação de cada instância de VNF e um custo fixo específico para cada máquina física para representar o custo associado ao consumo de energia por utilizar as máquinas físicas. A política de alocação ótima considera em sua função objetivo um par de pesos, w_l para representar a importância da latência e w_a para representar a importância da disponibilidade. A política tem como objetivo apresentar uma portância relativa baseada no tipo de serviço, sendo $w_l + w_a = 1$. Para solucionar o problema de alocação das instâncias de VNF, os autores utilizam o algoritmo genético inspirado no algoritmo de grupos genéticos proposto por Xu e Fortes [77].

Cziva *et al.* [20] propõem a solução *Edge vNF Placement* para o problema de alocação de instâncias de VNF em infraestruturas MEC. Os autores argumentam que as instâncias de VNF na borda da rede de telefonia móvel devem ser dinamicamente alocadas em sinergia com a dinâmicas da rede, ao considerar as flutuações na ocupação dos enlaces, as demandas dos usuários e a mobilidade. Como modelo de sistema, os autores assumem que qualquer máquina física possui CPU, memória e recursos de E/S para hospedar uma instância de VNF. Como há diferentes tipos (*flavors*) de VNFs, cada tipo de VNF apresenta seus próprios requisitos de recursos necessários para que a VNF seja instanciada. Os usuários também apresentam seus requisitos de atraso que devem ser respeitados para que o acordo a nível de serviço (*Service Level Agreement* - SLA) seja cumprido. A formulação para a alocação ótima considera a limitação dos recursos físicos das máquinas físicas, a tolerância máxima da latência, assim como os requisitos de largura de banda, no caminho entre a instância de VNF e o usuário. Entretanto, a formulação não considera as dinâmicas da rede, e após alguns instantes de tempo, a latência do caminho entre a instância de VNF e o usuário pode se tornar maior que a latência tolerada. Os autores argumentam que, por essa razão, a alocação inicial das instâncias de VNF devem ser avaliadas novamente após um certo tempo e as instâncias de VNF devem ser migradas de uma máquina física para outra que consiga respeitar a latência tolerada. A reavaliação possui um compromisso significativo, visto que se a reavaliação em um curto período de tempo, as instâncias de VNF são migradas com uma diferença de latência insignificante

com um alto custo de migração. Por outro lado, se houver uma espera muito longa para realizar a reavaliação, a latência pode ultrapassar a latência tolerada. Para encontrar o instante de tempo ótimo para realizar a reavaliação, os autores propõem o uso da teoria da regra de parada ótima. A proposta da regra de parada ótima seleciona o instante de tempo que maximiza a soma cumulativa do desvio entre a latência da alocação inicial da instância de VNF e a latência tolerada para realizar a reavaliação.

Capítulo 5

Aprendizado de Máquina

No Capítulo 4 propõe-se uma formulação baseada em programação linear inteira para minimizar o atraso de recuperação dos conteúdos que são armazenados em instâncias de *cache* VNF por parte dos UEs. Assume-se no modelo proposto que um nó controlador possui conhecimento sobre todos os conteúdos que estão sendo requisitados nas estações base. Tal suposição é otimista, visto que não é possível saber de antemão quais conteúdos serão requisitados às estações base, assim como não é possível saber a ordem na qual os conteúdos chegam nas estações base. Portanto, faz-se necessário uma solução que aprenda quais são os melhores conteúdos a serem armazenados e aonde eles devem ser armazenados a partir de interações com a rede de telefonia móvel, como ocorre no aprendizado de máquina por reforço. Este capítulo apresenta os conceitos fundamentais de aprendizado de máquina ao utilizar redes neurais artificiais nas Seções 5.1, 5.2 e 5.3, e finaliza com os conceitos de aprendizado de máquina por reforço que são utilizados para a proposta do agente apresentada no Capítulo 6.

5.1 Redes Neurais Artificiais *Feed-Forward*

As redes neurais artificiais são inspiradas no cérebro biológico e no sistema nervoso. O componente básico das redes neurais artificiais é chamado de neurônio (*neuron*). Um neurônio é considerado como uma unidade de processamento e os neurônios são conectados uns aos outros através de um peso sináptico (*weight*). Cada neurônio da rede neural recebe uma informação ponderada através do peso sináptico presente nas conexões e produz um valor da soma ponderada de todos os sinais de entrada através de uma função de ativação (*activation function*). A arquitetura das rede neurais *feed-forward* estabelecem que os neurônios são organizados em camadas e a Figura 5.1 ilustra sua arquitetura. As camadas

são divididas em três tipos, uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída.

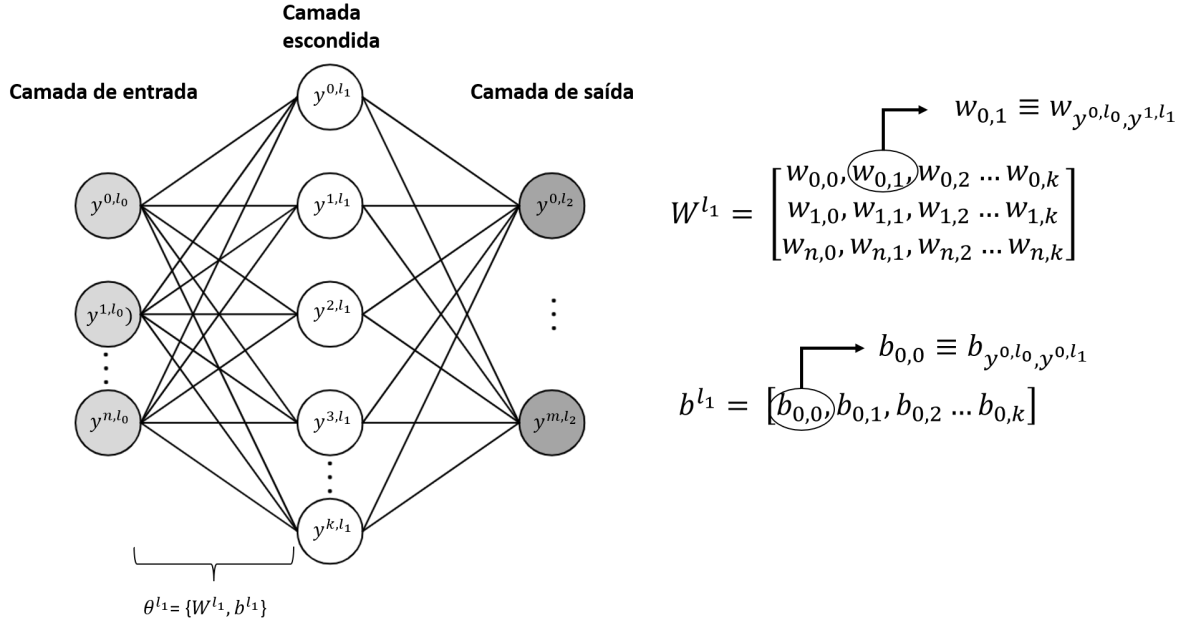


Figura 5.1: Rede neural artificial *feed-forward*.

Na arquitetura da rede neural artificial *feed-forward*, cada neurônio da camada l_i está conectado a todos os neurônios da camada l_{i-1} e o estado de cada neurônio da camada l_i no instante de tempo t é representado pela seguinte equação

$$y_t^{l_i} = a(W_t^{l_i} y_t^{l_{i-1}} + b_t^{l_i}). \quad (5.1)$$

Na Equação 5.1, $W_t^{l_i}$ representa os *weights* entre todas as conexões dos neurônios pertencentes à camada l_{i-1} com os neurônios pertencentes à camada l_i , $b_t^{l_i}$ representa o viés (*bias*) de cada neurônio pertencente a camada l_i , $y_t^{l_{i-1}}$ representa o vetor de neurônios pertencentes a camada l_{i-1} e $a(\cdot)$ representa a função de ativação. O espaço de valores da camada de entrada da rede neural artificial *feed-forward* são propagados para frente através da Equação 5.1, alcançando os neurônios das camadas escondidas até que produzam os valores finais da camada de saída. Esse processo é chamado de propagação para frente (*forward propagation*) e os valores produzidos pelos neurônios da camada de saída são utilizados como entrada para uma função chamada de *loss function* que produz um valor escalar de custo. Esse valor de custo é utilizado para atualizar os parâmetros $\theta_t = \{W_t, b_t\}$ da rede neural artificial *feed-forward* ao final de cada instante de tempo t . A atualização dos parâmetros é realizada através do método da Descida do Gradiente Estocástico (*Stochastic Gradient Descent* - SGD), cujo objetivo é reduzir o valor de custo

produzido pela *loss function* e aproximar os valores produzidos pela camada saída aos valores de treinamento.

5.2 O Método da Descida do Gradiente Estocástico

O método da Descida do Gradiente Estocástico (SGD) possui como objetivo minimizar o valor de custo produzido pela *loss function*, $L(\theta_t)$, em função dos parâmetros da rede neural artificial, $\theta_t = \{W_t, b_t\}$. Dessa forma, o SGD busca encontrar o valor mínimo de $L(\theta_t)$, que necessita ser uma função convexa e diferenciável, visto que as funções convexas diferenciáveis possuem apenas um único mínimo, que por sua vez é também um mínimo global [21]. Entretanto, a maioria das *loss functions* relaxam a condição de serem convexas e deixam de garantir que o mínimo global seja alcançado, mas produzem valores suficientemente próximos. O gradiente da *loss function*, $\nabla_{\theta_t} L(\theta_t)$, é responsável por apontar a direção na qual a função possui o maior crescimento e como o objetivo é alcançar o mínimo da função ao invés do máximo, é necessário apontar para o sentido oposto ao do gradiente, $-\nabla_{\theta_t} L(\theta_t)$. Portanto, com a descida do gradiente da *loss function* é possível estabelecer o seguinte algoritmo:

1. Define-se um valor inicial para os parâmetros da rede neural artificial;
2. Computa-se $-\nabla_{\theta_t} L(\theta_t)$ e atualiza-se os parâmetros da rede neural artificial com o valor calculado;
3. Verifica-se se o valor do gradiente está próximo de zero ou dentro de um limiar estabelecido, caso esteja o procedimento é encerrado. Caso contrário volta-se à Etapa 2.

A atualização dos parâmetros da rede neural artificial ocorre a partir da Equação 5.2 e o processo é ilustrado a partir da Figura 5.2.

$$\theta_{t+1} = \theta_t - \nabla_{\theta_t} L(\theta_t) \quad (5.2)$$

Ao utilizar a Equação 5.2, os parâmetros da rede neural artificial são atualizados, entretanto, sem regular adequadamente como o valor do gradiente impacta na atualização dos parâmetros, o tempo de convergência do algoritmo pode aumentar significativamente, como ilustra a Figura 5.3.

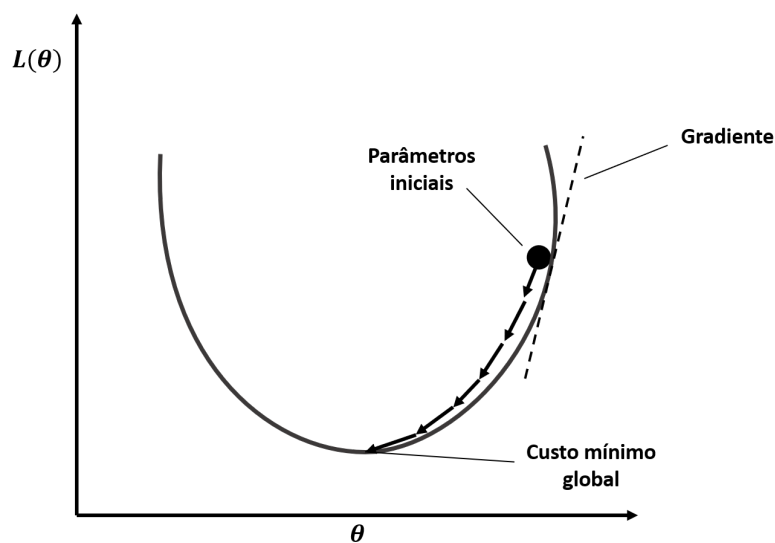


Figura 5.2: Processo de atualização dos parâmetros da rede neural artificial pelo algoritmo da descida do gradiente estocástico.

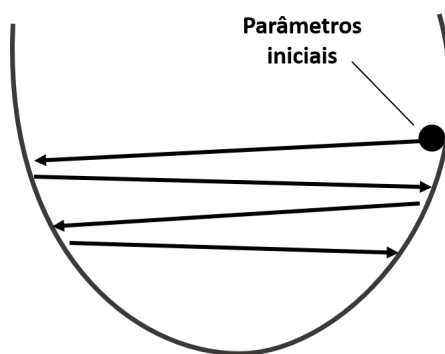


Figura 5.3: Atualização dos parâmetros da rede neural artificial sem o uso de um fator de regulação.

Utilizar inteiramente o valor calculado para a descida do gradiente dificulta o processo de convergência de alcançar o mínimo global, visto que a descida do gradiente pode realizar saltos enormes na função convexa. Dessa forma, emprega-se o fator η , chamado de taxa de aprendizagem (*learning rate*), para regular a proporção do valor calculado pelo gradiente que será utilizado para atualizar os parâmetros da rede neural artificial. A Equação 5.2 é modificada para incorporar a taxa de aprendizado η e passa a ser a seguinte equação

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} L(\theta_t). \quad (5.3)$$

A Equação 5.3 é empregada no treinamento da rede neural artificial a cada novo conjunto de dados fornecidos na camada de entrada, representada pelo instante de tempo t . Entretanto, a atualização dos parâmetros não é realizada sequencialmente. A atualização

realizada pela descida do gradiente estocástico ocorre pela amostragem aleatória uniformemente distribuída do conjunto de dados de treinamento. Ao realizar a atualização dos parâmetros da rede neural artificial de maneira aleatória, é possível evitar que ocorra alto grau de correlação entre as sequências de experiências no aprendizado do agente e consequentemente o problema de sobre-ajuste (*overfitting*) [53].

5.3 O Algoritmo *Back-Propagation*

O método da descida do gradiente estocástico tem como objetivo encontrar o conjunto de parâmetros θ que minimizem a função de custo $L(\theta)$. Entretanto, o gradiente computado sobre a *loss function* atualiza os parâmetros que são utilizados como entrada somente para a função de ativação para a camada de saída. Considere a Figura 5.4 como um exemplo de rede neural artificial, na qual l_e e l_s representam as camadas de entrada e de saída respectivamente, l_1 e l_2 representam as camadas escondidas, $\theta^l = \{W^l, b^l\}$ representa os parâmetros entre as camadas e $a(\cdot)$ a função de ativação.

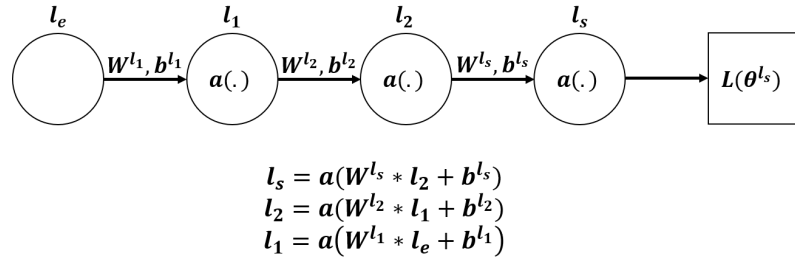


Figura 5.4: Encadeamento dos neurônios de cada camada da rede neural artificial.

