CENTRO UNIVERSITÁRIO FEI
CRISTIANO LOPES MOREIRA

MÉTODO DE OTIMIZAÇÃO DA ALOCAÇÃO DE REDES VIRTUAIS NA ESTRUTURA FÍSICA DE UMA REDE SUBSTRATO UTILIZANDO APRENDIZADO POR REFORÇO

São Bernardo do Campo
2021

CRISTIANO LOPES MOREIRA

MÉTODO DE OTIMIZAÇÃO DA ALOCAÇÃO DE REDES VIRTUAIS NA ESTRUTURA FÍSICA DE UMA REDE SUBSTRATO UTILIZANDO APRENDIZADO POR REFORÇO

Dissertação de Mestrado apresentada ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Reinaldo Augusto da Costa Bianchi.

São Bernardo do Campo

2021

centro universitário FEI

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA EXAMINADORA

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

Mestrado

PGE-10

Aluno: Cristiano Lopes Moreira **Matrícula:** 119103-0

Título do Trabalho: MÉTODO DE OTIMIZAÇÃO DA ALOCAÇÃO DE REDES VIRTUAIS NA ESTRUTURA FÍSICA DE UMA REDE SUBSTRATO UTILIZANDO APRENDIZADO POR REFORÇO..

Área de Concentração: Inteligência Artificial Aplicada à Automação e Robótica

Orientador: Prof.º Dr.º Reinaldo Augusto da Costa Bianchi

Data da realização da defesa: 03/08/2021

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

A banca foi realizada no dia 03 de agosto de 2021 às 09:00 horas, e se iniciou com a apresentação do aluno, que foi muito boa, e seguiu para a arguição, onde o aluno respondeu a todas as questões de forma adequada demonstrando conhecimento pleno do tema. Foram sugeridas melhorias em relação ao texto para a versão final. A aprovação foi por unanimidade.

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA	
Prof.º Dr.º Reinaldo Augusto da Costa Bianchi	Ass.: _____
Prof.º Dr.º Rodrigo Izidoro Tinini	Ass.: _____
Prof.º Dr.º Carlos Alberto Kamienski	Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO ☒ REPROVADO ☐

VERSÃO FINAL DA DISSERTAÇÃO

APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

Lopes Moreira, Cristiano.
Método de Otimização da Alocação de Redes Virtuais na Estrutura Física de uma Rede Substrato Utilizando Aprendizado por Reforço / Cristiano Lopes Moreira. SÃO BERNARDO, 2021.
142 f. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi.

1. Aprendizado por reforço. 2. redes de quinta geração (5G). 3.Fatiamento de Rede. 4. Alocação de Redes Virtuais (VNE) em rede heterogênea. 5. Computação de Borda. I. Augusto da Costa Bianchi, Reinaldo, orient. II.Título.

Dedico este trabalho à Ester de Souza Lopes,
onde quer que ela esteja

AGRADECIMENTOS

Agradeço primeiramente à minha esposa, por toda a ajuda e incentivo durante toda essa jornada. Agradeço, também, aos amigos que, de uma forma ou de outra, me deram forças para continuar seguindo meu caminho.

Agradeço aos colegas e amigos do mestrado, em especial àqueles que passaram comigo algumas horas fazendo os diversos trabalhos deste curso. Nossa convivência tornou essa jornada muito mais agradável, e a troca de experiências foi fundamental para o desenvolvimento deste trabalho.

Agradeço aos professores da FEI com quem tive o privilégio de aprender – em especial aos professores Reinaldo Bianchi, Flavio Tonidandel e Raul Cesar Gouveia Fernandes. A orientação e o apoio que recebi foram imprescindíveis para o meu crescimento pessoal, minha formação e, além disso, para garantir a qualidade deste trabalho.

“A complexidade, como a beleza, é difícil de definir, porém é fácil de identificar. Na área dos sistemas complexos há interessantes encontros entre as ciências aplicadas – engenharias, medicina, economia – e a física estatística.”

Constantino Tallis

RESUMO

As tecnologias 5G possibilitaram novas aplicações em uma infraestrutura de computação de borda que visa atender uma demanda heterogênea e distribuída que unifica hardware, rede e software voltados para habilitação digital. Baseada nos requisitos da Indústria 4.0, esta infraestrutura habilitadora introduz o conceito de fatiamento da rede (*Network Slicing*), um recurso fundamental que transforma a rede de um paradigma estático em um novo paradigma onde as redes são lógicas, utilizando o modelo de compartilhamento de computação em nuvem e névoa, que deve atender às necessidades de acordos de nível de serviço de forma conveniente e otimizada, exigindo um mecanismo de orquestração para a alocação dinâmica de recursos. Entre esses mecanismos, a incorporação de redes virtuais (VNE) e o gerenciamento dinâmico de recursos (DRM) têm mostrado uma maneira de definir onde e como a tecnologia de computação em névoa deve ser usada. Este trabalho propõe um algoritmo de alocação de recursos, o VNE_CRS, que utiliza uma técnica de inteligência artificial chamada aprendizado por reforço para orquestrar múltiplos domínios da infraestrutura de uma rede 5G, beneficiando-se de sua característica de considerar todo o problema, fim a fim, utilizando diferentes aspectos do Indice de Qualidade de Serviço 5G (5QIs). Experimentos foram realizados em simulação comparando o VNE_CRS com algoritmos do estado da arte para alocação VNE em ambiente Edge de múltiplos domínios. Os resultados mostraram que o uso de técnicas de aprendizado por reforço para alocação de recursos de VNE apresentou ganhos de desempenho. Ele pode não apenas simplificar a arquitetura VNE, mas também atuar como um sistema de orquestração completo que visa os resultados estratégicos de longo prazo.

Palavras-chave: Aprendizado por Reforço, Redes de Quinta Geração (5G), Fatiamento de Rede (*Network Slicing*), Alocação Dinâmica de Recursos em Rede Heterogênea, Virtualização de Redes, Alocação de Redes Virtuais (VNE – *Virtual Network Embedding*) em rede Heterogênea, Computação na Borda (*Edge Computing*), Computação em Névoa (*Fog Computing*).

ABSTRACT

5G technologies have enabled new applications on a heterogeneous and distributed Edge infrastructure which unifies hardware, network and software aimed at digital enabling. Based on requirements of Industry 4.0, this infrastructure introduces the concept of network slicing, a fundamental resource that transforms the network from a static paradigm into a new paradigm where networks are logical, developed using Cloud and fog computing sharing model, which should meet the needs of service level agreements in a convenient and optimized way, requiring an orchestration mechanism for the dynamic resource allocation. Among these mechanisms, virtual networks embedding (VNE) and dynamic resource management (DRM) have shown a way to define where and how Edge technology should be used. This paper proposes a resource allocation algorithm, VNE_CRS, which uses an artificial intelligence technique called reinforcement learning to orchestrate multiple domains, benefiting from its characteristic of considering the whole problem, end-to-end, using different aspects of 5G Quality of Service Indicator (5QIs). Experiments were carried out in simulation comparing VNE_CRS with state-of-the-art algorithms for multi domains Edge environment. Results have shown that the usage of reinforcement learning techniques to VNE resource allocation has shown performance gains. It can not only simplify the VNE architecture but also act as a full orchestration system that aims at the strategic long run results of whole infrastructure usage.

Keywords: Reinforcement learning, fifth generation networks (5G), Network Slicing, Dynamic Resource Allocation in Heterogeneous Network, Network Virtualization, Virtual Network Embedding (VNE), Edge Computing, Fog Computing.

LISTA DE ABREVIAÇÕES

1G	Primeira geração de sistemas de telefonia celular
2G	Segunda geração de sistemas de telefonia celular
3G	Terceira geração de sistemas de telefonia celular
3GPP	Projeto de Parceria de Terceira Geração – *3rd Generation Partnership Project*
4G	Quarta geração de sistemas de telefonia celular
5G	Quinta geração de sistemas de telefonia celular
5GNR	Rede de Acesso em Rádio 5G – *5G new radio*
5QI	Indicador QoS 5G
AM	Modulação por Amplitude – *Amplitude Modulation*
AMPS	Sistema avançado de telefonia móvel – *Advanced Mobile Phone System*
AP	Ponto de Acesso – *Access Point*
APIs	Interface de Programação de Aplicações – *Application Programming Interface*
APSK/APK	Modulação de chaveamento de fase e amplitude – *Amplitude Phase Keying*
AR	Realidade Aumentada – *Augmented Reality*
ARPU	Receita média por usuário – *Average Revenue Per User*
BSD	Licença de Uso de Programa pela Distribuição de Berkeley – *Berkeley Software Distribution*
BW	Largura de banda – *Bandwidth*
CN	Rede Núcleo – *Core Network*
CPU	Unidade de processamento Computacional – *Computer Processing Unit*
CSC	Controlador de Fatiamento da Rede Núcleo – *Core Slice Controllers*
CSP	Operadoras de Telecomunicações – *Communications Service Providers*
DPI	Inspeção e Classificação do Tipo do Tráfego – *Deep Packet Inspection*
DRM	Gerenciamento Dinâmico de Recursos de Rede – *Dynamic Resource Managment*
EDGE	Taxas de avançadas de transferência de dados GSM – *Enhanced Date Rates For GSM Evolution*
FCC	Comissão federal de comunicações (norte americano) – *Federal Communications Commission*
FM	Modulação por frequência – *Frequency Modulation*
GGSN	Equipamento de Ponto de Transferência e Suporte da Rede de Dados Móveis – *Gateway GPRS Support Node*
GPRS	Serviço de pacotes de dados da rede celular – *General Packet Radio Service*
GPU	Unidades de Processamento Gráfico – *Graphics Processing Unit*
GSM	Sistema Global para Comunicações Móveis – *Global System for Mobile Communication*
GSMA	Associação GSM
HetNet	Rede heterogênea – *Heteregeneous Network*
HLR	Base de Dados de Registro e Localização do Usuário – *Home Location Register*
HSPA	Rede de acesso de dados em alta velocidade – *High Speed Packet Access*
IA	Inteligência Artificial
IaaS	Infraestrutura Como Serviço – *Infrastructure as a Service*
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos – *Institute of Electrical and Electronic Engineers*

InPs	Provedores de Infraestrutura em Nuvem – *Infrastructure Providers*
IoT	Internet das coisas – *Internet of Things*
ISPs	Provedores de Serviço de Internet – *Internet Service Providers*
ITU	União Internacional de Telecomunicações
LAN	Conexões das Redes Internas – *Local Area Network*
LoRa	Rede de área ampla de baixa potência
LTE	Rede 4G de evolução de longo prazo – *Long Term Evolution*
M2M	Máquina-a-Máquina – *Machine to Machine*
Mbps	Megabits por segundo
MDP	Processos de Decisão Markoviano – *Markov Decision Process*
MEC	Multiacesso de computação de borda – *Multi-access Edge Computing*
MIP	Programação Inteira Mista – *Mixed-Integer Programming*
MLP	Rede Neural Artificial do tipo Perceptron Multicamada – *Multi Layer Perseptron*
MPI	Interface de Transferência de Mensagens – *Message Passing Interface*
MPLS	Comutação de Rótulos Multiprotocolo – *Multiprotocol layer switching*
ms	milissegundos
MSC	Central de Comutação de Telefonia Móvel – *Mobile Switching Center*
NaaS	Serviço de Infraestrutura de Rede Sob Demandada – *Network as a Service*
NF	Funções de Rede – *Network Functions*
NFV	Funções de Rede Virtualizada – *Network Function Virtualization*
NFVI	Infraestrutura de Virtualização de Função Rede – *Network Function Virtualization Infrastructure*
NIST	Instituto Nacional de Padrões e Tecnologia dos EUA – *National Institute of Standards and Technology*
NRM	Modelagem dos Recursos da Rede 5G – *Network Resource Model*
NS	Fatiamento de Rede – *Network Slicing*
NSaaS	Redes com Compartilhamento de Toda a Infraestrutura da Rede Como Serviço – *Network Slice as a Service*
NSC	Consumidor de Serviço de Fatiamento de Rede – *Network Slice Customer*
NSP	Provedor do Serviço de Fatiamento de Rede – *Network Slice Provider*
NV	Realmente Virtualizada – *Network Virtualization*
OFDM	Modulação por de divisão ortogonal de frequência – *Orthogonal Frequency Division Multiplexing*
OTT	Plataformas de Distribuição de Conteúdo Sobre a Infraestrutura da Rede – *Over The Top*
PaaS	Plataforma Como Serviço – *Platform as a Service*
PCM	Modulação por pulso codificado – *Pulse Code Modulation*
PDN	Rede Pública de Dados – *Public Data Network*
PM	Modulação por fase – *Phase Modulation*
PNF	Função de Rede Física – *Physical Network Function*
PSK	Modulação por chaveamento de fase – *Phase Shift Keying Modulation*
PSTN	Rede Pública de Telefonia – *Public Switched Telephone Network*
QAM	Modulação por amplitude em quadratura – *Quadrature Amplitude Modulation*
QFI	Identificação de QoS do Fluxo Fim a Fim – *QoS Flow ID*
QoE	Qualidade da experiência do usuário – *Quality of Experience*
QoS	Qualidade de serviço – *Quality of Service*

QoSE	Qualidade Experimentada e Percebida Pelo Usuário – *QoS experienced/perceived by customer/user*
RAN	Rede de Rádios de Acesso – *Radio Access Network*
RL	Aprendizado por Reforço – *Reinforcement Learning*
RNA	Rede Neural Artificial
RSC	Controlador de Fatiamento de Rádio – *Radio Slice Controllers*
RTT	Relatório de Tempo de Ida e Volta do Pacote Transmitido – *Round Trip Time*
SaaS	Programas de Computador Como Serviço – *Software as a Service*
SD-WAN	Rede WAN programável – *Software Defined Wide Area Network*
Sifgox	Operadora de rede que constrói redes sem fio para conectar objetos
SLA	Acordo de Nível de Serviço – *Service Level Agreement*
SMS	Serviço de mensagens de texto – *Short Message Service*
STA	Estação – *Station*
TCP	Protocolo de Controle de Transmissão da Camada de Transporte na Rede de Computadores – *Transmission Control Protocol*
TD	Aprendizado da Diferença Temporal – *Temporal-Difference Learning*
TDMA	Acesso por divisão no tempo – *Time-division multiple access*
TN	Rede Transporte – *Transport Network*
TSC	Controlador de Fatiamento da Rede de Transporte – *Transport Slice Controllers*
UE	Equipamento do Usuário – *User Equipment*
UMTS	Sistema de Telefonia Móvel Universal – *Universal Mobile Telecommunications System*
VLANs	Redes de Virtuais Locais – *Virtual Local Area Networks*
VM	Máquinas Virtuais – *Virtual Machine*
VNE	Incorporação de Redes Virtuais – *Virtual Network Embedding*
VNE_CRS	VNE por Estratégia de Custo de Recursos – VNE *Cost Resource System*
VNF	Virtualização das Funções de Rede dos Elementos – *Virtualized Network Function*
VNR	Requisição de Rede Virtual – *Virtual Network Request*
Volte	Voz sobre LTE – *Voice over LTE*
VPNs	Redes Privadas Virtuais – *Virtual Private Networks*
VR	Realidade Virtual – *Virtual Reality*
VRFs	Encaminhamento Virtual de Rede – V*irtual Routing and Forwarding*
WAN	Conexão até as Redes Externas – *Wide Area Network*
WCDMA	Acesso múltiplo por divisão de código – *Wideband Code Division Multiple Access*
Wi-Fi	Rede sem vio amigável – *Wireless Fidelity*
a	Ação
l^r_{BW}	Unidade de receita de BW – *BW Revenue Unit*
l^s_a	Capacidade Disponível no Enlace
n^s_a	Capacidade Disponível no Nó
n^r_{cpu}	Unidade de receita de CPU – *CPU Revenue Unit*
st	Estado do Ambiente – *Enviroment State*
Γ	Fator de Desconto

T	Função de Transferência
U^{π}	Função de Utilidade do Estado ou Função Valor
π	Política de ações
r	Recompensa
G^{r}	Conjunto de requerimentos de rede – *Set of Network Requeriment*
R	Conjunto de recompensas
S	Conjunto de Estados do Ambiente – *Set of States*
L^{S}	Conjuntos de Enlaces de Substrato – *Set of Substrate Links*
N^{S}	Conjuntos de Nós de Substrato – *Set of Substrate Node*
P^{s}	Conjuntos de Caminhos na rede de Substrato – *Set of Substrate Paths*
L^{v}	Conjuntos de Enlaces Virtuais – *Set of Virtual Links Request*
N^{v}	Conjuntos de Nós Virtuais – *Set of Virtual Node Request*
l^{s}_{c}	Unidade de custo do enlace de substrato – *Substrate Link Cost Unit*
l^{s}	Enlace de Substrato – *Substrate Links*
G^{S}	Rede de Substrato – *Substrate Network*
n^{s}	Nó de Substrato – *Substrate Node*
n^{s}_{c}	Unidade de custo do nó de substrato – *Substrate Node Cost Unit*
p^{s}	Caminho na rede de Substrato – *Substrate Path*
α	Taxa de aprendizado
l^{s}_{r}	Requerimento de Enlace Virtual – *Virtual Link Request*
l^{v}	Enlace Virtual – *Virtual Links*
G^{V}	Rede Virtual – *Virtual Network*
n^{v}	Nó Virtual – *Virtual Node*
n^{v}_{r}	Requerimento de nó Virtual *Virtual Node Request*
μ^{s}_{r}	Mapeamento de VNR na G^{S} – *VNR Mapping on Substrate Network*

LISTA DE ILUSTRAÇÕES

Figura 1 – Hierarquia das necessidades de Maslow x de Comunicação 24
Figura 2 – *Making 5G NR a reality leading the technology innovations* 26
Figura 3 – Modulações em quadratura 27
Figura 4 – Evolução do ARPU no Brasil Statista (2020) 28
Figura 5 – Contribuição esquemática para QoS fim a fim 29
Figura 6 – Quatro pontos de vista da qualidade de serviços 30
Figura 7 – Esquemático do teste de velocidade de *upload* e *download* 33
Figura 8 – Resultados de testes velocidade x distância dos servidores 35
Figura 9 – Necessidade de rede de dados por aplicação 37
Figura 10 – Arquitetura de computação em névoa e seus atributos 41
Figura 11 – Arquitetura fim a fim *Network Slicing* 43
Figura 12 – Estrutura NFV 45
Figura 13 – Estrutura SDN 47
Figura 14 – Grid de recursos UE/M2M para *uplink* e *downlink* 48
Figura 15 – Modelo básico de redes virtuais (VN) 53
Figura 16 – Exemplo esquemático de RL 58
Figura 17 – Ranqueamento do espectro de backups do TD 67
Figura 18 – Função de Generalização por ANN 69
Figura 19 – *Perceptron* 71
Figura 20 – *Multi Layer Perceptron* – MLP 72
Figura 21 – Sistema RL Multiagente DRM para Cooperação com a VNE 76
Figura 22 – Função de aproximação por uma rede neural. 79
Figura 23 – Arquitetura Orquestrador VNE_CRS 84
Figura 24 – Alg. 08, 09 e 10 na arquitetura do orquestrador VNE_CRS 90
Figura 25 – Alg. 11 na arquitetura do orquestrador VNE_CRS 94
Figura 26 – Alg. 12 na arquitetura do orquestrador VNE_CRS 96
Figura 27 – Servidor Rack PowerEdge R640 99
Figura 28 – Servidor Rack E9000H-4 99
Figura 29 – Transmissor óptico – *Gbic* 100
Figura 30 – Topologia da Rede 100
Figura 31 – Avaliação do problema de alocação de recursos e comparação dos cenários 105
Figura 32 – Distribuição de carga abordagem CSP convencional 105
Figura 33 – Distribuição de carga abordagem CSP + Nuvem Pública 105
Figura 34 – Distribuição de carga abordagem CSP + Nuvem Privada 106
Figura 35 – Distribuição de carga abordagem CSP Edge Base 106
Figura 36 – Taxa de Aceitabilidade na subida do tráfego – 2 minutos (treinamento) 109
Figura 37 – Taxa de Aceitabilidade na subida do tráfego – 5 minutos (treinamento) 110
Figura 38 – Resumo estatístico da taxa de aceitabilidade – 5 minutos (treinamento) 110
Figura 39 – Taxa de Aceitabilidade em carga de tráfego + 2 minutos (treinamento) 112
Figura 40 – Resumo estatístico da taxa de aceitabilidade – 5 minutos (treinamento) 112
Figura 41 – Comparativo da Taxa de Aceitabilidade de VNR por estratégia VNE 113
Figura 42 – Distribuição Carga CPU GCR 117
Figura 43 – Distribuição de carga CPU Hlog 117
Figura 44 – Comparativo de receita, conexões e resultado financeiro 120

LISTA DE TABELAS

Tabela 1 – Modulações 27
Tabela 2 – Evolução dos serviços em cada geração da telefonia móvel 28
Tabela 3 – Atributos do Fatiamento de Rede 44
Tabela 4 – Estado da Arte VNE 73
Tabela 5 – Matriz de transição de Markov de Cheng et al. 74
Tabela 6 – Estratificação de Estados de N e L 86
Tabela 7 – Ações de Estratégia de Custo 89
Tabela 8 – QoS x Slices 97
Tabela 9 – Relação de Velocidade x Latência x quadro 98
Tabela 10 – Treinamento 2 Minutos para definição de estratégia 108
Tabela 11 – Resultados dos algoritmos de VNE – 15 minutos de requisições VNR 114
Tabela 12 – Utilização de CPU nos nós primários pelas abordagens de alocação 118
Tabela 13 – Utilização de CPU nos nós secundários pelas abordagens de alocação 120
Tabela 14 – Descrição dos principais atributos para SLA da NS 140
Tabela 15 – Padronização do mapeamento de 5QI para características de QoS 141

SUMÁRIO

1 INTRODUÇÃO 17
1.1 DECLARAÇÃO DO PROBLEMA 19
1.2 HIPÓTESE E PERGUNTA DE PESQUISA 20
1.3 JUSTIFICATIVA 21
1.4 OBJETIVO 22
1.5 ORGANIZAÇÃO DESTE TRABALHO 22
2 REVISÃO BIBLIOGRÁFICA 23
2.1 REDES DE TELECOMUNICAÇÕES, UM BREVE HISTÓRICO 23
2.2 QUALIDADE NAS REDES DE TELECOMUNICAÇÕES 29
2.2.1 Qualidade de uma rede de transmissão de dados 32
2.2.2 Primeiras avaliações de qualidade em uma rede 5G 36
2.3 A NUVEM: *CLOUD COMPUTING* E *CLOUD NETWORKING* 39
2.3.1 Computação na borda e em névoa 40
2.3.2 A nuvem das redes (*Cloud Networking*) 42
2.3.3 Domínio da rede núcleo – NFV 45
2.3.4 Domínio da rede de transporte – SDN 46
2.3.5 Domínio da rede de acesso – RAN 47
2.4 ALOCAÇÃO DE RECURSOS DE REDE 49
2.4.1 Classificações das VNEs 50
2.4.2 Modelagem da VNE 52
2.5 APRENDIZADO POR REFORÇO 57
2.5.1 Conceitos do aprendizado por reforço 59
2.5.2 Métodos de otimização para problemas sequenciais decisórios 61
2.5.2.1 Programação Dinâmica 61
2.5.2.2 Método de Monte Carlo 64
2.5.2.3 Método das diferenças Temporais – TD-Learning 65
2.5.2.4 Q-Learning 68
2.6 REDES NEURAIS ARTIFICIAIS 70

3 TRABALHOS RELACIONADOS – O ESTADO DA ARTE 73
4 VNE_CRS - ORQUESTRAÇÃO POR APRENDIZADO POR REFORÇO 83
4.1 MODELAGEM E ABSTRAÇÕES DA VNE 83
4.2 O ALGORITMO DE ORQUESTRAÇÃO – VNE_CRS 89
5 METODOLOGIA DE EXPERIMENTAÇÃO 97
5.1 BASE DE DADOS 97
5.2 SIMULAÇÃO 99
5.3 ATIVIDADES, INPLEMENTAÇÕES E EXPERIMENTOS REALIZADOS 101
5.3.1 Implementação do método heurístico VNE não coordenado e VNE_GRC 101
5.3.1.1 Tratamento da base de dados sndlib.zib.de 101
5.3.1.2 Implementação da VNE não coordenado e VNE GRC 102
5.3.1.3 Simulação VNE convencional x VNE CGR 102
5.3.2 Desenvolvimento e implementação e simulação do VNE_CRS 102
5.4 CRITÉRIOS DE AVALIAÇÃO DOS RESULTADOS 103
6 RESULTADOS E CONSIDERAÇÕES 104
6.1 ABORDAGENS DE ALOCAÇÃO DE RECURSOS VNE 104
6.2 AVALIAÇÃO DOS RESULTADOS DO VNE_CRS NA ALOCAÇÃO VNE 108
6.2.1 Abordagem de treinamento VNE_CRS 108
6.2.2 Validação VNE_CRS e comparação de resultados 113
7 CONSIDERAÇÕES FINAIS 124
7.1 RESUMO DAS CONTRIBUIIÇÕES 125
7.1.1 MODELAGEM DOS DOMÍNIOS DE REDE PARA USO EM RL 126
7.1.2 OTIMIZAÇÃO DOS RECURSOS DA INFRAESTRTURA 127
7.1.3 REAPROVEITAMENTO DOS ALGORITMOS ESPECIALIZADOS 128
7.1.4 UTILIZAÇÃO DOS CRITÉRIOS DE 5QI – PARA REDES 5G 128
7.2 TRABALHOS FUTUROS 129
REFERÊNCIAS 130
APÊNDICE A – RECOMENDAÇÕES ITU E. 800 136
APÊNDICE B – TABELA DE DESCRIÇAO DE ATRIBUTOS PARA SLA 140
APÊNDICE C – TABELA DE PADRONIZAÇÃO 5QI 141

1 INTRODUÇÃO

Saracco, Harrow e Weihmayer 1997 (p. xxi) mostravam “como as telecomunicações estão se expandindo além do telefone para se tornarem apenas um recurso ‘invisível’ de um amplo espectro de pacotes de serviços.”

Nos anos 1990, o Sistema Global para Comunicações Móveis (GSM – *Global System for Mobile Communication*), a segunda geração de sistemas de telefonia celular – 2G, ampliou a cobertura geográfica que forneceu serviços fundamentais de comunicação que, na pirâmide de Maslow (1943), refletem as necessidades humanas da base da pirâmide, as necessidades básicas, comunicação necessária, e as necessidades de segurança, comunicação comum; avançando com o atendimento das necessidades sociais, de consumo de informação, através da evolução tecnológica para a terceira e quarta geração de sistemas de telefonia celular – 3G e 4G (Li, 2018).

Pela proposta de Saracco, Harrow e Weihmayer (1997), quando as necessidades básicas são alcançadas, assim como na indústria do asfalto ou qualquer serviço ou produto de outra indústria de infraestrutura essencial que alcançou e ultrapassou a demanda das necessidades básicas; essa indústria se torna “invisível”, uma *commodities* de mercado, ou seja, após a era do GSM o serviço de telecomunicação começou a ser visto apenas como um *commodities* de mercado, com muita demanda de investimento porém sem aumento de percepção de valor agregado por parte dos usuários, mesmo com o aumento exponencial da utilização das redes de telefonia móvel.

A confirmação da hipótese de Saracco, Harrow e Weihmayer (1997) pode ser observada pelo crescimento exponencial da demanda do uso de serviços de dados entre 2008 e 2019, projeção 2020, (PATRIK, 2019) e com a constatação no relatório, em 2017, da consultoria Stragy (2017) de que, apesar do aumento da utilização das redes móveis, e seus constantes investimentos, não está ocorrendo, por parte dos usuários, um aumento da percepção de valor agregado e consequentemente as operadoras estão perdendo sua rentabilidade, observada pela queda da receita média por usuário (ARPU – *Average Revenue Per User*).

A era da extensão sensorial e autoliberação trouxe novas aplicações que frearam a tendência de *comoditização* e transformaram a rede de telecomunicações em uma grande infraestrutura habilitadora para: serviços baseados em nuvem e computação de borda (NUNNA et all., 2015), comunicação veicular (RAISSI e YANGUI, 2019), serviços comunitários, telemedicina e atendimento remoto de emergências (LACY et all.,2019), monitoramento ambiental (SIMSEK et all.,2016), robôs e *drones* de entrega (HAYAT et all.,2021).

Na medida em que os dispositivos móveis com várias interfaces de rede se tornam comuns, os usuários demandam um serviço de comunicação e de infraestrutura computacional com parâmetros de qualidade distintos das métricas utilizadas até então, independentemente da localização ou tipo de acesso à rede. Há uma diversidade de aplicações que desafiam as futuras redes, para pessoas e para coisas, em termos de altíssima densidade de volume de tráfego e altíssima densidade de conexão (OSSEIRAN, MONSERRAT, MARSCH, 2016).

Como apontado por Osseiran, Monserrat e Marsch (2016), essas demandas de novas aplicações e de uma rede e infraestrutura flexível, heterogênea, com foco na qualidade técnica e percebida, culminaram no desenvolvimento da nuvem (*cloud*) das redes, iniciada nas conectividades fixas, SDN, e seguidas nas redes de quinta geração 5G implementadas entre os anos de 2019 e 2021.

O 5G propõe ser uma rede única: descentralizada e distribuída geograficamente (computação de borda – *edge computer/edge networking*, MEC - *Multi-access Edge Computing*); heterogênea (*HetNet – Heteregeneous Network*), interconectando usuários e coisas, por ethernet, fibra ótica, MPLS, SD-WAN, wifi, sifgox, lora, 2G, 3G, LTE, 5GNR, normalizadas, ou não, pelo Projeto de Parceria de Terceira Geração (3GPP – *3rd Generation Partnership Project*), que tenha uma conectividade com a capacidade de prover serviços especializados na qualidade de serviço (QoS – *Quality of Service*) e na experiência do usuário (QoE – *Quality of Experience*), determinadas por um acordo de nível de serviço (SLA – *Service Level Agreement*); vista como um serviço de infraestrutura computacional e de rede sob demandada (NaaS – *Network as a Service*); compartilhada, pública, porém segmentada na forma de fatias privativas de rede (NS – *Network Slices*), para cada necessidade, cada aplicação, cada indústria, ou ainda para cada cliente (3GPP TS 22.261, 2020).