Dessa forma, ao computar-se o gradiente $\nabla_{\theta^{l_s}} L(\theta^{l_s})$ da *loss function* $L(\theta^{l_s})$, obtém-se o valor que é utilizado para atualizar os parâmetros θ^{l_s} da camada l_s . Para atualizar os parâmetros θ^{l_2} da camada l_2 deve-se calcular o gradiente $\nabla_{\theta^{l_2}} L(\theta^{l_s}) = \nabla_{\theta^{l_s}} L(\theta^{l_s}) * \nabla_{\theta^{l_2}} l_s$. Isto é, calcula-se o gradiente da *loss function* em relação aos parâmetros da camada l_s , atualiza-se os parâmetros θ^{l_s} com a Equação 5.3, obtém-se o resultado da camada l_s com os parâmetros atualizados e então calcula-se o gradiente da camada l_s em relação aos parâmetros θ^{l_2} da camada l_2 . Esse processo de atualização dos parâmetros entre as camadas da rede neural artificial, ilustrado pela Figura 5.5, ao começar pelo resultado da *loss function* obtido a partir da camada de saída até chegar a camada de entrada é chamado de *back-propagation* [62].

O processo ilustrado pela Figura 5.5 é generalizado para quantidades de neurônios e camadas arbitrário e nada mais é que a aplicação da regra da cadeia para o cálculo

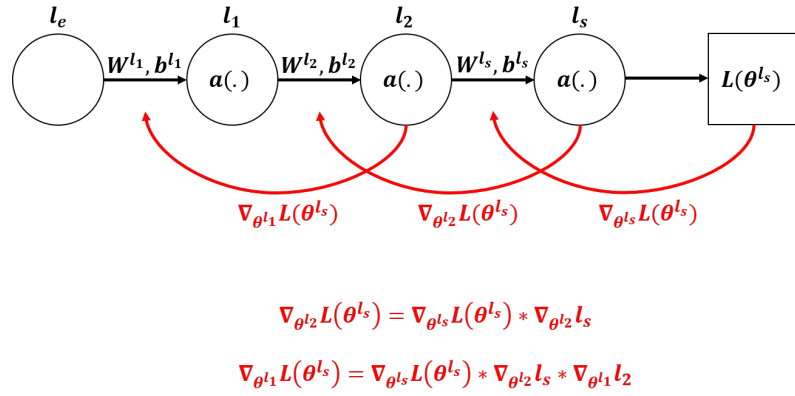


Figura 5.5: Processo do algoritmo de *back-propagation* para atualizar os parâmetros entre as camadas da rede neural artificial.

da derivada de funções compostas, como ocorre na redes neurais artificiais. Nas redes neurais artificiais, os valores estabelecidos para os neurônios de cada camada são obtidos pela função de ativação, que recebe como entrada o resultado da equação linear entre os valores dos neurônios da camada anterior e os parâmetros estabelecidos entre as duas camadas. Dessa forma, os neurônios da última camada são uma função composta das funções de ativação e dos parâmetros de todas as camadas anteriores e o cálculo do gradiente para atualizar os parâmetros da camada l_i pode ser obtido pela Equação 5.4

$$\nabla_{\theta^{l_i}} L(\theta^{l_n}) = \nabla_{\theta^{l_n}} L(\theta^{l_n}) \prod_{i=n}^1 \nabla_{\theta^{l_{i-1}}} l_i \quad (5.4)$$

na qual n representa o índice da camada de saída da rede neural artificial.

5.4 Aprendizado de Máquina por Reforço

Um agente de aprendizado de máquina por reforço possui como objetivo aprender quais são as melhores políticas para um problema de decisões sequenciais através de ações e das recompensas acumuladas que são obtidas como consequência das ações [68]. O agente não sabe de antemão qual a ação deve executar, dessa forma é necessário que o agente aprenda as melhores ações através de tentativa e erro. O método de aprendizado das melhores ações do agente pode então ser resumido como um laço fechado de selecionar uma ação, observar como a ação selecionada impacta no ambiente na qual a ação foi empregada e qual a recompensa associada ao alterar o ambiente com a ação selecionada através de um longo período de tempo. Nesse sentido, é possível descrever um problema de aprendizado por reforço como um processo de decisão de Markov (*Markov Decision*

Process - MDP), no qual um agente interage com ambientes observáveis em uma sequência de instantes de tempo discretos. Em cada instante de tempo t , o agente observa um estado $S_t \in \mathcal{S}$ do ambiente e seleciona uma ação $A_t \in \mathcal{A}(S_t)$, na qual $\mathcal{A}(S_t)$ é o conjunto de todas as ações possíveis quando o ambiente encontra-se no estado S_t . Como consequência da ação A_t , o agente recebe uma recompensa $R_{t+1} \in \mathcal{R}$ e altera o estado do ambiente para o estado S_{t+1} , como mostra a Figura 5.6.

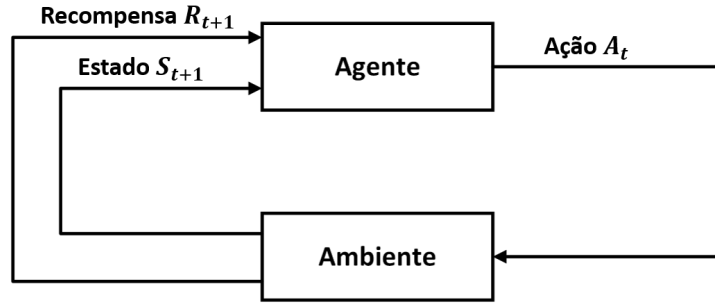


Figura 5.6: Interação entre agente e ambiente no processo de decisão de Markov.

Dessa forma, para cada instante de tempo t , o agente mapeia a probabilidade de selecionar cada possível ação $A_t \in \mathcal{A}(S_t)$ quando o ambiente encontra-se no estado S_t , e o mapeamento adotado pelo agente é chamado de política π_t . Portanto, $\pi_t(A_t|S_t)$ é a probabilidade do agente selecionar a ação A_t condicionada pelo estado S_t . O algoritmo de aprendizado de máquina por reforço mais amplamente empregado para que o agente estabeleça a política π_t para cada par estado-ação é o *Q-learning* [74]. No *Q-learning*, o agente estima o valor do par estado-ação $Q_\pi(S_t, A_t)$, que indica a recompensa esperada do agente ao selecionar a ação A_t quando o ambiente encontra-se no estado S_t sob a política π . O valor do par estado-ação $Q_\pi(S_t, A_t)$ é estimado pela experiência adquirida do agente ao interagir com o ambiente a longo prazo e a equação de Bellman apresentada na Equação 5.5 expressa a relação entre a recompensa imediata R_{t+1} do estado S_t a recompensa já observada através de interações passadas a partir dos estados que sucedem o estado S_t ao selecionar a ação A_t .

$$Q_\pi(S_t, A_t) = \mathbb{E}_\pi[R_{t+1} + \gamma \max_{A_{t+1}} Q_\pi(S_{t+1}, A_{t+1})], \quad (5.5)$$

na qual $\gamma \in (0, 1) \subset \mathbb{R}$ é um fator de desconto para reduzir a importância do valor esperado sobre as recompensas mais antigas em relação às recompensas mais atuais.

5.5 *Q-learning* Profundo

O algoritmo *Q-learning* pode ser implementado através de uma tabela de consulta ao armazenar os valores para todas as combinações de pares estado-ação computados pelo método de programação dinâmica [74]. A principal desvantagem em utilizar uma tabela de consulta para estabelecer os pares de estado-ação está na quantidade de combinações possíveis entre o espaço de valores que os estados e as ações podem assumir. Quando o espaço de valores dos estados ou ações é contínuo não é possível construir uma tabela e utilizar o método de programação dinâmica. Quando o espaço de valores dos estados ou das ações é muito grande, a tabela de consulta requer um uso enorme de memória para armazenar os pares estado-ação, além de exigir um número de iterações maior para que haja convergência nos valores dos pares estado-ação para o valor de recompensa esperado. Para superar essa limitação é possível substituir a tabela de consulta por uma função de aproximação parametrizada $Q_\pi(S_t, A_t; \theta_t)$ representada por uma rede neural artificial *feed-forward*, na qual θ_t representa os parâmetros da rede neural artificial, chamada *Q-learning* Profundo [53]. A Figura 5.7 ilustra como ocorre a interação entre o agente e o ambiente no *Q-learning* Profundo.

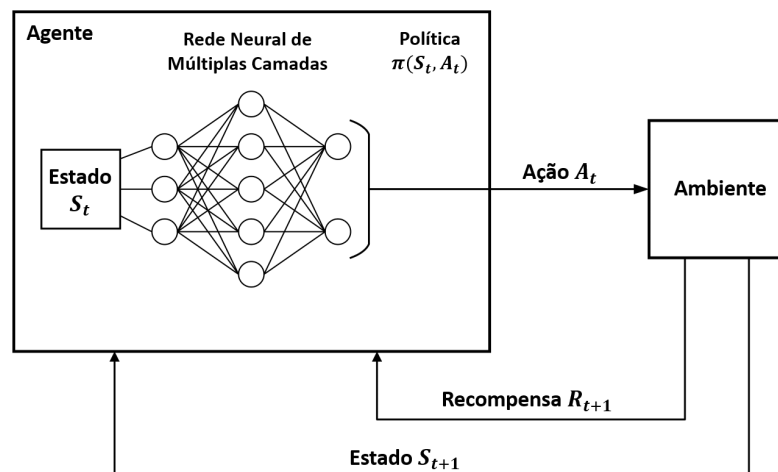


Figura 5.7: Interação entre o agente e o ambiente no *Q-learning* Profundo.

Diferentemente do algoritmo *Q-learning* realizado por tabela de consulta, no *Q-learning* Profundo o agente mapeia o espaço de valores dos estados de dimensão n no espaço de valores de ação de dimensão m utilizando uma rede neural artificial de múltiplas camadas como função de aproximação $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ [70]. Essa rede neural artificial é chamada de *Deep Q-Network* (DQN) [65].

Capítulo 6

Proposta de Agente de Aprendizado de Máquina por Reforço Profundo

Neste capítulo descreve-se o modelo do agente de aprendizado de máquina por reforço profundo proposto nesta tese e como o agente realiza o mapeamento do estado da *Next-Generation Network* em uma ação. A ação do agente consiste em selecionar um segmento de conteúdo para ser armazenado em um *cache* conectado à um nó NG-RAN. O capítulo começa descrevendo o modelo de sistema ao qual o agente interage na Seção 6.1. Em seguida, descreve o espaço do estado observado pelo agente na Seção 6.2 e o espaço de ações do agente na Seção 6.3. A Seção 6.4 descreve como são calculadas as recompensas utilizadas pelo agente para avaliar os pares de estado-ação. Por fim, a Seção 6.5 apresenta e discute os trabalhos relacionados.

6.1 Modelo de Sistema

O modelo de sistema é composto de três modelos que estão descritos nesta seção: (i) o modelo da rede, (ii) o modelo de comunicação, (iii) o modelo de *caching*. A Seção 6.1.1 introduz o modelo da rede ao descrever seus elementos e como estes elementos se conectam. Em seguida, introduz-se o modelo de comunicação na Seção 6.1.2 ao descrever como os UEs requisitam e recuperam os conteúdos. Por fim, introduz-se o modelo de *caching* ao descrever como os segmentos dos conteúdos são armazenados nos *caches* e recuperados na borda da rede de telefonia móvel.

6.1.1 Modelo da Rede

A Figura 6.1 ilustra a arquitetura lógica da NG-RAN que compõe o modelo da rede. O modelo da rede consiste de um ambiente NGN. A NGN é composta por um conjunto de nós NG-RAN que são representados pelas gNBs e ng-eNBs e pelo modelo como o conjunto $\mathcal{B} = \{b_1, b_2, \dots, b_B\}$. Os nós NG-RAN podem ser acessados por um conjunto de UEs que são representados no modelo como o conjunto $\mathcal{U} = \{u_1, u_2, \dots, u_U\}$. Um UE u se conecta a um nó NG-RAN b por meio de conexões de ondas milimétricas [79]. Os nós NG-RAN se conectam ao 5GC através da interface NG e a conexão entre nós NG-RAN e seus vizinhos ocorre pela interface Xn. Cada nó NG-RAN $b \in \mathcal{B}$ pode estar conectado a um servidor *cache* para armazenar segmentos de conteúdos. O conjunto de conteúdos disponível a ser requisitado é representado pelo conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_F\}$ e cada conteúdo $f \in \mathcal{F}$ é dividido em um conjunto de segmentos, sendo n_f o número de segmentos do conteúdo f . O agente proposto é implantado no 5GC e obtém as informações necessárias para estabelecer os estados e as ações através das funções de rede UPF e AMF. O UPF é responsável pelo encaminhamento, roteamento e inspeção dos pacotes, em garantir a qualidade do serviço e manter o estado do tráfego. Já o AMF atua como o ponto de entrada para as conexões do UEs e realiza as operações de gerenciamento de mobilidade, de registro e de conexão [6].

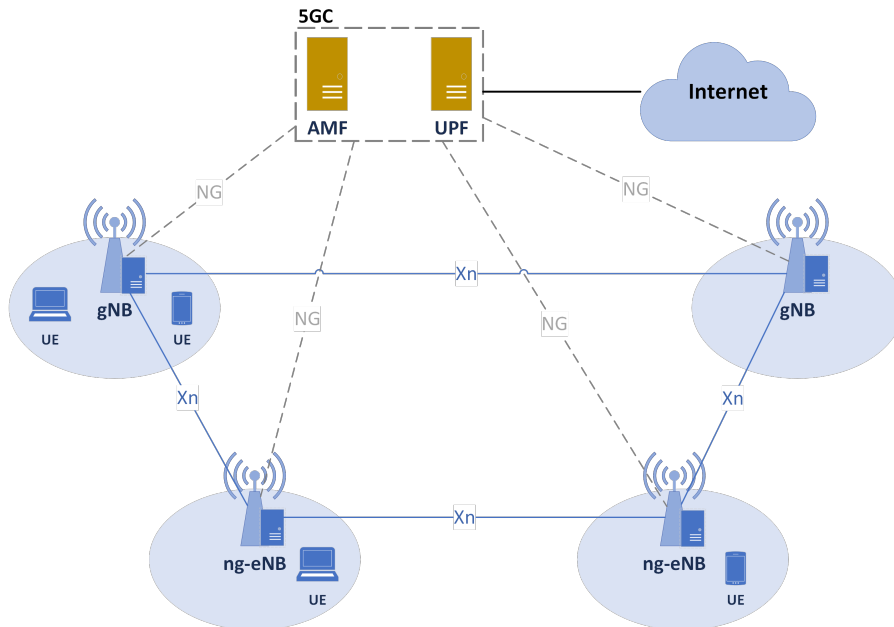


Figura 6.1: A arquitetura lógica da NG-RAN.