A capacidade virtualização e fatiamento da rede 5G (*Network Slicing*) é o recurso fundamental que transforma a rede de um paradigma estático de “tamanho único” em um novo paradigma em que as redes são lógicas, gerenciadas pelos orquestradores que buscam resolver o problema de incorporação de redes virtuais (VNE – *Virtual Network Embedding*) com privacidade (segurança e isolamento), com atendimento de requisitos de SLA diferenciados pelo QoE e QoS e de forma a otimizar a alocação dos recursos computacionais e de rede observados os critérios de latência, resiliência, densidade ou velocidade incorporados em redes virtuais, conforme descrito na especificação 3GPP TR 28.801(2018).

1.1 DECLARAÇÃO DO PROBLEMA

O problema de alocação de recursos tratado pelo VNE consiste em dispor recursos de uma rede física, chamada de rede substrato (física), para atender uma requisição de rede virtual (VNR – *Virtual Network Request*), que é composta por um conjunto de nós virtuais (VM – máquinas virtuais – poder computacional) e um conjunto de enlaces virtuais (*links* virtuais) que interconectam esses nós virtuais (ROBERTAZZI E SHI, 2020).

Já o problema da incorporação de VNRs heterogêneas é alocar o máximo possível de solicitações com tamanhos pacotes e características de SLA distintas, dentro de uma infraestrutura multidimensional, dividida em rede de transporte, rede de acesso, rede core e computação de borda distribuída.

Esse problema, em casos de redes pequenas, pode ser tratado como um problema de Programação Inteira Mista (MIP – *Mixed-Integer Programming*) usando métodos convencionais com algoritmos gulosos para encontrar caminhos com um único critério de otimização, que buscam minimizar a distância ou maximizar a largura de banda, como Dijkstra e Bellman-Ford, que se são eficientes para o paradigma de redes de perfil estático e parâmetros determinísticos.

Em redes grandes – com o perfil de tráfego proposto pelas redes 5G, que além de heterogêneo também atende as aplicações sensíveis à latência e tem o processamento das aplicações de usuário e de rede descentralizado e distribuídos geograficamente, (*edge computer/edge networking*), o que aumenta o tamanho da rede, sua complexidade e quantidade de nós e enlaces – a MIP torna-se um problema *NP-hard* (de complexidade computacional com tempo polinomial não determinístico) ineficiente na otimização da alocação de recursos da rede substrato como já amplamente estudado em Cheng et al.(2011), Chowdhury et al.(2012), Soualah et al. (2014), JMILA (2015), Haeri e Trajkovic (2018), Sciancalepore et al. (2018), Shahriar et al. (2018), Yao et al. (2018), Yu et al. (2018), Dehury et al. (2019), Dolati et al. (2019), Wang et al. (2019), Zheng et al. (2020), HE et al. (2020), Yaun et al. (2020), Yan et al. (2020) e (ROBERTAZZI E SHI, 2020).

O problema a ser estudado neste trabalho é alocação de recursos tratado pelo VNE em uma rede de 161 nós e 166 enlaces, considerada uma rede grande, ou seja, *NP-hard*. Por esse motivo as técnicas de inteligência artificial mostram-se com uma opção a ser estudada para encontrar um método de alocação quase ótimo.

1.2 HIPÓTESE E PERGUNTA DE PESQUISA

Nas redes virtuais e dinâmicas, um grande desafio é a utilização eficiente de recursos, em especial no caso de redes sem fio, pois a capacidade de espectro disponível é inerentemente limitada. Considerando a melhoria no desempenho e na eficiência, as técnicas de aprendizado de máquina mostraram suas vantagens na solução de problemas de rede por conter com algoritmos que podem aprender automaticamente o espaço da solução a partir de dados da própria rede (ambiente) e realizar uma nova busca de solução com base na experiência aprendida (ROBERTAZZI E SHI, 2020).

A hipótese formulada para guiar esta pesquisa de mestrado é que a utilização das técnicas de aprendizado por reforço atuando como orquestrador, aqui denominado VNE_CRS (VNE *Cost Resource System*), na alocação de recursos computacionais e de rede substrato (VNE), pela coordenação das ações dos algoritmos heurísticos convencionais, VNE não coordenado e VNE_GRC, mudando os parâmetros de custo dos recursos de rede e tendo como reforço a taxa de aceitação de requerimentos de rede virtual (VNR), observados os acordos de SLA, conseguem melhorar a eficiência do uso dos recursos da rede substrato de forma a alocar as requisições de redes virtuais heterogênea, maximizando a taxa de aceitação de requerimentos de rede virtual (VNR) e maximizando a receita pela utilização da rede de forma a otimizar a alocação de redes virtuais na estrutura da rede física.

Para tal se apresenta como pergunta de pesquisa: como tornar possível um algoritmo de aprendizado por reforço coordenar múltiplos algoritmos heurísticos, em múltiplos domínios de rede substrato, tendo como objetivo aumentar a eficiência do uso dos recursos em toda a rede?

Para responder essa pergunta, após a revisão bibliográfica realizada, foi desenvolvido uma modelagem matemática de cada domínio da rede, e seus recursos que compõem a rede substrato, em um ambiente de simulação de requerimento e alocação dos recursos de uma rede virtual e foram realizadas simulações de tipos distintos de tráfego, com alterações nas características das requisições de rede como: os quadros de dados (*frames*), a latência e a taxa de transferência de dados (*throughput*) ; sendo comparados os algoritmos tradicionais, VNE não coordenado e VNE_GRC, com e sem a orquestração utilizando a estratégia de custo de recurso de rede como mecanismo de controle.

1.3 JUSTIFICATIVA

Entre as revoluções – transformações estruturais abruptas – e as evoluções – mudanças graduais e adaptativas – aquelas que prevaleceram sempre tiveram em seu núcleo o atendimento de alguma necessidade humana.

As máquinas a vapor atenderam a necessidade básica de sobrevivência pela ampliação da produção e melhorias na logística para atender uma população com crescimento exponencial; a eletricidade atendeu as necessidades básicas de segurança com iluminação, aquecimento e melhorias nas máquinas do processo produtivo, o que atendeu a necessidade de uma energia mais limpa; e a tecnologia da informação atendeu as necessidades de pertencimento, sociais e de comunicação, o que possibilitou às pessoas aproximarem-se e se desenvolverem, como também atendeu as necessidades de automação no processo produtivo. Todas essas tecnologias somaram a característica de atender uma necessidade humana, mas também de atender a necessidade de otimização dos investimentos daqueles que as desenvolveram.

Uma nova revolução ou evolução, ainda não sabemos, se apresenta com o intuito de atender as necessidades de autoestima e de autorrealização pela extensão sensorial e autoliberação. Essa nova re/evolução propõe uma sociedade que poderá utilizar seu tempo para expandir seus conhecimentos e se libertar da necessidade de executar as tarefas essenciais para sua sobrevivência. Para isso, essa nova re/evolução introduz máquinas autogerenciáveis e completamente sensoriais ao ambiente em que as pessoas estiverem, robôs mecânicos que possam interagir com pessoas e ambientes de forma livre, segura e confiável; ela introduz a era da indústria 4.0.

Na sugestão de Li(2018), as aplicações de inteligência artificial necessitam de uma grande demanda computacional que trazem junto a necessidade espaço e energia que inviabilizam as iniciativas de sistemas autônomos móveis que possam realmente ser auto gerenciáveis e capazes de atender as necessidades de autoliberação de forma confiável e segura. Como Li (2018) explica, a única solução é retirar essa necessidade de processamento dos robôs e aplicações, transformando-os em dispositivos de sensoriamento e atuação mecânica, e levar a demanda computacional para uma infraestrutura remota que atenda às necessidades de capacidade de processamento, continuidade, velocidade e latência.

Em se tratando de uma infraestrutura, ela demanda investimentos, logo deve ser otimizada e preferencialmente compartilhada. Deve também atender as necessidades básicas da indústria 4.0, que hoje não são consideradas pelos algoritmos VNE e, portanto, esses algoritmos devem ser estudados e aprimorados.

1.4 OBJETIVO

O objetivo deste trabalho é propor um algoritmo de aprendizado por reforço que possibilite coordenar os algoritmos especializados de cada domínio, etapa, da alocação de recursos de uma rede substrato, mantendo a estrutura de algoritmos atuais, ou seja, coordenar ações que gerenciem esses algoritmos a encontrarem a melhor forma de alocar todas as subpartes das requisições de redes virtuais heterogêneas dentro da rede substrato para maximizar a taxa de aceitação de requerimentos de rede virtual (VNR) e minimizar o custo de recurso de rede, considerando os requerimentos 5QI.

As principais contribuições deste trabalho são:

- A modelagem dos domínios da rede para uso em aprendizado por reforço.
- A otimização dos recursos de infraestrutura das redes virtualizadas.
- A formalização de um algoritmo que permita os diversos domínios de uma rede virtualizada serem orquestrados com uma estratégia unificada para possibilitar o serviço de *Network Slicing* da rede 5G.
- A observação dos critérios de 5QI no processo de alocação de recursos das redes de quinta geração.

1.5 ORGANIZAÇÃO DESTE TRABALHO

Este trabalho está organizado em 7 capítulos. O primeiro, Introdução, traz um resumo sobre do tema aqui abordado; o segundo mostra uma revisão bibliográfica, que descreve a base das redes de telecomunicações, as metodologias de validação da qualidade das redes de telecomunicações, a evolução da infraestrutura de *cloud* e sua sinergia com a infraestrutura 5G, os métodos e modelagem da alocação de recursos da infraestrutura de rede por VNE, e as bases teóricas do aprendizado por reforço; o terceiro mostra trabalhos relacionados ao tema; o quarto descreve a formulação do algoritmo proposto neste trabalho, o VNE_CRS, como um método de orquestração para alocação de recursos em uma rede substrato utilizando aprendizado por reforço; o quinto apresenta a metodologia para a realização dos experimentos comparativos entre os algoritmos propostos na metodologia deste trabalho, o sexto a apresentação e discursão dos resultados e o sétimo as conclusões finais.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, em sua primeira seção, são apresentados aspectos gerais sobre o tema deste trabalho, incluindo uma introdução sobre as redes de telecomunicações, seu objetivo nas necessidades humanas, sua evolução e seus critérios de avaliação que são importantes para este trabalho, pois através dessas métricas é possível nos direcionar, sem nos deixar obscurecer pelo brilho da tecnologia, com conhecimento do que se quer otimizar, o motivo pelo qual estamos buscando otimizar e, com isto, consolidar as bases para a modelagem matemática do ambiente e suas possíveis abstrações, que é uma das partes mais complexas na tarefa de aprendizado por reforço. Na sessão seguinte, apresenta-se uma revisão bibliográfica sobre redes virtuais e o problema de alocação de recursos VNE, bem como artigos relacionados ao tema.

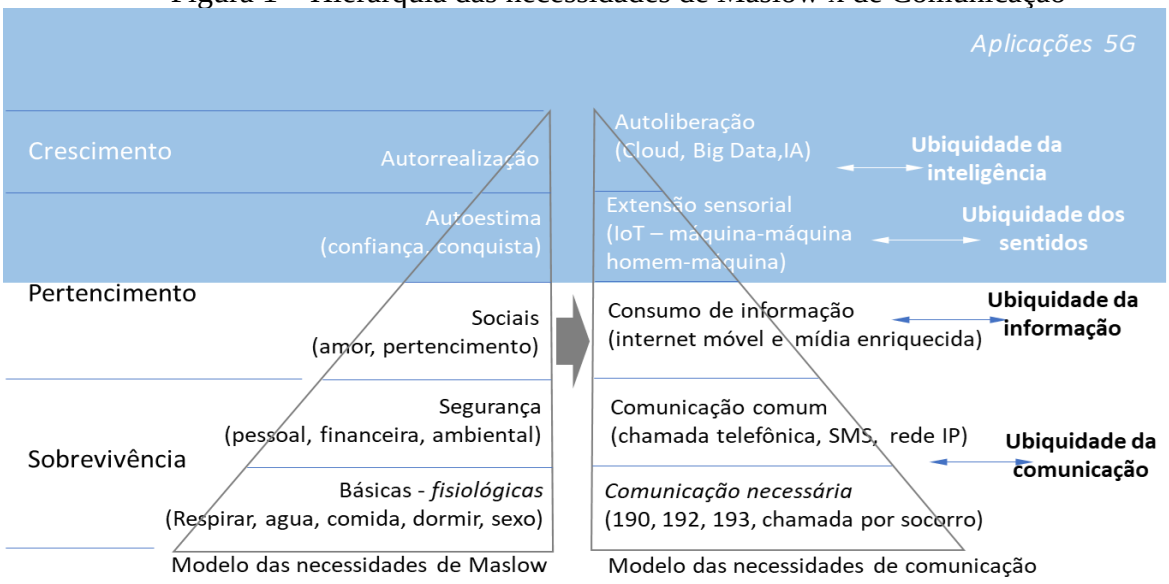
2.1 REDES DE TELECOMUNICAÇÕES, UM BREVE HISTÓRICO

Do grego “téle” à distância + “comunicação”, define-se telecomunicação como “designação genérica das comunicações a longa distância que abrange a transmissão, emissão ou recepção de sinais, sons ou mensagens por fio, rádio, eletricidade, meios ópticos ou qualquer outro processo eletromagnético” (HOUAISS, 2009). Telecomunicação é, portanto, a transmissão de informações de qualquer natureza à distância – que podem ser vídeos em alta resolução, imagens, mensagens de texto, até mesmo sinais de fumaça – por um meio portador dessa informação: cabo metálico, ondas de rádio, sistemas ópticos ou eletromagnéticos. A telecomunicação ocorre quando existe a troca de informações entre os participantes da comunicação incluindo o uso de alguma tecnologia para que ela ocorra.

Os sistemas de telecomunicações utilizando técnicas mais aprofundadas foram criados a partir da primeira metade do século XIX com o telégrafo elétrico, que permitia enviar mensagens com letras e números. Posteriormente surgiram o telefone, que veio acrescentar a possibilidade de comunicar com a própria à voz humana, as comunicações com as ondas de rádio, que fizeram uma autêntica revolução nos hábitos da humanidade, e, finalmente, as comunicações entre máquinas e entre máquinas e homens, que apontam para uma nova revolução tecnoindustrial.

A indústria de telecomunicações foi desenvolvida, inicialmente, para atender às necessidades primárias, de segurança, como uma ferramenta para pessoas e militares informarem perigo e localização em tempo real, e secundárias, sociais, por meio da aproximação de pessoas e grupos.

Figura 1 – Hierarquia das necessidades de Maslow x de Comunicação



Fonte: Autor “adaptado de” Li, 2018, p. 18.

Zhengmao LI (2018), vice-presidente executivo de pesquisa e desenvolvimento na China Mobile (desde maio 2008) mostra que a indústria de telecomunicação não desapareceu, distinto do mencionado por Sarraco, Harrow e Weihmayer, mas evoluiu para o atendimento de outras necessidades. Pelo advento dos *smartphones* e das redes sociais, as telecomunicações alcançaram o atendimento de necessidades que estão na parte superior da pirâmide de Maslow (sensação de pertencimento, estima) sendo:

- Comunicações necessárias: concentrar-se nas necessidades de comunicação necessárias do ser humano. Este nível só atende às comunicações mais urgentes e necessárias (era do telegrafo, por exemplo). Normalmente, ocorre quando o custo da comunicação é elevado ou existe escassez dos meios de comunicação.
- Comunicações comuns: concentra-se nas necessidades gerais de comunicação dos seres humanos. Observadas com a disseminação do 2G/GSM, mencionadas por Saraco, Harrow e Weihmayer (1997), em que ocorre um aumento da oferta e quantidade de meios de telecomunicações com consequente redução de seus custos e o tráfego de comunicação se espalha rapidamente (era telefonia comutada).
- Consumo de informação: concentra-se nas necessidades de informação. As camadas de necessidades mais básicas na pirâmide de comunicação de Maslow já foram supridas e as pessoas desejam mais que a comunicação entre si. A

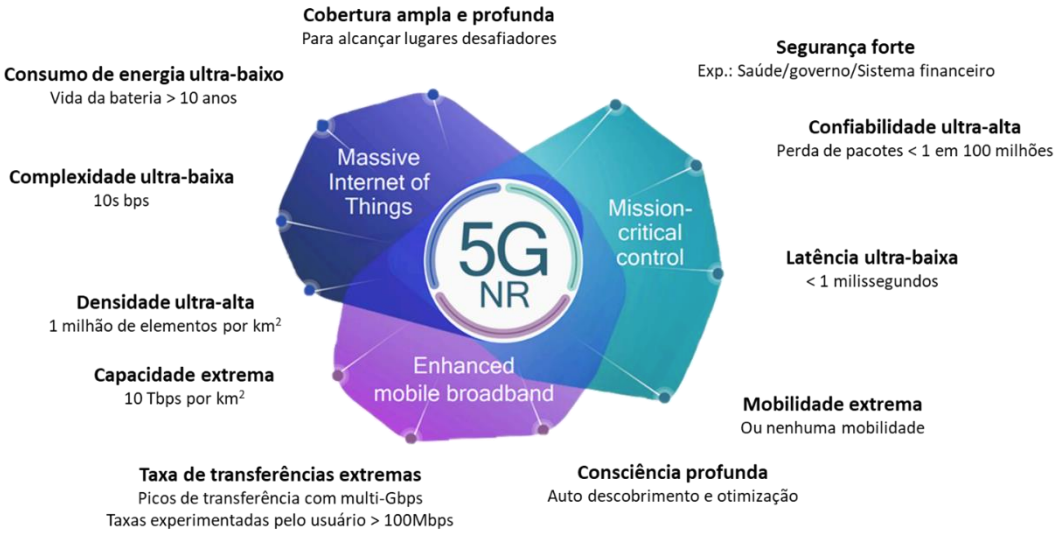
principal necessidade neste nível é o consumo de informação e o aumento da variedade de ferramentas de comunicação, o que levará ao aumento acentuado das necessidades de banda larga da Internet (era da internet móvel).

- Extensão sensorial: concentra-se nas necessidades de comunicação homem-máquina e máquina-a-máquina (M2M/IoT) e ampliação da realidade. Com os dispositivos de sensoriamento de clima, qualidade do ar, da água e de monitoramento cardíaco; e a ampliação da realidade percebida, a experiência sensorial do homem é estendida juntamente com interconexão entre o humano e a máquina. O número de conexões de rede aumentará rapidamente (era do M2M, IoT e da Realidade Virtual).
- Autolibertação: os quatro níveis anteriores se concentram na escala e no tipo de conexão. A autolibertação foca mais no conhecimento produzido pelas conexões anteriores. O sistema de comunicação torna-se inteligente, autoconfigurável e auto-otimizável, para si e para as coisas interconectadas, com isto permitirá que que as pessoas se libertem destas tarefas, deixando espaço para o conhecimento e o controle pelas máquinas (era do aprendizado por máquina – *Machine Learning* – da Computação em Nuvem – *Cloud Computing*, da inteligência cognitiva, da análise de massiva de dados – *Analytics/Big Data* e da robótica)

Vemos que na escalada da pirâmide de Maslow das telecomunicações, Figura 1, a geração tecnológica no topo da pirâmide é tomada pelas necessidades de conectividades vinculadas às máquinas, chamando a atenção para os seguintes casos: Indústria 4.0, Agricultura 4.0, aplicações automotivas e cirurgias remotas; impulsionadores da 5G, com crescimento das interações com as máquinas, do sensoriamento e do uso de mecanismos de inteligência artificial.

Nesse paradigma da subida na pirâmide de Maslow, é necessário atender não apenas o aumento da demanda por conectividade, mas também a necessidade de redes cada vez mais personalizadas para cada tipo de cliente e/ou aplicação, ou seja, as operadoras deverão aumentar a capacidade de rede e também segmentar essa rede em características e SLA de acordo com a demanda do cliente (menor latência, maior *throughput*, maior quantidade de *devices*, maior disponibilidade, redes heterogêneas/mistas).

Figura 2 – *Making 5G NR a reality leading the technology innovations*



Fonte: Autor “adaptado de” Qualcomm, 2016, p. 4.

As novas necessidades da Indústria de Telecomunicações se concretizam em uma nova especificação de rede, 5G , na qual a rede deve se preparar para criar uma rede para cada cliente (*Network Slices*), um conceito que não está necessariamente vinculado ao acesso de alta capacidade ou baixa latência, e sim à criação de um novo serviço pela especialização da rede para cada necessidade, uma rede para cada assinante, uma rede para cada camada de aplicação, heterogênea (HetNet), customizada de acordo com a necessidade da aplicação e dos usuários, principalmente corporativos; uma rede que permita aplicações e serviços em tempo real com alta criticidade (baixa latência), como medicina remota e transportes autônomos; ou uma rede de conexões massivas, como controle por internet das coisas (IoT – *Internet of Things*) massivo, cidades inteligentes e indústria 4.0; ou que permita o uso de alta capacidade, como plataformas de distribuição de conteúdo sobre a infraestrutura da rede (OTT – *Over The Top*), realidade virtual (VR – *Virtual Reality*) e realidade aumentada (AR –*Augmented Reality*) com mobilidade, representados na Figura 2 (QUALCOMM, 2016)).

Para tal é necessário construir e suportar essa nova infraestrutura de rede 5G que tem uma grande demanda de investimento em uma indústria na qual já se observa a queda da rentabilidade, como indicado por Stragy (2017).

Ao se avaliarem as possibilidades e necessidades de um investimento, normalmente, busca-se uma visão de retorno desse investimento no futuro, com base em elementos observados no passado e no presente, acrescidos de tendências e predições para mitigar os riscos desses investimentos. Assim, ao se avaliarem critérios de investimentos na indústria de telefonia móvel faz-se mister compreender seu passado, sua evolução e suas tendências.

Um ponto comum na avaliação da evolução tecnológica das redes móveis é afirmar que a evolução ocorre no aumento de capacidade ou de velocidade na transmissão da informação. Em um primeiro momento essa afirmação pode até parecer correta, principalmente quando observamos que a área das telecomunicações está sempre sendo contornada pelos limites dos meios de transmissão e suas tecnologias.

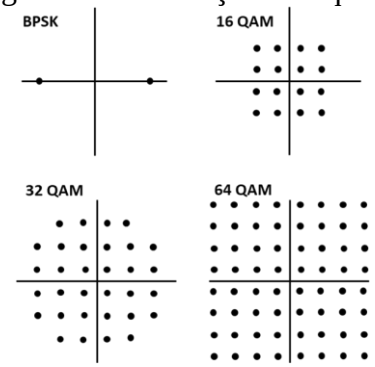
Para obter um melhor aproveitamento e reuso desses meios as técnicas de modulações (forma de transmitir uma informação sobre uma portadora), evoluíram e continuam evoluindo sendo por modulações analógicas – amplitude (AM) e frequência (FM) – ou digitais – pulso codificado (PCM), fase (PM), chaveamento de fase (PSK), de amplitude em quadratura (QAM), de divisão ortogonal de frequência (OFDM), de chaveamento de fase e amplitude (APSK/APK) – observadas na Tabela 1 e na Figura 3.

Tabela 1 – Modulações

Modulação	BITS por símbolo
BPSK	1
QPSK	2
8PSK	3
16QAM	4
32QAM	5
64QAM	6
128QAM	7
256QAM	8

Fonte: Autor

Figura 3 – Modulações em quadratura



Fonte: Autor

Essas evoluções na modulação trazem consigo uma eficiência maior do uso do meio de transmissão e consequentemente mais capacidade, normalmente atrelados à necessidade de materiais mais sensíveis e algumas vezes gerando ruídos e taxas de erro maiores. Outro método de ampliar a capacidade é pelo reuso dos meios de transmissão em espaços distintos, com adensamento de antenas (reduzir a área de cobertura de um transmissor) que trazem grandes ganhos de capacidade, mas necessitam de muito investimento em equipamentos.

Porém, se observarmos de forma mais aprofundada, a evolução das gerações tecnológicas está mais vinculada a novos serviços, vide Tabela 2, e a velocidade e a capacidade aos passos entre gerações (2.5, 3.5 e 4.5). Essa noção da evolução por novos serviços é importante quando se compara o decréscimo do ARPU, Figura 4, com os serviços antigos e sua sustentação através de novas aplicações e serviços.

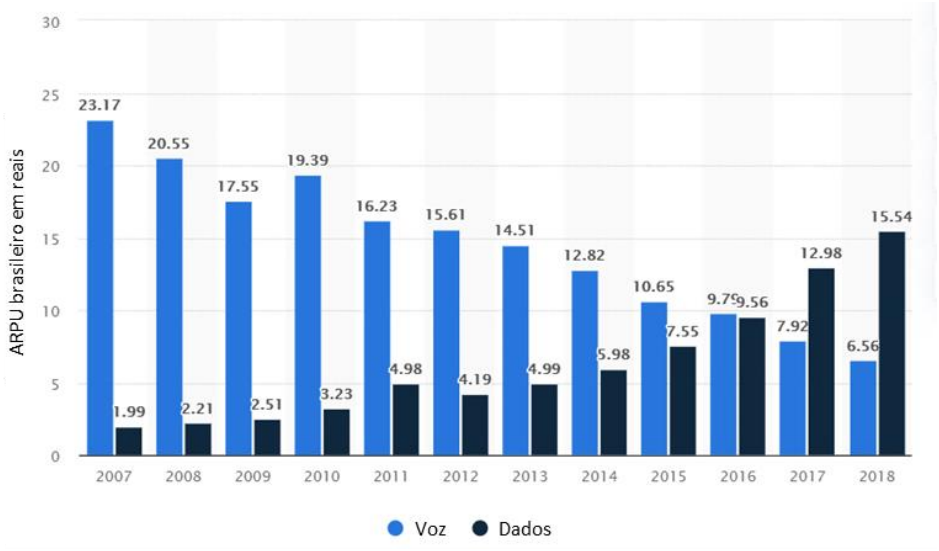
Tabela 2 – Evolução dos serviços em cada geração da telefonia móvel

	Geração									
	1G	2G		2.5G	3G		3.5G	4G	4.5G	5G
	AMPS	TDMA	GSM	GPRS/EDGE	UMTS	HSPA	HSPA+	LTE	LTE-Advanced	
Ano	1990	1998	2001	2003	2004		2012	2013	2018	2020
Mobilidade	X	X	X	X	X	X	X	X	X	X
Voz	X	X	X	X	X	X	X	X	X	X
SMS		X	X	X	X	X	X	X	X	X
Identificação		X	X	X	X	X	X	X	X	X
Roaming			X	X	X	X	X	X	X	X
Simcard			X	X	X	X	X	X	X	X
Dados				X	X	X	X	X	X	X
M2M				X	X	X	X	X	X	X
Multimedia					X	X	X	X	X	X
Internet Banda larga							X	X	X	X
IoT								X	X	X
Volte								X	X	X
Fatiamento de Rede									Core	X
throughput (max)	N/D	14,4 Kbps	14,4 Kbps	473,6 Kbps	2,0 Mbps	7,2/14,4 Mbps	21/42 Mbps	100 Mbps	1,0 Gbps	20 Gbps
throughput (med)	N/D	10-40 Kbps	10-40 Kbps	100-130 Kbps	128-384 Kbps	0,3-1 Mbps	3-6 Mbps	5-12 Mbps	5-42 Mbps	100 Mbps
Latência (ms)	N/D	500	500	300	250	70	30	10	10	<1

Fonte: Autor (informações compiladas das normas 3GPP e dados da GSMA)

Observa-se no relatório apresentado pala consultoria Statista (2020) que é pelo surgimento de novas aplicações, como no caso do acesso à rede de dados observado na Figura 4, que a receita das operadoras tem se sustentado, ou seja, é para o advento de novos serviços e aplicações que os maiores investimentos normalmente são destinados. Nesse caso é importante observar que a transmissão de dados de um ponto a outro, somente em quantidade, torna-se apenas *commodities* e que as aplicações agregadas, com características como latência, segurança, abrangência, dentre outras, devem nortear os investimentos nas próximas aplicações e serviços da rede.

Figura 4 – Evolução do ARPU no Brasil Statista (2020)



Fonte: Statista, 2020

Essas novas aplicações trazem junto de si novas métricas de avaliação do uso do recurso e da qualidade percebida do serviço prestado, vinculadas às necessidades mapeadas por Zhengmao LI (2018) em sua pirâmide de Maslow, que são cada vez mais um fator que leva à decisão de consumidores escolherem um prestador de serviço de telecomunicações.

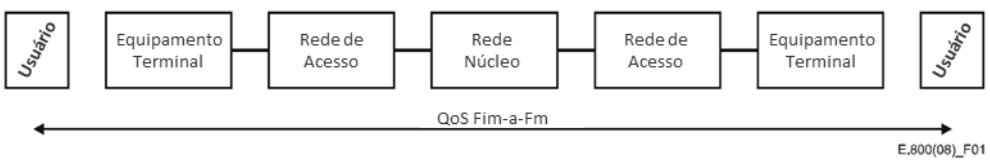
Esta sessão nos leva a compreender primeiro a importância de otimização dos recursos de telecomunicações por serem recursos limitados e escassos, e que a modelagem a ser utilizada por um orquestrador de recursos compartilhados de rede hibrida deve considerar tanto as necessidades básicas providas pelas aplicações das redes antigas (necessidade de comunicação para segurança, de comunicação para atender necessidades sociais e de consumo de informação) como também as aplicações das redes de quinta geração que atendem às necessidades das camadas superiores (extensão sensorial e altoliberação) considerando, porém, pesos distintos entre elas, já que cada uma delas atende a uma camada distinta das necessidades de Maslow e que camadas superiores geram uma percepção de valor agregado superior às camadas inferiores.

A próxima sessão avalia as métricas atuais e futuras da avaliação de qualidade e percepção dos serviços prestados para serem consideradas na modelagem do ambiente a ser orquestrado por aprendizado por reforço.

2.2 QUALIDADE NAS REDES DE TELECOMUNICAÇÕES

Pela ITU (2008) a qualidade nos sistemas de comunicação é conhecida pela capacidade de um provedor de telecomunicações de fornecer um serviço satisfatório a seus clientes, o que inclui: potência do sinal, qualidade da voz do serviço de telefonia, baixa taxa de bloqueio e de queda de chamadas, disponibilidade de serviço, baixo *delay* e altas taxas nas transmissões dados, estabilidade para a utilização dos serviços na transmissão de mídia ou uso de aplicativos; ou seja, a qualidade da comunicação fim a fim percebida pelo usuário como pode-se ver na Figura 5.