6.1.2 Modelo de Comunicação

O modelo de comunicação consiste na associação entre os UEs e os nós NG-RAN, e as requisições realizadas pelos UEs em cada instante de tempo $t \in \mathcal{T} = \{t_1, t_2, \dots, t_T\}$. A associação entre um UE $u \in \mathcal{U}$ e um nó NG-RAN $b \in \mathcal{B}$ no instante de tempo t é representado por $a_t^{u,b} \in \{0, 1\}$, sendo que o UE u deve estar associado a pelo menos um nó NG-RAN e no máximo um nó NG-RAN, como está representado pela Equação 6.1

$$\sum_{b \in \mathcal{B}} a_t^{u,b} = 1, \forall u \in \mathcal{U}, t \in \mathcal{T}. \quad (6.1)$$

Assume-se que a associação é baseada na distância entre o UE e o nó NG-RAN e a cada instante de tempo t_i , o UE u está associado a um nó NG-RAN b e essa associação se mantém sem alterações até o início do próximo instante de tempo t_{i+1} , como é representado pela Equação 6.2

$$a_t^{u,b} = a_{t'}^{u,b}, \forall t \neq t' \in [t_i, t_{i+1}), \forall i \in [1, T]. \quad (6.2)$$

Cada UE u requisita no máximo um conteúdo f a cada instante de tempo t , $\sum_{f \in \mathcal{F}} r_t^{u,f} \leq 1, \forall u \in \mathcal{U}, t \in \mathcal{T}$, e o número de requisições por conteúdo f recebido pelo nó NG-RAN b no instante de tempo t é equivalente ao somatório das requisições de cada UE u associado ao nó NG-RAN b , como é representado pela Equação 6.3

$$r_t^{f,b} = \sum_{u \in \mathcal{U}} r_t^{u,f} a_t^{u,b}, \forall b \in \mathcal{B}, f \in \mathcal{F}, t \in \mathcal{T}. \quad (6.3)$$

Assume-se que todas as requisições recebidas pelo nó NG-RAN b no instante de tempo t chegam em uma ordem aleatória, são enfileiradas e satisfeitas através da política *First Come First Served* - (FCFS). Assume-se também que todo nó NG-RAN b é capaz de satisfazer todas as $r_t^{f,b}, \forall f \in \mathcal{F}$ requisições recebidas até o início do próximo instante de tempo. Dessa forma, o número de requisições satisfeitas pelo nó NG-RAN b no instante de tempo t é representado pela Equação 6.4

$$s_t^b = \sum_{f \in \mathcal{F}} r_t^{f,b}, \forall b \in \mathcal{B}, t \in \mathcal{T}. \quad (6.4)$$

O atraso $d_t^{b,f}$ para que o nó NG-RAN b satisfaça a requisição por um segmento do

conteúdo f pode ser dividido em local, cooperação e remoto. O atraso local $d_{local}^{b,f}$ ocorre quando o nó NG-RAN b satisfaz a requisição recebida pelo servidor *cache* que está conectado ao nó. O atraso de cooperação $d_{coop}^{b,f}$ ocorre quando o nó NG-RAN b satisfaz a requisição através da interface Xn e recupera o segmento de conteúdo utilizando o servidor *cache* de nó NG-RAN $v \neq b$. Quando o segmento de conteúdo não está armazenado no servidor *cache* local do nó NG-RAN b e também não está armazenado em um servidor *cache* conectado em nó NG-RAN $v \neq b, \forall v \in \mathcal{B}$, o segmento de conteúdo deve ser recuperado remotamente pelo provedor do conteúdo, resultando no atraso remoto $d_{remoto}^{b,f}$. Considere $c_t^{b,f} = 1$ quando o servidor *cache* conectado ao nó NG-RAN b armazena o conteúdo f e $c_t^{b,f} = 0$ caso contrário. O atraso $d_t^{b,f}$ pode ser representado pela Equação 6.5

$$d_t^{b,f} = \begin{cases} d_{local}^{b,f}, & c_t^{b,f} = 1, \\ d_{coop}^{b,f}, & c_t^{b,f} = 0 \text{ e } \sum_{v \neq b | \forall v \in \mathcal{B}} c_t^{v,f} \geq 1, \\ d_{remoto}^f, & \text{caso contrário.} \end{cases} \quad (6.5)$$

6.1.3 Modelo de *Caching*

Na NGN, cada nó NG-RAN b está conectado a um servidor *cache* com capacidade de armazenar n_b MB para armazenar segmentos de conteúdos temporariamente e prover esses conteúdos aos UEs fisicamente mais próximos. A capacidade de armazenamento de n_b MB é limitada. Entretanto, os nós NG-RAN formam uma rede de *caches* distribuídos na borda da rede de telefonia móvel ao utilizar a interface Xn. Dessa forma, os nós NG-RAN vizinhos conseguem prover um conteúdo requisitado que não esteja armazenado no servidor *cache* local de um nó NG-RAN. Seja n_f o número de segmentos do conteúdo f , a Equação 6.6 representa o limite de segmentos que cada servidor *cache* é capaz de armazenar

$$\sum_{f \in \mathcal{F}} n_f c_t^{b,f} \leq n_b, \forall b \in \mathcal{B}, t \in \mathcal{T}. \quad (6.6)$$

A Equação 6.6 define o número máximo de conteúdos completos que cada servidor *cache* é capaz de armazenar no instante de tempo t , entretanto, os servidores *cache* podem armazenar parcialmente os conteúdos para prover maior diversidade de conteúdos. Seja $n_t^{b,f}$ o número de segmentos do conteúdo f que o servidor *cache* armazena no instante de tempo t . A Equação 6.7 define o número de segmentos e quais conteúdos o servidor *cache* armazena no instante de tempo t

$$\sum_{f \in \mathcal{F}} n_t^{b,f} c_t^{b,f} \leq n_b, \forall b \in \mathcal{B}, t \in \mathcal{T}. \quad (6.7)$$

O UPF é responsável por manter controle do posicionamento dos conteúdos e por encaminhar as requisições por conteúdos nos nós NG-RAN para o servidor *cache* que provê o menor atraso ou para encaminhar a requisição até o provedor do conteúdo [6]. Quando o servidor *cache* alcança sua capacidade máxima, a política de substituição *Least Recently Used* (LRU) [45] é acionada para armazenar um novo segmento de conteúdo.

6.2 Espaço de Estados

O espaço de estados observável pelo agente define as informações que o agente observa para selecionar uma ação e pode ser dividido em parcialmente observável e completamente observável. Define-se como espaço de estados parcialmente observável aquele no qual o agente é capaz de observar somente parte das informações que compõem o estado S_t do ambiente em cada instante de tempo t . Já o espaço de estados completamente observável é aquele no qual o agente obtém todas as informações que compõem o estado S_t . Define-se como um dos elementos do estado S_t neste trabalho a quantidade de requisições $r_t^{f,b}$ recebidas em cada nó NG-RAN $b \in \mathcal{B}$ por cada conteúdo $f \in \mathcal{F}$ em cada instante de tempo $t \in \mathcal{T}$. Visto que $r_t^{f,b}$ só é completamente observável ao final de cada instante de tempo t , se faz necessária uma representação da popularidade de cada conteúdo no início de cada instante de tempo t .

Assuma $\hat{r}_t^{f,b}$ como a estimativa do estado de requisições em cada nó NG-RAN b para cada conteúdo f no início de cada instante de tempo t . Dessa forma, é possível estipular a popularidade global $\hat{p}_t$ na NGN em cada instante de tempo t ao realizar o somatório do estado de requisições de todos os nós NG-RAN para cada conteúdo f como demonstra a Equação 6.8, sendo $\hat{p}_t = \{\hat{p}_t^1, \hat{p}_t^2, \dots, \hat{p}_t^F\}, \forall t \in \mathcal{T}$

$$\hat{p}_t^f = \sum_{b \in \mathcal{B}} \hat{r}_t^{f,b}, \forall f \in \mathcal{F}, t \in \mathcal{T}. \quad (6.8)$$

Com a estimativa global de popularidade dos conteúdos $\hat{p}_t$ no início de cada instante de tempo t o agente é capaz de antecipar os conteúdos mais populares para os servidores *cache* e, com isso, a NGN é capaz de satisfazer a maior parte das requisições sem a necessidade de encaminhar as requisições para os provedores dos conteúdos.

Uma vez que os conteúdos são divididos em segmentos e os UEs requisitam os segmentos do conteúdo sequencialmente, o valor de $\hat{r}_t^{f,b}$ depende dos estados de requisições anteriores. Dessa forma, a predição do estado de requisição no instante de tempo t é definido como uma distribuição condicional $P(\hat{r}_t^{f,b} | r_{t-1}^{f,b}, \dots, r_0^{f,b})$. Para lidar com a natureza sequencial das requisições, aproxima-se o valor de $\hat{r}_t^{f,b}$ em cada nó NG-RAN b com um *Gated Recurrent Unit* (GRU) [18], um tipo de rede neural artificial recorrente (*Recurrent Neural Network* - RRN). Diferentemente das redes neurais artificiais *feed-forward* nas quais as informações só são propagadas de uma camada antecessora para uma camada sucessora, as redes neurais artificiais recorrentes também possuem conexões da camada sucessora de volta para a camada antecessora. GRUs são capazes de aprender através de uma sequência de dados de tamanho variável ao decidir de maneira adaptativa quais informações devem ser lembradas e quais informações devem ser esquecidas. GRUs gerenciam a sequência de dados de tamanho variável ao empregar uma ou mais unidades escondidas (*hidden unit*) h_t cuja a ativação depende dos estados antecessores, de um portão de redefinição (*reset gate*) rg_t , e um portão de atualização (*update gate*) zg_t , a cada instante de tempo t , como está ilustrado na Figura 6.2. Dessa forma, o valor da estimativa do estado de requisição em cada nó NG-RAN é o valor de saída $\hat{r}_t^{f,b} = g(r_{t-1}^{f,b}; \phi_t)$ da função parametrizada $g(\cdot)$ que define o GRU, na qual $\phi_t = \{W_t, b_t\}$ é o conjunto de parâmetros da RRN.

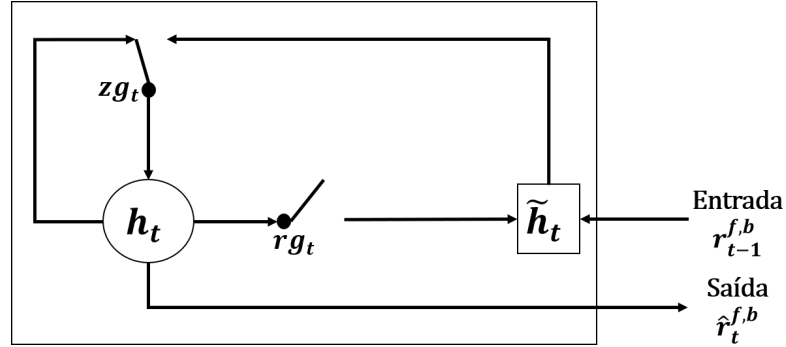


Figura 6.2: Esquema do *Gated Recurrent Unit*.

O portão de atualização zg_t é responsável em determinar o quanto de informação do passado o modelo deve lembrar e é definido a partir da Equação 6.9,

$$zg_t = \sigma(W_t^{e,zg} r_{t-1}^{f,b} + b_t^{e,zg} + W_t^{h,zg} h_{t-1} + b_t^{h,zg}), \quad (6.9)$$

enquanto que o portão de redefinição rg_t é responsável por determinar o quanto de informação o modelo deve esquecer e é definido a partir da Equação 6.10,

$$rg_t = \sigma(W_t^{e,rg} r_{t-1}^{f,b} + b_t^{e,rg} + W_t^{h,rg} h_{t-1} + b_t^{h,rg}), \quad (6.10)$$

sendo $W_t^{i,j}$ e $b_t^{i,j}$ a matriz de pesos e o vetor de viés entre os neurônios das camadas i e j respectivamente no instante de tempo t e $\sigma(\cdot)$ é a função sigmoide. A unidade escondida candidata (*candidate hidden unit*) $\tilde{h}_t$ é a célula que representa o novo conteúdo de memória, que define pelo portão de redefinição em manter somente a informação relevante do passado sobre o conteúdo da célula de memória atual h_t , e adiciona a informação presente na nova sequência de dados de entrada. A Equação 6.10 define como a unidade escondida candidata $\tilde{h}_t$ é atualizada, na qual $\odot$ é o produto de Hadamard

$$\tilde{h}_t = \tanh(W_t^{e,h} r_{t-1}^{f,b} + b_t^{e,h} + rg_t \odot (W_t^{\tilde{h},rg} h_{t-1} + b_t^{\tilde{h},rg})). \quad (6.11)$$

A unidade escondida h_t é a célula de memória final responsável por realizar a predição do estado de requisição $\hat{r}_t^{f,b}$ e é atualizada através da interpolação linear entre o estado anterior da unidade escondida do instante de tempo $t-1$ e a unidade escondida candidata $\tilde{h}_t$ do instante de tempo t , como demonstra a Equação 6.12

$$h_t = (1 - zg_t) \odot \tilde{h}_t + zg_t \odot h_{t-1}. \quad (6.12)$$

A matriz do estado de requisições estimadas é representada por $\hat{r}_t^{f,b} = g(r_{t-1}^{f,b}; \phi_t) = s(h_t)$, sendo $s(\cdot)$ a função softmax [28], como uma distribuição de probabilidade cuja a soma de todas as células da matriz é igual a 1, $\sum_{b \in \mathcal{B}} \sum_{f \in \mathcal{F}} \hat{r}_t^{f,b} = 1, \forall t \in \mathcal{T}$. Os parâmetros $\phi_t = \{W_t, b_t\}$ do GRU são atualizados para maximizar a *loss function* da máxima verossimilhança, representada pela Equação 6.13, através do método da descida do gradiente estocástico e do algoritmo *back-propagation*

$$L(\phi_t) = \max_{\phi_t} \frac{1}{T} \sum_{t=1}^T \log p_{\phi_t}(\hat{r}_t^{f,b} | r_{t-1}^{f,b}). \quad (6.13)$$

O espaço de estados completamente observável S_t do agente também é composto pelo estado global de cada servidor *cache* conectado a um nó NG-RAN b definido pela Equação 6.14

$$c_t^b = n_t^{b,f} c_t^{b,f}, \forall b \in \mathcal{B}, f \in \mathcal{F}, t \in \mathcal{T}, \quad (6.14)$$

sendo $c_t = \{c_t^1, c_t^2, \dots, c_t^B\}$ e o estado S_t definido como $S_t = \{\hat{p}_t, c_t\}, \forall t \in \mathcal{T}$.

6.3 Espaço de Ações

Ao receber o estado completamente observável S_t da NGN, o agente precisa selecionar uma ação que ou maximize o acerto em *cache* das requisições recebidas em cada nó NG-RAN, ou minimize o atraso de recuperação das requisições. O agente seleciona a ação no início de cada instante de tempo e antes do momento no qual os nós NG-RAN satisfazem as requisições recebidas. Dessa forma, o agente armazena antecipadamente (*prefetching*) um segmento de conteúdo em cada instante de tempo e a ação pode ser dividida em três decisões principais: (i) qual conteúdo deve ser selecionado para ser armazenado no servidor *cache*, (ii) qual segmento do conteúdo selecionado será armazenado, (iii) qual o servidor *cache* que será responsável por armazenar o segmento de conteúdo selecionado. Assuma a_t^{cont} como o conteúdo selecionado para ser armazenado no instante de tempo t , a Equação 6.15 define que o conteúdo selecionado é aquele que possui a maior popularidade ao considerar as requisições recebidas por todos os nós NG-RAN

$$a_t^{cont} = \underset{f}{\operatorname{argmax}} \hat{p}_t, \forall t \in \mathcal{T}. \quad (6.15)$$

Uma vez que o conteúdo seja selecionado, o agente deve selecionar qual segmento do conteúdo selecionado será armazenado, representado por a_t^{seg} . Para decidir qual o segmento a ser armazenado, o agente precisa olhar para o estado dos servidores *cache* e verificar quais segmentos já estão armazenados na rede de *caches* distribuída da NGN. Assuma que o conteúdo f é dividido em K segmentos, sendo $f = \{f_0, f_1, \dots, f_K\}$ o conjunto de segmentos. A Equação 6.16 define que o segmento de conteúdo a ser armazenado é aquele que possui o menor índice dentre os segmentos do conteúdo que ainda não estão armazenados no servidores *cache*

$$a_t^{seg} = \min(f_k | f \setminus c_t). \quad (6.16)$$

A última decisão que compõe a ação do agente é a escolha de qual servidor *cache* a ser selecionado para armazenar o segmento de conteúdo de acordo com a política aprendida pelo agente através da DQN apresentada na Seção 5.5. O espaço de ações aprendido pelo agente é proporcional à quantidade de servidores *cache* presentes na NGN e quando o agente seleciona um índice que não pertence aos índices dos servidores *cache* nenhuma

ação é realizada. Assuma que a_t^{serv} representa a decisão do servidor *cache* escolhido para armazenar o segmento de conteúdo, a Equação 6.17 define o servidor *cache* selecionado pelo agente

$$a_t^{serv} = \underset{a_t^{serv} \in A_t}{\operatorname{argmax}} Q_\pi(S_t, A_t; \theta_t). \quad (6.17)$$

A composição da ação no instante de tempo t é resumida como $A_t = \{a_t^{cont}, a_t^{seg}, a_t^{serv}\}$.

6.4 Cálculo da Recompensa

Para atualizar os parâmetros θ_{t+1} da DQN no instante de tempo subsequente, o agente precisa quantificar como a ação A_t , baseada no estado completamente observável S_t , é capaz de recuperar a maioria dos segmentos de conteúdo requisitados pelos servidores *cache* ao invés de encaminhar a requisição para o provedor do conteúdo externo. Para quantificar a ação do agente, propõem-se duas funções de recompensa. A função de recompensa $rw_{b,t}^{atraso}$ calcula o inverso da razão entre o atraso de recuperação experimentado pelo UE sobre a requisição do conteúdo f no nó NG-RAN b , $d_t^{b,f}$ e o atraso de recuperar o segmento de conteúdo pelo provedor do conteúdo, d_{remoto}^f , definido pela Equação 6.18

$$rw_{b,t}^{atraso} = \sum_{f \in \mathcal{F}} \sum_1^{r_t^{b,f}} \left(\frac{d_t^{b,f}}{d_{remoto}^f} \right)^{-1}, \forall b \in \mathcal{B}, t \in \mathcal{T}, \quad (6.18)$$

e a recompensa total para o instante de tempo $t + 1$ é a soma da função de recompensa sobre todos os nós NG-RAN $R_{t+1}^{atraso} = \sum_{b \in \mathcal{B}} rw_{b,t}^{atraso}, \forall t \in \mathcal{T}$. A função $rw_{b,t}^{acerto}$ tem como objetivo estabelecer a quantidade de acertos e faltas em *cache* para cada segmento de conteúdo requisitado, e está definida pela Equação 6.19

$$rw_{b,t}^{acerto} = \sum_{f \in \mathcal{F}} \sum_1^{r_t^{b,f}} I(d_t^{b,f}), \forall b \in \mathcal{B}, t \in \mathcal{T}, \quad (6.19)$$

na qual $I(\cdot)$ retorna o valor $+1$ quando $d_t^{b,f} < d_{remoto}^f$ e o valor -1 quando $d_t^{b,f} = d_{remoto}^f$. A recompensa total para o instante de tempo $t + 1$ é $R_{t+1}^{acerto} = \sum_{b \in \mathcal{B}} rw_{b,t}^{acerto}, \forall t \in \mathcal{T}$.

6.5 Trabalhos Relacionados

He *et al.* [34] e Hu *et al.* [37] propõem uma política de alocação de recursos para redes veiculares baseada em aprendizado por reforço profundo. Os autores argumentam que o armazenamento em *cache* nas redes veiculares auxilia na redução do atraso e na carga do tráfego. O atraso entre os veículos e os serviços que são executados na infraestrutura de nuvem é muito alto, o que pode fazer com que as aplicações que exigem baixas latências não possuam seus requisitos garantidos. A arquitetura MEC é responsável por implantar recursos computacionais próximos aos veículos para lidar com as restrições de baixa latência. A política de alocação de recursos considera como modelo de recursos a comunicação, a computação e o armazenamento em *cache*. O modelo de comunicação é formulado como canais que possuem L níveis discretos como uma cadeia de Markov de estados finitos. O modelo de computação também é formulado como uma cadeia de Markov de estados finitos, na qual a capacidade computacional de cada servidor MEC em um determinado instante de tempo pode estar em N possíveis estados. O modelo de armazenamento em *cache* é formulado como uma cadeia de Markov de dois estados, na qual o estado 0 corresponde ao servidor MEC não armazenar em *cache* o conteúdo requisitado, ou ao estado 1 quando o servidor MEC armazena em *cache* o conteúdo requisitado. O estado observável do agente no instante de tempo t é a composição dos estados de comunicação, computacional e de armazenamento em *cache* de cada servidor MEC. A ação do agente para cada veículo é a composição de três ações: (i) a alocação da estação base baseando-se nas condições dos canais, (ii) a alocação do servidor MEC baseando-se na capacidade computacional, (iii) a alocação do armazenamento em *cache* baseando-se na disponibilidade dos conteúdos. Apesar da formulação dos autores apresentar todos os elementos para otimizar a alocação de recursos para cada veículo na rede, a ação do agente não modifica o estados dos *caches* para que se adaptem à flutuação da popularidade dos conteúdos.

Sadeghi *et al.* [63] e Qiao *et al.* [59] propõem um *framework* de armazenamento de conteúdos na borda da rede de telefonia móvel baseado na cooperação entre as estações base, as unidades de beira de estrada (*Road-Side Unit* - RSU), e os veículos. Os autores argumentam que a maioria das propostas de armazenamento em *cache* na borda das redes de telefonia móvel não se aplica às redes veiculares por causa da alta velocidade de movimentação dos veículos. Para os cenários nos quais a mobilidade é lenta, é válido assumir que o usuário é capaz de recuperar o conteúdo completo dentro da cobertura da mesma estação base. No caso de veículos, há uma grande probabilidade que um veículo

entre e saia da cobertura de diversos RSUs até que o conteúdo completo seja recuperado. Para lidar com o problema da alta velocidade de movimentação dos veículos, os autores assumem que o modelo da rede possui diferentes tipos de nós de armazenamento na borda da rede de telefonia móvel, incluindo estações base de Macro-célula, RSUs, e veículos que armazenam conteúdos. Assume-se que as estações base de Macro-célula possuem capacidade de armazenamento abundante e, portanto, são capazes de armazenar todos os conteúdos disponíveis. As RSUs e os veículos que armazenam conteúdos possuem a capacidade de armazenamento limitada e, portanto, são capazes de armazenar somente uma parcela dos conteúdos disponíveis. O processo de armazenamento cooperativo é dividido em duas granularidades de tempo distintas: (i) o horizonte de armazenamento do conteúdo, (ii) o horizonte de entrega do conteúdo. A variação na escala de tempo para o armazenamento do conteúdo é mais larga ao compará-la com a variação na escala de tempo da entrega do conteúdo. Dessa forma, dentro de dois instantes de tempo do horizonte de armazenamento de um conteúdo, há diversas oportunidades de entrega de conteúdos. No início de cada período de armazenamento de conteúdos, o agente de aprendizado por reforço profundo coleta as informações sobre o estado do sistema. O estado observável do sistema considera qual o conteúdo que está sendo requisitado, o prazo de entrega do conteúdo, o tamanho restante para recuperar completamente o conteúdo, a região de transporte, a popularidade do conteúdo requisitado e o nó de armazenamento do conteúdo requisitado. No início de cada horizonte de entrega de conteúdos, a ação do agente agenda e aloca a largura de banda para o veículo. A ação considera se o conteúdo requisitado está armazenado em *cache*, se o veículo é capaz de se conectar a um nó de armazenamento que possua o conteúdo requisitado armazenado e a quantidade de largura de banda capaz de ser alocada. Os autores consideram que a popularidade do conteúdo é calculada baseando-se na quantidade de requisições realizadas em cada horizonte de armazenamento. Entretanto, o que os autores não consideram, é que o agente somente consegue saber a popularidade dos conteúdos no final do horizonte de armazenamento. Portanto, a cada horizonte de entrega de conteúdo entre os horizontes de armazenamento, o agente é capaz de obter somente a popularidade parcial dos conteúdos para a sua decisão.

Zhong *et al.* [82, 83] propõem um *framework* sobre aprendizado por reforço profundo baseado na arquitetura Wolpertinger [22] para o armazenamento em *cache* na borda das redes de telefonia móvel. O *framework* proposto é implantado em cada estação base e a política do agente baseia-se na popularidade dos conteúdos. Os autores assumem que os usuários estão aleatoriamente distribuídos na rede de telefonia móvel estão cobertos por pelo menos uma estação base. As estações base recebem requisições dos usuários que

estão em sua cobertura simultaneamente e, dentro de um mesmo instante de tempo, os usuários não alteram sua área de cobertura. Cada usuário possui suas próprias preferências para requisitar conteúdos e requisita somente um único conteúdo a cada instante de tempo. Cada requisição é transmitida para todas as estações base que o usuário possui cobertura e a estação base responsável por satisfazer a requisição é aquela capaz de transmitir o conteúdo com o menor atraso. O estado observável do agente compreende três subconjuntos, que representam os estados de requisição curtos, médios e longos. O espaço de ação do agente está em selecionar o conteúdo que será substituído ou em não alterar o estado do *cache*. Embora o espaço de estados considere o estado de requisição em cada estação base, os autores não esclarecem o significado de requisições de curto, médio, e longo prazo. Além disso, o espaço de ações cresce proporcionalmente com a capacidade do *cache*, o que pode aumentar consideravelmente o tempo de treinamento.

Wang *et al.* [72, 73] propõem uma solução cooperativa para o armazenamento em *cache* na borda das redes de telefonia móvel baseada em aprendizado por reforço profundo federada. Os autores consideram uma arquitetura na qual ambos UEs e estações base são capazes de prover armazenamento em *cache* para aprimorar a eficiência espectral. Além disso, os autores assumem que a popularidade global dos conteúdos segue a distribuição Mandelbrot-Zipf. A popularidade local dos conteúdos em cada estação base é definida como a soma das preferências individuais de cada UE. A preferência de cada UE é definida como a razão entre o conteúdo requisitado pelo UE e as requisições de todos os demais UEs associados à estação base. O estado observável do agente em cada instante de tempo considera os estados de requisição e de armazenamento em *cache* da estação base. O estado do armazenamento em *cache* é representado por uma matriz entre conteúdos e estações base, na qual cada célula contém o valor 0 ou 1. O valor 0 representa que a estação base não armazena o conteúdo correspondente à célula, enquanto que o valor 1 representa que o conteúdo está armazenado na estação base. O estado de requisição é representado por uma matriz entre conteúdos e UEs, na qual cada célula contém o valor 0 ou 1. O valor 0 representa que o UE não requisita o conteúdo correspondente à célula, enquanto que o valor 1 representa que o conteúdo é requisitado pelo UE. A ação do agente é composta de três ações diferentes: (i) a ação de processamento local, (ii) a ação do processamento cooperativo, (iii) a ação do processamento remoto, sendo cada ação representada pelo valor 0 ou 1 para indicar a sua execução. A ação do processamento local define que o conteúdo requisitado em uma estação base será processado localmente e o conteúdo requisitado deve substituir um conteúdo já armazenado em *cache* caso este tenha sua capacidade exaurida. A ação do processamento cooperativo define que uma estação base diferente

da estação base que recebe a requisição para o conteúdo é selecionada para satisfazer a requisição. A ação do processamento remoto define que nem a estação base que recebe a requisição para o conteúdo e nem uma estação base vizinha é responsável por satisfazer a requisição. Nesse caso, a requisição é encaminhada para o provedor do conteúdo para que seja satisfeita. Embora os autores considerem a flutuação na popularidade dos conteúdos localmente nas estações base com o passar do tempo, a popularidade local é derivada de uma popularidade local global estacionária e bem conhecida. Além disso, não está esclarecido no trabalho como o agente executa a ação final baseada na composição das três ações, visto que agente pode decidir por realizar as três ações simultaneamente e estas três ações são mutualmente exclusivas.

Chen *et al.* [16] propõem uma solução multi-agente cooperativa para o armazenamento em *cache* na borda das redes de telefonia móvel. Os autores afirmam que a maior parte da largura de banda do *backhaul* das redes de telefonia móvel é ocupado para transmitir o mesmo subconjunto de conteúdos populares. Dessa forma, empregar a arquitetura MEC possui a capacidade de reduzir a ocupação dos enlaces do *backhaul* ao trazer os conteúdos populares para a borda da rede de telefonia móvel. Entretanto, estimar a popularidade dos conteúdos é um problema complexo visto a sua dependência espaço-temporal. Para lidar com a dependência espaço-temporal, os autores propõem utilizar uma rede neural artificial recorrente (*Variational Recurrent Neural Network* - VRNN) para estimar a popularidade dos conteúdos e uma rede neural artificial recorrente (*Long Short-Term Memory* - LSTM) para estimar qual estação base os usuários estarão associados. O estado observável de cada estação base é formulado como a composição do estado de seu *cache*, a estimativa da popularidade dos conteúdos requisitados e a estimativa de associação dos usuários. O estado observável é independente em cada estação base e a ação do agente consiste em selecionar quantos segmentos de um conteúdo devem ser armazenados em *cache*. A ação é influenciada pelo estado observável local da estação base e do estado observável global da rede de telefonia móvel que é transmitido entre as estações base através de um módulo de comunicação. Embora a solução multi-agente seja bem formulada, o espaço de ações é proporcional ao número de conteúdos disponíveis para requisição. Além disso, a ação especifica uma quantidade de segmentos que devem ser armazenadas em *cache*, mas não especifica quais são esses segmentos. As estações base também precisam transmitir o seu estado local para as demais estações base para constituir o estado global, o que implica uma sobrecarga de transmissão dos enlaces de *backhaul*.

A Tabela 6.1 mostra a comparação entre os trabalhos relacionados apresentados e a solução proposta nesta tese. A maior parte dos trabalhos desconsidera que os conteúdos

Tabela 6.1: Comparação dos trabalhos relacionados em aprendizado por reforço profundo.