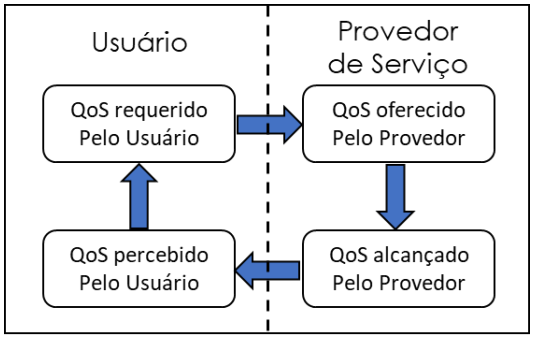
Figura 5 – Contribuição esquemática para QoS fim a fim



Fonte: Autor “adaptado de” ITU, 2008, p 1.

Em 2008, pelo documento de recomendação ITU-T E.800 (2008), a União Internacional de Telecomunicações (ITU) definiu QoS como a totalidade das características de um serviço de telecomunicações que tem a capacidade de satisfazer necessidades declaradas e implícitas do usuário do serviço. Nesse documento, a ITU já havia norteado a necessidade de distinguir e correlacionar a qualidade objetiva (QoS) ofertada pelo provedor de serviços de telecomunicações e a qualidade percebida pelo usuário do serviço (QoE), mapeada na Figura 6.

Figura 6 – Quatro pontos de vista da qualidade de serviços



Fonte: Autor "adaptado de" ITU, 2008, p 2.

Os mecanismos de QoS das redes sempre foram a base para o planejamento e dimensionamento das redes públicas de telecomunicações. A quantidade restrita de aplicações, com previsibilidade do comportamento do tráfego, suas projeções de crescimento e expectativas claras dos usuários possibilitaram por muitos anos um mecanismo de investimentos orgânicos com ampla utilização de métodos estatísticos e algoritmos para otimizar os recursos e garantir a previsibilidade e rentabilidade do sistema.

Existe uma clara diferença entre QoS e QoE. Enquanto a QoS lida com aspectos de desempenho do sistema físico, a QoE lida com a avaliação do usuário final de um sistema. A QoS possui abordagem orientada à tecnologia e depende de abordagens analíticas e medições empíricas ou simuladas, já a QoE requer uma abordagem multidisciplinar e multimetodológica para sua compreensão. É crucial lembrar que a QoE é altamente dependente da QoS, porque o desempenho técnico de um sistema tem um impacto significativo em algumas dimensões da QoE (CALLET, MOLLER e PERKIS, 2013).

Para a avaliação da qualidade de serviços prestados e sua comparação entre os prestadores de serviço, faz-se necessária a normalização dos termos e métodos de verificação dessa qualidade. Observando a importância desse tema para os usuários, operadores, fornecedores e governos, nas redes públicas de telefonia e Internet, a União Internacional de

Telecomunicações (ITU) desenvolveu um método normalizado de comparação e verificação das redes públicas de telecomunicações e estratificou seus temas nos seguintes documentos:

- E.800 (09/2008): *Quality of telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services*
- E.801 (10/1996): *telephone network and ISDN - framework for service quality agreement*
- E.802 (02/2007): *Framework and methodologies for the determination and application of QoS parameters*
- E.803 (12/2011): *Quality of service param for supporting service aspects*
- E.804 (02/2014): *QoS aspects for popular services in mobile networks*
- E.807 (02/2014): *Definitions, associated measurement methods and guidance targets of user-centric parameters for call handling in cellular mobile voice service*
- P.809 (06/2018): *Subjective evaluation methods for gaming quality*
- E.806 (06/2019): *Measurement campaigns, monitoring systems and sampling methodologies to monitor the quality of service in mobile networks*

A normativa apresentada segmentou a avaliação da qualidade em parâmetros objetivos (quantitativos), mensurados por instrumentos de análise de desempenho (QoS), e subjetivos (qualitativos), expressados pelo julgamento do entendimento humano do usuário (QoE). Os parâmetros quantitativos são definidos pela ITU em 3 grupos: parâmetros de qualidade de rede, de serviço e de gestão da operação, totalizando 94 parâmetros de qualidade, listados no Apêndice A, a serem avaliados para se verificar a qualidade fim a fim dos serviços de telecomunicações prestados ao usuário.

Dos parâmetros propostos pela ITU, os mais comuns a serem observados nas operadoras, excluídos os de negócio (provisionamento, bilhetagem e tarifação), são: os referentes ao acesso à própria rede (*network accessibility*), os de inicialização de serviços (*call set-up, connection establishment success ratio, voice interconnection, streaming e end-to-end ip network interconnection*), os de perseverança e integridade dos serviços (*call progress, service integrity performance, bit error ratio, interruption duration*) e os de desempenho (*conversational quality, end-to-end IP network performance e speed*), de forma a monitorar todo o fluxo de transmissão da comunicação do usuário, tanto os serviços de transmissão de

voz quanto os de transmissão de dados, fim a fim, desde seu ponto de acesso até sua aplicação final.

Como já mencionado, o documento de definições E.800 de 2008 da ITU já distingue a visão objetiva de indicadores de qualidade entregues pela operadora (QoS) daqueles indicadores percebidos pelo usuário do sistema, definidos como qualidade experimentada e percebida pelo usuário (QoSE – QoS *experienced/perceived by customer/user*). Desde então, apesar de não ter sido detalhado, já é possível observar uma tendência do normalizador de considerar a ótica do usuário ao que tange, mesmo que subjetivamente, à observação da qualidade. Essa tendência é comprovada pelos estudos e apresentações propostas na series de *workshops* da ITU sobre os temas de qualidade nos anos de 2019 e 2020 (INTERNATIONAL TELECOMMUNICATION UNION, 2020).

2.2.1 Qualidade de uma rede de transmissão de dados

Com toda a dimensionalidade dos tipos de aplicações e as variações na ótica da qualidade dos serviços prestados pelas redes de telecomunicações, as operadoras dos serviços tentam simplificar as pluralidades da qualidade em um único indicador. Dentre as propostas, aquela que teve, até o momento, uma grande aceitação é a dos medidores de velocidade da rede de dados.

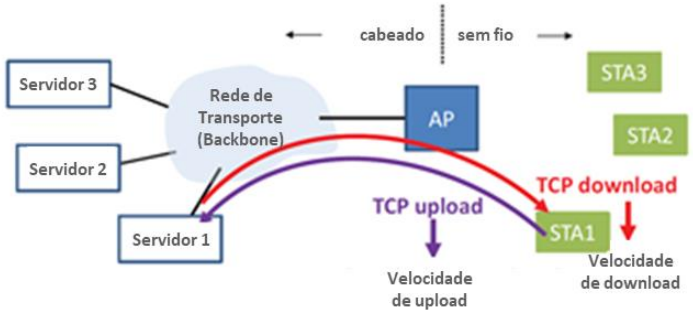
As aplicações de medição de velocidade, como a Ookla Speedtest, utilizam o tráfego de protocolo de controle de transmissão da camada de transporte na rede de computadores (TCP – *Transmission Control Protocol*) para avaliar a qualidade da rede de dados baseando-se na observação do histórico de que 80% do tráfego de dados da Internet é TCP, por isto tomou-se como premissa a medição da taxa de transmissão de TCP como critério do desempenho de vários aplicativos *on-line* dependente da taxa de transferência TCP máxima possível (NAYAK, PANDEY e KNIGHTLY, 2019).

Os estudos de Nayak, Pandey e Knightky (2019) mostram que a taxa de transmissão de dados (velocidade) é mensurada pela realização de testes ativos de transferência de pacotes de dados entre um terminal cliente e um servidor de testes. Esses testes são divididos em duas etapas: uma fase de configuração durante a qual os parâmetros do teste de velocidade são configurados e uma fase de medição que envolve um *upload* e *download* ativo de TCP.

A etapa inicial, de configuração, começa com um processo de seleção de servidor que pode ser manual ou orientada pela aplicação (o mais convencional). Quando é orientada pela aplicação, um servidor de teste é selecionado através da análise de um conjunto de servidores

disponíveis, de modo que o atraso na rede de transporte (*backbone*) entre o servidor de teste e o terminal seja o menor possível para garantir um rendimento máximo de TCP (NAYAK, PANDEY e KNIGHTLY, 2019). Assim, o teste de velocidade do cliente usa o servidor mais próximo, ou seja, um servidor com latência mínima possível para o ponto de acesso, como observado na Figura 7, onde a STA 1 (estação 1) executa um teste de velocidade (nesse caso o servidor 1 é selecionado a partir de um conjunto de servidores por possuir um atraso menor que os demais).

Figura 7 – Esquemático do teste de velocidade de *upload* e *download*



Fonte: NAYAK, PANDEY e KNIGHTLY, 2019, p. 2307

Na etapa de medição, o aplicativo de teste de velocidade estabelece vários fluxos TCP paralelos entre o servidor e o cliente, a partir de então, o aplicativo inicializa vários *downloads* e realiza o cálculo da taxa de transferência agregada em todos os fluxos para garantir resultados robustos. Durante a sessão de *upload*, o terminal cliente faz um *upload* ativo no servidor selecionado e mede a taxa de transferência TCP calculando a média do total de dados transmitidos de ponta a ponta no tempo total.

O principal objetivo dessas plataformas de medição de desempenho da rede de dados é identificar os índices de qualidade e desempenho, de forma objetiva, pelos seguintes parâmetros (NAYAK, PANDEY e KNIGHTLY, 2019):

A taxa de transferência (***Throughput*****)** é quantidade de dados que podem ser transferidos entre dois pontos da rede em um determinado intervalo de tempo, medido em megabits por segundo (Mbps), ele é diretamente relacionados à largura de banda (**BW – *Bandwidth***), enquanto o *throughput* é taxa de transferência real, BW é a taxa transferência máxima teórica;

A latência *(Latency)* é tempo que um único pacote de dados leva para viajar para um destino, mensurado com o tempo de ida e retorno do pacote, normalmente apresentado no relatório de tempo de ida e volta do pacote transmitido (RTT – *Round Trip Time*) em milissegundos (ms);

O distúrbio de rede (*Jitter*) é a variação entre duas medições de latência, também em milissegundos (ms);

A continuidade (*Packet Loss*) é taxa de perda de pacotes, tipicamente calculada como o número de pacotes perdidos dividido pelo número de pacotes transmitidos, apresentada nos relatórios em percentual (%) que auxilia a mensurar as interrupções/cortes em uma transmissão.

Como já observado no estudo realizado por McClellan (2019, p. 13) , "O projetista de um teste de medição de rede seleciona fatores com base nas suposições sobre o estado da rede, topologia e demandas de recursos. Os testes são normalmente realizados entre a máquina de um usuário, um computador pessoal e o servidor do provedor de teste.".

As características de rede como velocidade e latência geralmente variam com a distância e com a arquitetura do caminho do tráfego na rede de um ponto a outro. Um sistema de medição da qualidade de rede deve levar em consideração toda a arquitetura da rede e suas nuances para poder ser preciso ao encontrar os gargalos da rede e expressar a qualidade experimentada pelo usuário.

O trabalho de McClellan (2019, p.52) questiona a eficiência das ferramentas de verificação de velocidade da Internet: "Os testes de velocidade de rede existentes, como o Speedtest.net atribuem o servidor mais próximo por padrão, e os resultados relatados retratam uma imagem falsa do desempenho da rede com base na suposição imprecisa de que a maioria dos usuários de conteúdo da Internet vai estar próximo do conteúdo".

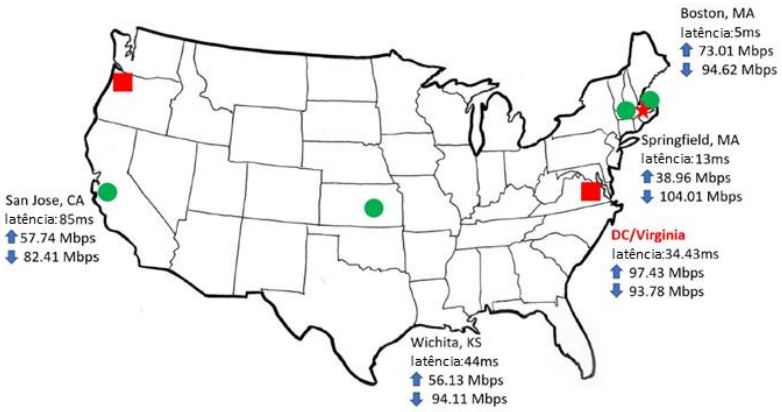
McClellan (2019) mostra as deficiências do *Speedtest* na validação do uso das Aplicações Fim, que não são o próprio teste de velocidade, pois essas aplicações estão em ambientes distintos dos servidores de teste de velocidade; elas são aplicações ponto a ponto (*peer to peer*), com comunicação direta entre usuários do sistema de telecomunicações, ou são aplicações para destinos diversos como Netflix, Google e outros, que não são as rotas avaliadas pelas aplicações de testes de velocidade.

Tanto o *throughput* quanto a latência são sensíveis a essa distância, o que confirma que quanto mais próximo o servidor de teste estiver do usuário, maior será sua medição de velocidade. McClellan (2019, p.20) também aponta que os gargalos na rede continuam existindo e são difíceis de serem localizados: "crescentes pontos de conexão entre ISPs e redes corporativas para serviços de Internet como Google ou Netflix podem enfrentar gargalos que são difíceis de localizar e podem exigir acordos de várias corporações para mitigar o problema desses gargalos."

Nos seus experimentos, McClellan (2019) observou que as medidas de velocidade da Internet realizadas pelos aplicativos como *Ookla Speedtest* são sensíveis à distância entre o

usuário que realiza o teste e o servidor do teste. Distância essa distinta daquela entre o usuário até a aplicação desejada, que leva ele a concluir que a experiencia do usuário terá uma realidade de latência e velocidade distintas daquelas observadas pela medição apontada pelo Speedtest.

Figura 8 – Resultados de testes velocidade x distância dos servidores



Fonte: McCLELLAN (2019, p. 54).

A velocidade apresentada pelos medidores de velocidade, quando eram externos às operadoras, mostrava uma aproximação dos gargalos existentes nas interconexões de rede dos provedores de serviço de internet (ISPs – *Internet Service Providers*).

Porém, após as grandes ampliações realizadas nesse segmento da rede, a velocidade de conexão até as redes externas (WAN – *Wide Area Network*) deixou de ser um ponto de estrangulamento. Uma nova realidade se apresentou: surgiram novos gargalos entre o usuário e as aplicações da rede nas conexões ponto a ponto (*peer to peer*), nas conexões das redes internas (LAN – *Local Area Network*) ou até mesmo nas aplicações pelas interconexões externas da Internet (McCLELLAN, 2019).

Além dos apontamentos apresentados por McClellan, Jordan (2019) indica a interferência nos resultados dos medidores de velocidade pela localização dos servidores de testes, que muitas vezes, como relatado por ele, se encontram dentro das operadoras: “Alguns servidores de medição estão localizados nas redes dos quatro maiores provedores de serviços de banda larga móvel e, portanto, a rota geralmente não cruza um ponto de interconexão. A metodologia escolhe o servidor de medição com a latência mais baixa de e para o assinante.” (JORDAN, 2019, p. 8).

Os medidores de velocidade, além de apresentarem medições pontuais, não avaliam a velocidade e a experiência do usuário pelos pontos de interconexão, já que os servidores se encontram dentro das operadoras. Ou seja, ter mais servidores de teste espalhados

geograficamente, na estrutura da rede, “encurta a distância” entre o usuário que realiza o teste e o servidor de testes que, consequentemente, apresenta valores melhores, mas não necessariamente alcança uma qualidade mais bem percebida pelo usuário no uso da aplicação desejada como *YouTube*, *Netflix*, *Facebook*, *WhatsApp*.

Jordan (2019) conclui em seus experimentos que quanto mais próximo dos servidores de teste estão os usuários, maior é a discrepância entre resultados da experiência do usuário, ou ainda, prestadores de serviço de Internet com maior quantidade de servidores de teste tendem a ter resultados melhores no *Speedtest*, porém esse resultado não reflete necessariamente a experiência do usuário.

Para este trabalho observa-se que a modelagem do orquestrador deve observar os parâmetros de latência, velocidade, distância e continuidade, mas não de uma forma centralizada nas aplicações de medidores de velocidade, e sim entre cada segmento dos enlaces que interconectam os elementos de cada aplicação.

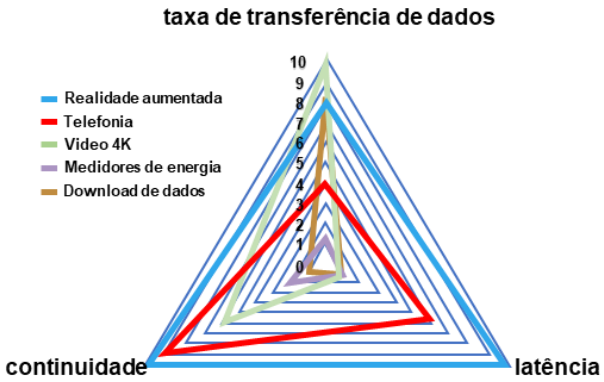
O conceito de rede 5G abrange a tecnologia *edge computing* e *edge networking*, que descentraliza o núcleo da rede e as aplicações que fazem uso dessa infraestrutura, ou seja, as avaliações de qualidade por velocidade e latência necessitam de uma verificação descentralizada e, por isto, o orquestrador deve ser hábil em validar os trechos e cada tipo de aplicação, e não somente o resultado final do enlace de internet.

2.2.2 Primeiras avaliações de qualidade em uma rede 5G

No *Mobile World Live Webinar: 5G network quality validation – today and the future*, ocorrido em fevereiro de 2020, foram apresentadas as primeiras avaliações de qualidade de uma rede de quinta geração em Manchester, Inglaterra. Nessa avaliação da qualidade da rede e da experiência do usuário, foi indicado, com base nas verificações de uso do 5G e no documento de padronização da arquitetura 3GPP TS 23.501 (2018) que, principalmente nas novas tecnologias e aplicações, a perspectiva de qualidade passa a ser um modelo para cada caso de uso, um triangulo entre velocidade da conexão, latência e continuidade do serviço com relevância diferenciada de acordo com cada aplicação, representado na Figura 9.

No exemplo de Manchester, foram verificadas as métricas de velocidade, latência e continuidade, correlacionadas com a nota de interatividade de usuários utilizando o serviço de conectividade para jogos *on-line*, diferenciando-se nas tecnologias de acesso 3G, 4G e 5G.

Figura 9 – Necessidade de rede de dados por aplicação



Fonte: Mobile World Live, Michael, Arnd, 2020

Dos resultados da utilização das redes para jogos em Manchester, é possível obter as seguintes conclusões: esse tipo de aplicação não necessita muita velocidade, a qual não é relevante no *score* de qualidade; a aplicação é muito impactada por latência (RTT); as interrupções na continuidade da transmissão de dados (*packet loss*) são o principal motivo de degradação do *score* de qualidade; as tecnologias 4G e 5G possibilitam uma latência melhor, porém é principalmente a estabilidade da transmissão o que torna o *score* do serviço mais elevado.

Na mesma linha dos testes realizados em Manchester, aplicações distintas estão sendo validadas em características específicas de cada uma delas. Alguns artigos de 2019 abordam essa segmentação dos parâmetros de qualidade de acordo com a aplicação, como por exemplo:

Cirurgias telementorizadas (Baixa latência, aplicação de alto risco e precisão.): o artigo publicado em 2019 no British_Journal_of_Surgery (LACY et all.,2019) mostra que a falta de profissionais em algumas áreas médicas cria a necessidade da realização de procedimentos cirúrgicos monitorados remotamente por médicos especialistas. Nesse segmento, a latência e a continuidade da transmissão são os principais critérios para avaliar a qualidade percebida pelos usuários.

A percepção do olho do tomador de decisão, o médico, é estipulada em 300 milissegundos o que torna esse experimento um sucesso para a experiência dos agentes que participaram. Porém, as especificações neste segmento estipulam, pela sua missão crítica, uma latência inferior a 100 milissegundos, 5QI 67 - 3GPP TS 23.501 (2018), para evitar quaisquer riscos ou perdas de informação.

A evolução dos procedimentos de cirurgias telementorizadas é no sentido de que esses procedimentos sejam executados remotamente em localidades em que não há profissionais para

realizá-los sob supervisão remota, sendo executados por braços mecânicos controlados remotamente com uma latência inferior a 5 milissegundos.

A importância do artigo de uso das telecomunicações em missão crítica neste trabalho é observar que a evolução das métricas de investimentos com base em qualidade percebida pelo usuário de segmentos médicos, com altíssimo valor agregado, requer novas tipologias de qualidade que não se restringem à velocidade da transmissão e nem mesmo somente à latência. Nesses modelos de alta criticidade, além da velocidade, continuidade e baixa latência, estão sendo inseridas necessidades de ampla cobertura e um novo item, a prioridade na transmissão dos pacotes de missão crítica, o que é previsto pela especificação dos 5QI.

Já para o mercado de games (Ultrabaixa latência*:* Jogos em tempo real, jogos em nuvem), um dos que mais cresce, desdobra-se em jogos em nuvem (*Cloud Gaming*), um novo paradigma de provisionamento de serviços que hospeda videogames na nuvem e transmite de forma interativa os jogos para usuários via Internet (GUO et al., 2019).

Para atender os jogos em nuvem objetivo do 5G é reduzir a demanda por equipamentos nos consoles de jogos e equalizar as diferenças entre os jogadores. Para esse segmento a qualidade da imagem e principalmente a latência na transmissão, e ainda a diferença de latência entre os competidores, são fatores decisivos nos resultados dos jogos em grupos (*multiplayer*).

A importância do segmento de jogos *on-line* para este trabalho é observar que novas aplicações, utilizando a mesma infraestrutura de rede, tem dependências distintas. A competitividade equiparada, com a latência e qualidade da imagem no segmento de jogos *on-line* são os critérios de qualidade experimentada, assim como a qualidade da voz foi, na história das telecomunicações.

A quantidade de tráfego de vídeo na rede já alcança (*Continuity Hight capacity: Edge Caching.*), como informado na publicação de Astudillo, Fonseca e Borin (2016), 70% do tráfego total carregado pela rede de transmissão entre o conteúdo e o usuário (*Backhaul*). Esse tráfego tem uma tendência de se ampliar com o crescimento das aplicações de realidade virtual e de realidade expandida.

Os efeitos da falta de capacidade de rede para as aplicações de vídeo, além do impacto nos demais segmentos de uso, são uma experiência degradada pela qualidade da imagem, interrupção completa do vídeo e o efeito de *buffering*. Algumas técnicas para reduzir tais impactos, são apresentadas nos artigos (RAHMAN, HONG e HUH, 2019) e (VIGNERI, SPYROPOULO e BARAKAT, 2019), que desenvolvem uma estratégia de levar o conteúdo cada vez mais próximo do assinante pelos mecanismos de Edge Caching para possibilitar o acesso ao conteúdo pelo usuário de forma direta e mais próxima e, com isso, reduzir a latência,

a carga no *Backhaul* e, consequentemente, ampliar a percepção de qualidade do usuário pelo uso do cache.

Para este trabalho, o segmento de acesso massivo a vídeo, *Edge Caching*, mostra que observar os tipos de tráfego, seus conteúdos e locais de absorção aproximam a operação da rede do seu usuário, possibilita desenvolver políticas de qualidade para a distribuição dos *Edge Caching* – infraestrutura computacional *cloud/fog networking* – baseados na interação do usuário com o conteúdo disponibilizado pela a rede, distanciando a rede da visão de *commodities* de velocidade e de volume, e agregando valor pela experiência de uso da aplicação.

Observa-se, com as abordagens sobre a qualidade da rede, que a modelagem do ambiente para o orquestrador deve levar em consideração: os tipos diferentes de tráfego, atendendo todas as aplicações porém obtendo uma remuneração de acordo com a cadeia de valor que a aplicação atende, sendo maior nas aplicações de quinta geração e menor nas aplicações legadas; os critérios de SLA baseados em prioridade (5QI), velocidade (*throughput/BW*), latência (RTT) e continuidade (*% Drop*) na distância entre o usuário e a aplicação (quantidades de saltos – *hops* – incluindo computação de borda – *edge computing*) e, se possível, na experiência subjetiva informada pelo usuário (QoE) de forma a alhanar as ações do agente ao proferir uma estratégia de custo, observadas as restrições sobre a capacidade da rede fim a fim e não somente de forma pontual.

2.3 A NUVEM: *CLOUD COMPUTING* E *CLOUD NETWORKING*

Talvez a palavra mais marcante no mercado nos últimos 10 anos seja compartilhamento. Pode-se observar o crescente número de empresas que estão utilizando a estratégia de compartilhamento como: Uber (carros), *Yellow* (bicicletas), *Green* (patinetes), *Airbnb* (imóveis), *WeWork* (escritórios), *Aws e Azure* (computadores). Toda essa economia compartilhada passou por um processo de conceituação, modelagem e abstração dos recursos físicos para que, com o uso de mecanismos matemáticos e de inteligência artificial, fosse possível obter a maior eficiência no uso desses recursos. No caso específico da nuvem, o conceito que já foi defendido de várias formas tem sua definição clara introduzida pelo Instituto Nacional de Padrões e Tecnologia dos EUA (NIST):

> "A computação em nuvem é um modelo para permitir acesso conveniente, sob demanda e pela rede, a um pool compartilhado de recursos de computação configuráveis (por exemplo redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com mínimo esforço de gerenciamento, ou serviço de interação, pelo provedor" (MELL et al., 2011, p. 3).

De fato, o conceito de nuvem vem sendo cunhado desde sua primeira idealização, em 1960, por John McCarthy que introduziu o compartilhamento de uso de *mainframes* no tempo, passando pelos conceitos da abstração via virtualização dos computadores, da computação em grade (*Grid*) (KRAUTER, BUYYA e MAHESWARAN, 2002) até receber seu nome de nuvem (*cloud computing*) em 1997 em uma palestra, na *INFORMS meeting, Dallas* (CHELLAPPA, 1997), na qual. No mesmo ano, Chellappa, Barua e Whinston (1997) publicaram sua visão de uma universidade virtual, eletrônica, que reduziria as barreiras da distância física do conhecimento através de uma infraestrutura de pesquisa e estudos compartilhada que denominaram "E^3".

De certa forma foi em meados de 2009 que surgiu o conceito de *Utility Computing* como um modelo de provisionamento de serviço em que os recursos de computação são fornecidos sob demanda e cobrados de acordo com o uso (*pay as you go*) e o conceito da virtualização, que juntos possibilitaram, de forma flexível, uma versão modelada, virtual, dos recursos de hardware com forma simples de serem gerenciados, culminando em *Cloud Computing*, uma computação em grade que usa tecnologias de virtualização em vários níveis para possibilitar o conceito de *Utility Computing* (ZHANG, CHENG, BOUTABA, 2010): .

Em 2010 já haviam sido concretizados três modelos de serviços utilizando a infraestrutura da Nuvem (ZHANG, CHENG, BOUTABA, 2010): o Programas de computador como serviço (SaaS – *Software as a Service*), em que a Nuvem provê programas de computador em formato de infraestrutura, acessados pela Internet, mensurados e cobrados sobre demanda, como *Microsoft 365*, banco de dados *Oracle*.; a plataforma como serviço (PaaS – *Platform as a Service*), em que a Nuvem provê infraestrutura para realizar a instalação de aplicações, como *Amazon AWS, Windows Azure*; e finalmente a infraestrutura como serviço (IaaS – *Infrastructure as a Service*), pela abstração e virtualização dos equipamentos através de máquinas virtuais (VM – Virtual Machine), a Nuvem provê infraestrutura computacional física como serviço, cpu, memória e disco rígido, como *Amazon AWS, Microsoft Azure* e *Google Compute Engine*.

2.3.1 Computação na borda e em névoa

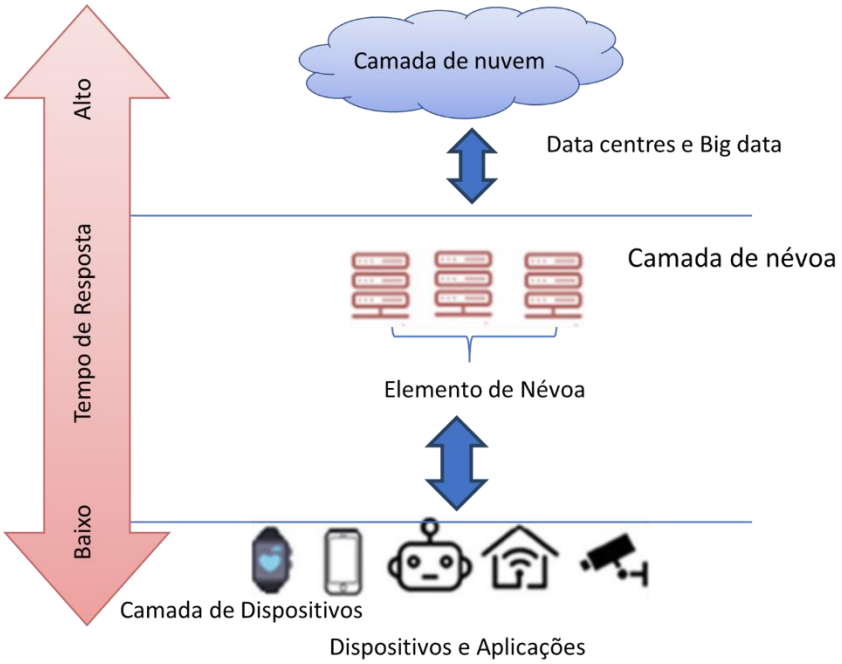
Devido à alta latência e lacuna de privacidade na computação em nuvem, uma abordagem denominada computação de borda (*Edge Computing*) propõe transferir o processamento de dados para dispositivos na borda da rede. Essa abordagem busca resolver o

problema de processamento com alta latência, porém, nos casos que a aplicação necessita de grandes volumes de dados, é necessária uma quantidade proporcional de espaço para o armazenamento desses dados. Para esse caso, a proposta é exercitar a computação completa de dados em dispositivos localizados na borda, uma abordagem denominada computação em névoa, que oferece o acesso aos dados aos usuário que possua acesso a esses dispositivos (TANWAR, 2021).