Referências	Armazenamento Cooperativo	Solução Distribuída	Conteúdos Heterogêneos	Conteúdos Segmentados	Modelo de Popularidade	Estima o Estado das Requisições	Seleção da Ação
He <i>et al.</i> [34]	✓	×	×	×	Dinamicidade Espaço-Temporal	×	Estocástica
Hu <i>et al.</i> [37]	✓	×	×	×	Dinamicidade Espaço-Temporal	×	Estocástica
Sadeghi <i>et al.</i> [63]	×	×	×	×	Dinamicidade Temporal	×	Estocástica
Zhong <i>et al.</i> [82, 83]	×	✓	✓	×	Dinamicidade Espaço-Temporal	×	Estocástica
Qiao <i>et al.</i> [59]	×	×	×	×	Dinamicidade Temporal	×	Estocástica
Wang <i>et al.</i> [72, 73]	✓	×	×	×	Estacionário	×	Estocástica
Chen <i>et al.</i> [16]	✓	✓	✓	✓	Dinamicidade Espaço-Temporal	✓	Estocástica
Solução proposta	✓	×	✓	✓	Dinamicidade Espaço-Temporal	✓	Estocástica + Determinística

são compostos por tamanhos diferentes e são segmentados em pedaços menores para serem recuperados. Além disso, quase todos os trabalhos consideram que o modelo de popularidade dos conteúdos apresenta dinamicidade espaço-temporal. Entretanto, tais trabalhos, com exceção de Chen *et al.* [16], assumem que é possível para o agente obter um estado de requisições completamente observável desde o início de todo e qualquer instante de tempo. O agente proposto nesta tese, ainda que seja uma solução logicamente centralizada, considera a cooperação entre os nós NG-RAN para a recuperação dos conteúdos e que os conteúdos requisitados possuem tamanhos distintos e são segmentados em pedaços menores. Assim como Chen *et al.* [16], o agente proposto assume que a popularidade dos conteúdos apresenta dinamicidade espaço-temporal e que a maneira de obter a informação de popularidade dos conteúdos em cada nó NG-RAN em cada instante de tempo é através de sua estimativa ao considerar os instantes de tempo antecessores.

Em todos os trabalhos relacionados, a ação do agente é completamente estocástica, o que centraliza o processo de decisão completamente no método de aprendizado por reforço profundo empregado pelo agente. Ao tornar o espaço de ações complexo, aumenta-se consideravelmente o tempo de treinamento do agente. Para lidar com a complexidade do espaço de ações, propõem-se para o agente apresentado nesta tese dividir a ação em três sub-ações, sendo uma sub-ação para selecionar o conteúdo popular, uma sub-ação para selecionar o segmento do conteúdo popular a ser armazenado, e uma sub-ação para selecionar o local de armazenamento. O conteúdo popular é selecionado deterministicamente pela estimativa da popularidade dos conteúdos realizada no início de cada instante de tempo. O segmento do conteúdo popular a ser armazenado é deterministicamente selecionado ao observar os segmentos do conteúdo que já estão armazenados nos servidores *cache* da rede. Dessa forma, somente a sub-ação de selecionar o servidor *cache* é selecionada estocasticamente pelo agente, o que reduz significativamente o espaço de ações e o tempo de treinamento do agente.

Capítulo 7

Avaliações do Agente Proposto

Neste capítulo apresentam-se as avaliações realizadas sobre o agente proposto. A Seção 7.1 discute a metodologia empregada para realizar as avaliações. São apresentados o cenário, os parâmetros utilizados e as métricas empregadas para a análise. Na Seção 7.2 são apresentados os resultados obtidos ao avaliar o agente através da metodologia apresentada na Seção 7.1.

7.1 Metodologia de Avaliação

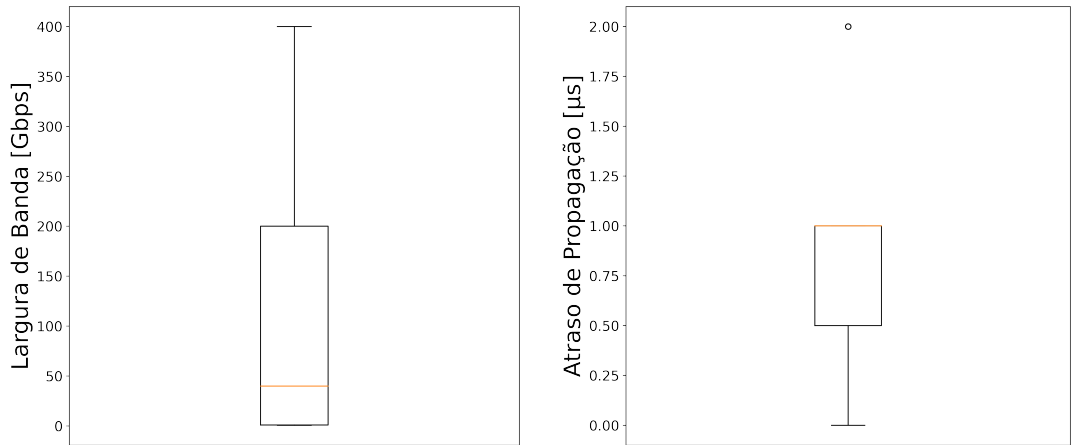
O cenário considerado para avaliar o agente proposto é composto por características presentes em redes de telefonia móvel reais. A Figura 7.1 mostra os nós NG-RAN considerados no cenário de avaliação como círculos pretos e a movimentação de cada UE ao redor dos nós NG-RAN é representada por uma cor. Para selecionar os nós NG-RAN são obtidas aleatoriamente 4 estações base pertencentes ao conjunto de dados do *OpenCellID Project* [69] que fornecem as posições de latitude e longitude de diversas estações base ao redor do mundo. A mobilidade é definida sinteticamente para 99 UEs ao redor dos 4 nós NG-RAN, na qual a cada instante de tempo todos os UEs se locomovem para uma nova posição e ao final da simulação toda a área ao redor dos 4 nós NG-RAN é coberta pela mobilidade dos UEs.

A topologia é composta por 20 elementos de rede no total, sendo 4 nós NG-RAN, 7 servidores *cache* e 8 elementos encaminhadores. A Figura 7.2 mostra as características dos enlaces que compõem a topologia. A Figura 7.2(a) mostra que a largura de banda dos enlaces apresenta uma grande variabilidade, visto que 50% dos valores encontram-se no intervalo entre 1 Gbps e 200 Gbps. Por outro lado, a Figura 7.2(b) mostra que há uma pequena variabilidade no atraso de propagação dos enlaces, visto que o maior valor



Figura 7.1: Posicionamento dos nós NG-RAN e mobilidade dos UEs.

observado é de $2 \mu s$ e 50% dos valores estão no intervalo entre $0,5 \mu s$ e $1 \mu s$.



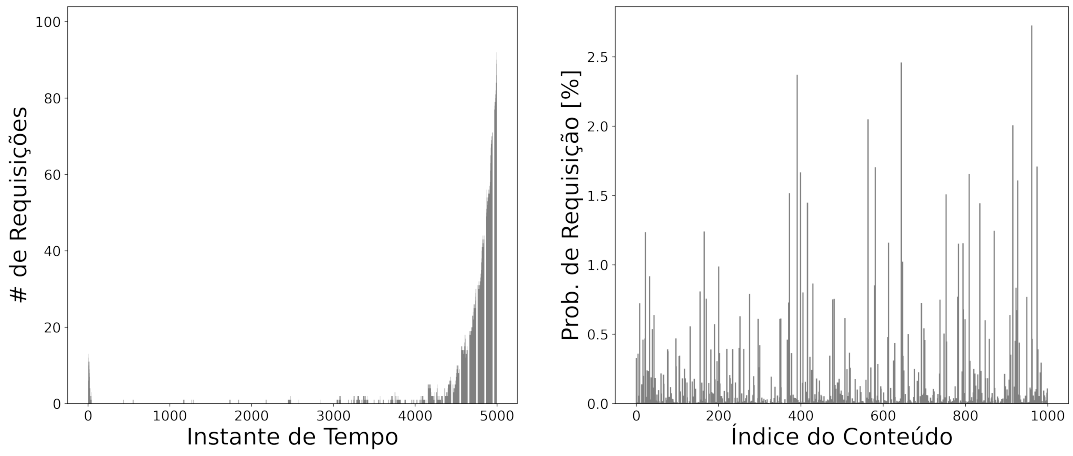
(a) Distribuição da largura de banda dos enlaces.

(b) Distribuição dos atrasos de propagação dos enlaces.

Figura 7.2: Características dos enlaces que interconectam os elementos de rede na topologia.

A simulação é composta por um conjunto $\mathcal{T}$ de instantes de tempo discretos e em cada instante de tempo $t \in \mathcal{T}$ ocorre um conjunto de eventos. Dentre o conjunto de eventos que ocorrem em cada instante de tempo estão as requisições dos UEs, a ação do agente e a recuperação do segmento de conteúdo requisitado por cada UE. O primeiro evento é a ação do agente, e é responsável por prever a popularidade dos conteúdos no instante de tempo e realizar o *prefetching* de n segmentos de conteúdo e armazená-los em um servidor *cache* selecionado pelo agente. Cada segmento de conteúdo dentre os n segmentos de conteúdo armazenados pelo agente corresponde respectivamente aos n conteúdos mais populares previstos no instante de tempo. Em seguida, ocorrem as requisições dos UEs. A Figura 7.3(a) mostra a quantidade de requisições realizadas em cada instante de tempo,

que apresenta o processo de Poisson com a taxa de chegada $\lambda = 0,1$. O tamanho dos conteúdos segue uma distribuição Gamma com os parâmetros $\alpha = 1,8$ e $\beta = 5441,93$, que produzem um tamanho médio de 9809,52 KB, mediana de 8480,33 KB, e uma covariância de 0,93 KB [7]. Por fim, todas as requisições realizadas em um instante de tempo são satisfeitas no mesmo instante de tempo. A seleção do conteúdo a ser requisitado pelo UE segue a distribuição de popularidade derivada do catálogo da Netflix [12] e apresentada na Figura 7.3(b), na qual um catálogo de 1000 conteúdos está disponível para a escolha dos UEs. A simulação apresenta ao todo 5000 instantes de tempo.



(a) Quantidade de requisições realizadas pelos UEs em cada instante de tempo da simulação. (b) Distribuição de probabilidade de requisição de cada conteúdo do catálogo de conteúdos na simulação.

Figura 7.3: Características das requisições realizadas pelos UEs na simulação.

Para avaliar o desempenho do agente proposto na simulação, consideram-se duas métricas, o acerto em *cache* médio (ACM) e a redução média do atraso de recuperação (RMAR) em relação ao atraso de recuperação pela Internet. O acerto em *cache* médio é calculado ao desconsiderar a primeira requisição realizada por cada UE e estabelecer a razão entre a quantidade de requisições atendidas com o atraso de recuperação inferior ao da Internet sobre a quantidade de requisições satisfeitas no instante de tempo, como mostra a Equação 7.1

$$\text{ACM} = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{f \in \mathcal{F}} \sum_{b \in \mathcal{B}} \sum_1^{r_t^{b,f}} \frac{C(d_t^{b,f})}{s_t^b} \quad (7.1)$$

na qual a função $C(\cdot)$ retorna o valor +1 quando $d_t^{b,f} < d_{remoto}^f$ e o valor 0 quando $d_t^{b,f} = d_{remoto}^f$. A métrica da redução média do atraso de recuperação é calculada ao estabelecer a razão entre a diferença entre o atraso de recuperação pela Internet e o atraso

de recuperação do segmento de conteúdo sobre o atraso de recuperação pela Internet para cada requisição realizada pelos UEs e em cada instante de tempo, como mostra a Equação 7.2

$$\text{RMAR} = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{f \in \mathcal{F}} \sum_{b \in \mathcal{B}} \sum_1^{r_t^{b,f}} \frac{\text{Internet} - d_t^{b,f}}{\text{Internet}}. \quad (7.2)$$

A Tabela 7.1 resume os parâmetros utilizados nas simulações realizadas para produzir os resultados que são gerados a partir de 10 rodadas de simulação e para todos os pontos apresentados calcula-se um intervalo de confiança para um nível de confiabilidade de 95% representado por barras verticais.

Tabela 7.1: Resumo dos parâmetros de simulação.

Parâmetro	Valor
Instantes de Tempo	5000
# UEs	99
Atraso da Internet	[100, 200, 300, 400, 500] ms
Capacidade do Cache	[100, 200, 300, 400, 500] MB
Tamanho dos Segmentos	1 MB
# de Rodadas	10
Confiança	95%

7.2 Resultados

Nesta seção apresenta-se os resultados obtidos a partir da avaliação do agente proposto. O método de aprendizado de máquina por reforço é um método de aprendizado não supervisionado, no qual não é necessário um conjunto de dados de treinamento. Uma vez que o agente proposto nesta tese utiliza uma rede neural artificial para mapear os estados da NGN em suas ações, se faz necessário uma etapa de treinamento para ajustar o parâmetro $\theta = \{W, b\}$ da DQN através da *loss function*. Dessa forma, a Seção 7.2.1 apresenta e discute os resultados obtidos a partir do treinamento da DQN e a Seção 7.2.2 apresenta e discute os resultados obtidos ao validar o modelo treinado da DQN. Por fim, a Seção 7.2.3 apresenta os resultados de desempenho obtidos pelo agente após o treinamento e a validação da DQN através das métricas apresentadas na Seção 7.1.

7.2.1 Treinamento

O treinamento do agente é realizado através da própria simulação através de episódios. Um episódio ocorre quando o agente interage com o ambiente em seu estado inicial e passa por todos os estados até chegar ao seu estado terminal. No contexto desta tese, o agente completa um episódio do treinamento ao observar o estado da NGN em seu instante de tempo inicial ($t = 0$) e passar por todos os instantes de tempo até o final da simulação ($t = 4999$). Além disso, há um período de experimentação $\tau \subset E$ do agente no qual o agente somente explora o espaço de ações e armazena a recompensa das ações selecionadas e os estados nas quais foram selecionadas em um *buffer* de experiências $e_t = \{S_t, A_t, R_{t+1}, S_{t+1}\} \in \mathcal{E}$, sendo $\mathcal{E} = \{e_1, e_2, \dots, e_E\}$.

Após as τ experiências, o agente passa a cada instante de tempo t explorar o espaço de ações ao selecionar uma ação aleatoriamente seguindo uma distribuição uniforme com probabilidade ε_t , ou selecionar a ação que resulta na melhor recompensa cumulativa com probabilidade $(1 - \varepsilon_t)$. No início o agente deve empregar uma alta taxa de exploração para adquirir o conhecimento sobre quais são as melhores ações em determinados estados. Com o passar do tempo, o agente adquire esse conhecimento e sua taxa de exploração deve ser reduzida para selecionar com mais frequência as ações que resultam na maior recompensa cumulativa. Dessa forma, utiliza-se um fator de decaimento $\xi \in (0, 1) \subset \mathbb{R}$ para atualizar a taxa de exploração a cada instante de tempo, como mostra a Equação 7.3, na qual ε_{min} é a taxa mínima de exploração utilizada pelo agente

$$\varepsilon_{t+1} = \max(\varepsilon_t \xi, \varepsilon_{min}). \quad (7.3)$$

O agente além de continuar preenchendo o *buffer* de experiências a cada instante de tempo após as τ experiências, também seleciona um subconjunto (*batch*) do *buffer* de experiências para realizar o processo de aprendizado da DQN ao atualizar o parâmetro θ . Uma vez o *batch* selecionado, utiliza-se o estado S_t e a ação A_t para obter o valor atual $Q_\pi(S_t, A_t; \theta_t)$ e o estado seguinte S_{t+1} e a recompensa imediata R_{t+1} para obter o valor esperado Y_t^Q pela equação de Bellman apresentado na Equação 7.4

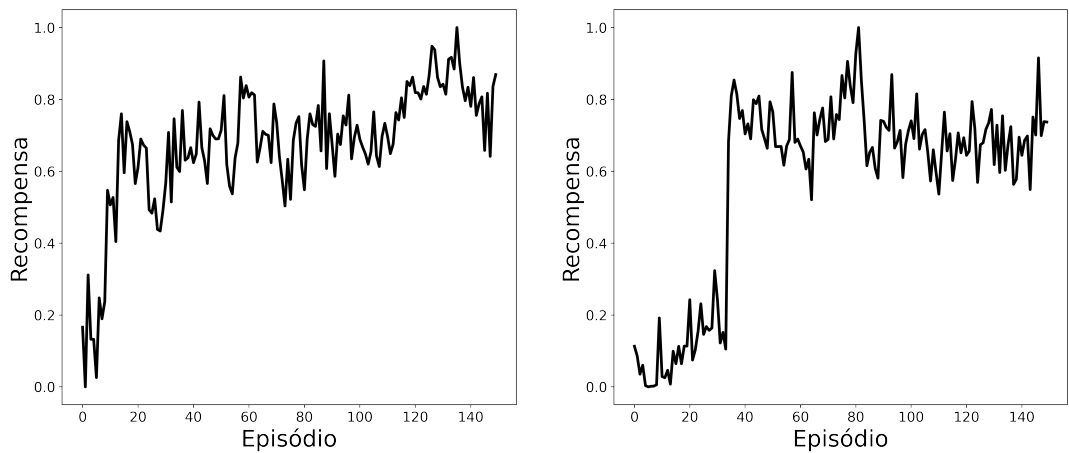
$$Y_t^Q \equiv R_{t+1} + \gamma \max_{A_{t+1}} Q_\pi(S_{t+1}, A_{t+1}; \theta_t). \quad (7.4)$$

Ambos os valores $Q_\pi(S_t, A_t; \theta_t)$ e Y_t^Q são utilizados para medir o desempenho do treinamento através da *loss function* $L(\theta_t)$ apresentada pela Equação 7.5

$$L(\theta_t) = \begin{cases} 0.5(Q_\pi(S_t, A_t; \theta_t) - Y_t^Q)^2/\beta, & \text{se } |Q_\pi(S_t, A_t; \theta_t) - Y_t^Q| < \beta, \\ |Q_\pi(S_t, A_t; \theta_t) - Y_t^Q| - 0.5\beta, & \text{caso contrário.} \end{cases} \quad (7.5)$$

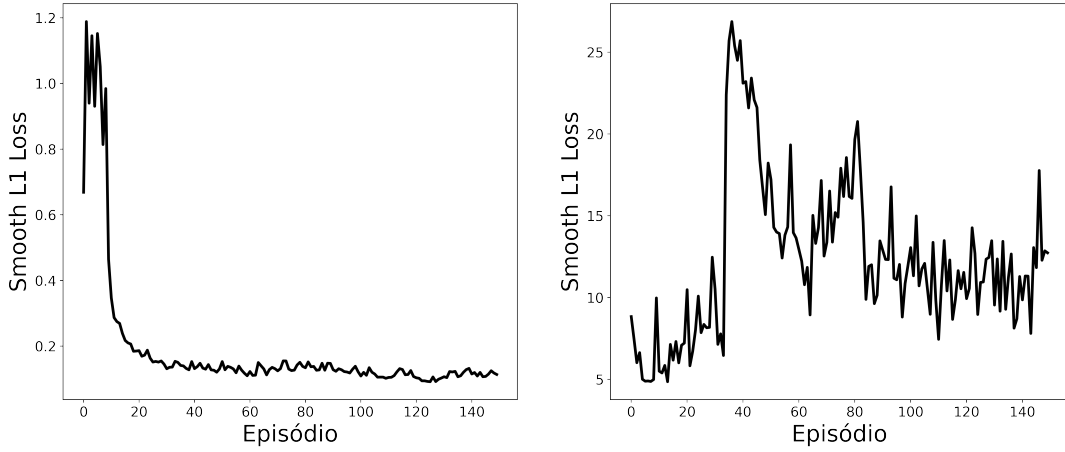
A *loss function* apresentada pela Equação 7.5 é chamada de *Smooth L1 loss* [71]. A *Smooth L1 loss* é uma combinação das *loss functions* *Least Absolute Deviations* (*L1 loss*) e *Least Square Errors* (*L2 loss*). A *L1 loss* busca minimizar a diferença absoluta entre o valor atual obtido pela DQN $Q_\pi(S_t, A_t; \theta_t)$ e o valor desejado Y_t^Q , e portanto é mais robusta e menos afetada por valores atípicos. A *L2 loss* busca minimizar a diferença quadrática entre o valor atual obtido pela DQN $Q_\pi(S_t, A_t; \theta_t)$ e o valor desejado Y_t^Q e, portanto, é mais sensível a valores atípicos, sendo mais apropriada para capturar a diferença de valores que não estejam muito distantes.

Dessa forma, o aprendizado do agente pode ser medido pela evolução dos valores obtidos para a recompensa acumulada e para a *loss function* ao longo de diversos episódios. O objetivo do treinamento está em maximizar o valor de recompensa obtido e minimizar o valor da *loss function* ao longo dos episódios. Os valores obtidos pela *loss function* em cada instante de tempo e, acumulados em cada episódio, são utilizados pelos algoritmos da descida do gradiente estocástico e do *back-propagation* para ajustar o parâmetro θ da DQN que são executados pelo algoritmo Adam [43]. A Figura 7.4 mostra a evolução dos valores obtidos a partir das funções de recompensas acumuladas propostas na Seção 6.4, e a Figura 7.5 mostra a evolução dos valores obtidos a partir da *Smooth L1 loss* para ambas as funções de recompensas acumuladas.



(a) Evolução da recompensa acumulada R_{t+1}^{acerto} ao longo dos episódios de treinamento. (b) Evolução da recompensa acumulada R_{t+1}^{atraso} ao longo dos episódios de treinamento.

Figura 7.4: Evolução das funções de recompensas acumuladas propostas durante o processo de treinamento.



(a) Evolução da *Smooth L1 loss* para a função de recompensas acumuladas R_{t+1}^{acerto} ao longo dos episódios de treinamento. (b) Evolução da *Smooth L1 loss* para a função de recompensas acumuladas R_{t+1}^{atraso} ao longo dos episódios de treinamento.

Figura 7.5: Evolução da *Smooth L1 loss* para as funções de recompensas acumuladas propostas.

A Figura 7.4 apresenta os valores normalizados para as duas funções de recompensas acumuladas para comparação. Observa-se que ambas as funções de recompensa acumuladas apresentam um crescimento após os episódios iniciais e se mantêm dentro de um padrão de crescimento e redução dos valores obtidos em cada episódio. Entretanto, a Figura 7.5 não apresenta esse mesmo padrão de estabilidade a respeito da *Smooth L1 loss*. A Figura 7.5(a) mostra a evolução da *Smooth L1 loss* para a função de recompensa acumulada de acerto em *cache* R_{t+1}^{acerto} e observa-se que após os episódios iniciais os valores mantêm estabilidade e estão próximos de zero, que é o objetivo em utilizar a *loss function* e a descida do gradiente estocástico. Por outro lado, a Figura 7.5(b) mostra que a função de recompensas acumuladas de redução do atraso R_{t+1}^{atraso} não apresenta estabilidade e com valores muito distantes de zero, o que se traduz em uma ausência de convergência no processo de treinamento. Após o treinamento, os resultados obtidos pelas funções de recompensas acumuladas e da *loss function* devem passar pelo processo de validação dos modelos treinados. Os hiperparâmetros utilizados para o treinamento do agente estão resumidos na Tabela 7.2.

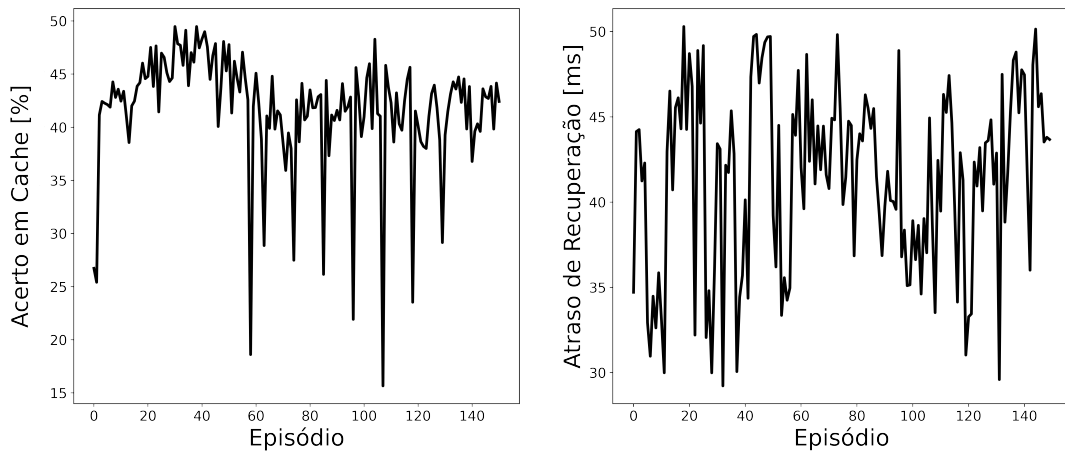
7.2.2 Validação

A validação é um processo realizado após o processo de treinamento e tem como objetivo avaliar o desempenho do modelo treinado. Durante o processo de treinamento o parâmetro θ da DQN é ajustado em cada instante de tempo através do *buffer* de experiências, da aplicação da *loss function*, pelo algoritmo da descida do gradiente estocástico

Tabela 7.2: Resumo dos parâmetros de treinamento.

Parâmetro	Valor
Camadas	Linear = [50, 30]
Taxa de Aprendizado (η)	0,00025
Buffer de experiências	200 experiências
Batch	32 experiências
Fator de Desconto das Recompensas (γ)	0,9
Limiar da Smooth L1 loss (β)	1
Taxa de Exploração Inicial (ε_0)	1
Taxa de Exploração Mínima (ε_{min})	0,02
Decaimento da Exploração (ξ)	0,99

e pelo algoritmo *back-propagation*. Para verificar o desempenho do parâmetro θ ajustado durante o processo de treinamento, realiza-se o processo de validação ao fazer com que o agente seja submetido aos mesmos estados do processo de treinamento, entretanto o parâmetro θ não é ajustado em nenhum momento durante a validação. A Figura 7.6 apresenta os resultados de validação obtidos através das métricas ACM e RARM ao longo dos episódios.



(a) Métrica ACM ao longo dos episódios de validação. (b) Atraso de recuperação médio ao longo dos episódios de validação.

Figura 7.6: Resultados das métricas ACM e atraso de recuperação médio durante o processo de validação.

Observa-se pela Figura 7.6(a) que com exceção dos episódios nos quais há uma queda acentuada na métrica ACM, há estabilidade nos resultados obtidos ao longo dos episódios. A queda acentuada ocorre devido a um fenômeno chamado de esquecimento catastrófico (*catastrophic forgetting*) [61], no qual há perda das informações aprendidas anteriormente quando novas informações são aprendidas. Esse fenômeno ocorre por dois principais motivos: (i) dois estados muito semelhantes levam a valores de recompensas muito distantes

quando o agente realiza a mesma ação sobre esses estados; (ii) o *buffer* de experiências é completamente preenchido e experiências antigas são substituídas por experiências novas.

Já a Figura 7.6(b) não apresenta estabilidade na métrica recuperação de atraso médio obtida ao longo dos episódios no processo de validação, o que demonstra um efeito semelhante nos resultados apresentados na Seção 7.2.1 durante o processo de treinamento. Dessa forma, para os resultados apresentados na Seção 7.2.3, somente o modelo do episódio que apresenta o maior valor para a métrica ACM é utilizado durante as avaliações.

7.2.3 Desempenho

As avaliações de desempenho consideram as variações de atraso de recuperação do segmento pela Internet, as variações das capacidades dos servidores *cache* e a quantidade n de segmentos de conteúdos populares que são armazenados pelo agente no início de cada instante de tempo. A Figura 7.7 apresenta os resultados obtidos pelo agente através da métrica ACM para as capacidades de armazenamento definidas na Seção 7.1 e ao realizar o *prefetching* de $n = \{1, 3, 5\}$ segmentos de conteúdos populares. A linha vermelha tracejada representa um oráculo que já possui conhecimento sobre os conteúdos populares desde o início da simulação e atua na política de substituição dos servidores *cache*. Uma vez que o servidor *cache* tenha sua capacidade exaurida, somente segmentos de conteúdos que são mais populares do que os que já estão armazenados são candidatos a ativarem a política de substituição. O segmento de conteúdo substituído é aquele com o menor índice de popularidade.