De acordo com Tanwar (2021) a computação de névoa, localizada na borda da rede, foi inicialmente projetada com o objetivo de oferecer recursos computacionais adicionais com o aumento do tamanho do armazenamento, como sendo uma extensão da computação em nuvem.

A Figura 10 mostra a arquitetura da computação em névoa e seus atributos. Essa arquitetura é dividida em três partes principais: camada de dispositivo, camada de névoa e camada de nuvem.

Figura 10 – Arquitetura de computação em névoa e seus atributos



Fonte: Autor “adaptado de” Tanwar. (2021, p. 5)

A camada de dispositivo funciona na visualização de dados, no processamento em tempo real, no armazenamento de micro dados e como um sistema de *gateway* incorporado. Já a camada de névoa consiste em um nó, ou elemento, de névoa que é usado para a análise e para a redução de dados. Essa camada é conectada com a camada de dispositivo (físico) para detectar e processar os dados. A última camada dessa arquitetura é a camada de nuvem que é utilizada como um *data center* (TANWAR, 2021).

Uma estrutura de computação em névoa é distribuída pelas bordas da rede com uma variedade de diferentes números de dispositivos. Esses dispositivos são universalmente conectados ao terminal da rede para fornecer: comunicação adaptável, serviços de armazenamento, variáveis colaborativas e capacidade de processamento computacional.

Com essa arquitetura, a computação em névoa propõe transferir alguns serviços de computação em nuvem para a borda da rede, perto dos dispositivos do usuário e possivelmente também parcialmente contando com os recursos dos dispositivos dos usuários, distribuindo assim a carga entre os dispositivos finais e *data centers* tradicionais em nuvem. Essa abordagem em névoa traz, entre outros, baixa latência e capacidade de resposta mais rápida, para a melhorar a escalabilidade e desempenho em todo o sistema (TANWAR, 2021).

Em resumo, a computação em névoa melhora a entrega pontual do serviço e reduz alguns problemas relacionados à computação em nuvem, como atrasos e instabilidade enquanto as informações são transferidas para a nuvem.

2.3.2 A nuvem das redes (*Cloud Networking*)

Apesar dos usuários do ambiente de *cloud* terem o controle da utilização dos recursos computacionais, eles não possuíam nenhuma visibilidade ou controle sobre os recursos da rede de interconexão entre os elementos das aplicações, ou dos clientes até as aplicações.

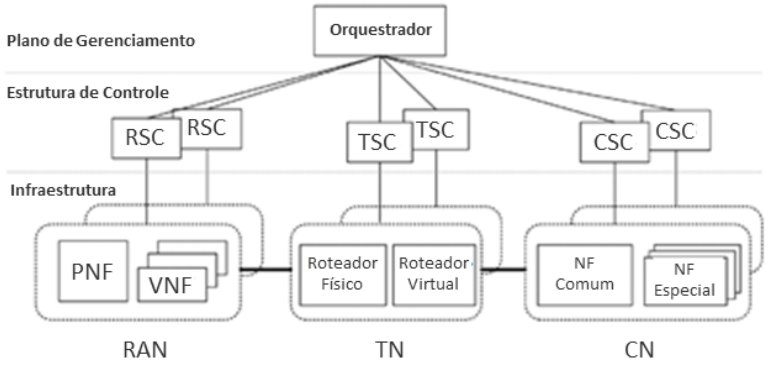
Embora a ideia de redes programáveis tenha sido concebida em meados da década de 1990 (CALVERT, 2006), somente em 2011, impulsionada pelo cenário da falta de visibilidade na *cloud*, que foi iniciada a idealização da virtualização dos recursos de rede com o objetivo possibilitar flexibilidade e controle, o NaaS, como sendo uma tecnologia chave para se criar uma nuvem de redes (*Cloud Networking*), que conecta os usuários aos serviços de nuvem de computadores e interconecta os serviços da nuvem (COSTA, 2012).

A transformação da rede em um recurso de nuvem permite aos usuários colocarem suas aplicações na nuvem com a topologia de interconexão das VMs que lhes convier, mesmo em localidades distintas dentro da infraestrutura da *cloud*, como uma aplicação servidora localizada nas proximidades de um escritório na cidade de São Paulo e sua aplicação cliente localizada na cidade de Belo Horizonte, por exemplo. Os usuários podem acessá-las por endereçamentos privativos e customizados, sendo garantida a segurança e os critérios de qualidade acordados por SLA. De fato, NaaS prove o serviço de uma rede virtual (COSTA, 2012).

Nas redes 5G, é a facilidade de fatiamento de rede (NS – *Network Slicing*) que transforma a rede em paradigma onde as redes são lógicas, virtualizadas e gerenciadas por orquestradores que buscam prover o SLA fim a fim em toda a infraestrutura. Essa nova

facilidade possibilita um novo formato de nuvem (*cloud*) das redes, com compartilhamento de toda a infraestrutura da rede como serviço (NSaaS).

Figura 11 – Arquitetura fim a fim *Network Slicing*



Fonte: Autor “adaptado de” Xu et al. (2020, p. 29527)

Como observado na Figura 11 (XU et al., 2020), o fatiamento de rede está dividido em 3 domínios: a virtualização da rede de transporte (que pode ser realizada através das técnicas de virtualização do SDN), responsável por interconectar os elementos computacionais dentro e fora da Nuvem; seguindo pela virtualização das funções de rede dos elementos (VNF – *Virtualized Network Function*, uma técnica de virtualização de funções de rede para agilizar as implementações de elementos da rede de telecomunicações explorada em gerações anteriores e também utilizada na quinta geração) da Rede Núcleo (CN – *Core Network*), responsável pelo gerenciamento e roteamento dos serviços de rede; e pela virtualização da nova rede de acesso em rádio 5G (5GNR – *5G new radio*), responsável pela entrada do usuário na rede. Sendo: a rede de rádios de acesso (RAN – *Radio Access Network*) constituída por torres rádios segmentada em função de rede física (PNF – *Physical Network Function*) e virtual (VNF), responsável por conectar os equipamentos dos usuários (UE – *User Equipment*) à Rede Núcleo (CN) através de conexões pela rede transporte (TN – *Transport Network*); a TN, constituída por roteadores enlaces (*links*) de transmissão e roteadores físicos e virtuais, sendo esse último representado pela SDN, é responsável por interligar a rede RAN à rede CN; a CN, constituída funções de rede comuns (ponto de transferência de serviços de voz, dados e mensagens – *gateways* voz e dados) e funções de rede especiais, virtualizada (NFV – *Network Function Virtualization*), responsável pela gerencia da prestação de serviço de telecomunicações, realiza a autenticação, o provisionamento dos serviços e SLA do usuário, e a interconexão ao serviço final através das saídas para a rede pública de telefonia (PSTN –*Public Switched Telephone Network*), ou para a internet, rede pública de dados (PDN – *Public Data Network*), ou interconexão a um destino privado.

A NS é a própria possibilidade da criação de uma rede para cada tipo de aplicação ou ainda para cada cliente, de forma personalizada. É essa flexibilidade que impulsiona as redes 5G a atingirem os níveis de necessidade mais elevados da pirâmide de Maslow e a atender os requerimentos para as redes da extensão sensorial e autoliberação. Sua especificação prevê que ela deve ter os princípios: de automação, com configuração sob demanda, sem intervenções manuais, além de contratos flexíveis; de ser programável, de forma a permitir que pessoas de fora do provedor de serviço controlem os recursos alocados; de visibilidade fim a fim, desde os provedores de serviços até o usuário final da aplicação; de isolamento, com garantias de desempenho e segurança; de elasticidade, necessária para garantir SLA; e de customização, com alocações customizadas de recursos e garantida dos requisitos adequados ao SLA.

De acordo com a especificação 3GPP TS 28.530 (2020) o fatiamento de rede (NS) deve ter a possibilidade de se adaptar com base nos requisitos específicos do SLA acordado entre o consumidor (NSC – *Network Slice Customer*) e o provedor do serviço (NSP – *Network Slice Provider*) e, ainda, o provedor de NS pode também adicionar requisitos não derivados diretamente dos SLAs, sendo somente associados aos objetivos (de negócios) internos do provedor, como por exemplo, para este trabalho, adicionar uma relação de custo na decisão de alocação dos recursos da rede. Esses requisitos seguem os atributos definidos no documento 3GPP TS 28.541 V16.5.0 (2020) sobre a modelagem dos recursos da rede 5G (NRM – *Network Resource Model*) apresentados na Tabela 3 com descrição dos principais atributos da Tabela 14 do Apêndice B. Essa tabela mostra que o modelo de consumo da rede de telecomunicações pode ser transformado, do modelo estático, de tamanho único, de consumo por volume de minutos e *megabytes*, em um modelo de sublocação da infraestrutura como serviço com definições dos atributos dessa infraestrutura.

Tabela 3 – Atributos do Fatiamento de Rede

Nome do atributo	Qualificador de Suporte	Nome do atributo	Qualificador de Suporte
serviceProfileId	MandatóriOpcional	uLThptPerSlic	Opcional
sNSSAIList	MandatóriOpcional	uLThptPerUE	Opcional
pLMNIdList	MandatóriOpcional	maxPktSize	Opcional
maxNumberoFalsoUEs	Opcional	maxNumberoFalsoConns	Opcional
coverageArea	Opcional	kPIMonitoring	Opcional
latency	Opcional	supportedAccessTech	Opcional
uEMobilityLevel	Opcional	userMgmtOpen	Opcional
resourceSharingLevel	Opcional	v2XCommModels	Opcional
sST	MandatóriOpcional	termDensity	Opcional
availability	Opcional	activityFALSOactor	Opcional
delayTolerance	Opcional	uESpeed	Opcional
deterministicComm	Opcional	jitter	Opcional
dLThptPerSlice	Opcional	survivalTime	Opcional
dLThptPerUE	Opcional	reliability	Opcional

Fonte: Autor “adaptado de” 3GPP TS 28.541 V16.5.0 (2020. p. 152)

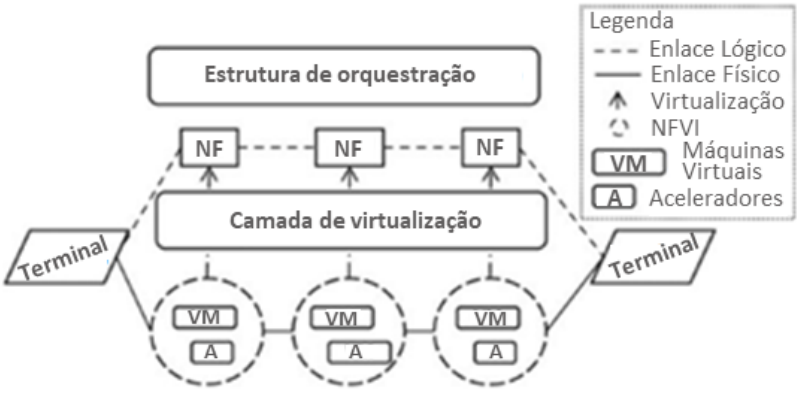
2.3.3 Domínio da rede núcleo – NFV

Entre o 1G e o 3G os serviços da rede núcleo eram realizados exclusivamente por Equipamentos (*hardware appliance*) dedicados e especializados no serviço a que ele se destinava, a central de comutação de telefonia móvel (MSC – *mobile switching center*), o equipamento de ponto de transferência e suporte da rede de dados móveis (GGSN – *gateway gprs support node*) e a base de dados de registro e localização do usuário (HLR – *home location register*); normalmente com custos e tempo de integração bem elevados (OSSEIRAN, MONSERRAT, MARSCH, 2016).

A partir no 4G, e concretizado no 5G, iniciou-se, já como uma forma de realizar o fatiamento da rede núcleo, a virtualização da CN através do NFV que transportou a função dos Equipamentos de rede para os computadores (servidores) convencionais da indústria, com custo e tempo de integração menores. Esses servidores realizam funções de distintos elementos na rede núcleo (OSSEIRAN, MONSERRAT, MARSCH, 2016) e passam a ser distribuídos em diversas localizações (computação de borda), até mesmo dentro do estabelecimento de usuários ou corporações, para aproximar o fluxo de dados (*payload*) das aplicações e usuários e, com isso, reduzir a latência necessária para as novas aplicações da indústria 4.0 e para as aplicações de controles emergências.

O NFV, de certa forma, se assemelha à virtualização de servidores da *Cloud Computing*, porém se difere da virtualização tradicional porque a virtualização de funções de rede (VNF) consiste em uma ou mais máquinas virtuais executando programas e processos distintos para substituir um Equipamento.

Figura 12 – Estrutura NFV



Fonte: Autor “adaptado de” Osseiran, Monserrat e Marsh (2016, p. 50)

Essas funções de rede virtualizas, representados na Figura 12 como VNF, são alocadas em uma infraestrutura de virtualização de função rede (NFVI – *Network Function Virtualization Infrastructure*) por uma estrutura de orquestração, uma espécie de VNE que, mapeia e aloca as VNFs (aplicações com necessidade de infraestrutura) na NVFI (camada física); em alguns casos, por ser inerente de telecomunicações, algumas funções de rede (NF – *Network Functions*) e aplicações necessitem de resposta em tempo real, nesses casos é necessário adicionar alguns *hardwares (chipset)* de aceleração. A orquestração de NFV se difere do VNE porque além de alocar os recursos na infraestrutura física (CPU e memória) ele também aloca as aplicações virtuais das funções de rede, ou seja, o processo de VNE é parte, está incluído dentro, do processo de orquestração NFV.

2.3.4 Domínio da rede de transporte – SDN

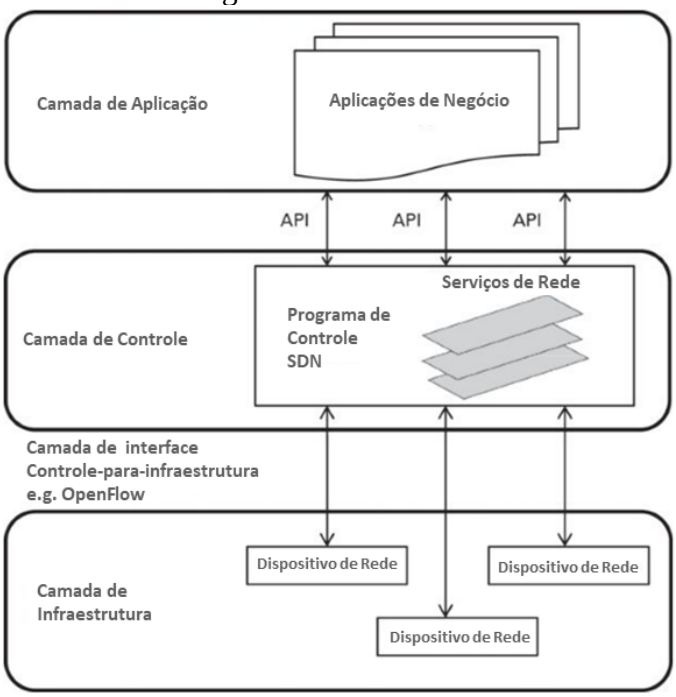
Uma rede é um "entrelaçado de fios (...) formando uma espécie de tecido de malha aberto, composto em losangos ou em quadrados de diversos tamanhos" (HOUAISS, 2009). Essa primeira definição da indústria têxtil inspirou a nomeação do "sistema constituído pela interligação de dois ou mais computadores e seus periféricos, com o objetivo de comunicação, compartilhamento e intercâmbio de dados" (HOUAISS, 2009) , seja ela de computadores, de informática ou simplesmente de telecomunicações.

De fato, mesmo com os avanços tecnológicos das redes sem fio, somente o acesso à infraestrutura ocorre sem fio. Após concluído o acesso, toda conexão da rede é realizada por um entrelaçado de cabos e fibras (fios), com os roteadores sendo os nós dessa grande malha de informação.

A rede de transporte é essa malha que realiza o elo de toda a comunicação à distância, ou talvez, é ela a própria rede, que é acessada de diversas formas.

O princípio básico da SDN é pela separação da rede em 2 planos, Figura 13, de controle e de dados (*payload*) que, de forma programável por interfaces (APIs – *Application Programming Interface*), possibilita aos desenvolvedores interagirem com a infraestrutura da rede para personalizar as características do tráfego com novos caminhos, velocidades e tipo de segurança, de uma forma virtual/encapsulada e mapeada pelo orquestrador VNE (OSSEIRAN, MONSERRAT, MARSCH, 2016).

Figura 13 – Estrutura SDN



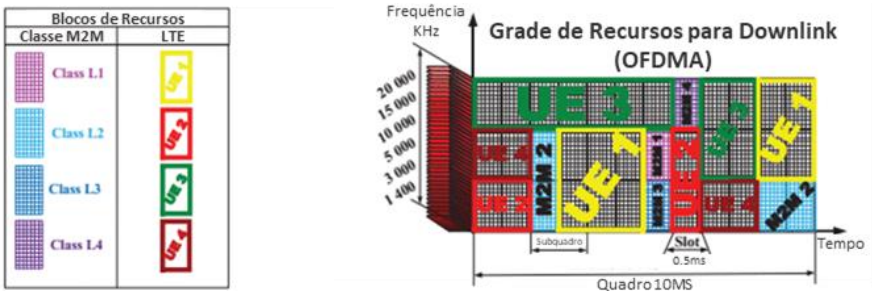
Fonte: Autor: “adaptado de” Osseiran, Monserrat e Marsh (2016 , p. 51)

Enquanto o orquestrador de NFV cria e exclui instâncias VNF, o orquestrador SDN cria e exclui caminhos de roteamento entre PNFs e VNFs e, em vez de permitir que os roteadores tomem decisões de roteamento, gerencia seus controladores responsáveis por determinar uma rota para cada fluxo na rede. Com um conhecimento mais abrangente sobre a rede, os controladores são capazes de realizar roteamento com objetivos de atender aos requisitos de QoS/SLA de forma a alocar um caminho, origem e destino, com uma determinada uma largura de banda, latência máxima ou continuidade (taxa de erro) mais adequada para a conclusão da requisição do fluxo.

2.3.5 Domínio da rede de acesso – RAN

Assim como os demais domínios da rede, para possibilitar ser programável e flexível de acordo com a especificação do NSaaS, a rede de acesso RAN introduz a virtualização do uso das capacidades da rede sem fio 5GNR pela a abstração do espectro de frequência e por sua alocação através mapeamento entre a identificação do fluxo fim a fim (QFI –QoS *Flow* ID) para o requerimento de NS e a modulação e alocação dinâmica de canais de rádio no tempo e na frequência, Figura 14. É essa forma de alocação de recursos que possibilita a criação de redes heterogêneas que compartilham o mesmo espectro (KLYMASH et al., 2017).

Figura 14 – Grid de recursos UE/M2M para *uplink* e *downlink*



Fonte: Autor: “adaptado de” Mykhailo et al (2017, p. 3)

Todos esses recursos de uma infraestrutura multiforme e distribuída, como afirmaram Mell e Grance (2011), vêm com a necessidade de ser compartilhada de forma conveniente (flexível, oportuna, apropriada, otimizada), configurável e rapidamente provisionada, o que nos leva ao problema a ser estudado por este trabalho, os algoritmos VNEs.

Necessidade essa que, de acordo arquitetura 5G apresentada na Figura 11, está dividido nos 3 domínios da camada dos controladores de fatiamento de rede, sendo: o controlador de fatiamento de rádio (RSC –*Radio Slice Controllers*), responsável por realizar a alocação e gerenciamento dos recursos da interface aérea como canais de frequência, rádios e capacidade de processamento da estação de rádio; o controlador de fatiamento da rede de transporte (TSC –*Transport Slice Controllers*), responsável pela alocação dos recursos da rede SDN, gerenciando a capacidade dos links, memória e processamento dos elemento de roteamento; e o controlador de fatiamento da rede núcleo (CSC –*Core Slice Controllers*), responsável por controlar a alocação as NFVs e seus VNFs de acordo com os requerimentos de SLA de cada *slice.*

Nessa seção podemos observar que, para obter uma orquestração dos recursos de uma rede 5G com fatiamento de rede (NS), é necessário orquestrar a solicitação do requerimento em três domínios: o da própria aplicação, que deve ser alocada em algum lugar próximo ou até dentro dos elementos de rede núcleo; o da rede de transporte com o gerenciamento da interconexão entre os elementos das rede núcleo, aplicação e seu usuário; e da alocação do recurso espectral de acordo com os requerimentos e atributos do SLA da aplicação/*slice*.

Nesse último podemos abstrair o equipamento do usuário (UE – *user equipment*) como sendo um elemento computacional de necessidade de recurso nula, já que não é demandada alocação cpu e memória (que já são parte da unidade móvel do usuário), e a interface espectral como sendo um enlace de capacidade compartilhada entre os diversos usuários que estão acampados no determinado espectro como representado na Figura 14. Essa abstração nos permite reduzir a solicitação do requerimento em dois domínios, porém necessita de uma gestão

rígida da localização de origem fixa da requisição da banda espectral e da largura de banda disponível/utilizada como sendo a entrada das características do enlace.

2.4 ALOCAÇÃO DE RECURSOS DE REDE

A imprevisibilidade da demanda pelos usos de recursos de rede gerada por muitos usuários da nuvem que não conseguem estimar a quantidade de recursos a ser utilizada no futuro, leva à infraestrutura ter que suportar uma demanda elástica via alocações dinâmicas, adaptando-se e mapeando os requerimentos em sua camada física.

Como vistos nas sessões anteriores, é a com virtualização de rede, que consiste em compartilhar recursos de dispositivos físicos de rede (enlaces, espectro, roteadores ou *switches*) entre diferentes redes virtuais, que é possível permitir a coexistência de várias redes, possivelmente heterogêneas, em uma única infraestrutura física, e adaptar-se a essa demanda imprevisível.

Observando da ótica de uma rede virtual, os roteadores e enlaces virtuais são vistos como fisicamente dedicados, mas, na prática, eles são compartilhados e por isso devem ter uma estrutura de compartilhamento e virtualização que garanta o isolamento (segurança) e mapeamento dos recursos de rede de forma a impedir a invasão de privacidade, os conflitos de utilização e a indisponibilidade do serviço (JMILA, 2015).

Algumas tecnologias de virtualização precursoras do NaaS/NSaaS possibilitaram o compartilhamento dos recursos de uma rede de forma segura através das redes virtuais locais (VLANs –*Virtual Local Area Networks*), do roteamento e encaminhamento virtual de rede (VRFs –*Virtual Routing and Forwarding*) e das redes privadas virtuais (VPNs – *Virtual Private Networks*). Essas são as tecnologias que, no ano de 2020, foram bastante difundidas e trouxeram ganhos no roteamento e compartilhamento de redes, porém, esses modelos são estáticos, pouco flexíveis, demandam de muito trabalho operacional de configuração por parte das operadoras e não gerenciam os recursos da rede de forma a otimizar a capacidade, a qualidade do serviço e os investimentos, ou seja, essas tecnologias são pouco convenientes.

Como já mencionado nas sessões anteriores, e por Mell e Grance (2011), toda essa virtualização traz a necessidade de ser compartilhada de forma conveniente, o que nos leva ao problema de alocação de redes virtuais, muito bem definido por Robertazzi e Shi (2020): “O problema da incorporação de redes virtuais, VNEs, é alocar o recurso de uma rede física, chamada de rede substrato, para uma requisição de rede virtual (VNR), que é composta por um conjunto de nós virtuais (máquinas virtuais) e um conjunto de enlaces virtuais que

interconectam esses nós virtuais" (ROBERTAZZI E SHI, 2020, p.183). Para tal é necessário compreender os tipos e estratégias de VNEs, a sua evolução até o estado da arte e como as VNEs são tratadas no NSaaS.

2.4.1 Classificações das VNEs

Existem diversas formas para abordar o problema da VNE: pela forma de atendimento das requisições, pela quantidade de domínios de rede, pela coordenação dos estágios de alocação, ou ainda se é centralizado ou distribuído (JMILA, 2015).

A abordagem descentralizada possibilita o aumento de eficiência pela computação distribuída com múltiplos VNEs para obter ganho de escalabilidade (HOUIDI et al., 2008), porém, essa abordagem é sensível à comunicação entre os agentes distribuídos e gera problemas com o sincronismo das informações do mapeamento da rede substrato. Independente dos possíveis ganhos, o risco de uma falha de sincronismo para operações críticas propostas pelo 5G nos faz optar por uma abordagem centralizada na qual a VNE tem uma visão global da rede substrato e toma as decisões de acordo com a informação atualizada dos recursos disponíveis, assim como nas propostas de orquestração centralizada do 5G no 3GPP TS 28.530 (2020). A abordagem centralizada gera problemas de escalabilidade que podem ser contornados por uma arquitetura federalizada (Wei, 2012), que divide a rede em domínios que são orquestrados, cada um, por VNEs independentes.

Alguns modelos de VNEs fazem a previsão da alocação de recursos em diversos domínios de rede, (HOUIDI et al., 2011), como se uma requisição pudesse, previamente, armazenar os recursos dentro de mais de uma rede de substrato, de mais de operadora de telecomunicação, aquela que atende o usuário na origem da requisição e aquela que hospeda a aplicação requerida.

A abordagem chamada multidomínios é pouco explorada por não ser, hoje, viável coordenar ações de alocação com diferentes operadoras, sendo mais comum a abordagem intra-domínio ou mono-domínio, quando a VNE aloca recursos em uma única rede substrato. Uma aplicação para a abordagem de multidomínios é no aumento de escalabilidade pela abordagem federalizada mencionada anteriormente, em que toda a infraestrutura está dentro de um mesmo operador de serviços de rede.

Pela caracterização da forma de atendimento às VNRs, a VNE pode ser classificada pela abordagem *off-line* ou *on-line*. De fato, a maioria dos casos reais de utilização de redes ocorrem *on-line* e, quando as requisições chegam, a VNE não identifica qual o perfil da próxima

requisição (DI et al., 2012). No caso das VNEs *off-line*, assume-se que todas as VNRs são conhecidas previamente (HOUIDI et al., 2011), o que em um primeiro momento pode parecer algo muito irreal, mas se observarmos a previsibilidade do comportamento do tráfego ao longo de grandes intervalos, plotado em pequenos intervalos de tempo, esses modelos podem tornar-se úteis.

Como a rede substrato é composta por mais de uma camada de alocação, no mínimo de nós e de enlaces, foram desenvolvidas abordagens utilizando o conceito de divisão e de conquista, em que se divide o problema maior em problemas menores para ser possível solucionar o problema como um todo. Essa abordagem ramifica-se na condução da VNE, pela execução coordenada entre os subproblemas e pela abordagem não coordenada.

A abordagem não coordenada, apresentada no Algoritmo 01, resolve cada problema de uma forma isolada e independente. Primeiramente ela realiza o mapeamento dos elementos de nó computacional e provê os insumos – localização dos nós mapeados – para, na sequência, realizar o mapeamento dos enlaces (ZHU e AMMAR, 2006). A cada interação a abordagem busca associar a demanda de nó virtual com a maior necessidade recursos computacionais ao nó de substrato com mais recursos disponíveis – um algoritmo de busca gulosa – e na sequência os enlaces são mapeados com um algoritmo que busca a rota mais curta entre os nós mapeados.

Algoritmo 01	Alocação básica VN (após a chegada do i-ésimo VN) não coordenada
	Entradas
	$G^S = (N^S, L^S)$: topologia da rede substrato;
	$N(n_a^S), \forall v \in N^S$: capacidade disponível de nós
	$L(l_a^S), \forall v \in L^S$: capacidade disponível de nós
	$G^r = (N^r, L^r)$: topologia VNR;
	Saídas
	$M: N^V \rightarrow N^S, n^v \forall N^V$ e $M: L^V \rightarrow L^S, l^v \forall L^V$
01:	$N^G = \emptyset$ {Nota: é um agrupamento de nós de substrato para a mesma VN}
02:	Para i = 1, ..., N_i^r faça:
03:	$N^G = N^G \cup \{argmin_{v \in N^S} n_a^s\}, n_a^s \geq n_r^v \ \forall n^v \in N^V$
04:	fim do loop
05:	Mapear nós em N^G de forma que N^Gtenha recursos para atender a VNR
06:	Encontre o caminho de menor distância para todos l(v)s no VNR

Fonte: Autor "adaptado de" Zhu e Ammar (2006)

O Algoritmo 01 recebe a estrutura da rede pelo grafo G^S juntamente com as informações de capacidade disponível, $n_a^S e\ l_a^S$, e as requisições de recursos da toda a rede virtual pelo grafo G^r. Ele inicializa um grafo N^G com valores "vazio" para receber um apontamento dos nós com a capacidade disponível, ordenados por uma política que preserva os nós com maior capacidade

e utiliza os nós com a menor capacidade, para serem todos mapeados aos n^v de acordo com os requerimentos da n_r^v.

Concluída a etapa de busca por recurso computacional para todos os nós, o grafo N^G é utilizado como insumo na etapa responsável por buscar e mapear todos os enlaces com a menor distância entre os VNs. Quando existem os recursos computacionais e de largura de banda, o algoritmo realiza os mapeamentos dos nós $M: N^V$ e enlaces $M: L^V$, caso contrário a requisição é rejeitada.

Entretanto, a falta de coordenação entre os estágios resulta na ineficiência do mapeamento dos enlaces e, portanto, utilizar uma abordagem coordenada, que incorpore as duas fases da alocação de uma forma conjunta, ou em "cenas" interligadas, possibilita a obtenção de melhorias de desempenho na alocação dos recursos de rede (CHOWDHURY, RAHMAN E BOUTABA, 2011).

Na coordenação em dois estágios, o mapeamento é realizado levando-se em conta o conhecimento da topologia dos enlaces virtuais, já a coordenação de estágio unificado propõe que os elementos da VNR, os enlaces e os nós computacionais, sejam conectados ao mesmo instante, o nós computacionais são ranqueados de acordo com os recursos disponíveis na posição da topologia e quando o primeiro par de nós computacionais é mapeado o enlace virtual (*link*) também é mapeado (CHOWDHURY, RAHMAN e BOUTABA, 2011).

2.4.2 Modelagem da VNE

O provisionamento de redes virtuais é uma das áreas que ainda requerem atenção já que afeta a utilização de recursos físicos (cabos e equipamentos) e eletromagnéticos escassos (espectro de frequência). Assim esse processo necessita de um constante aumento de eficiência com a garantia da qualidade do serviço prestado, uma relação custos x benefícios.

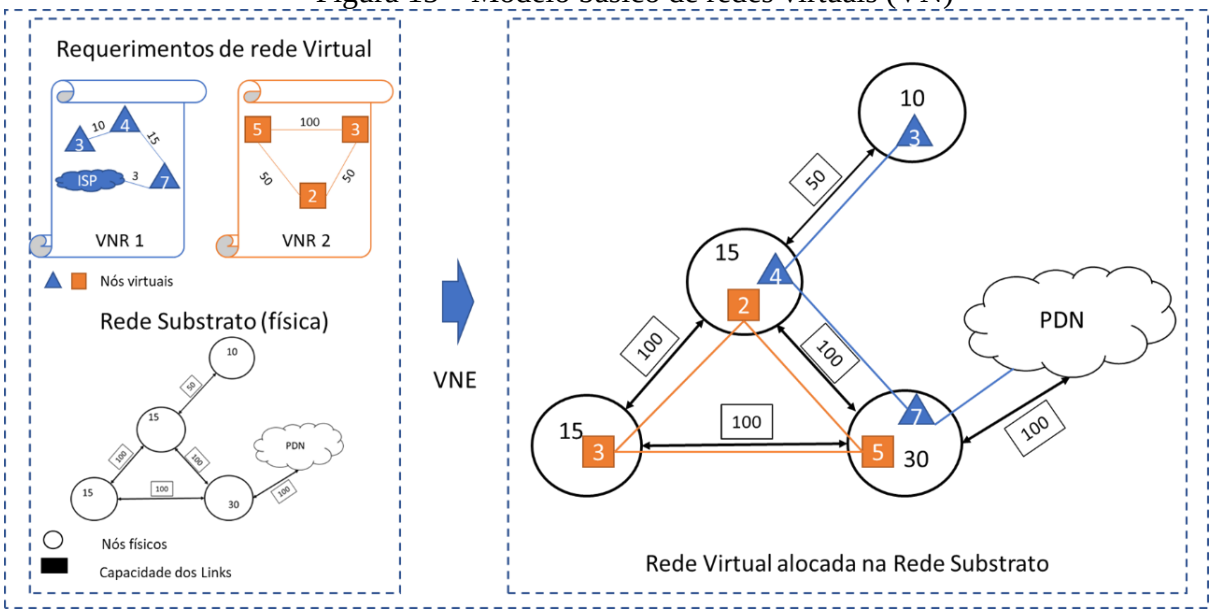
Na literatura os estudos do provisionamento de recursos de rede estão divididos em dois grandes grupos estratégicos: o de alocação de recursos de rede por VNE, e o de gerenciamento dinâmico de recursos de rede (DRM – *Dynamic Resource Managment*) (JMILA, 2015). Enquanto o objetivo da VNE é realizar o mapeamento eficiente das VNRs na rede substrato (**G^S** – ***substrate network*)**, o objetivo da DRM é trabalhar a demanda de recursos flutuante pela reorganização das redes virtuais na G^S.

Quando é realizada uma solicitação de VNR aos orquestradores da rede, a VNE realiza a tentativa de alocar os recursos e, quando obtém sucesso na operação de provisionamento, o DRM é acionado para otimizar o processo de alocação do recurso de forma continuada no

tempo. De certa forma, o DRM não existe sem a VNE, o que acaba sendo uma estratégia complementar e cooperativa de otimização.

A proposta de uma rede com os recursos abstratos e realmente virtualizada (NV – *network virtualization*) é definida pela abstração do hardware para computação, armazenamento e rede, tendo como principal mérito o compartilhamento seguro de múltiplas redes na mesma infraestrutura física realizada em duas entidades: as redes virtuais (VN – Virtual Networks) e a rede substrato (G^S). Sua forma mais comum de representação é por grafos, Figura 15, onde os elementos de rede (roteadores, servidores) são modelados e representados como vértices (nós) do grafo e o relacionamento entre eles, os enlaces físicos e lógicos, são representados pelas arestas desse grafo, sendo atribuído aos vértices e arestas as características de capacidade disponível e consumida de cada elemento.

Figura 15 – Modelo básico de redes virtuais (VN)



Fonte: Autor

A rede substrato (G^S – *substrate network*) é a infraestrutura física, que forma a parte essencial do ser, sobre a qual repousam os atributos da rede; é composta por um conjunto de nós de substrato (N^S – *set of substrate nodes*) e por um conjunto de enlaces de substrato (L^S – *set of substrate links*) que interconectam esses nós. Caracteriza-se por sua quantidade limitada de recursos disponíveis, tipicamente CPU e memória para os nós N^S e largura de banda para os enlaces físicos L^S.

Geralmente a rede substrato é representada por um grafo não direcional $G^S = (N^S, L^S)$; em que N^S é o conjunto de nós de substrato (n^s) e L^S é o conjunto de enlaces de substrato (l^s).

Representação da rede Substrato:
$G^S = (N^S, L^S)$ (1)

Um nó de substrato (n^s – *substrate node***)** é um elemento eletrônico capaz de enviar, receber e transferir informações para outros nós de substrato. Ele pode hospedar um ou mais nós virtuais (n^v) da mesma rede, ou de redes diferentes. A virtualização do nó de substrato pode ocorrer com diversas técnicas, chamadas *hipervisors*, que permitem a existência de múltiplas instâncias de espaços de uso. Assim, diferentes nós virtuais podem se hospedar em um nó de substrato e utilizar as mesmas funcionalidades da máquina hospedeira.

Cada nó de substrato $n^s \in N^S$ é associado a uma quantidade de capacidade de recursos disponíveis (n_a^s) e a uma unidade de custo do recurso (n_c^s).

Um enlace de substrato (l^s – *substrate link*) é o meio físico que conecta dois nós de substrato. Para virtualizar esse meio físico, o enlace de substrato é dividido em canais distintos e assim, o usuário que envia e recebe os dados tem a ilusão de que é dono do enlace (*link*).

Um enlace de substrato pode ser compartilhado por muitos enlaces virtuais da mesma rede virtual ou de redes virtuais diferentes; ele pode hospedar todo o enlace virtual, quando o enlace físico conecta fim a fim a demanda da rede virtual, ou somente parte do enlace virtual, quando o enlace de substrato se encontra no caminho do enlace virtual sendo, assim, necessários mais de um enlace de substrato para atender a demanda do enlace virtual.

Cada enlace de substrato $l^s \in L^S$ é associado a uma quantidade de capacidade de recursos disponíveis (l_a^s), que denota a capacidade de largura de banda (BW – *Bandwidth*) disponível no enlace e uma unidade de custo da BW (l_c^s).

A notação de caminho *P* (*Path*) é utilizada para representar o caminho percorrido pelo enlace virtual entre dois ou mais enlaces de substrato. P^S é o conjunto de caminhos (p^s) sem *loop* na rede substrato (G^S). A capacidade de BW do caminho (p^s) é denotada como (p_a), que pode avaliar a menor capacidade disponível, gargalo, nos enlaces do caminho de substrato.

A rede virtual (G^V –*virtual network*), assim como a rede substrato, é composta por um conjunto de nós (N^V –*set of virtual nodes*) e enlaces (L^V – *set of virtial links*), agora virtuais. É caracterizada pela quantidade de recursos requerida (VNR), podendo ser alterada durante o ciclo de vida da VN, normalmente representada por um grafo não direcional G^V (N^V, L^V) em que N^V é o conjunto de nós de substrato (n^v) e L^V é o conjunto de enlaces de substrato (l^v).

Representação da rede virtual:
$G^V = (N^V, L^V)$ (2)

Representação da requisição de rede virtual:
$VNR = (N^R, L^R)$ (3)

O nó de rede virtual (n^v – *virtual node*) é um componente de software, por exemplo uma máquina virtual, encapsulando CPU, memória, sistema operacional e dispositivo de rede. Em alguns casos pode também vincular algumas características restritivas como localização geográfica ou tipo do nó (roteador ou *switch*).

Um nó virtual pode ser hospedado por um e somente um nó de substrato; eles são interconectados pelos enlaces virtuais formando a topologia de rede virtual (VN).

Cada nó virtual $n^v \in N^V$ é associado a uma quantidade de capacidade de recursos requerida na VNR (n^r).

O enlace de rede virtual (l^v – *virtual link*) é a interconexão lógica de dois nós virtuais, normalmente definida pela quantidade de banda e a latência máxima aceitável. Em alguns casos podem também vincular algumas características restritivas como quantidade de sessões simultâneas, ou a necessidade de acesso a redes externas ou, ainda, endereçamentos de rede específicos.

Um enlace de rede virtual pode ser hospedado por um enlace de substrato ou dividido entre dois ou mais enlaces, sendo que cada divisão amplia, pelo processamento e enfileiramento do roteamento, a latência percebida no enlace lógico.

Cada enlace virtual $l^v \in L^V$ é associado a uma quantidade de capacidade de recursos requeridos na VNR (l^r), que denota a capacidade de largura de banda (BW – *Bandwidth*) requerida no enlace. A notação *P* (*Path*) é utilizada para representar o caminho percorrido pelo enlace virtual entre dois ou mais enlaces de substrato. P^V é o conjunto de caminhos (p^v) na rede virtual (G^V).

E o mapeamento de recursos (M) realizado pela VNE busca satisfazer a requisição VNR alocando os recursos da rede G^S de forma a minimizar os custos da alocação. Os recursos da alocação retornam à G^S quando o tempo da requisição expira ou quando a sessão de rede é terminada.

A Modelagem da VNE é definida como:

$$VNE\ min(n_c^S, l_c^S) \rightarrow M: G^V \rightarrow G^S = M: N^V \rightarrow N^S\ +\ M: L^V \rightarrow P^S \quad (4)$$

O problema do mapeamento em múltiplos domínios, o domínio de elementos e o de enlaces, mostrado na equação (4) é um problema conhecido por ser *NP-hard*. Alguns algoritmos heurísticos são utilizados para buscar de forma gulosa a alocação de menor custos em cada um de seus domínios, de forma geral, o problema é divido em dois subproblemas: o subproblema de mapeamento de elementos e o subproblema de mapeamento de enlaces (ROBERTAZZI e SHI, 2020). Representados nas equações e restrições abaixo:

Subproblema inicial, mapeamento de nós virtuais:

$$M: N^V \rightarrow N^S, n^v \rightarrow M(n^v) \quad (5)$$

Restrição básica: o nó de substrato tem que ter recursos disponíveis,

$$n_a^s \geq n_r^v\ \ \forall n^v \in N^V \quad (6)$$

Restrição básica: o nó virtual só pode ser alocado em um nó de substrato,

$$M(n^v) = M(m^v)\ \ se\ \ n^v = \ m^v, \qquad \forall n^v, m^v \in N^V \quad (7)$$

Subproblema secundário, mapeamento de enlaces:

$$M: L^V \rightarrow P^S, l^v \rightarrow M(l^v) \quad (8)$$

Tem que existir um caminho de substrato que interligue o ponto w ao j,

$$M(l^v) = p_{w,j}^v, \exists_p \in P^S,\ \ \forall l^v \in L^V \quad (9)$$

Restrição básica: caminho deve ter BW suficiente para a requisição,

$$l_a^s \geq l_r^v\ \ \forall l^v \in p_{w,u}^v \quad (10)$$

O objetivo dos algoritmos heurísticos executados em duas etapas é alcançar o maior número de alocações pelo atendimento das VNRs – responsáveis pelo conceito de receita de um operador de redes virtuais – com a menor utilização de recursos da rede física, substrato.

A receita é calculada pela utilização dos recursos da rede mapeados,

$$Receita = N_r^v + L_r^v \quad (11)$$

sendo a Receita de CPU(N) da requisição VNR,

$$N_r^v = \sum_{r \in G^r} \sum_{n \in N^r} \begin{cases} se\ \mu_{n^r}^{n^S} = 1, \quad n_{rCPU} \cdot receita_CPU \\ se\ não, \qquad 0 \end{cases} \quad (12)$$

e a Receita de BW da requisição VNR, $L_r^v =$

$$\sum_{r \in G^r} \sum_{l_{(w,u)} \in L^r} \begin{cases} se\ contagem\ \mu_{l_{(w,u)}}^{l^S} > 1, & l_{rBW} \cdot receita_BW \\ se\ não, & 0 \end{cases} \quad (13)$$

Onde $\mu_{n^r}^{n^S} = 1$ implica que o nó virtual foi mapeado em um nó de substrato e $\mu_{l_{(w,u)}}^{l^S} = 1$ implica que o enlace da VNR foi mapeado.

Os Conceitos mais abrangentes de otimização da rede substrato consideram não apenas a utilização dos recursos, mas também a qualificação desses recursos de acordo com a relação de SLA e QoS.

2.5 APRENDIZADO POR REFORÇO

Naturalmente, muitos problemas de rede podem ser resolvidos com as técnicas de aprendizado por máquina, desde os processos de inspeção e classificação do tipo do tráfego (DPI – *Deep Packet Inspection*) por redes neurais que auxiliam a balizar a qualidade percebida versus a característica de rede entregue, a regressões para previsão e antecipação do perfil do tráfego e de falhas que influenciam no balanceamento de carga da rede e, em especial para este trabalho, nos processos de decisão para tomada de ações de alocação de recurso e roteamento de tráfego, em que o Aprendizado por Reforço (RL – *Reinforcement Learning*) se mostra eficiente (ROBERTAZZI E SHI, 2020).

Aprendizado por Reforço é a ideia de a aquisição do conhecimento ocorrer sem um professor explicito, usando somente a própria interação com o ambiente – interação essa realizada pelo exercício das conexões com o ambiente e pela compreensão da relação de causa e efeito das ações realizadas no ambiente com o necessário para alcançá-las (SUTTON E BARTO, 2018).

Sutton e Barto (2018) descrevem um tipo de aprendizado por reforço que ocorre pela aquisição de conhecimento através das interações por ações que um agente de RL realiza em um determinado ambiente que ele deseja conhecer. Nessa descrição, RL é distinto dos casos de aprendizado supervisionado por não ser conduzido com um conjunto de amostras (*data set*) que fazem a função de professor do que se deseja aprender, e ainda se distingue do aprendizado não supervisionado porque não buscar apenas encontrar padrões para agrupamentos e classificações em um conjunto de dados.

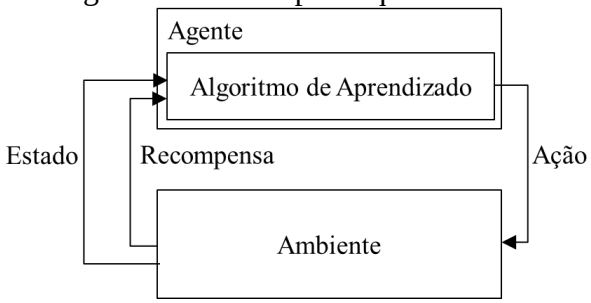
Na forma mais comum, o agente de RL está conectado ao ambiente por seus sensores de percepção e por suas ações. Em cada interação, o agente recebe a indicação do estado atual do ambiente s_t e então decide realizar uma de suas ações a para interferir no ambiente alterando-o a um estado subjacente $s_{(t+1)}$ que é percebido pelo agente juntamente com uma recompensa $r(s_{(t+1)})$ indicando se a ação foi boa ou ruim.

O trabalho do agente de RL é encontrar uma política π que mapeia quais ações devem ser tomadas em determinados estados do ambiente – estados → ações – de forma a maximizar a recompensa R no longo prazo, ou seja, o agente RL deve encontrar ações que não sejam boas apenas pontualmente.

Para encontrar a melhor política π é necessário conhecer os melhores caminhos e, para isto, é necessário pesquisar, explorar, e experimentar o seu ambiente tanto quanto possível, o agente RL tem que aprender com o resultado de suas ações e utilizar o conhecimento adquirido para, em suas próximas interações, escolher realizar as ações que obtiveram um maior benéfico no passado.

O agente de RL deve realizar várias tentativas e erros de forma a aprender como se comportar no ambiente e criar uma estimativa confiável sobre o ambiente e o prêmio esperado. Ele aprende através de suas interações em troca da obtenção de uma vantagem concretizada no acúmulo de um benefício numérico, ou seja, é um agente que realiza uma troca entre exploração e obtenção de vantagem, representada por uma recompensa.

Figura 16 – Exemplo esquemático de RL



Fonte: Autor

Suas denotações básicas, mostradas na Figura 16, são representadas por um conjunto de estados do ambiente (*S* – *States*), por um conjunto de possíveis ações *A* e um conjunto de recompensas *R*.

Como o agente de RL tem um objetivo claro de aumentar a longo prazo a recompensa R sobre todo o ambiente *S*, ele consegue observar o problema globalmente. Essa é uma de suas principais características em contraste com outras abordagens que consideram os subproblemas sem abordar como eles possam entrar em um quadro maior, ou seja, o resultado final de RL

comparado com outras abordagens pode levar a ganhos maiores na otimização fim a fim de sistemas complexos como o caso do problema VNE estudado neste trabalho.

2.5.1 Conceitos do aprendizado por reforço

Os processos para tomada de decisão quando o ambiente é observável, de decisão sequencial, estocástico, com modelo de transição em que o próximo estado do ambiente depende somente do estado atual e da ação tomada nele, ou seja, Markoviano, e tem recompensas aditiva sobre suas ações, esses processos são chamados de decisão de Markov (MDP – *Markov Decision Process*), que é a forma mais tradicional de tratamento dos problemas por aprendizado por reforço quando se tem uma fundamentação matemática do problema bem estabelecida.

O MDP é formalmente constituído por: um grupo de estados do ambiente; um grupo de ações que podem ser realizadas no ambiente; um modelo do ambiente (ou somente ambiente); uma função de recompensa sobre as ações do agente; e uma política que especifique qual ação o agente deve tomar estando em um determinado estado.

O Estado s_t são todas percepções que o agente tem das características do ambiente com a descrição suficiente para o agente tomar suas decisões. Para utilizar RL, a relação entre o estado atual e o estado futuro deve depender somente do estado atual; essa relação não pode ter interferência do histórico ou da sequência de ações que o levaram ao estado atual, ou seja, ela deve ser Markoviana.

A ação *a* é a interação que o agente realiza para alterar o estado s_t do ambiente, e o conjunto de suas possibilidades é representado por A.

O **modelo do ambiente** permite ao agente fazer inferências sobre o comportamento do ambiente; é utilizado no planejamento para auxiliar na decisão de qual ação deve ser tomada considerando-se previamente os possíveis resultados futuros.

A **função de transição de estados**, $T(s_t, a) \rightarrow s_{(t+1)}$, indica o estado seguinte, ou sua probabilidade nos ambientes não determinísticos, em que o ambiente se encontrará após o agente ter tomado uma ação a partindo de um estado s_t.

A **função de recompensa** $r(s_t, a) \rightarrow R$ é um resultado numérico – o prêmio – que o ambiente reporta ao agente após cada interação pela tomada de uma ação *a* quando o ambiente se encontra em um estado s_t. É ela que define o que é bom ou ruim para o agente em um dado estado e para uma dada ação, sendo responsável por definir o objetivo no aprendizado por reforço. O agente só pode influenciar o resultado de ganho do prêmio através de suas ações.

Enquanto a função de recompensa observa o que é bom de imediato, **a função de Valor ou Utilidade** U^{π}, na equação (14), especifica o que é bom no longo prazo; é definida como a soma das recompensas esperadas ao se seguir uma política π, ou seja, ela é a predição das recompensas futuras:

$$U^{\pi}(s_t) = \sum_{k=0}^{t} r_t \tag{14}$$

Ainda na função Utilidade, um fator de desconto γ (onde $0 \leq \gamma < 1$), que descreve a preferência que o agente tem pelas recompensas recentes sobre as recompensas futuras, pode ser adicionado à equação (14) para amortizar as recompensas e auxiliar na convergência no longo prazo; esse fator amplia a influência do conhecimento recém adquirido, representado na equação (15).

$$U^{\pi}(s_t) = \sum_{k=0}^{t} (r_t \cdot \gamma^t) \tag{15}$$

A **política de ações $\boldsymbol{\pi}$** define a maneira como o agente se comporta em um determinado momento através do mapeamento dos possíveis estados para as possíveis ações, π ($s_t \rightarrow$ a). Ela representa o núcleo do agente de aprendizado por recorço, podendo ser vista como um conjunto de estímulos, e é a responsável por mapear os estados percebidos no ambiente com as ações a serem tomadas quando estiverem nesses estados.

A política π pode ser representada por uma função simples, uma tabela ou cálculo extensos que devem ser suficientes para determinar o comportamento do agente. Buscar as ações para maximizar U^{π} no longo prazo, na forma do termo argmax na equação (16), é buscar as políticas de ações π^* que alcancem essa finalidade.

$$\pi^* = argmax_{\pi} U^{\pi}(s), \forall(s) \tag{16}$$

Ou ainda, na notação de probabilidades, π^* é o mapeamento $s_t \rightarrow a$ que maximiza a somatória das probabilidades, vezes seus respectivos ganhos, de uma ação a no estado s.

$$\pi^*(s) = argmax_{a \in A(s)} \sum_{S\prime} P(s'|s,a)U(s') \tag{17}$$

O principal objetivo do agente RL é resolver o problema de maximizar o ganho total no longo prazo, ou seja, maximizar U^{π}, que é o valor da quantidade total de recompensa esperado no futuro partindo do estado atual; é a Utilidade de um estado, ou ainda, é a recompensa total esperada a partir desse estado em diante. Para maximizar U^{π} o agente deve encontrar política ótima π^* pela solução da equação (17). Os principais métodos para encontrar essa solução encontram-se na subseção a seguir.

2.5.2 Métodos de otimização para problemas sequenciais decisórios

A literatura aponta alguns tipos de métodos de otimização para problemas sequenciais decisórios, desses, para este trabalho, é observado três tipos: a abordagem por programação dinâmica (*dynamic programming approach*), o método de Monte Carlo e o aprendizado pelo método das diferenças temporais (*temporal difference learning*). De acordo com as características da modelagem do problema, cada tipo de RL é mais adequado. Eles também se diferenciam em eficiência e formas de convergência (SUTTON E BARTO, 2018).

2.5.2.1 Programação Dinâmica

Na abordagem por **programação dinâmica** os algoritmos são utilizados para encontrar a melhor política quando se tem uma modelagem completa do ambiente. Seu funcionamento é baseado no aumento contínuo, ação por ação, da função U^{π} até convergir para a melhor política que maximiza U^{π} no longo prazo.

Essa abordagem inicia-se pela realização de uma ação de acordo com uma dada política, seguida da alteração do valor da utilidade do estado inicial, anterior à ação, com a recompensa dessa ação acrescida dos valores de utilidade de todos os estados possíveis e atingíveis do estado sucessor, ponderados pela probabilidade de alcançá-los.

Uma ação é ótima quando maximiza o somatório dos ganhos a longo prazo, ou seja, é aquela ação que alcança os estados de maior valor utilidade partindo de um estado s, e de todos

os estados possíveis e atingíveis do estado sucessor com essa ação. Se assumirmos que o agente irá tomar a ação ótima da política, podemos descrever a função de utilidade, ou função valor, levando a equação (17) na equação (15), pela equação (18), chamada equação de Bellman.

$$U(s) = R(s) + \gamma \cdot argmax_{a \in A(s)} \sum_{s\prime} P(s'|s,a)U(s') \qquad (18)$$

Porém, encontrar o argumento máximo que a ação a, pertencente ao grupo de ações A, realiza no ambiente em s não é uma operação linear, e para isto é utilizada uma abordagem interativa com o uso da equação (18). Essa é a base para um algoritmo de interação por valores que resolve os problemas de MDP.

O Algoritmo 02, de interação por valores, é utilizado na fase de treinamento do agente para calcular o valor da utilidade dos estados do ambiente. Ele gera um valor de recomendação para uma política π que, uma vez concluído o processo de treinamento, será utilizada para a tomada de decisões.

Algoritmo 02	MDP_interação de valor
	Entradas
	mdp: com estados S, ações $A(s)$, modelo de transição $P(s'\|s,a)$. recompensas $R(s)$, desconto γ, ε, erro máximo permitido na utilidade de qualquer estado;
	Saídas
	U: valores de utilidade dos estados
01:	repita
02:	$U \leftarrow U'$: nota{vetor utilizado para estados em S iniciado com zero}
03:	$\delta \leftarrow 0$: nota{variável local para validação de erro máximo}
04:	Para cada estado s em S faça: (interação pela equação de Bellman)
05:	$U(s) = R(s) + \gamma \times argmax_{a \in A(s)} \sum_{S\prime} P(s'\|s,a)U(s')$
06:	Se $\|U'(s) - U(s)\| > \delta$ então $\delta \leftarrow \delta + \|U'(s) - U(s)\|$
07:	Até $\delta < \varepsilon(1-\gamma)/\gamma$
08:	Retorna U

Fonte: Autor “adaptado de” Russeill e Norvig (2010, p.665).

Através das interações dos ciclos, representadas na linha 01 do algoritmo MDP por interação, é realizada a atualização do valor de $U(s)$ pela equação (18) de valor da utilidade de Bellman, ou seja, o valor de $U(s)$ é atualizado com o valor da recompensa atual adicionado o valor da esperança de recompensa futura ótima, reduzida a expectativa futura pelo desconto γ. Este processo se repete enquanto a variação δ for maior que um o erro aceitável sobre a taxa de

interesse $\varepsilon(1-\gamma)/\gamma$, o que de fato pode ser representado somente por ε, uma vez que a cada interação a diferença entre $U(s)$ e $U'(s)$ diminui e se torna muito pequena, levando o algoritmo MDP por interação a convergir.

Até este ponto o MDP realizou interações com o valor da utilidade do estado para encontrar a política ótima. Uma outra forma de encontrar essa política é diretamente pela interação com a própria política π que, se for considerada como sendo a política ótima a convergir, permite simplificar a equação de Bellman para o valor da utilidade U do estado s' estando no estado s, realizando a política π, representada na equação (19).

$$U(s) = R(s) + \gamma \sum_{S'} P\big(s'|s,\pi(s)\big)U(s') \tag{19}$$

Durante a etapa de treinamento, o MDP por interação da política, representado no Algoritmo 03, alterna em duas etapas, a primeira calculando a função utilidade U^{π} usando a política π, linha 02 do Algoritmo 03, e a segunda validando a necessidade de uma nova política no ciclo da linha 08 do Algoritmo 03. Essas etapas se repetem com os ciclos da linha 01 até convergir para uma política ótima π, quando na linha 05 não existirem mais ações que levem a um futuro melhor, ou seja $\pi = \pi'$ e $U^{\pi}(s) = U^{\pi'}(s)$.

Algoritmo 03	MDP_interação por política
	Entradas
	mdp: com estados S, ações $A(s)$, modelo de transição $P(s'\|s,a)$.
	Saídas
	π: política, vetor estado – ação
01:	repita
02:	$U \leftarrow (\pi, U, mdp)$: avaliação da política – Etapa 01
03:	$\delta \leftarrow Verdadeiro$: indicador se a política estabilizou - Etapa -02
04:	Para cada estado s em S faça: (interação pela equação de Bellman)
05:	Se $max_{a\in A(s)} \sum_{S'} P(s'\|s,a)U(s') > \sum_{S'} P\big(s'\|s,\pi(s)\big)U(s')$
06:	$\pi(s) \leftarrow max_{a\in A(s)} \sum_{S'} P(s'\|s,a)U(s')$
07:	$\delta \leftarrow Falso$
08:	Até $\delta = verdadeiro$
09:	Retorna π

Fonte: Autor “adaptado de” Russeill e Norvig, (2010, p.669).

Pode ser observado que, no algoritmo MDP por interação, o valor da utilidade U^{π} é atualizado somente ao passar pela avaliação de política pela Etapa 1 na linha 01 do Algoritmo 03. Essa abordagem funciona com uma primeira etapa que gera e atualiza o valor utilidade e uma segunda etapa que valida se esse valor de utilidade conseguiu gerar uma política ótima

para todos os estados do ambiente em S, repetindo esse processo até estabilizar na política ótima. Uma vez concluído o processo de treinamento, o algoritmo retorna a política para ser utilizada na tomada de decisões.

Com uma matemática muito bem desenvolvida para alcançar a melhor política, a programação dinâmica traz consigo a dificuldade de necessitar de uma modelagem precisa do ambiente, o que geralmente não é algo realista, e ainda a dificuldade em sistemas com variáveis lineares, que aumentam o espaço de estados com um crescimento exponencial, chamado "a maldição da dimensionalidade de Bellman" (SUTTON e BARTO, 2018, p. 17), que faz da programação dinâmica um problema computacionalmente difícil de ser tratado nesses casos.

2.5.2.2 Método de Monte Carlo

Dentre os métodos que não necessitam da função de transição bem definida (*Model-Free*), o método de **Monte Carlo** não requer nenhuma modelagem e é conceitualmente simples; é uma abordagem estatística para o estudo de problemas e equações diferenciais com muitas variáveis que tenham o cálculo analítico inviável. "O termo Monte Carlo é frequentemente usado de forma mais ampla para qualquer método de estimativa cuja operação envolva um componente aleatório significativo" (SUTTON e BARTO, 2018, p. 113).

A proposta de Monte Carlo é sortear variáveis de uma equação e realizar a interação dessas variáveis sorteadas com o ambiente para obter um resultado estatístico; o resultado deve ser anotado e o processo deve ser repetido *k* vezes até que suas amostras sejam suficientes para representar o ambiente. Sua exatidão depende do número de amostras geradas.

Monte Carlo torna-se mais factível com a introdução da computação que possibilita realizar diversos cálculos com muitas amostras aleatórias, assumindo-se que o erro gerado pelas amostras irá convergir para o valor real quanto maior for a quantidade de amostras. Monte Carlo possibilita obter resultados por avaliação diretas, *on-line*, com o problema, mesmo que não se tenha o modelo do ambiente – pela realização do cálculo algébrico da experimentação ou por sensoriamentos de um ambiente – e pode ser utilizado, *off-line*, com simulação ou amostras do problema ou ambiente.