Os resultados apresentam a tendência esperada no aumento da métrica ACM ao aumentar a capacidade dos servidores *cache*. Observa-se, para a capacidade de armazenamento em 100 MB, que os valores de ACM produzidos em todas as curvas do agente apresentam o mesmo resultado com o nível de confiabilidade de 95% e esse comportamento se repete até a capacidade de armazenamento em 300 MB. Os maiores valores de ACM aproximam-se de 47%, 60% e 64% para as capacidades de armazenamento em 100 MB, 200 MB e 300 MB respectivamente. A partir da capacidade de armazenamento em 400 MB, as curvas deixam de apresentar o mesmo resultado e as curvas para $n = 3$ e $n = 5$ apresentam a métrica ACM superior à curva $n = 1$. Tal resultado, indica um benefício em armazenar diversos segmentos de conteúdos populares a cada instante de tempo quando o servidor *cache* é capaz de armazenar muitos segmentos de conteúdos. Os conteúdos possuem em média 9,8 MB, dessa forma para 100 MB de capacidade de armazenamento, o servidor *cache* é capaz de armazenar no máximo 10 conteúdos com-

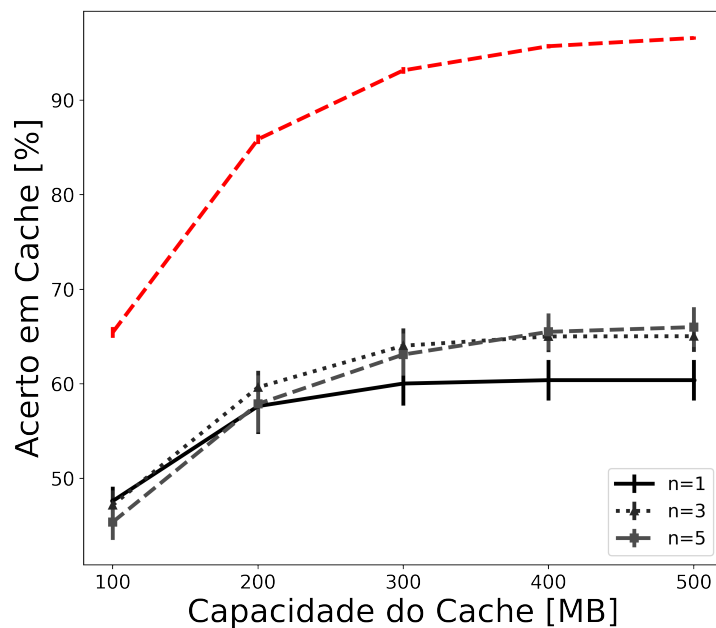
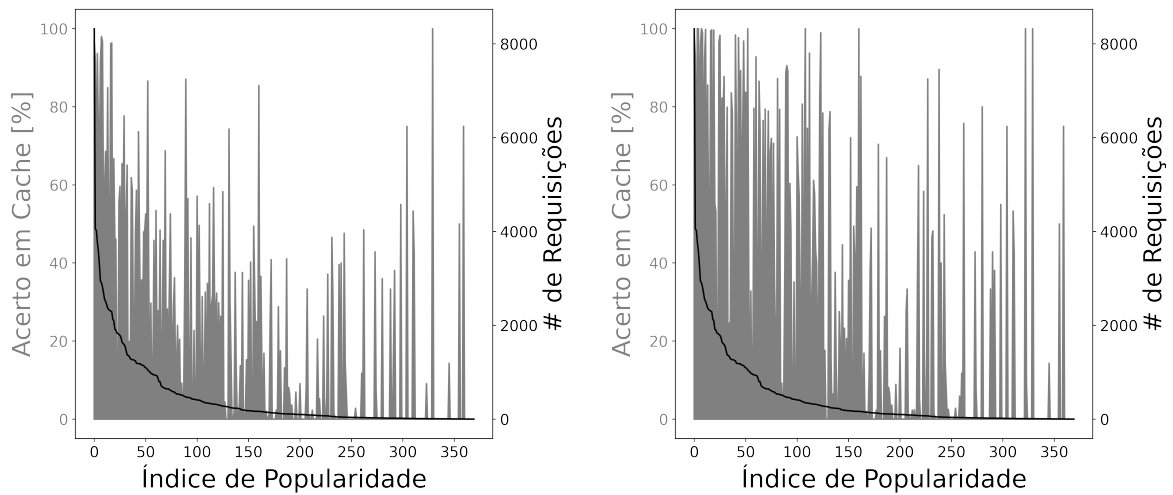


Figura 7.7: Resultados da métrica ACM para as diferentes capacidades de armazenamento do servidor *cache* e para cada quantidade n de segmentos armazenados através do *prefetching*.

pletos. Portanto, tentar manter diversos conteúdos populares a cada instante de tempo ativa frequentemente a política de substituição e, conseqüentemente, altera o estado dos servidores *cache*. Outro fator que sugere esse comportamento é a distância entre o valor de ACM produzido pelo agente e o valor de ACM produzido pelo oráculo. As distâncias são de 17%, 25%, 28%, 30% e 31% para as capacidades de armazenamento em 100 MB, 200 MB, 300 MB, 400 MB e 500 MB, respectivamente. Portanto, a distância entre o oráculo e o agente aumenta em função da capacidade de armazenamento e esse comportamento é explicado pelo funcionamento do oráculo. O oráculo só realiza substituições quando o segmento de conteúdo candidato a ser armazenado é mais popular que algum segmento de conteúdo já armazenado. Dessa forma, reduz-se o número de substituições e mantém-se disponíveis os conteúdos populares por mais instantes de tempo para serem recuperados pelo servidor *cache*.

A Figura 7.8 apresenta em barras verticais cinzas a métrica ACM para cada conteúdo separadamente e pela curva em preto a quantidade de requisições para o conteúdo em função da popularidade dos conteúdos para as capacidade de armazenamento 100 MB e 500 MB na curva $n = 1$. A popularidade dos conteúdos é ordenada de forma crescente. Observa-se que a Figura 7.8(b) apresenta maior densidade de barras verticais próximas a 100% de ACM para os conteúdos mais populares quando comparada com a Figura 7.8(a). Ao aumentar a capacidade de armazenamento do servidor *cache* adia-se a necessidade de



(a) A métrica ACM em função da popularidade do conteúdo para a capacidade de armazenamento em 100 MB e $n = 1$.
 (b) A métrica ACM em função da popularidade do conteúdo para a capacidade de armazenamento em 500 MB e $n = 1$.

Figura 7.8: A métrica ACM em função da popularidade do conteúdo.

acionar a política de substituição e mantém-se durante um período de tempo maior os conteúdos mais populares e obtém-se um efeito semelhante ao produzido pela política de substituição do oráculo. Os resultados apresentados e discutidos para as Figuras 7.7 e 7.8 indicam que a decisão de qual conteúdo deve ser substituído no servidor *cache* uma vez que a sua capacidade seja exaurida deve estar incorporada no espaço de ações do agente. Dessa forma, o agente é capaz de se aproximar da curva apresentada pelo oráculo mantendo a mesma capacidade de armazenamento.

A Figura 7.9 mostra os resultados da métrica RAMR para todas as combinações de atraso da Internet e capacidade de armazenamento, e para as curvas com $n = \{1, 3, 5\}$. Observa-se que, quanto maior a capacidade de armazenamento e quanto maior o atraso da recuperação dos segmentos de conteúdo pela Internet, maior a redução do atraso experienciada pelos UEs. O menor valor de RAMR é de 28% e é observado para a capacidade de armazenamento em 100 MB, atraso da Internet em 100 ms e $n = 5$. Já o maior valor de RAMR é de 61% e é observado para a capacidade de armazenamento em 500 MB, atraso da Internet em 500 ms e $n = 5$. Esse contraste entre o menor e o maior valor observado para a métrica RAMR com $n = 5$ demonstra a ligação entre a redução do atraso na recuperação dos segmentos e do acerto em *cache*, além de reforçar os resultados anteriores sobre a sensibilidade entre armazenar muitos segmentos de conteúdos populares a cada instante de tempo sem que a política de substituição seja ciente da popularidade dos conteúdos.

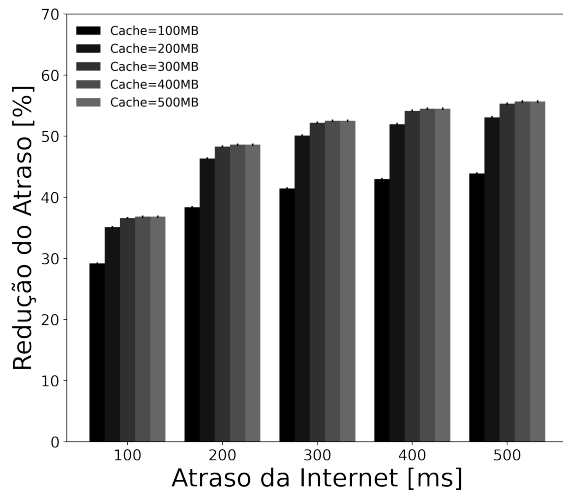
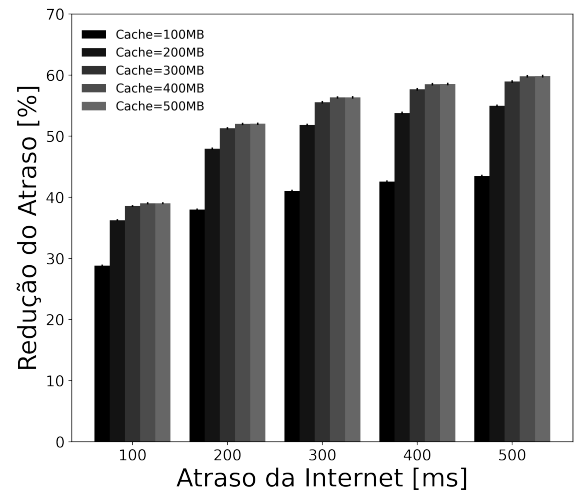
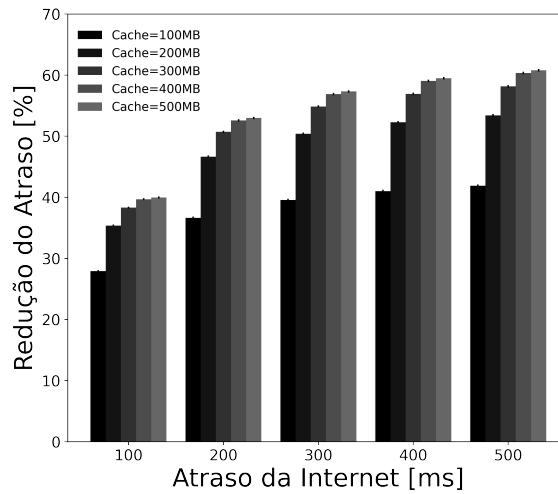
(a) Métrica RARM para a curva $n = 1$.(b) Métrica RARM para a curva $n = 3$.(c) Métrica RARM para a curva $n = 5$.

Figura 7.9: Métrica RARM para todas as combinações de atraso da Internet e capacidade de armazenamento.

Capítulo 8

Conclusão

As atuais redes de telefonia móvel de comunicação 4G LTE estão amplamente implantadas para entregar serviços de banda larga móvel. A comunicação por voz e a banda larga móvel são as principais aplicações das redes de telefonia móvel de comunicação, enquanto aplicações que exigem comunicações confiáveis de baixa latência exigem a incorporação de novas tecnologias e sistemas na nova geração de redes de telefonia móvel. Esta tese apresenta uma revisão da tecnologia das redes 4G LTE atualmente utilizadas para as comunicações móveis e em seguida apresenta as principais alterações a serem incorporadas para as redes 5G, assim como as arquiteturas que estarão presentes para prover as comunicações confiáveis de baixa latência.

O Capítulo 4 apresenta a proposta de formulação para minimizar a alocação de instâncias de *cache* VNF na arquitetura *Multi-access Edge Computing* além de minimizar o atraso de recuperação dos conteúdos que são armazenados nessas instâncias de *cache* VNF por parte dos UEs. Os resultados apresentados indicam que três decisões são de grande importância para o desempenho da rede de telefonia móvel na recuperação de conteúdos em servidores MEC: (i) quais conteúdos devem ser escolhidos para serem armazenados em servidores MEC, (ii) em qual servidor MEC o conteúdo selecionado para ser armazenado deve ser mantido, (iii) em qual servidor MEC que armazena uma cópia do conteúdo requisitado por um UE deve ser selecionado para a sua recuperação.

Assume-se no modelo proposto que o nó controlador possui conhecimento sobre todas os conteúdos que estão sendo requisitados nas estações base. Tal suposição é otimista, visto que não é possível saber de antemão quais conteúdos serão requisitados às estações base e mesmo que o nó controlador possua o conhecimento sobre quais conteúdos serão requisitados, não é possível saber a ordem na qual os conteúdos chegam nas estações base. Dessa forma, para a decisão em (i), uma abordagem que considera o passado de requisições

nas estações base é mais eficiente para selecionar quais mídias devem ser armazenadas nos servidores MEC.

Pelos resultados gerados pelo modelo na Seção 4.3, as decisões (ii) e (iii) são consideradas conjuntamente pelo nó controlador. Os resultados mostram que a grande maioria dos conteúdos armazenados são atribuídos aos servidores MEC que possuem os menores atrasos de transferência das imagens a partir do nó controlador. Em casos nos quais o tamanho das imagens não é muito superior ao tamanho médio das mídias, servidores MEC que possuam baixa dispersão em sua distribuição na vazão entre as estações base também são considerados para armazenar os conteúdos, visto que apresentará um menor valor médio no atraso de recuperação.

Visto que os modelos apresentam soluções otimistas ao desconsiderar a ordem na qual as requisições dos usuários chegam às estações base para abordar o problema de alocação das instâncias de *cache* VNF e recuperação das mídias, propõem-se um agente baseado em aprendizado de máquina por reforço profundo para solucionar as decisões discutidas em (ii) e (iii). Para propor o agente são formulados todo o modelo de sistema no qual o agente atua, assim como o espaço de estados e o espaço de ações do agente sobre o modelo de sistema. São propostas também duas funções de recompensa para o aprendizado do agente, uma função de recompensa que busca maximizar o acerto em *cache* e outra função de recompensa que busca minimizar o atraso de recuperação dos conteúdos. Além disso, o agente proposto utiliza uma rede neural artificial recorrente chamada *Gated Recurrent Unit* para capturar a flutuação espaço-temporal da popularidade dos conteúdos e armazená-los nos *caches* antes que as requisições dos UEs sejam recebidas pelas estações base. Dessa forma, a maioria das requisições que chegam às estações base podem ser satisfeitas sem a necessidade de serem encaminhadas até o provedor do conteúdo.

O Capítulo 7 apresenta e discute os resultados obtidos ao avaliar o agente proposto. As avaliações demonstram que utilizar uma função de recompensa baseada em maximizar o acerto em *cache* produz convergência para o aprendizado do agente durante o treinamento. Por outro lado, uma função de recompensa que busca minimizar os atrasos de recuperação dos conteúdos não produz convergência e durante o processo de validação se mostra instável para o treinamento do agente. Os resultados de desempenho demonstram que o agente é capaz de produzir até 65% de acertos em *cache*, assim como reduzir em 61% o atraso de recuperação dos conteúdos requisitados nas redes de telefonia móvel em relação ao atraso de recuperação pela Internet ao encaminhar as requisições até o provedor do conteúdo.

8.1 Publicações

Esta tese gerou as seguintes publicações:

- I. Bastos, I. Moraes, N. Thi-Mai-Trang, and G. Pujolle. “Modelo e Avaliação da Recuperação de Conteúdos Através de Funções de Rede Virtuais na Arquitetura de Computação na Borda em Redes Móveis”, in Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Gramado, 2019, pp. 749-762.
- Bastos, I. V., Moraes, I. M., Nguyen, T. M. T., and Pujolle, G. “Content Media Retrieval using Virtual Network Functions in Multi-access Edge Computing architecture”, International Journal of Network Management, e2208.

Além dos dois trabalhos já publicados, mais um trabalho intitulado “A Deep Reinforcement Learning Agent for Content Placement in Next Generation Networks” está em produção com a co-autoria dos professores Aline Marins Paes e Igor Monteiro Moraes.

8.2 Trabalhos Futuros

Nas avaliações de desempenho do agente proposto, as curvas obtidas para a métrica de acerto em *cache* são comparadas com uma curva de um oráculo. O oráculo é desenvolvido para atuar nas políticas de admissão e substituição do *cache* ao possuir o conhecimento da distribuição de popularidade dos conteúdos requisitados pelos UEs. O oráculo só realiza substituições quando um segmento de conteúdo candidato é mais popular que algum segmento de conteúdo já armazenado. Ao realizar a admissão e substituição dos conteúdos considerando o conhecimento sobre suas popularidades, o oráculo reduz o número de substituições e mantém disponíveis os conteúdos populares por mais instantes de tempo para serem recuperados pelo servidor *cache*.

O agente proposto não é responsável pelo conteúdo a ser substituído nos *caches*. O espaço de ações do agente compreende as decisões determinísticas sobre o conteúdo e o segmento do conteúdo a ser armazenado e a decisão estocástica de qual *cache* será responsável por armazenar o conteúdo. Os resultados demonstrados pelo oráculo se mostram interessantes para incluir no espaço de ações do agente a ação estocástica de selecionar o conteúdo já armazenado em *cache* que será substituído quando a capacidade do *cache* for exaurida.