Em RL, o algoritmo de Monte Carlo explora o ambiente pelos *k* sorteios das possíveis ações e sua aplicação direta no ambiente, ou simuladas, e calcula o valor de utilidade pela média dos valores das recompensas de cada sorteio na mesma relação $s_t - a$ durante vários ciclos. Nesse processo não existe um ponto de convergência definido; ele tende a convergir com infinitos sorteios, sendo que o erro do valor utilidade de um estado $\left(U^{\pi}(s') - U^{\pi}(s)\right)$ reduz a

cada sorteio com a mesma relação s_t – a pelo cálculo da média do valor utilidade nessa relação como observado na equação (20) e (21) de atualização do valor de utilidade por Monte Carlo.

$$U^{\pi}(s) \leftarrow \frac{1}{k}\left(\sum_{i=1}^{k} r_i\right) \tag{20}$$

Que equivale a:

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \frac{1}{k}\left(r_s - U^{\pi}(s)\right) \tag{21}$$

Em se tratando de uma média entre todos os sorteios realizados, o valor da utilidade do estado U^{π} no método de Monte Carlo só é atualizado após ter concluído todos os *k* ciclos do episódio.

Quin (2000) mostra que a computação paralela com uso de interface de transferência de mensagens (MPI – *Message Passing Interface*) e de unidades de processamento gráfico (GPUs – Graphics Processing Unit) é perfeita para a utilização do método de Monte Carlo por ser possível gerar vários sorteios e cálculos pela divisão da tarefa entre os diversos processadores e, na sequência, centralizar as informações geradas no elemento Mestre (*Master*) que concluirá o cálculo da abordagem. Porém, Monte Carlo não é adequado para computação interativa, onde a política é aprendida passo a passo, e pode demandar muito tempo de processamento para casos com muitos estados e possibilidades (SUTTON e BARTO, 2018).

2.5.2.3 Método das diferenças Temporais – TD-Learning

O Método de **aprendizado da diferença temporal (TD – *Temporal-Difference Learning*)** é uma combinação do método de programação dinâmica, no sentido de que é totalmente incremental e interativo, com o método de Monte Carlo, já que pode aprender diretamente da experiência com o ambiente sem que tenha um modelo bem formatado. Nesse método, a função de utilidade e a função de política são atualizadas de maneira interativa após cada ação entre os estados sucessivos e, por isto, é chamado diferença temporal. Ele considera a atualização do valor utilidade do estado como parte incremental de uma estimativa já existente, de acordo com a equação (22), uma outra forma de utilizar as equações de Bellman, onde α é considerado como uma taxa de aprendizado.

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha\left(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right) \tag{22}$$

A principal diferença de TD dos algoritmos de programação dinâmica, observada na diferença sutil da equação (22) que considera somente o estado futuro da utilidade $U^{\pi}(s')$ pela estimativa da política π, é realizar um único ajuste da utilidade por observação em cada interação. TD exige apenas que o próximo passo seja observado, enquanto a programação dinâmica, observado em $\sum_{S'} P(s'|s,a)U(s')$ na equação (18), realiza tantos passos quanto necessário e visita todos os estados para computar o valor utilidade. A abordagem de TD reduz muito o espaço de estados, além de não requerer um modelo completo do ambiente; na verdade, ela não requer nenhum modelo, podendo atuar pela percepção direta do ambiente (RUSSEILL e NORVIG, 2010).

TD também é diferente dos métodos de Monte Carlo porque não exige que a tarefa seja organizada em episódios ou que o a função de utilidade seja atualizada somente quando o episódio seja concluído.

O Algoritmo 04, o TD (0), é um caso especial de uma classe mais geral de algoritmos TD (n) quando n=0 (SUTTON e BARTO, 2018). No TD (n) a atualização do estado atual depende somente do estado futuro, porém ele pode utilizar *n* estados futuros para atualizar a utilidade do estado atual e ser mais assertivo para encontrar uma opção ótima.

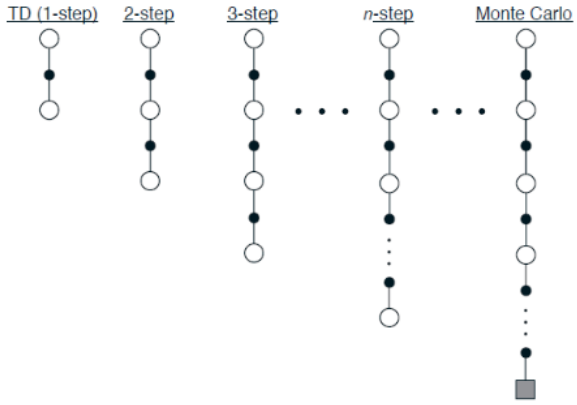
Algoritmo 04	Agente TD (0)
	Entradas
	π; a política a ser avaliada
	$U(s)$: inicializado com 0 $\forall s \in S$;
	Saídas
	$U^{\pi}(s)$
01:	Inicializa o estado s aleatoriamente, ou por ocasião
02:	Repita para cada passo do episódio de treinamento
03:	$a \leftarrow$ ação da política π no estado s
04:	Realize a ação a, observe a recompensa r e o novo estado s'
05:	$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha\big(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\big)$
06:	$s \leftarrow s'$
07:	Até s terminal

Fonte: Autor “adaptado de” Sutton e Barto, (2018, p.145).

No primeiro ciclo do Algoritmo 04 da linha 02, como todos os valores utilidade U são inicializados na linha 01, o algoritmo irá gerar as estimativas pela interação com a política π , assim como os algoritmos de interação, Algoritmo 02 e Algoritmo 03, vistos anteriormente. A partir de então, pela linha 05 do algoritmo 04, pode ser observado que a utilidade U^{π} é atualizada gradualmente através da taxa de aprendizado α. Em ambientes determinísticos, o valor de α pode ser 1, uma vez que o valor da utilidade será sempre o mesmo. Já em ambientes

não determinísticos, α deve ser menor que 1 para ponderar a relação entre a estimativa futura e a estimativa atual.

Figura 17 – Ranqueamento do espectro de backups do TD



Fonte: Sutton e Barto, (2018, p.169).

Como observado na Figura 17, as atualizações podem depender de uma profundidade de *n* estados futuros, até, inclusive, alcançar o estado terminal e se assemelhar a Monte Carlo. Porém é pouco usual, por ser necessário esperar os *n* passos para tomar a decisão. Através TD (0) podemos introduzir TD (λ) com a função $E(s)$ como um fator de elegibilidade do estado futuro e, ao invés de observar somente um estado a frente como TD (0). É possível ter uma visão que se aproxima daquela de Monte Carlo. Com $E(s)$ a equação do valor da utilidade para TD (λ) é reescrita na forma da equação (23):

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha\big(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\big)\, E(s) \tag{23}$$

A elegibilidade $E(s)$ de um estado é um "rastro" contendo a informação de quais estados foram atualizados recentemente, ela pode ser utilizada como um método para atualizar um reforço recebido aos demais estados visitados proximamente. Para calcular $E(s)$ o método mais simples é através da própria interatividade do algoritmo pela acumulação das vezes em que o estado foi "visitado" vinculado a uma variável do estado s.

$$E(s) = \begin{cases} \gamma\lambda E(s) + 1, & se\ s = s_t \\ \gamma\lambda E(s), & caso\ contrário \end{cases} \tag{24}$$

Onde γ , sendo $0 \leq \gamma < 1$, é o fator de desconto para reforços futuros e λ, sendo $0 \leq \lambda < 1$, um fator de desconto das diferenças temporais. Esses fatores são utilizados para TD (λ) convergir e fazer a relação $\gamma\lambda$ decair proporcionalmente na propagação para estados mais recentemente atualizados.

2.5.2.4 Q-Learning

Apesar de ser um algoritmo que utiliza o método *TD-Learning*, o *Q-Learning* está classificado, neste trabalho, em uma subseção separada por ele ter a característica de ser desvinculado da necessidade de uma política. O algoritmo desenvolvido por Chris Watkins (1989) observa a transição ocorrida nas interações entre os estados, e não propriamente o valor da utilidade do estado como os outros algoritmos observados até aqui.

Nesse tipo de aprendizado por reforço é muito comum utilizar uma outra notação para as funções que derivam da equação (22) de Bellman, sendo uma função $Q(s,a)$ que vincula diretamente a ação-estado ao valor das funções de utilidade do estado:

$$U(s) = argmax_{a \in A(s)} Q(s,a) \tag{25}$$

A proposta do Q-Learning, mostrado no Algoritmo 05, é obter uma maneira interativa para aprender a política ótima π^* diretamente quando o modelo não é conhecido. O algoritmo propõe que, ao invés de buscar o valor U^* da utilidade do estado, ele aprenda uma função da relação ação-valor, ou função Q, diretamente. Chris Watkins (1989) apresentou a variação das funções de Bellman, equação (26), para a relação $Q(s,a)$:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma\, max_a Q(s',a) - Q(s,a)] \tag{26}$$

Sendo s estado, a, ação a ser realizada em *t* estando em s, $r(s,a)$ a recompensa por se realizar a ação a estando em *s*, s' o estado futuro, a' ação a ser realizada no estado futuro, γ fator de desconto, sendo $0 \leq \gamma <1$ e α , sendo $0 \leq \alpha <1$ a taxa de aprendizado do algoritmo.

Algoritmo 05	Q-Learning
	Entradas
	*P*ercepção do estado s e da recompensa r após uma ação a
	Saídas
	π: política, vetor estado – ação
01:	Inicialize $Q(s,a) \forall s \in S, a \in A(s)$, de forma arbitrária, e $Qterminal$=0
02:	repita para cada passo do episodio
03:	$a \leftarrow A(s)$: uma ação pela política derivada de Q(e-greedly)
04:	Execute a ação e observe o estado futuro s' e a recompensa r
05:	$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma\, max_a Q(s',a) - Q(s,a)]$
06:	$s \leftarrow s'$
07:	Até s for terminal

Fonte: Autor “adaptado de” Sutton e Barto, (2018, p.158).

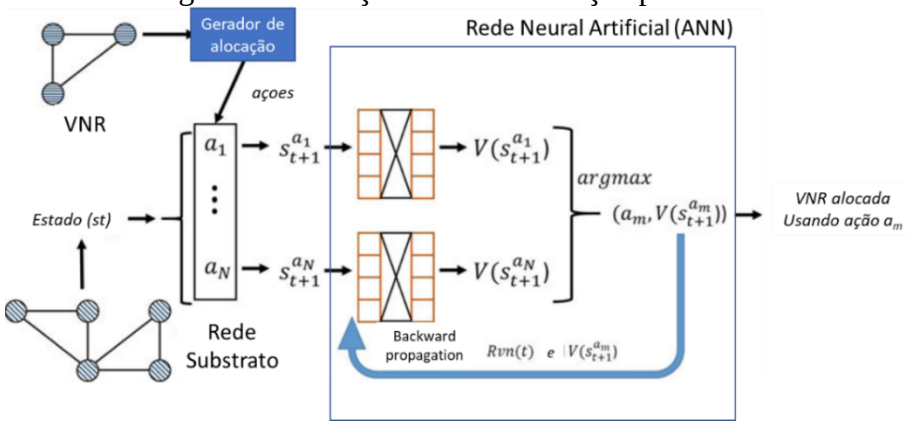
Q-Learning aproxima o valor de Q^* interativamente a partir da experiencia do agente. O algoritmo atualiza o valor da relação ação-valor na linha 05 somente com as informações que ele tem do estado atual e do valor do estado futuro após realizar a ação. A ação é escolhida por uma política aleatória, uma estratégia de escolha das ações para a exploração *e-greedy*. Apesar de convergir para a política ótima, o Q-Learning necessita de muito tempo para isso, principalmente por explorar por essa estratégia *e-greedy*.

Nos casos em que o aprendizado por reforço encontra uma quantidade extensa de estados, por ter variáveis contínuas ou sensoriamentos complexos, o uso convencional de RL fica limitado, não apenas pelo uso de memória, mas também pelo tempo e informação necessária para ter uma boa acurácia. Para que haja algum aprendizado, é necessário fazer generalizações que possam proceder uma aproximação boa do valor de utilidade do estado futuro com apenas algumas amostras de experiencias passadas (SUTTON e BARTO, 2018, p. 225).

Existem muitas formas de generalização, aquela necessária para o aprendizado por reforço é chamada de função de aproximação, porque ela obtém exemplos da função desejada *U(s)* e tenta generalizar por meio de uma função de aproximação, ou seja, um aprendizado supervisionado.

Uma Rede Neural Artificial do tipo *Perceptron* Multicamada (MLP – *Multi Layer Perceptron*) pode ser usada como um método de generalização para aproximação de uma função utilidade (valor), como observado na Figura 18, onde a entrada da rede neural é um vetor contendo a solicitação VNR juntamente com o estado s_t da Rede Substrato para que a rede neural generalize, incluso para estados não conhecidos, a possível melhor opção de alocação de recursos com base no histórico de aprendizado.

Figura 18 – Função de Generalização por ANN



Fonte: Autor “adaptado de”, Robertazzi e Shi (2020, p. 185).

Resumindo, RL pode ser vista como uma abordagem computacional para se entender e automatizar aprendizados direcionados a objetivos de tomadores de decisão. Distingue-se das demais abordagens por enfatizar o aprendizado pela interação direta do agente com o ambiente sem supervisão ou modelagem completa deste ambiente, tendo os conceitos de reforço *r* e função utilidade *U* como a chave para seu método de aprendizado, que é uma parte importante da modelagem de um agente tomador de decisão de alocação de VNE neste trabalho.

2.6 REDES NEURAIS ARTIFICIAIS

A busca de uma modelagem matemática, pela observação da neurociência, das atividades sinápticas do cérebro humano, em especial das células chamadas neurônios, auxiliou a construção de técnicas voltadas à ideia de uma inteligência artificial (IA) inspirada por hipóteses de como se criar uma rede neural artificial (RNA) pela modelagem de um neurônio, o *perceptron* como um classificador binário linear. Essa ideia foi classificada por Minsky e Papert (1969) como romantismo cibernético

Minsky e Papert (1969) trouxeram afirmações sobre a impossibilidade do modelo do *perceptron* de realizar aprendizado distinto de uma simples regressão linear por não ser possível, de acordo com os autores, transferir conhecimento para as camadas ocultas em uma rede de múltiplas camadas de *perceptron*s. Na afirmação de Minsky e Papert (1969), o erro entre a saída esperada e a saída estimada é objetivo e observável na camada de saída da rede, porém, os erros nas camadas ocultas são desconhecidos, uma vez que os dados de treinamento não informam quais valores os elementos ocultos devem ter, o que gera uma incerteza sobre o modelo e um desafio científico sobre o tema.

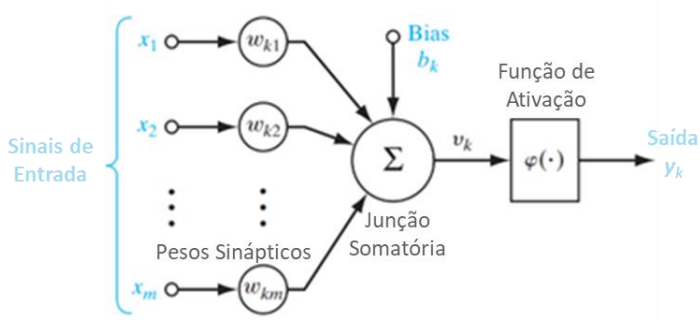
O desenvolvimento do algoritmo de retropropagação (*Back-Propagation*) pela derivação do gradiente do erro total (RUMELHART, HINTON e WILLIAMS, 1986) – erro que é diferença entre a saída da RNA e a amostra conhecida pelo treinamento supervisionado – propõe que: para que as camadas ocultas de uma rede neural artificial possam armazenar conhecimento, basta conhecer e propagar o erro da saída para a entrada da RNA através do gradiente do erro proporcional a cada peso e a cada elemento oculto da rede.

A descoberta de Rumelhart, Hinton e Williams (1986) proporcionou o retorno das pesquisas em aprendizado de máquina que culminaram em diversas aplicações nas seguintes áreas: reconhecimento e classificação de padrões de escrita e fala; reconhecimento de faces em visão computacional; controle e previsão de ações no mercado financeiro; e identificação de anomalias em imagens médicas, dentre outras.

A rede neural artificial *perceptron*, assim como o método de regressão linear, busca encontrar os coeficientes da reta que melhor representem as amostras. Porém, distinto do método de regressão linear pelos mínimos quadrados, o coeficiente de estimativa calculado pela RNA *perceptron* é obtido através de diversas interações, chamadas de treinamento ou aprendizado, que minimizam o erro do coeficiente com base na confirmação do resultado esperado, balizado por um coeficiente de aprendizado.

Uma vez concluído o processo de treinamento, o *perceptron* pode ser utilizado para estimar casos que não fazem parte do conjunto de dados do treinamento através da somatória de todas as variáveis de entrada, balizadas pelos pesos sinápticos e pela verificação de uma função de ativação, que quando alcançam seu nível de saturação acionam a saída do *perceptron* com 1 ou 0 (classificação binária), que, ao final, estima qual é a classe que o item pesquisado pertence:

Figura 19 – *Perceptron*



Fonte: Haykin (2009, p.11)

Matematicamente os coeficientes da reta, ou pesos sinápticos, são vetorialmente representados por *w*, com seu aprendizado realizado pelas diversas interações nos ciclos de aprendizado (épocas) com as equações (27), (28) e (29):

$$w_i = w_i + \Delta w_i \tag{27}$$

Sendo:

$$\Delta w_i = \eta(t - o)x_i \tag{28}$$

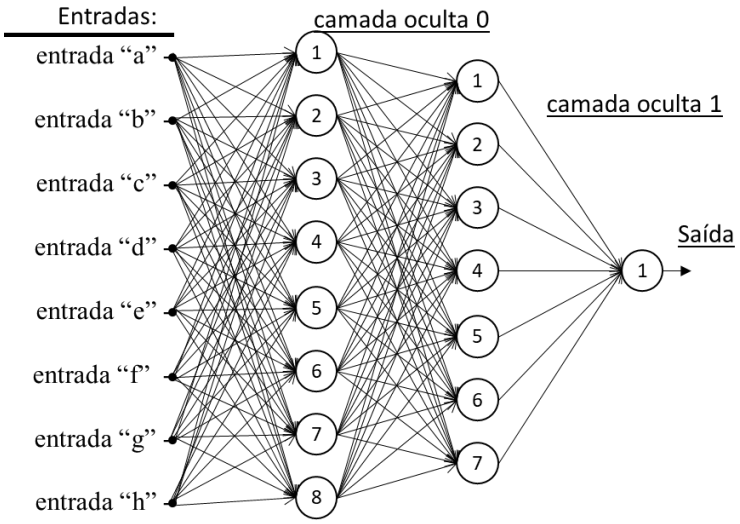
Onde 't' é o valor desejado, 'o' é a saída do *perceptron*, 'x_i' é a entrada do *perceptron*, e 'η' é a taxa de aprendizado

A função de ativação,

$$f(W^T X) = \begin{cases} 1 \; se \sum_{i=0}^{n} w_i x_i > 0 \\ 0 \; caso \; contrário \end{cases} \quad (29)$$

A MLP é uma rede de múltiplas camadas com conexões progressivas (*feedfoward*) sem realimentação; é um arranjo de um conjunto de *perceptrons* em diversas camadas com pelo menos uma camada oculta, como apresentado na Figura 20 sendo uma MLP com oito entradas e duas camadas ocultas, o que possibilita, através do agrupamento/soma de diversas redes lineares, um modelo computacional de classificação não linear (Haykin, 2009) .

Figura 20 – *Multi Layer Perceptron* – MLP



Fonte: Autor

O aprendizado da MLP é realizado pelo algoritmo *back-propagation*, que propaga o erro da saída aos elementos ocultos por uma regra de atualização dos pesos sinápticos, onde *e* são os erros propagados em cada camada (Haykin, 2009).

3 TRABALHOS RELACIONADOS – O ESTADO DA ARTE

No meio científico, entre os anos de 2018 e 2021, o tema de alocação de recursos em uma rede virtual tem ganhado mais atenção por ser um problema que se destaca com a implementação das redes de quinta geração que pretendem atender às necessidades de comunicação das camadas mais altas da pirâmide de Maslow: comunicação máquina-máquina, máquina-homem e autoliberação. Além de uma demanda especializada de rede, essas aplicações necessitam de atribuições como baixa latência e altíssima disponibilidade, a serem utilizadas em ações críticas, com riscos à segurança de pessoas e à continuidade operacional de companhias, já mencionados nas sessões anteriores.

Os trabalhos considerados mais relevantes para esta pesquisa estão listados na Tabela 4, sendo observados os atributos de rede, a localização da alocação do recurso, qual a metodologia utilizada para resolver o problema da VNE e se fazem uso ou não de aprendizado por reforço (RL):

Tabela 4 – Estado da Arte VNE

Autor	Ano	Metodologia	Atributos	Atributos espaciais (Localizacao)	Uso de RL
CHENG et al.	2011	Ranqueamento de elemento	CPU e BW	Não	Não
CHOWDHURY et al	2011	Modelamento MIP e relaxamento LP	CPU e BW	Sim	Não
MIJUMBI et al.	2014	3 Camadas de rede neural	CPU e BW	Sim	Não
MIJUMBI et al.	2014	Q-Learning Tabular	Status da alocação de recurso	Sim	Não
SOUALAH et al.	2014	Teoria de jogo	CPU e BW	Não	Não
JMILA	2015	Reorganização pelo Fluxo	CPU e BW	Não	Não
BENKACEM et al.	2018	2 inteiro LPs e teoria de jogo de barganha	Custo de pagamento e QoS Streaming	Sim	Não
HAERI e TRAJKOVIĆ	2018	Monte Carlo	CPU e BW	Não	Não
SCIANCALEPOREYOUSAF	2018	Q-Learning	CPU e BW	Sim	Não
SHAHRIAR et al.	2018	Formulação Heuristica LP	balanceamento da rede e caminho	Sim	Não
YAO et al	2018	abordagem por aprendizado profundo	CPU e BW	Sim	Sim
YU et al.	2018	Computação paralela	Alocação de recurso, latencia e drop	Sim	Não
DEHURY e SAHOO	2019	Formulação MIP	CPU, memória e BW	Sim	Não
DOLATI et al.	2019	Aprendizado profundo	CPU e BW	Sim	Sim
KIBALYA et al.	2019	Aprendizado profundo	CPU, BW e QoE	Sim	Não
CAO et al.	2019	Modelamento MIP e relaxamento LP	CPU, BW, QoS	Sim	Não
WANG et al.	2019	Aprendizado profundo	CPU, memória e BW	Não	Sim
WANG et al.	2019	diferença temporal	CPU e QoS	Não	Sim
ZHENG et al.	2020	Monte Carlo	CPU e BW	Não	Não
CAO et al	2020	Modelamento MIP e relaxamento LP	CPU, BW, Latencia, QoS	Sim	Não
YUAN et al.	2020	Aprendizado profundo	CPU e BW	Sim	Sim
HE et al.	2020	Formulação Heuristica LP	CPU, BW e QoE	Sim	Não
YUAN et al	2020	Q-Learning	CPU e BW	Sim	Não
YAN et al.	2020	Aprendizado profundo	CPU e BW	Sim	Sim

Fonte: Autor

Dentre esses, a seguir são detalhados os mais relevantes para este trabalho.

Cheng et al. (2011), desenvolveram um algoritmo (RW-MaxMatch) inspirado no mecanismo de busca da Google (*PageRank*) que realiza o ranqueamento de nós de rede (*NR – Node Ranking*) antes de iniciar o mapeamento da VNR. O RW-MaxMatch inicia com a criação de um enfileiramento (*buffering*) das VNRs até atingir uma quantidade *th* de requerimentos

para, então, realizar o ranqueamento dos nós virtuais e dos nós de substrato, considerando o nó de substrato de maior capacidade e de maior acessibilidade como sendo aquele com mais pontos, equivalente aos votos no *PageRank*.

Os autores desenvolvem uma métrica que, além da capacidade disponível, verifica os nós de substrato vizinhos com sua capacidade computacional e de links, para que, durante a alocação, o RW-MaxMatch verifique a probabilidade de que a requisição seja transferida para um nó de substrato vizinho (pular) ou de que o tráfego seja encaminhado como ponte para um nó de substrato vizinho reencaminhar ao destino final (*forwarding*). Com estas métricas, demostradas na equação (30), Cheng et al. (2011) formam a lista de ranqueamento de nós NR:

$$NR^{(t+1)}(v) = \sum_{u \in V} p^{J}_{u,v} \cdot p^{J}_{u} \cdot NR^{(t)}(u) + \sum_{u \in nbr(v)} p^{F}_{u,v} \cdot p^{F}_{u} \cdot NR^{(t)}(u) \tag{30}$$

onde $p^{J}_{u,v}$ é a probabilidade do nó virtual "pular" – ser realocado – do ponto *u* para o ponto *v* na vizinhança, $p^{F}_{u,v}$ é a probabilidade de encaminhar o trafego através do ponto u ao ponto v na vizinhança, $NR^{(0)}$é a somatória das capacidades dos elementos disponíveis $CPU \times \sum BW(l)$ e p^{J}_{u} e p^{F}_{u} são fatores de viés (*bias*) para balancear a relação entre a probabilidade do "pulo" a um nó de substrato e encaminhamento por um nó de substrato vizinho. Uma relação entre os recursos locais e globais da rede substrato, que eles determinaram como sendo 0.15 e 0.85 respectivamente, os mesmos valores do PageRank que, de acordo com eles, são os valores que experimentalmente atingem o resultado ótimo.

Tabela 5 – Matriz de transição de Markov de Cheng et al.

$$\mathbf{T} = \begin{pmatrix} p^{J}_{11} & p^{J}_{12} & \cdots & p^{J}_{1n} \\ p^{J}_{21} & p^{J}_{22} & \cdots & p^{J}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p^{J}_{n1} & p^{J}_{n2} & \cdots & p^{J}_{nn} \end{pmatrix} \cdot \begin{pmatrix} p^{J}_{1} & 0 & \cdots & 0 \\ 0 & p^{J}_{2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p^{J}_{n} \end{pmatrix} + \begin{pmatrix} 0 & p^{F}_{12} & \cdots & p^{F}_{1n} \\ p^{F}_{21} & 0 & \cdots & p^{F}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p^{F}_{n1} & p^{F}_{n2} & \cdots & 0 \end{pmatrix} \cdot \begin{pmatrix} p^{F}_{1} & 0 & \cdots & 0 \\ 0 & p^{F}_{2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p^{F}_{n} \end{pmatrix}$$

Fonte: Cheng et al., (2011, p. 43).

Cheng et al. (2011), desenvolveram uma matriz de transição de Markov *T* com todos os passos na matriz de transição para simplificar a equação (30) na equação (31):

$$NR^{(t+1)}(v) = \boldsymbol{T} \cdot NR^{(t)} \tag{31}$$

Os resultados do experimento de Cheng et al. (2011) apontam, pela observação dos gráficos, um aumento no longo prazo de cerca de 12,5% na utilização dos recursos da rede de substrato (receita) e na taxa de aceitabilidade de VNR, comparados com os algoritmos básicos de VNE (BL-SP e BL-MCF).

Chowdhury, Rahman e Boutaba (2011) propuseram um método de coordenação entre o mapeamento dos enlaces e dos nós através dos algoritmos ViNEYard (*Deterministic VN Embedding*, D-ViNE, e *Randomized VN Embedding*, R-ViNE) que levam em conta o conhecimento da topologia dos enlaces virtuais.

A proposta de Chowdhury, Rahman e Boutaba (2011) é uma abordagem na qual a cada VNR é recebida pelo algoritmo que amplia o grafo de rede substrato com a criação de "meta-nós" que representam a agregação de nós virtuais. Esses "meta-nós" são conectados a todos os nós de substrato considerando a restrição de latência, representada pelo quão distante a localização dos VNs pode ficar da VNR. Sobre o grafo formado, um algoritmo relaxado de propagação inteira mista (MIP) é utilizado para solucionar o problema de mapeamento do nó virtual e, após isso, o enlace é encontrado seguindo uma solução do kth caminho mais curto (*k-shortest*), quando o fluxo tem que ser contínuo, ou um algoritmo para solucionar o problema de fluxo multimercadoria (*multicommodity flow*), quando os pacotes da requisição podem ser segmentados.

Para as redes *mesh*, as simulações de Chowdhury, Rahman e Boutaba (2011) alcançaram, em comparação com os algoritmos de etapas da VNE não coordenadas (G-SP e G-MCF), um aumento no longo prazo de cerca de 15,5% na utilização dos recursos da rede de substrato (receita) e 7,5% na taxa de aceitabilidade de VNR.

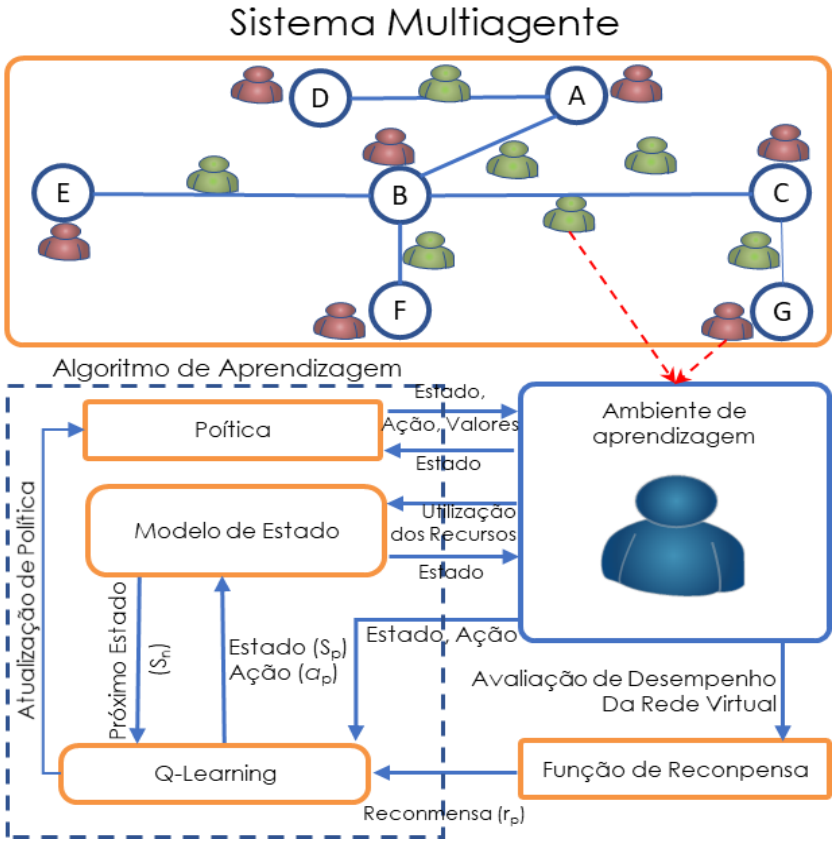
Mijumbi et al.(2014) propuseram um algoritmo, com o objetivo de aumentar a taxa de aceitação de VNR e a receita de alocação de recursos, que utiliza a dinâmica das redes como uma oportunidade para realocar, também dinamicamente, os recursos ativos, dentro do ciclo de vida da VN através do uso de aprendizado por reforço com multiagentes distribuídos, que os autores classificaram como sendo um algoritmo para gerenciamento de recursos dinâmicos (DRM – *Dynamic Resource Managment*).

A primeira diferença no trabalho de Mijumbi et al.(2014) está na forma de interagir com o problema: a proposta apresentada não é de otimização direta da VNE e nem de coordenação das etapas da VNE; é um algoritmo de cooperação que utiliza o conhecimento do comportamento dinâmico da utilização das redes para realocar os recursos disponíveis da rede

substrato e aumentar a disponibilidade, temporária, que permite aos algoritmos de alocação tradicionais observar uma capacidade disponível ampliada e, assim, alocarem mais VNRs.

Com uma abordagem que utiliza aprendizado por reforço (RL) Mijumbi et al.(2014) desenvolveram um sistema multiagente em que cada elemento da rede substrato, representados na Figura 21 como nós de substrato em vermelho e enlaces em verde, é mapeado como um agente de RL e que se comunicam de forma a possibilitar a coordenação das alocações dinâmicas nos casos em que a VNR utiliza recursos de mais de um elemento (agente), por exemplo nas rotas virtuais que utilizam mais de um enlace físico para atender a interligação de dois nós virtuais.

Figura 21 – Sistema RL Multiagente DRM para Cooperação com a VNE



Fonte: Autor “adaptado de” Mijumbi et al.(2014)

A alocação realizada pela VNE disponibiliza uma quantidade de recursos R_a à VN que não são completamente utilizados durante todo o ciclo de vida da VN. Os agentes RL de Mijumbi et al.(2014) monitoram a ocupação dinâmica R_x^V para ajustarem a alocação dos recursos da VN, de forma a flexibilizar a alocação acordada, com ações discretas de aumento ou redução dos recursos em passos de 12,5% da capacidade da VNR.

A proposta de Mijumbi et al.(2014) modela o estado de ambiente (S – *State*) percebido pelo agente RL com uma tupla $S(R_a, R_x^V, R_x^S)$ que denota a abstração da utilização dos recursos

alocados, R_a, dos recursos em uso pela VN, R_x^V e dos recursos da rede substrato R_x^S, classificados, de forma discreta, em sete grupamentos de ocupação com blocos de 12,5%.

Retirar recursos alocados e não utilizados por uma VNR aumenta a disponibilidade de recursos e consequentemente a possibilidade de alocar novas VNEs, mas também traz consigo o risco da demanda negociada pela VN crescer e não existirem recursos alocados para ela. Mijumbi et al. (2014) inseriram na função de recompensa do RL, equação (32), um prêmio negativo para evitar que os agentes disponibilizem mais de 75% da capacidade negociada pela VN. Os autores entendem que até 25% de recursos reservados já é suficiente para o algoritmo de RL ampliar a capacidade da VN no ritmo do crescimento da demanda.

$$recompensa(v) = \begin{cases} -100 & se\ R_a \leq 0{,}25 \\ v \cdot R_u - (k\widehat{D}_{ij} + \hat{P}_i)\ de\ outra\ forma \end{cases} \quad (32)$$

O objetivo do algoritmo de RL de Mijumbi et al.(2014) é encontrar uma política π^* ótima, ou quase ótima, de ações de flexibilização da alocação de recursos da VN, em um determinado estado de utilização dos recursos, que possibilite maximizar a utilização R_u (receita da rede substrato), garantido os índices de qualidade das VNRs, latência, $\widehat{D}_{ij}$ e continuidade (*drop*) $\hat{P}_i$. Para isso, utilizou-se o Q-Learning com explotação de acordo com a função de reforço da equação (32) e exploração do ambiente por uma função exponencial normalizada do valor de Q (*softmax*) de forma reduzir o tempo de convergência do RL.

Nos resultados de simulações do RL de Mijumbi et al.(2014) observa-se que a taxa de queda de pacotes (continuidade) e o atraso na transmissão (latência) convergem para os valores próximos aos do sistemas sem cooperação (aceitáveis) apenas nos casos com mais de 500 mil episódios de aprendizado. Observa-se também um aumento da taxa de aceitação das VNRs, em relação aos mesmos sistemas VNE não cooperados, entorno de 33% no longo prazo.

Jmila (2015), que também classificou seu trabalho como sendo um DRM, assim como a linha de pesquisa de Mijumbi et al.(2014) que se baseia na dinâmica do tráfego de uma rede, observou que as pesquisas até então não consideravam que os recursos ativos da G^V pudessem ser transportados para outros elementos da G^S e assim liberar recursos críticos da infraestrutura e possibilitar a aceitação de novas VNRs.

O trabalho de Jmila (2015) é focado nas VNs já alocadas, mas não observa somente a possibilidade de realocar uma G^V na G^S, propõe uma arquitetura que considera a dinamicidade da VN podendo ser expandia ou reduzida pelo próprio requerimento do usuário como aumento

de requerimentos das aplicações ou mesmo de volume de tráfego, o que gera a necessidade de ampliar a alocação de recursos para manter o serviço ativo.

Wang et al.(2019) realizaram a proposta coordenada e preditiva que integra uma rede neural (DNN – *Deep Neural Network*) a um RL para desenvolver o VNE_TD, um algoritmo de alocação de rede virtual baseado no aprendizado por diferença temporal. Eles modelaram o problema da alocação de recursos de uma rede virtual como sendo um problema em que sua solução deve maximizar os ganhos no futuro, mas que a tomada de decisão ocorre quando o VNE recebe as solicitações (*on-line*). De acordo com eles, isto formata um ambiente perfeito para aprendizado por reforço.

O modelo estado de ambiente (S – *State*) apresentado por Wang et al.(2019), na equação (33), é a capacidade residual normalizada dos nós de substrato e enlaces:

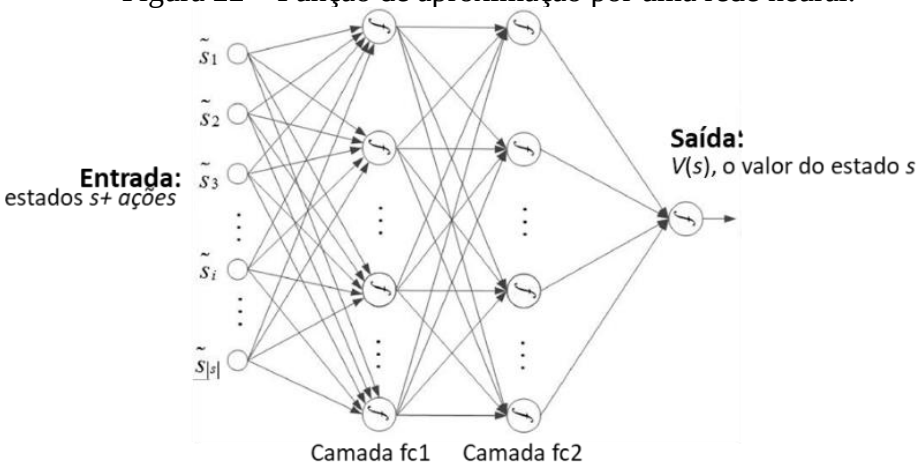
$$S_t = \begin{cases} \frac{n_{a_1}^{S^t}}{c_{max}^t}, \frac{n_{a_2}^{S^t}}{c_{max}^t}, \dots\dots, \frac{n_{a_N}^{S^t}}{c_{max}^t}, \frac{l_{a_1}^{S^t}}{b_{max}^t}, \frac{bl_{a_2}^{S^t}}{b_{max}^t} \dots\dots \frac{l_{a_L}^{S^t}}{b_{max}^t} \end{cases} \tag{33}$$

onde $n_{a_1}^{S^t}$ é a capacidade disponível do nó de substrato n^S no instante t, $l_{a_1}^{S^t}$ é a capacidade disponível do enlace l^S no instante t e c_{max}^t e l_{max}^t são as capacidades totais remanescentes de nós de substrato e enlaces que ponderam o estado de ambiente.

Na proposta de Wang et al. (2019) é identificada a dificuldade de encontrar a função U^*ou política π^*, ponto chave nos problemas de RL, devido ao tamanho do estado de espaços S ; não sendo possível verificar $s, \forall s \in S$ na memória. A quantidade de possibilidades de utilização da capacidade de todos os enlaces e nós de substrato pode tornar-se computacionalmente inviável, além disso, como o estado de espaços incluem variáveis contínuas, ele pode ter estados que jamais irão ocorrer.

Assim a única forma de uma inteligência artificial aprender o problema do VNE é pela generalização das experiências passadas e, de acordo com Wang et al.(2019), a generalização pode ser vista como um problema de aproximação, que seleciona exemplos para construir uma aproximação de toda a função, logo pode ser considerado um problema de aprendizado supervisionado resolvido por uma rede neural utilizando *feed-forwarding*, uma função de aproximação por meio de uma rede neural, observada na Figura 22.

Figura 22 – Função de aproximação por uma rede neural.



Fonte: Autor “adaptado de”,Wang et al. (2019, p. 256).

Wang et al.(2019) desenvolveram um modelo de ranqueamento com uma variação da proposta de ranqueamento de elementos de Gong et al. (2014), o CG_GRC (Algoritmo 06), que é utilizado com um parâmetro L (defino neste trabalho como *nCand* para evitar conflitos com outras denotações deste trabalho) de redução do tamanho do estado de espaço a ser explorado.

Algoritmo 06	GC_GRC – Ranqueamento de Elemento
	Entradas
	$G^S = (N^S, L^S)$: topologia da rede substrato;
	$G^r = (N^r, L^r)$: topologia VNR;
	Saídas
	$Canditatos$
01:	$Candidatos \leftarrow \emptyset$
02:	$NodesSet \leftarrow N^S$ normalizado por GRC (um ranqueamento de interesse)
03:	Para $i \ de\ 0\ a\ nCand$ faça: {nota: nCand é número de candidatos a serem gerados)
04:	$combina \leftarrow \emptyset$
05:	Para cada N_j^r em VNR faça:
06:	Selecione aleatoriamente n^k em $NodesSet$
07:	$NodesSet \leftarrow NodesSet - n^k$
08:	$combinação \leftarrow combinação \cup < n_j^r, n^k >$
09:	fim do ciclo
10:	$Candidatos \leftarrow Candidatos \cup combinação$
11:	fim do ciclo
12:	Retorna $Candidatos$

Fonte: Autor “adaptado de” Wang et al., (2019, p. 256).

O GC_GRC recebe as requisições para ranqueá-las e gera, na linha 04, um grupo de nós de substrato normalizados e ranqueados de acordo com os critérios de importância do elemento. Com uma quantidade de *nCand*, pelos ciclos da linha 05, o GC_GRC busca de forma aleatória uma combinação a cada requisição de um elemento na rede substrato para gerar uma lista de candidatos pré-mapeados que será o insumo para simulações realizadas pelo RL antes de efetivamente serem alocadas na rede substrato.

O VNE-TD, Algoritmo 07, é um aprendizado por reforço por diferença temporal (*temporal-diference*) que aprende diretamente com a experiencia do ambiente sem ser necessário uma modelagem da dinâmica de alocação das VNEs.

Algoritmo 07	VNE-TD – Ranqueamento de Elemento
01:	**Entradas**
02:	$G^S = (N^S, L^S)$: topologia da rede substrato;
03:	$G^r = (N^r, L^r)$: topologia VNR;
04:	**Saídas**
05:	$Alocação\ da\ VNR$
06:	$S_c \leftarrow (N_a^S, L_a^S)$: Obtém o estado do ambiente de acordo com equação (33)
07:	$Candidatos \leftarrow$ Algoritmo 06
08:	$U_{max} \leftarrow -infinito$
09:	$m = -1$
10:	Para cada $combinação$ j em $Candidatos$ faça:
11:	Simule a alocação do VNR_j na G^S (ação)
12:	Se a simulação falhar então:
13:	Continue com o próximo item das combinações
14:	Final do Se
15:	$S_p^j \leftarrow$ estado do ambiente com a ação de alocar a VNR_j na G^S
16:	$U_p^j \leftarrow$ U pela aproximação da rede neural $U_{ANN}(\text{VNR}_j, S_p^j)$
17:	Se $U_p^j > U_{max}$ então:
18:	$U_{max} \leftarrow V_p^j$
19:	$m \leftarrow j$
20:	Final do Se
21:	Final do ciclo
22:	Se $m \neq -1$ então:
23:	Aloque o VNR na G^S de acordo com a ação m da $combinação$ em $Candidatos$
24:	$r \leftarrow$ recompensa (VNR)
25:	$s_n \leftarrow s_n^m$ estado do ambiente após a ação de alocação m
26:	Se não:
27:	$r \leftarrow 0$
28:	$s_n \leftarrow s_c$
29:	Final do Se
30:	Armazena cópia (*backup*) do (S_c, r, S_n)
31:	Em intervalos de tempo treine novamente a rede neural com o backup

Fonte: Autor “adaptado de” Wang et al. (2019, p. 257).

O algoritmo de Wang et al.(2019) realiza simulações com o ambiente através da geração pseudoaleatória das ações candidatas, com o algoritmo 06, e compara o resultado com o de uma função generalizada por uma ANN, na linha 16, que é atualizada constantemente através das experiências com o ambiente, caracterizando-se como um algoritmo de diferença temporal (TD – *temporal-diference*) .

Apesar de o VNE_TD, não deixar claro o momento da alocação dos enlaces, subentende-se que ela ocorre na etapa de simulação, linha 11, e posteriormente na etapa de alocação direta na rede substrato na linha 23. O VNE_TD tem como recompensa a rotina entre as linhas 22 e 29, que retornam 0 se a alocação falhar, ou uma recompensa positiva, caso o agente consiga alocar a VNR.

O algoritmo conclui sua rotina armazenando o valor do estado S_c, o estado de destino S_n e sua recompensa r, para ampliar e atualizar o espaço de estado incrementalmente, e constantemente treina a rede neural artificial com os novos valores utilidade do estado.

Nos resultados de simulações do RL-TD de Wang et al.(2019), o desempenho do VNE_TD foi comparado com os algoritmos Rand-GRC e GRC-VNE através dos indicadores de bloqueio de VNR, receita da VNR, entre outros.

Pode-se observar pelos gráficos apresentados no artigo (WANG et al., 2019) que o VNE_TD tem uma taxa de bloqueio sete pontos percentuais menor que o Rand-GRC, o que indica que o VNE_TD conseguiu, no longo prazo, aceitar mais VNRs que os demais algoritmos, o que é justificado pelo fato de o VNE_TD utilizar estimativas futuras para alocar os recursos no presente, e os demais algoritmos observarem somente o momento da solicitação da VNR e as lógicas heurísticas de alocação.

Da mesma forma que ocorreu uma melhora na taxa de bloqueio de VNRs, é possível observar que a receita do VNE_TD é 10 pontos superior à dos demais algoritmos, fato justificado pela maior taxa de alocação das VNRs.

A principal diferença entre os trabalhos apresentados nesta seção e a proposta deste trabalho é que os algoritmos apresentados aqui propõem somete a visão de otimização uniforme dos recursos da infraestrutura, como se todo o tráfego fosse igual, assim como é a proposta das redes atuais. Eles não consideram as características da rede 5G que prevê um modelo de redes heterogêneas, com requerimentos não regulares definidos pelos 5QI com necessidades distintas e garantias que atendam as camadas de extensão sensorial e autoliberação da pirâmide de Maslow (capítulo 2.1) ou seja, desconsideram o motivo pelo qual o 5G foi desenvolvido.

Este trabalho se dedica a melhorar o desempenho na alocação de recursos em uma rede de 5G com NSaaS através da orquestração por aprendizado por reforço dos algoritmos de alocação de recursos utilizando uma abordagem de coordenação estratégica, dos algoritmos especializados responsáveis por alocar recursos da rede, levando-se em conta as necessidades de uma rede heterogênea que, na pirâmide de Maslow (capítulo 2.1) atende às necessidades de extensão sensorial – uma rede com quadros de dados pequenos – às necessidades de autoliberação – uma rede com baixa latência e autoconfigurável e autogerenciável – e também as necessidades básicas de comunicação e informação: múltiplos tipos distintos de rede, múltiplos fatias de rede, em múltiplos domínios da rede.

4 VNE_CRS - ORQUESTRAÇÃO POR APRENDIZADO POR REFORÇO

Orquestração é definida como a “arte de compor para cada instrumento da orquestra; adaptação de uma melodia ou composição a uma estrutura orquestral; conjunto de ideias, procedimentos etc. que visam à realização de algo; articulação, planejamento.” (HOUAISS, 2009).

Lunenburg (2012), em seu trabalho sobre a estrutura organizacional de Mintzberg, mostra a coordenação como sendo a chave que possibilita uma organização dividir e delegar as tarefas que podem ser, de acordo com o estágio de amadurecimento da organização, por supervisão direta, indireta ou ainda, apenas uma coordenação por direcionamento estratégico, ou seja, coordenar é orquestrar para alcançar um o objetivo estratégico da organização.

Assim como na estrutura de Mintzberg (LUNENBURG, 2012), sistemas de inteligência artificial podem interagir com uma coordenação direta, em que um sistema centralizado toma todas as ações, uma coordenação indireta, em que o sistema supervisiona as ações do agente e influência na condução do resultado, ou ainda com agentes independentes que se alinham com um objetivo comum de atingir uma meta numérica.

Esta sessão versa sobre: a modelagem do ambiente da rede 5G para o uso nos algoritmos VNE, sessão 4.1; e a formulação e descrição do algoritmo de orquestração utilizando RL, o VNE_CRS, descrito na sessão 4.2.

4.1 MODELAGEM E ABSTRAÇÕES DA VNE

Nesta sessão encontram-se a arquitetura e a abstração matemática do ambiente, das ações e da recompensa para o uso no algoritmo de aprendizado por reforço pela orquestração dos algoritmos heurísticos VNE não coordenado e VNE_GRC.

Inicialmente o processo de Markov (MDPs) pode parecer o melhor caminho nos problemas de otimização de recursos. Sua modelagem de recompensas serve para definir políticas ótimas que convergem para maximizar a recompensa total, mas para isto é necessária uma boa modelagem do ambiente e da função de recompensa. A tarefa de orquestrar múltiplos algoritmos é complexa e de difícil modelagem, principalmente nos ambientes em que o próprio estado contém variáveis contínuas e amplia exponencialmente sua dimensionalidade.

Alguns trabalhos, como Fajjari (2011), já utilizaram algoritmos evolutivos, como colônia de formigas, para resolver o problema da VNE. Mas, como já observado por Sutton e Barto (2018), os algoritmos evolutivos, ou genéticos, não levam em conta que as políticas nos

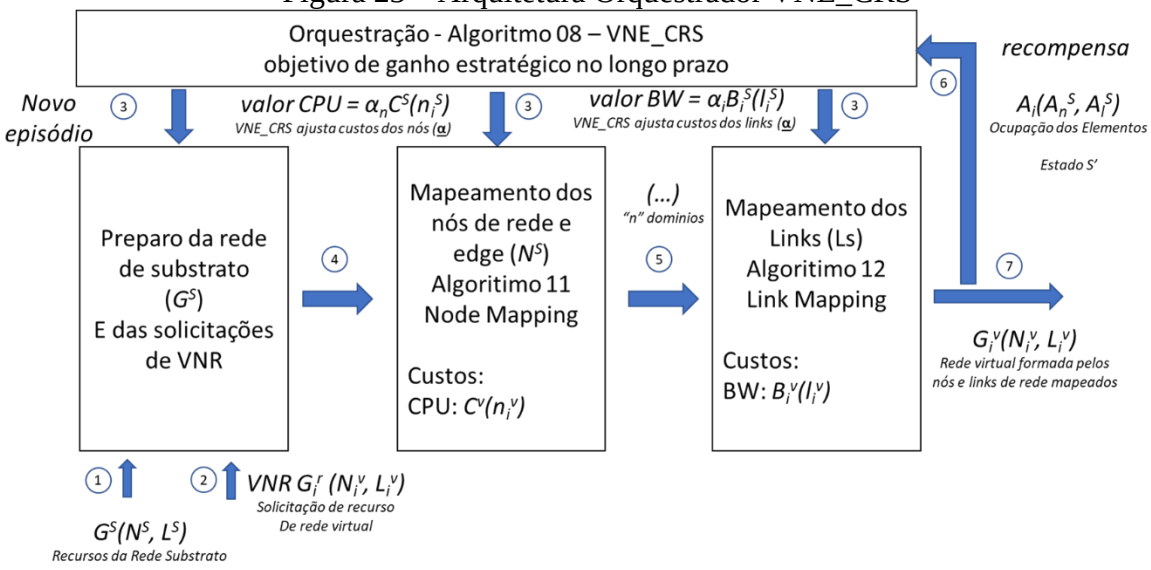
problemas de otimização são uma relação entre o estado atual e as ações a serem tomadas para maximizar os ganhos. Esses algoritmos também não levam em consideração os ganhos em aprender pela percepção e interação com o ambiente, o que pode ser uma boa opção para problemas em que não se conhece nada sobre o ambiente e o tamanho do espaço de estados é pequeno. Porém, no caso do VNE, além de o espaço de estados poder crescer exponencialmente, temos conhecimento sobre o ambiente, e temos como interagir e perceber o ambiente, de tal maneira que algoritmos evolutivos não são a melhor opção.

"Em muitos domínios complexos, a aprendizagem por reforço é o único caminho possível para treinar um programa com desempenho de alto nível" (RUSSELL e NORVIG, 2010, p. 846). A tarefa da aprendizagem por reforço consiste em usar recompensas observadas para aprender uma política ótima (ou quase ótima) para o ambiente.

Assim, pela bibliografia, observa-se que neste trabalho RL é uma opção adequada para orquestrar e tratar o problema da VNE, como também já mencionado por Wang et al.(2019) p, 255: "A natureza do objetivo de longo prazo e da tomada de decisão *on-line* do problema VNE forma um ambiente perfeito para o envolvimento de RL".

A proposta desta pesquisa foi desenvolver um modelo de RL que se integre aos algoritmos especializados em cada domínio da infraestrutura de forma a influenciar no custo dos elementos dessa infraestrutura – item mais comumente utilizado entre esses algoritmos como fator de escolha de elemento – para ranquear esses elementos e alocá-los às solicitações VNR, seguindo a arquitetura abaixo:

Figura 23 – Arquitetura Orquestrador VNE_CRS



Fonte: Autor

Para a proposta deste trabalho foi necessário desenvolver as abstrações do **estado do ambiente** da VNE**,** ou simplesmente **ambiente,** as **ações** de interações entre o agente e o ambiente (algoritmos de VNE + rede substrato), uma **função de recompensa** para direcionar as ações do agente de RL e a relação de sensoriamento com este ambiente.

O **ambiente** (S – *State*) do VNE_CRS é mapeado por um vetor de disponibilidade de cada elemento da rede substrato, equação (34), seguindo uma taxa de disponibilidade dos elementos, de acordo com a equação (35), baseada na proposta de Wang et al.(2019), que usa a relação de estados da capacidade disponível pela equação (33), confluída com a proposta de estratificação de Mijumbi et al. (2014).

$$S_t = \{Ds(n_1), \dots, Ds(n_n), \dots, Ds(l_0), \dots, Ds(l_n)\} \tag{34}$$

$$Disponibilidade_Ds(X) = \begin{cases} se\ \dfrac{d_a^{S^t}}{c_{max}^t} \leq 0{,}3 \quad , \quad 0 \\ se\ \dfrac{d_a^{S^t}}{c_{max}^t} > 0{,}8 \quad , \quad 2 \\ caso\ contrário \quad , \quad 1 \end{cases} \tag{35}$$

onde $d_a^{S^t}$ representa o $n_a^{S^t}$ para os nós de substrato ou o $l_a^{S^t}$ para os enlaces, sendo o $n_a^{S^t}$ a capacidade disponível do nó de substrato n^S no instante t, definida pela equação (36), o $l_a^{S^t}$ a capacidade disponível do enlace l^S no instante t, definida pela equação (37), e o c_{max}^t e o l_{max}^t são as capacidades totais remanescentes de nós de substrato e enlaces que ponderam o estado de ambiente.

$$n_a^S = n_t^S - \sum_{r \in G^r} \sum_{n \in N^r} \begin{cases} se\ \mu_{n^r}^{n^S} = 1, \quad n_{rCPU} \\ se\ não, \qquad 0 \end{cases} \tag{36}$$

$$l_{(i,j)_a}^S = l_{(i,j)_t}^S - \sum_{r \in G^r} \sum_{l_{(w,u)} \in L^r} \begin{cases} se\ \mu_{l_{(w,u)}}^{l^S} = 1, \quad l_{rBW} \\ se\ não, \qquad 0 \end{cases} \tag{37}$$

onde $\mu_{n^r}^{n^S}$ e $\mu_{l_{(w,u)}}^{l^S}$ representam se o elemento está mapeado ou não para uma VNR, n_t^S e $l_{(i,j)_t}^S$.

Para reduzir o tamanho do espaço de estados, foram utilizadas três classes de estratificação da disponibilidade de nós de substrato e enlaces definidas pelo percentual de

disponibilidade de recursos, Tabela 6, formando uma quantidade de 3^n estados, com n sendo o número de elementos, o que é uma relação exponencial que limita a quantidade de elementos da rede a serem tratados, mas reduz o tamanho do estado de espaços quando comparado com uma relação de uma variável linear mencionada na subseção 2.5.2.1. Para redes maiores, pode ser utilizada uma estratégia de federação, como sugerido por Wei (2012), que é uma estratégia convencional das operadoras de infraestrutura de telecomunicações.

Tabela 6 – Estratificação de Estados de N e L

%Disponibilidade		Classe
0,0%	30,0%	0
30,0%	80,0%	1
80,0%	100,0%	2

Fonte: Autor

Em RL um dos principais fatores de sucesso é a definição da **Função de Recompensa** (r) pois um agente que não tem estímulo não realiza a ação. Neste trabalho o objetivo é realizar um equilíbrio entre os recursos disponíveis na infraestrutura e as necessidades de uma rede especializada do 5G. De acordo com a proposta da pesquisa, a equação (38) – equação de recompensa ou reforço – busca uma relação entre a alocação exitosa da VNR, que é o principal ponto de avaliação dos algoritmos de VNE, em conjunto com sua receita.

$$r = \begin{cases} se\ o\ VNR\ foi\ alocado, & 2^{tp5QI} \cdot receita \\ se\ não, & -1 \end{cases} \quad (38)$$

Essa função de recompensa (r) retorna valores positivos somente se o agente conseguir alocar os recursos solicitados. Caso contrário, ela retorna um valor negativo para ensinar ao agente que o caminho tomado não traz retornos exitosos e para influenciar o agente em decisões futuras, nas quais ele se encontre no mesmo estado do ambiente.

Na equação (38) foi adicionada uma relação diretamente proporcional à receita para induzir o agente a querer alocar as requisições que possibilitem uma receita maior, com o objetivo de atender os requisitos de qualidade apontados na seção 2.2 deste trabalho. Existe também uma relação de recompensa que inclui os 5QI das redes 5G, o tp5QI (Tabela 8), inserida em uma forma exponencial com pesos maiores às solicitações de maior criticidade.

Para a modelagem da **função da receita** com a alocação da infraestrutura, representada na equação (39), é considerada a somatória da receita em todos os domínios da infraestrutura,

sendo, no caso deste trabalho, o domínio da infraestrutura computacional e da infraestrutura de enlaces.

$$receita = \sum_{r \in VNR} \rho \cdot N_r^v + \partial \cdot L_r^v \tag{39}$$

onde ρ e ∂ são fatores para uma relação de valores entre CPU e BW, sendo N_r^v e L_r^v respectivamente os valores CPU e do enlace (*link*) da requisição de VNR alocada com sucesso, dada por:

$$N_r^v = \sum_{r \in G^r} \sum_{n \in N^r} \begin{cases} se\ \mu_{n^r}^{n^S} = 1, \quad n_{rCPU} \cdot receita_CPU \\ se\ não, \qquad 0 \end{cases} \tag{40}$$

onde $\mu_{n^r}^{n^S} = 1$ são as requisições de nós virtuais mapeadas com sucesso e $receita_CPU$ é o valor cobrado por cada unidade de CPU da requisição, e a alocação do enlace dada por:

$$L_r^v = \sum_{r \in G^r} \sum_{l_{(w,u)} \in L^r} \begin{cases} se\ contagem\ \mu_{l_{(w,u)}}^{l^S} > 1, \quad l_{rBW} \cdot receita_BW \\ se\ não, \qquad 0 \end{cases} \tag{41}$$

onde $\mu_{l_{(w,u)}}^{l^S} = 1$ são as requisições de enlaces mapeadas com sucesso e $receita_BW$ é o valor cobrado por cada unidade de BW da requisição.

O modelo de enlace se distingue da visão por CPU, que tem uma relação direta entre utilização e receita, porque uma requisição de enlace lógico pode necessitar de múltiplos enlaces físicos na rede substrato, mas, como receita, considera-se somente a requisição.

Já a modelagem do custo é uma relação inversamente proporcional à disponibilidade. O objetivo dessa modelagem é fazer com que os elementos proporcionalmente mais ocupados gerem uma percepção de um custo superior, que reflete o comportamento regionalizado de procura e demanda dos provedores de infraestrutura, para priorizar a alocação dos recursos em locais com maior disponibilidade e, com isso, balancear o tráfego na rede, aumentar a disponibilidade da infraestrutura e evitar investimentos precoces.

Sendo o custo o axioma responsável por alhanar as ações dos agentes heurísticos, as **equações de custo** (42), (43), (44) e (45) foram utilizadas para avaliar a eficiência dos

algoritmos básicos VNE não coordenado e VNE_GRC, tendo essas equações um índice de estratégia de custo α com o intuito de influenciar na alocação fim a fim do modelo das VNE.