Também como trabalhos futuros, é interessante explorar outras soluções para estimar a popularidade dos conteúdos e realizar o aprendizado de máquina por reforço profundo. Além do *Gated Recurrent Unit* para realizar a predição da popularidade dos conteúdos, há outras redes neurais artificiais recorrentes com estruturas mais simples, como a *Variational Recurrent Neural Networks*, assim como outras com estruturas mais complexas como a *Long Short-Term Memory*. Já o aprendizado de máquina por reforço profundo possui outros métodos com estruturas mais complexas a serem investigados como o *Double Deep Q-Network*, no qual duas rede neurais artificiais são utilizadas ao invés de uma. Com as duas redes neurais artificiais, uma é utilizada para selecionar as ações enquanto que a segunda é utilizar para avaliar as ações selecionadas. Ao separar a seleção da avaliação, evita-se a seleção de ações superestimadas.

Referências

- [1] From 1G to 5G: A brief history of the evolution of mobile standards. <https://www.brainbridge.be/news/from-1g-to-5g-a-brief-history-of-the-evolution-of-mobile-standards>. Acessado em 07-02-2020.
- [2] From 1G to 5G – A 40-year journey. <https://apistraining.com/news/from-1g-to-5g/>. Acessado em 07-02-2020.
- [3] LTE evolution: Standardization and deployment: The long run to 5G. <https://www.telit.com/lte-evolution-standardization-and-deployment-the-long-run-to-5g/>. Acessado em 07-02-2020.
- [4] Mobile data traffic outlook. <https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast>. Acessado em 22-08-2022.
- [5] Mobile subscriptions Q3 2019. <https://www.ericsson.com/en/mobility-report/reports/november-2019/mobile-subscriptions-q3-2019>. Acessado em 07-02-2020.
- [6] 3GPP. NR and NG-RAN overall description, 2022. 3GPP Technical Specification TS 38.300, Release 17.
- [7] ABHARI, A.; SORAYA, M. Workload generation for YouTube. *Multimedia Tools and Applications* 46, 1 (2010), 91.
- [8] AHMED, E.; REHMANI, M. H. Mobile edge computing: opportunities, solutions, and challenges, 2017.
- [9] AKYILDIZ, I. F.; GUTIERREZ-ESTEVEZ, D. M.; REYES, E. C. The evolution to 4G cellular systems: LTE-Advanced. *Physical communication* 3, 4 (2010), 217–244.
- [10] ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. *Computer networks* 54, 15 (2010), 2787–2805.
- [11] BAŞTUĞ, E.; BENNIS, M.; DEBBAH, M. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Communications Magazine* 52, 8 (2014), 82–89.
- [12] BENNETT, J.; LANNING, S., ET AL. The Netflix prize. In *Proceedings of KDD cup and workshop* (2007), vol. 2007, New York, NY, USA, p. 35.

- [13] BONOMI, F.; MILITO, R.; ZHU, J.; ADDEPALLI, S. Fog computing and its role in the Internet of Things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (2012), pp. 13–16.
- [14] CAICEDO, O.; CERRONI, W.; LIMAM, N. Series editorial: Network softwarization and management. *IEEE Communications Magazine* 60, 2 (2022), 11–11.
- [15] CHECKO, A. Cloud radio access network architecture. towards 5G mobile networks.
- [16] CHEN, S.; YAO, Z.; JIANG, X.; YANG, J.; HANZO, L. Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks. *IEEE Transactions on Communications* 69, 4 (2020), 2441–2456.
- [17] CHIARAVIGLIO, L.; FIORE, M.; ROSSI, E. The 5G italy book 2019: a multiperspective view of 5G, 2019.
- [18] CHO, K.; VAN MERRIËNBOER, B.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
- [19] COX, C. *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications*. John Wiley & Sons, 2012.
- [20] CZIVA, R.; ANAGNOSTOPOULOS, C.; PEZAROS, D. P. Dynamic, latency-optimal vnf placement at the network edge. In *IEEE International Conference on Computer Communications, INFOCOM* (2018), IEEE, pp. 693–701.
- [21] DATA SCIENCE ACADEMY. *Deep Learning Book*. Data Science Academy, 2022. <https://www.deeplearningbook.com.br>(Acessado: 15/07/2022).
- [22] DULAC-ARNOLD, G.; EVANS, R.; VAN HASSELT, H.; SUNEHAG, P.; LILLICRAP, T.; HUNT, J.; MANN, T.; WEBER, T.; DEGRIS, T.; COPPIN, B. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679* (2015).
- [23] EKTI, A. R.; SHAKIR, M. Z.; SERPEDIN, E.; QARAQE, K. A.; IMRAN, M. A. On the traffic offloading in wi-fi supported heterogeneous wireless networks. *Journal of Signal Processing Systems* 83, 2 (2016), 225–240.
- [24] ETSI-GS-MEC-001. Mobile Edge Computing (MEC) Terminology. https://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/01.01.01_60/gs_MEC001v010101p.pdf. Acessado em 19-02-2020.
- [25] ETSI-GS-MEC-003. Mobile Edge Computing (MEC) Framework and Reference Architecture. https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf. Acessado em 19-02-2020.
- [26] ETSI-GS-NFV-SWA-001. Network Functions Virtualisation (NFV) Virtual Network Functions Architecture. https://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf. Acessado em 19-02-2020.

- [27] FONSECA, F. F.; CORREA, S. L.; CARDOSO, K. V. Optimizing allocation and positioning in a disaggregated radio access network. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2019), SBC, pp. 791–804.
- [28] GAO, B.; PAVEL, L. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805* (2017).
- [29] GARCIA-SAAVEDRA, A.; SALVAT, J. X.; LI, X.; COSTA-PEREZ, X. Wizhaul: On the centralization degree of cloud RAN next generation fronthaul. *IEEE Transactions on Mobile Computing* 17, 10 (2018), 2452–2466.
- [30] GESBERT, D.; KOUNTOURIS, M.; HEATH, R. W.; CHAE, C.-B.; SALZER, T. Shifting the MIMO paradigm. *IEEE signal processing magazine* 24, 5 (2007), 36–46.
- [31] GOLOMB, M. The road to 5G: A brief history of mobile networks. <https://medium.com/@michaelgolomb/the-road-to-5g-a-brief-history-of-mobile-networks-172e50d44c2d>, 2019. Acessado em 07-02-2020.
- [32] GOZALVEZ, J. Tentative 3GPP timeline for 5G [mobile radio]. *IEEE Vehicular Technology Magazine* 10, 3 (2015), 12–18.
- [33] GUPTA, P. Evolvment of mobile generations: 1G to 5G. *International Journal for Technological Research in Engineering* 1 (2013), 152–157.
- [34] HE, Y.; ZHAO, N.; YIN, H. Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology* 67, 1 (2017), 44–55.
- [35] HONG, D. P.; ALBUQUERQUE, C.; OLIVEIRA, C.; SUDA, T. Evaluating the impact of emerging streaming media applications on TCP/IP performance. *IEEE communications Magazine* 39, 4 (2001), 76–82.
- [36] HOYDIS, J.; TEN BRINK, S.; DEBBAH, M. Massive MIMO: How many antennas do we need? In *2011 49th Annual Allerton conference on communication, control, and computing (Allerton)* (2011), IEEE, pp. 545–550.
- [37] HU, R. Q., ET AL. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Transactions on Vehicular Technology* 67, 11 (2018), 10190–10203.
- [38] HU, Y. C.; PATEL, M.; SABELLA, D.; SPRECHER, N.; YOUNG, V. Mobile edge computing—a key technology towards 5G. *ETSI white paper* 11, 11 (2015), 1–16.
- [39] HWANG, I.; SONG, B.; SOLIMAN, S. S. A holistic view on hyper-dense heterogeneous and small cell networks. *IEEE Communications Magazine* 51, 6 (2013), 20–27.
- [40] INTEL CORPORAT. Smart Cells Revolutionize Service Delivery, 2013.
- [41] JIN, Y.; WEN, Y.; WESTPHAL, C. Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 12 (2015), 1914–1925.

- [42] KARDARAS, G.; LANZANI, C. Advanced multimode radio for wireless & mobile broadband communication. In *2009 European Wireless Technology Conference* (2009), IEEE, pp. 132–135.
- [43] KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [44] LECONTE, M.; PASCHOS, G.; GKATZIKIS, L.; DRAIEF, M.; VASSILARAS, S.; CHOUVARDAS, S. Placing dynamic content in caches with small population. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE* (2016), IEEE, pp. 1–9.
- [45] LEE, D.; CHOI, J.; KIM, J.-H.; NOH, S. H.; MIN, S. L.; CHO, Y.; KIM, C. S. LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies. *IEEE transactions on Computers* 50, 12 (2001), 1352–1361.
- [46] LI, X.; WANG, X.; LI, K.; LEUNG, V. C. CaaS: Caching as a service for 5G networks. *IEEE Access* 5 (2017), 5982–5993.
- [47] LI, X.; WANG, X.; ZHU, C.; CAI, W.; LEUNG, V. C. Caching-as-a-service: Virtual caching framework in the cloud-based mobile networks. In *INFOCOM Workshops* (2015), pp. 372–377.
- [48] LIN, X.; LEE, N. *5G and Beyond Fundamentals and Standards*. Springer, 2021.
- [49] LIN, Y.; SHAO, L.; ZHU, Z.; WANG, Q.; SABHIKHI, R. K. Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development* 54, 1 (2010), 4–1.
- [50] LUIZELLI, M. C.; BAYS, L. R.; BURIOL, L. S.; BARCELLOS, M. P.; GASPARY, L. P. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (2015), IEEE, pp. 98–106.
- [51] MACH, P.; BECVAR, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1628–1656.
- [52] MAO, Y.; YOU, C.; ZHANG, J.; HUANG, K.; LETAIEF, K. B. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2322–2358.
- [53] MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G., ET AL. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [54] MOBILE, C. C-RAN: the road towards green RAN. *White paper, ver 2* (2011), 1–10.
- [55] MOGENSEN, P.; NA, W.; KOVÁCS, I. Z.; FREDERIKSEN, F.; POKHARIYAL, A.; PEDERSEN, K. I.; KOLDING, T.; HUGL, K.; KUUSELA, M. LTE capacity compared to the shannon bound. In *2007 IEEE 65th vehicular technology conference-utc2007-spring* (2007), IEEE, pp. 1234–1238.

- [56] MOON, S.; SHIN, Y.; CHUNG, S.; KIM, S. Design and analysis of virtualized caching service on cellular infrastructure. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access* (2017), ACM, pp. 45–54.
- [57] PASCHOS, G.; BASTUG, E.; LAND, I.; CAIRE, G.; DEBBAH, M. Wireless caching: Technical misconceptions and business barriers. *IEEE Communications Magazine* 54, 8 (2016), 16–22.
- [58] PENTTINEN, J. T. *The LTE/SAE deployment handbook*. John Wiley & Sons, 2011.
- [59] QIAO, G.; LENG, S.; MAHARJAN, S.; ZHANG, Y.; ANSARI, N. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks. *IEEE Internet of Things Journal* 7, 1 (2019), 247–257.
- [60] QIU, L.; CAO, G. Popularity-aware caching increases the capacity of wireless networks. *IEEE Transactions on Mobile Computing* 19, 1 (2019), 173–187.
- [61] RATCLIFF, R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review* 97, 2 (1990), 285.
- [62] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.
- [63] SADEGHI, A.; SHEIKHOLESAMI, F.; GIANNAKIS, G. B. Optimal and scalable caching for 5G using reinforcement learning of space-time popularities. *IEEE Journal of Selected Topics in Signal Processing* 12, 1 (2017), 180–190.
- [64] SATYANARAYANAN, M.; BAHL, P.; CACERES, R.; DAVIES, N. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing* 8, 4 (2009), 14–23.
- [65] SAZLI, M. H. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 50, 01 (2006).
- [66] SESIA, S.; TOUFIK, I.; BAKER, M. *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.
- [67] SOARES, C. L.; MORAES, I. M. Sobre a influência dos nós egoístas no descarregamento de tráfego com redes oportunistas. In *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais* (2016), SBSeg.
- [68] SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.
- [69] UNWIREDLABS. Opencellid project. <https://opencellid.org/>.
- [70] VAN HASSELT, H.; GUEZ, A.; SILVER, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (2016), vol. 30.
- [71] WANG, Q.; MA, Y.; ZHAO, K.; TIAN, Y. A comprehensive survey of loss functions in machine learning. *Annals of Data Science* 9, 2 (2022), 187–212.

- [72] WANG, X.; HAN, Y.; WANG, C.; ZHAO, Q.; CHEN, X.; CHEN, M. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network* 33, 5 (2019), 156–165.
- [73] WANG, X.; WANG, C.; LI, X.; LEUNG, V. C.; TALEB, T. Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching. *IEEE Internet of Things Journal* 7, 10 (2020), 9441–9455.
- [74] WATKINS, C. J.; DAYAN, P. Q-learning. *Machine learning* 8, 3 (1992), 279–292.
- [75] WAXMAN, B. M. Routing of multipoint connections. *IEEE journal on selected areas in communications* 6, 9 (1988), 1617–1622.
- [76] WHAIDUZZAMAN, M.; GANI, A.; ANUAR, N. B.; SHIRAZ, M.; HAQUE, M. N.; HAQUE, I. T. Cloud service selection using multicriteria decision analysis. *The Scientific World Journal* 2014 (2014).
- [77] XU, J.; FORTES, J. A. Multi-objective virtual machine placement in virtualized data center environments. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing* (2010), IEEE, pp. 179–188.
- [78] YALA, L.; FRANGOUDIS, P. A.; KSENTINI, A. Latency and availability driven VNF placement in a MEC-NFV environment. In *2018 IEEE Global Communications Conference (GLOBECOM)* (2018), IEEE, pp. 1–7.
- [79] YANG, J.; YANG, B.; CHEN, S.; ZHANG, Y.; ZHANG, Y.; HANZO, L. Dynamic resource allocation for streaming scalable videos in SDN-aided dense small-cell networks. *IEEE Transactions on Communications* 67, 3 (2018), 2114–2129.
- [80] ZHANG, H.; CHU, X.; GUO, W.; WANG, S. Coexistence of wi-fi and heterogeneous small cell networks sharing unlicensed spectrum. *IEEE Communications Magazine* 53, 3 (2015), 158–164.
- [81] ZHANG, L.; ZHAO, L.; WANG, Z.; LIU, J. Wifi networks in metropolises: From access point and user perspectives. *IEEE Communications Magazine* 55, 5 (2017), 42–48.
- [82] ZHONG, C.; GURSOY, M. C.; VELIPASALAR, S. A deep reinforcement learning-based framework for content caching. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)* (2018), IEEE, pp. 1–6.
- [83] ZHONG, C.; GURSOY, M. C.; VELIPASALAR, S. Deep reinforcement learning-based edge caching in wireless networks. *IEEE Transactions on Cognitive Communications and Networking* 6, 1 (2020), 48–61.
- [84] ZHOU, S.; ZHAO, M.; XU, X.; WANG, J.; YAO, Y. Distributed wireless communication system: a new architecture for future public wireless access. *IEEE Communications Magazine* 41, 3 (2003), 108–113.