O primeiro par de equações de custo, equação (42) e equação (43), para os nós e os enlaces respectivamente, é denominado como equações de custo lineares que consideram o custo unitário da CPU e a somatória do custo unitário da largura de banda consumida nos diversos enlaces no caminho da interconexão entre os nós de substrato :

Equação linear de custo de CPU

$$N_c^v = \sum_{r \in G^r} \sum_{n \in N^r} \begin{cases} se\ n_a^S \geq n_r^v,\ \ \forall\, n_r^v \nexists\, \mu^{n^S}, & n_{rCPU} \cdot \alpha_{ns} \\ se\ não, & 0 \end{cases} \tag{42}$$

Equação linear de custo de BW no enlace

$$l_{(i,j)_c}^v = \sum_{r \in G^r} \sum_{l_{(w,u)} \in L^r} \begin{cases} se\ l_{(i,j)_a}^S \geq l_{r_{(w,u)}}^v, \forall\, l_{r(w,u)}^v \nexists\, \mu^{l_{(w,u)}^S}, l_{rBW} \\ se\ não, \quad 0 \end{cases} \tag{43}$$

O segundo par de equações de custo, uma proposta deste trabalho, equação (44) e equação (45), para os nós de substrato e os enlaces respectivamente, é denominado como equações pela razão da disponibilidade logarítmica.

Essas equações têm o objetivo de equilibrar a relação entre o aumento de custo com a receita produzida pela alocação do recurso através da multiplicação do custo (n_{rCPU} ou l_{rBW}) pelo logaritmo do inverso da percentual da disponibilidade ($\ln 1/n_a$ ou $\ln 1/l_a$), ou seja, quanto maior o percentual da disponibilidade de recursos (n_a ou l_a), menor será o fator que que será multiplicado pelo custo (n_{rCPU} ou l_{rCPU}), sendo logarítmicas para suavizar a relação de custo versus disponibilidade nos casos em que a disponibilidade do recurso *a* seja muito baixa:
Equação de custo pela razão da disponibilidade logarítmica de CPU

$$N_c^v = \sum_{r \in G^r} \sum_{n \in N^r} \begin{cases} if\ n_a^S \geq n_r^v,\ \ \forall\, n_r^v \nexists\, \mu^{n^S}, & n_{rCPU} \cdot \alpha_{ns} \cdot \ln \dfrac{1}{n_a + \beta} \\ otherwise, & 0 \end{cases} \tag{44}$$

Equação de custo pela razão da disponibilidade logarítmica de BW no enlace

$$l_{(i,j)_c}^v = \sum_{r \in G^r} \sum_{l_{(w,u)} \in L^r} \begin{cases} if\ l_{(i,j)_a}^S \geq l_{r_{(w,u)}}^v, \forall\, l_{r(w,u)}^v \nexists\, \mu^{l_{(w,u)}^S}, l_{rBW} \cdot \alpha_{ls} \cdot \ln \frac{1}{l_{a(i,j)} + \beta} \\ otherwise, \quad 0 \end{cases} \tag{45}$$

onde $n_a = n_a^{S^t}/n_{max}^t$ e $l_a = l_a^{S^t}/l_{max}^t$, representando respectivamente o percentual da capacidade disponível do nó de substrato e do enlace de rede, n_r^v e l_r^v a requisição da VNR, β uma constante pequena para não zerar a equação, $\mu_{n^r}^{n^S}$ e $\mu_{l_{(w,u)}}^{l^S}$ representam se o elemento está mapeado ou não para uma VNR e α_{ns} e α_{ls} fatores de estratégia de custo que serão utilizados para tomada de ação, sendo que para o cálculo do custo real estes valores são considerados como valor unitário igual a 1 (um).

Nas **possíveis ações**, último item da modelagem, o agente pode inflacionar ou deflacionar o custo da alocação variando o parâmetro α das equações de custo, a cada elemento (*n* ou *l*), de forma a influenciar na decisão dos algoritmos heurísticos VNE não coordenado e GRC-VNE, seguindo a Tabela 7 de ações.

Tabela 7 – Ações de Estratégia de Custo

Tipo	α
0	-25%
1	0
2	25%
3	infinito

Fonte: Autor

No longo prazo, o objetivo do agente RL é maximizar o resultado da equação (46), a taxa de aceitabilidade das requisições de rede virtual, que é a métrica mais presente nos artigos utilizados como base deste estudo como: Zhu e Ammar (2006), Cheng et al. (2011), Chowdhury, Rahman e Boutaba (2011) , Mijumbi et al.(2014) Jmila (2015), Wang et al.(2019) e outros relacionados na Tabela 4 – Estado da Arte VNE.

$$taxa\ de\ aceitaçao\ de\ VNE = \frac{Total\ de\ requisições\ aceitas}{Total\ de\ requisições} \quad (46)$$

4.2 O ALGORITMO DE ORQUESTRAÇÃO – VNE_CRS

Inspirado nos ganhos obtidos com as metodologias de gestão por coordenação estratégica das empresas (LUNENBURG, 2012), o VNE_CRS, Algoritmo 08, propõe uma abordagem para coordenar algoritmos especializados em seus domínios.

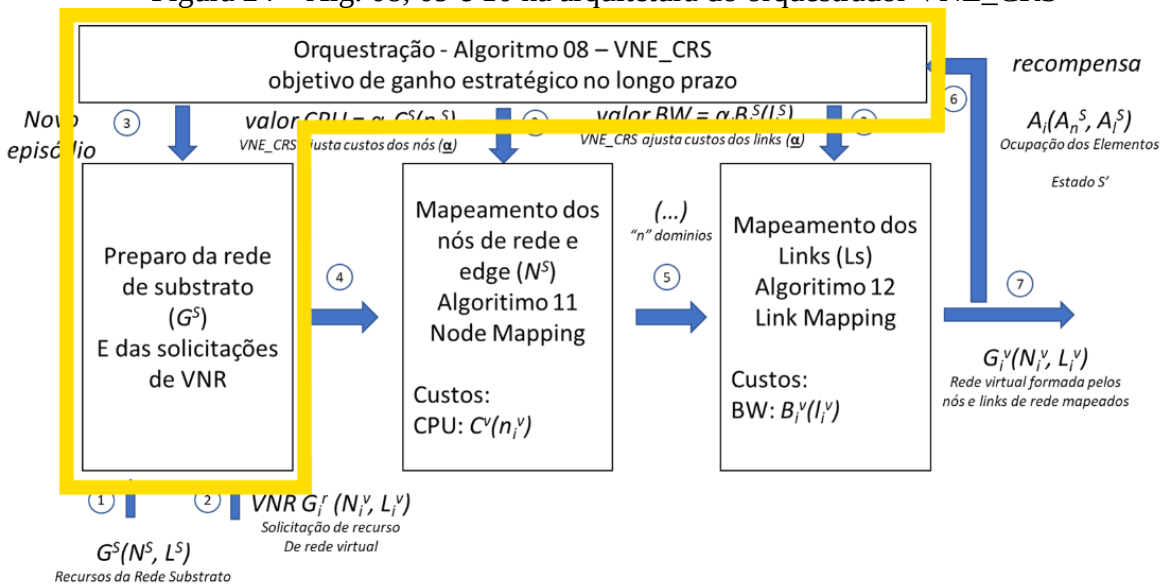
Para influenciar na alocação de recursos da infraestrutura, e com isso orquestrar o desempenho dos algoritmos envolvidos, o VNE_CRS (Figura 24) utiliza a coordenação indireta pela supervisão do resultado geral, atuando na estratégia de custos dos recursos da infraestrutura G^s, inflando ou reduzindo esses custos, com intuito de alterar a conduta de cada algoritmo especializado durante o processo de alocação de recursos da infraestrutura.

A abordagem de alterar a conduta de cada algoritmo especializado difere daquelas apresentadas no capítulo 3, por propor reutilizar todo o conhecimento de cada algoritmo especializado, visto que muitos estudos foram realizados para melhorar o desempenho desses algoritmos de domínio, e difere das outras abordagens apresentadas, por induzir a alocação via um índice de custos, ao invés de alocar diretamente os recursos.

O VNE_CRS lida com os conflitos de interesse que cada algoritmo de domínio gera ao buscar de forma gulosa o melhor desempenho somente para os recursos de seu domínio, ele orquestrando o resultado de forma a obter a maior recompensa no longo prazo, equivalente aos objetivos estratégicos das organizações de Mintzberg (LUNENBURG, 2012), através da influência pela coordenação das ações de estratégia de custo dos recursos, antes da atuação de cada algoritmo de domínio, utilizando a previsão dos ganhos acumulados no futuro, após a atuação de todos os algoritmos especializados.

Ou seja, o VNE_CRS tem uma visão fim a fim de toda caleia de alocação de recursos, uma característica dos RL apresentada na seção 2.5, e utiliza-se dessa característica para balizar as decisões dos algoritmos coordenados através de suas ações, que visam os resultados a longo prazo.

Figura 24 – Alg. 08, 09 e 10 na arquitetura do orquestrador VNE_CRS



Fonte: Autor

Algoritmo 08 VNE_CRS – Abordagem *on-line*

Entradas

$G^S = (N^S, L^S)$ – rede substrato
$M:G^V$– mapeamento da rede virtual
S_t – espaço de estado do ambiente (quando conhecido)
Q^π – valor utilidade do ambiente (quando conhecido)
π – política de ações (quando conhecida)
G^r – requerimentos da infraestrutura em distribuição de poisson

Saídas

$M:G^V$, Q^π, π e S_t

01: **Inicializa** G^S, $M:G^V$, Q^π, π e S_t utilizando os valores de entrada
02: **repita**:
03: **se** existe uma G^v com ciclo de vida no fim **então**:
04: Libere os recursos alocados para a G^v na G^s
05: **final do se**
06: VNR ← próxima requisição na G^r
07: Q^π, π ← simulação_&_atualização_política (Algoritmo 09)
08: execute uma ação de estratégia de custo *a* no estado *st* usando π
09: tmp_status, tmp_mapping ← vazio
10: **para cada** domínio da G^S **faça o ciclo**:
11: tmp_status, tmp_mapping ← Algoritmo do domínio (11, 12)
12: **se** tmp_status == Falha **então**:
13: marque a VNR como Falha
14: tmp_mapping ← vazio
15: **interromper o ciclo**
16: **final do Se**
17: **final do ciclo**
18: **se** tmp_mapping não está vazio **então**:
19: aloque os recursos de tmp_mapping na G^S e mapeie a $M:G^v$
20: **final do se**
21: retorne os valores de custo dos recursos para o normal
22: armazena cópia (*backup*) do $M:G^V$, Q^π, π and S_t
23: **enquanto** existirem VNR não mapeadas na G^r (N^r, L^r)
24: **retorne** ($M:G^V$, Q^π, π , S_t)

O método de alocação dos recursos no VNE_CRS ocorre em duas etapas: na primeira etapa o algoritmo realiza uma simulação do resultado obtido através de sua política π, avalia os resultados, atualiza sua política π, na linha 07 do Algoritmo 08, e, antes da execução dos algoritmos de domínio, o VNE_CRS executa sua Ação de influência dos algoritmos de domínio via ações de estratégia de custo dos recursos na linha 08 do Algoritmo 08; na segunda etapa o VNE_CRS verifica a possibilidade de alocar os recursos no Algoritmo 08 entre as linhas 10 e 17, e caso, existam os recursos necessários em todos os dominós da requisição, verificados na linha 18 do Algoritmo 08, ele inicializa a alocação de recursos através da linha 19 e, na linha 21, o VNE_CRS retorna os valores originais de custos dos recursos sem sua atuação. Se em

qualquer domínio o requisito da infraestrutura não puder ser atendido, o VNE_CRS marca a VNR como bloqueada, linha 13 do Algoritmo 08, para identificar uma falha na alocação e não alocar nenhum recurso em nenhum dos domínios.

Sendo agnóstico quanto aos algoritmos especializados a serem utilizados, pois coordena as ações do algoritmo de domínio por meio de uma estratégia de valores de custos de recursos, o VNE_CRS permite utilizar qualquer algoritmo de domínio, desde que tenha um parâmetro de custo dos recursos que possa influenciar esse algoritmo.

Por ser um algoritmo de aprendizado por reforço por diferença temporal (TD), o VNE_CRS aprende diretamente pela experiência com ambiente, sem a necessidade de uma modelagem completa da dinâmica de alocação de VNEs. Esse aprendizado pela experimentação pode ser visto na geração e atualização da política na linha 07 do Algoritmo 08, que ocorre a cada ciclo entre as linhas 02 e 23 do Algoritmo 08, enquanto existirem requisições VNR.

É importante observar que a primeira ação do Algoritmo 08 é de verificar, pela linha 03, se existe algum recurso virtualizado a ser liberado na G^s para liberá-lo antes de verificar os próximos requerimentos. Essa ação torna-se relevante porque uma requisição, como descrito na sessão 5.1, pode perdurar ativa por até 5 minutos, mantendo retida a infraestrutura alocada.

Uma característica do aprendizado por reforço RL é a possibilidade de realizar simulações no ambiente para ampliar o conhecimento sobre este mesmo ambiente e atualizar de suas políticas de atuação. Assim, o VNE_CRS, no Algoritmo 08 na linha 07, realiza, através do Algoritmo 09 na linha 03, as simulações com o ambiente por meio de uma política π de ações de estratégia de custo *A*, definidas na Tabela 7, avalia o resultado utilizando uma função de recompensa generalizada pela equação (38), no Algoritmo 09 linha 04 e linha 05, atualiza sua política, através do Algoritmo 09 linha 12, com a ação que retorna o maior valor utilidade acumulado no estado $Q^{\pi}(s)$, e armazena uma cópia *backup*, através Algoritmo 08 na linha 22, que permite o VNE_CRS explorar o ambiente e ampliar seu conhecimento a cada interação.

O CRS_VNE, pode ser tratado em duas abordagens: totalmente *on-line*, pelo algoritmo 08, em que o algoritmo aprende tudo sobre o ambiente VNE durante as interações de alocação de recursos e, com base nos conhecimentos adquiridos, escolhe as ações de estratégia de custo, cria e atualiza a política de RL; ou, uma vez que o uso da infraestrutura de telecomunicações é, de certa forma, previsível (ERLANG, 1909), pode ser treinado o com informações de conhecimento prévios, *off-line*, representado no Algoritmo 10, para gerar um espaço de estado de ambiente de conhecimento básico S_t, valores de utilidade de ambiente Q^{π} e a política π a ser usada na abordagem *on-line* do algoritmo durante o processo de inicialização.

Algoritmo 09 simulação_&_atualização_política

Entradas

$G^S = (N^S, L^S)$ – rede substrato
VNR – requerimento de rede virtual
S_t – espaço de estado do ambiente
s_t –estado do ambiente
Q^π – valor utilidade do ambiente
π – política de ações

Saídas

Q^π, π e S_t

01: **Inicializa** $s_1 \leftarrow st$
02: escolha uma ação de estratégia de custo *a* partindo do estado *st* usando π
03: simule a ação *a* e a alocação VNE nos domínios da infraestrutura
04: observe os resultados da simulação, recompensa *r* e novo estado s_2
05: **se** s_1 for diferente de s_2 e s_2 não se encontra em S_t **então**
06: adicione o novo estado s_2 em S_t
07: $U^\pi(s_2) \leftarrow 0$
08: $\pi(s_2) \leftarrow 0$
09: **final do se**
10: append_temporal $\leftarrow r + \gamma \cdot \max_{a2} Q(s_2, a_2) - Q(s_1, a)$
11: $Q(s_1, a) \leftarrow Q(s_1, a) + \alpha \cdot$ append_temporal
12: atualize $\pi \leftarrow max_{a \in A(s)} Q^\pi(s), \forall(s)$
13: **retorne** (S_t, Q, π)

Através da abordagem *off-line* o VNE_CRS executa todas as requisições simuladas da base de treinamento, Algoritmo 10 entre as linhas 06 e 08, e, ao final de cada episódio, ele atualiza o valor utilidade Q^π que é confrontado com o seu valor anterior ao ciclo de treinamento, com o objetivo de verificar se o aprendizado estabilizou ou alcançou um erro aceitável, linha 11 Algoritmo 10, para, com isso, armazenar a política e estado de espaços aprendidos, que serão utilizados para determinação de ações durante a abordagem *on-line*.

Na abordagem mista, *off-line* + *on-line*, é utilizada uma base de dados com requisições simuladas para a criação e treinamento da política de atuação π, a criação da matriz de valor utilidade Q^π e formação da base de estados de espaços do ambiente S_t; que são inicializados com valores nulos. Essa abordagem faz chamadas ao Algoritmo 08, através do Algoritmo 10 na linha 07, para realizar as interações e simulações com a base de dados a cada ciclo de treinamento, denominado episódio, no Algoritmo 10 entre as linhas 02 e 11.

Algoritmo 10 VNE_CRS – Abordagem pré-treinamento *off-line*

Entradas

$G^S = (N^S, L^S)$ – rede substrato

G^r – conjunto de requerimentos (VNRs) para treinamento

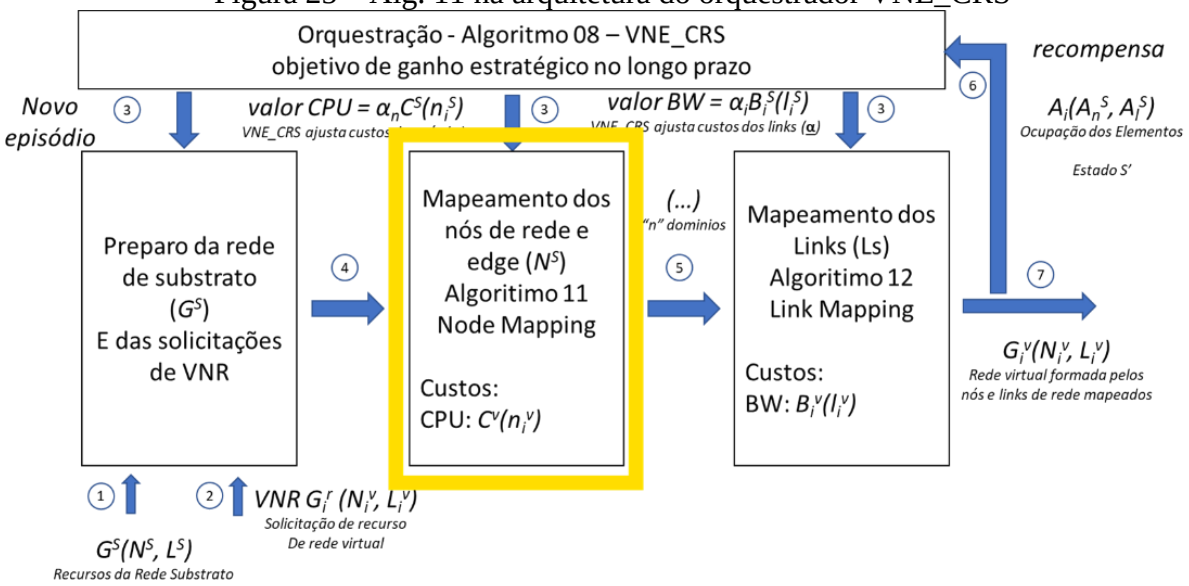
Saídas

Q^π, π e S_t

01: **Inicializa** G^S com os valores da entrada e Q^π, π and S_t com vazio
02: **repita para cada episódio**
03: $M:G^V \leftarrow$ vazio
04: libere todos os recursos da G^S
05: $BaseQ^\pi \leftarrow Q^\pi$
06: **para cada** VNR na Gr **faça o ciclo**
07: $M:G^V, Q^\pi, \pi, S_t \leftarrow$ Algoritmo 08 (VNE_CRS)
08: **final do ciclo**
09: $deltaQ \leftarrow Q^\pi - BaseQ^\pi$
10: $BaseQ^\pi \leftarrow Q^\pi$
11: **enquanto** deltaQ < acceptable error
12: armazena cópia (backup) do $M:G^V$, Q^π, π and S_t
13: **execute os próximos requerimentos utilizando a base de conhecimento**

O Algoritmo 11, apresentado na Figura 25, utiliza o princípio dos "meta-nós" de Chowdhury, Rahman e Boutaba (2011). A cada VNR recebida pelo Algoritmo 11, ele amplia o grafo da rede substrato com a criação dos "meta-nós", na linha 03 do Algoritmo 11, que representam a agregação de nós virtuais, ordenados de forma decrescente por uma das funções de custo das equações (42) e (44), considerando a restrição de latência entre os nós computacionais permitida pela VNR.

Figura 25 – Alg. 11 na arquitetura do orquestrador VNE_CRS



Fonte: Autor

Algoritmo 11 Mapeamento de nós

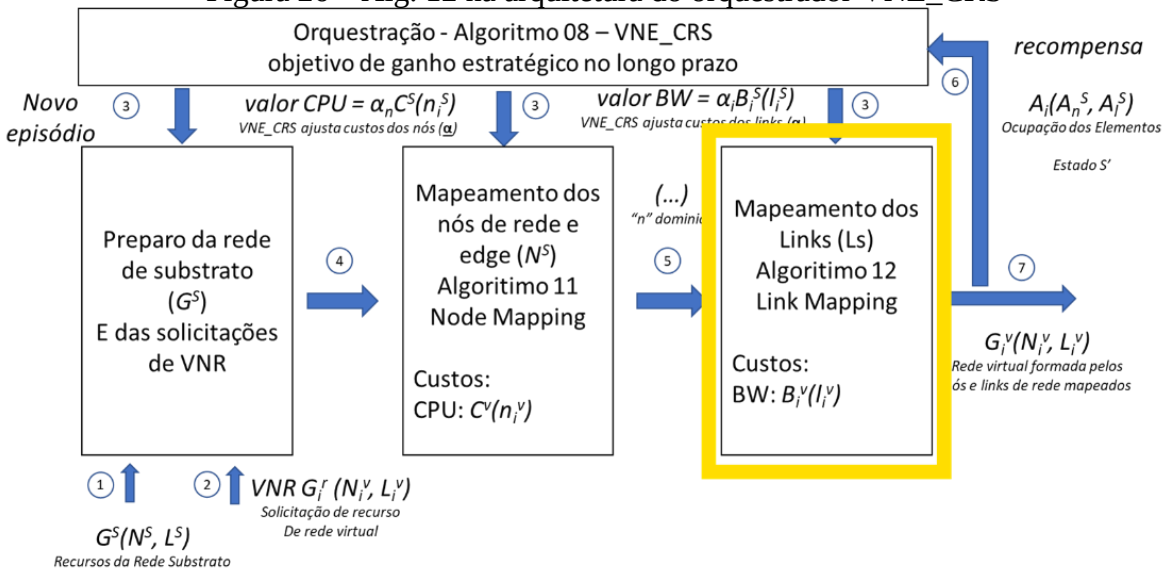
```
                Entradas
                        G^S = (N^S, L^S) – rede substrato
                        VNR – requerimento de rede virtual
                        S_t – espaço de estado conhecido
                Saídas
                        node mapping status (tmp_status),  map_vector (M:n^V)
01:             Inicializa tmp_status ← Falso
02:             repita para cada n^r não mapeado no VNR faça o cliclo
03:                     gerar uma lista de m_nós N^S (calc o custo de n^s com a Eq.(44))
04:                     para cada m_node, em  ordem decrescente de custo faça o ciclo
05:                             se as restrições das eq. (6), (7) e da VNR são verdadeiras:
06:                                     M:n^v  ←  m_node(n^r)
07:                                     tmp_status ← Verdadeiro
08:                                     interromper o ciclo
09:                             final do se
10:                             tmp_status ←  Falso
11:                     final do ciclo
12:             final do ciclo
13:             retorna tmp_status,  M:n^v
```

A busca pelos nós computacionais ocorrem no ciclo entre as linhas 04 e 11 do Algoritmo 11 e o mapeamento ($M:n^v$), linha 06 do Algoritmo 11, considera a restrição de quantidade de recursos disponíveis para atender a VNR e que um nó virtual não pode ser dividido em dois nós de substrato, representada pelas equações (6) e (7), sendo que o ciclo de busca pelos nós ocorre até que seja encontrada uma opção de nó da rede substrato válida, quando o elemento é mapeado e o ciclo de busca interrompido, linha 08 Algoritmo 11, ou, caso não encontre um nó de substrato que atenda à necessidade da VNR, o processo é finalizado enviando a informação de alocação não exitosa, falso, na linha 10 do Algoritmo 11.

Já para o mapeamento do domínio de enlaces é utilizado um algoritmo de busca do k-ésimo melhor caminho, representado pelo Algoritmo 12, considerando as restrições de latência e banda requeridas pela VNR.

O algoritmo 10, apresentado na Figura 26, busca o melhor caminho de acordo com a equação de custo dos enlaces no caminho (L:03), considera as restrições (5QI) de latência e banda requeridas pela VNR e ordena de forma decrescente de custo (L:03-04) e, caso atenda às necessidades (L:05), ele retorna um ponteiro de mapeamento (L:06-08). Caso contrário o algoritmo seleciona o próximo k-caminho até encontrar um que atenda a VNR (L:02-12). Se não encontrar o recurso o algoritmo marca como indisponível (L:10) a solicitação.

Figura 26 – Alg. 12 na arquitetura do orquestrador VNE_CRS



Fonte: Autor

Algoritmo 12	Mapeamento de enlaces
	Entradas
	$G^S = (N^S, L^S)$ – rede substrato
	VNR – requerimento de rede virtual
	S_t – espaço de estado conhecido
	Saídas
	mapping status (tmp_status), map_vector ($M:n^V, l^V$)
01:	**Inicializa** tmp_status ← Falso
02:	**repita para cada** l^r não mapeado no VNR **faça o cliclo**
03:	gerar uma lista de camihos P^S (calc custo de p^s com a Eq. (45))
04:	**para cada** P na lista, em ordem decrescente de custo **faça o ciclo**
05:	**se** as restrições das eq. (9), (10) e da VNR são verdadeiras:
06:	$M:n^v, l^v$ ← path_links(p^r)
07:	tmp_status ← Verdadeiro
08:	**interromper o ciclo**
09:	**final do se**
10:	tmp_status ← Falso
11:	**final do ciclo**
12:	**final do ciclo**
13:	**retorna** tmp_status, $M:n^v, l^v$

5 METODOLOGIA DE EXPERIMENTAÇÃO

Neste capítulo é apresentada a metodologia de experimentação utilizada neste trabalho, que inclui a descrição da base de dados utilizada nas simulações, a arquitetura da topologia e as aplicações utilizadas nas simulações dos algoritmos de VNE e os critérios de medição e comparação de desempenho entre os algoritmos utilizados.

5.1 BASE DE DADOS

Foi utilizada a base de dados da SNDlib (ORLOWSKI et al. 2010), uma biblioteca com dados reais de utilização de rede, usando a topologia e dados da rede alemã *Brain*, por ser a rede que tem a menor granularidade de dados para treinamento do RL disponível.

Como a tecnologia 5G está em desenvolvimento e ainda não existem casos reais de tráfego utilizando a possibilidade de *slicing* de rede heterogêneas, o que não diminui a importância de serem realizados ensaios com base em previsões de consultorias e perfil de modelagem de tráfego, foi realizada uma projeção baseada na SNDlib dos perfis de dados de acordo com as referências do APÊNDICE C – TABELA DE PADRONIZAÇÃO 5QI, extraído da norma 3GPP TS 23.501 V16.5.1, 2018, sendo considerados para este trabalho os três perfis listados na Tabela 8.

Tabela 8 – QoS x Slices

Slice	**Tip5QI**	**5QI**	**Mbps**	**Distribuição**	**Latência [ms]**
Consumidor	0	2	20,4	55,7%	120
Empresarial	1	60	28	39,1%	60
5GIoT	2	82	128	5,2%	10

Fonte: Autor.

A distribuição de tráfego entre os *slices* de rede foi realizada com a projeção da consultoria *iot-analytics research* (2018); os dados de latência foram baseados na classificação 5QI caracterizados por uma camada de uso convencional de internet, uma de aplicações empresariais e uma de serviços críticos.

O *throughput* do *slice* de consumidor foi baseado nas informações da consultoria STRATEGY (2020), e o *throughput* dos demais *slices*, calculado utilizando o tamanho do quadro de dados *JumboFrame* (9216) e um quadro de dados de 512 para representar as

aplicações 5GIoT, com a equação (47), do atraso na propagação de dados, sendo a relação de latência = tamanho do frame / *throughput*:

$$U = \frac{1}{\left(1 + \frac{2BLe}{cF}\right)} \tag{47}$$

Fonte: Tanenbaum (2003, p.222)

onde:

U: é a utilização ou eficiência do uso do meio de transmissão

B: é a banda em kb/s

c: é a velocidade da luz

L: é o comprimento em metros do meio de transmissão

F: é o tamanho do quadro de dados em bits

e: exponencial de 1 = 2,71828182845905

Tabela 9 – Relação de Velocidade x Latência x quadro

Utilização Máxima	Latência [ms]	Throughput Kb/s	Distância	Vel.Luz	Quadro	
					Cabec	MSS
100,00%	40000000	0,013	1000	199.861.639	24	488
100,00%	4000000	0,128	1000	199.861.639	24	488
99,99%	400000	1,280	1000	199.861.639	24	488
99,93%	40000	12,800	1000	199.861.639	24	488
99,32%	4000	128,000	1000	199.861.639	24	488
93,63%	400	1.280,000	1000	199.861.639	24	488
59,52%	40	12.800,000	1000	199.861.639	24	488
12,82%	4	128.000,000	1000	199.861.639	24	488
1,45%	0,4	1.280.000,000	1000	199.861.639	24	488

Fonte: Autor

A escolha do *throughput* para as aplicações críticas, 5QI 82, foi realizada considerando-se a tabela da taxa de utilidade do meio de transmissão, Tabela 9, somada à necessidade de re-roteamentos que acrescentam latência na transmissão, sendo essa latência definida neste trabalho como 2 ms por salto (*holp*) baseada no tempo de tratamento dos pares (transmissão e recepção) de transceptores em cada salto.

5.2 SIMULAÇÃO

A implementação de soluções de pesquisa em redes públicas reais é economicamente inviável e todos os artigos referenciados aqui neste trabalho realizam simulações. Alguns trabalhos utilizam ambientes totalmente aleatórios, e outros têm como base de dados as informações de redes reais. Assim, neste trabalho, foi implementado em ambiente de simulação orientado a eventos com *Python* pela biblioteca *networkx*, com licença de uso de programa pela distribuição de Berkeley (BSD – *Berkeley Software Distribution*), utilizada nas pesquisas de redes, usando dados da rede *Brain*.

A rede substrato tem 161 nós e 166 enlaces, a topologia da rede *Brain*, o número de CPUs por VNs seguiu uma distribuição normal entre 0.01 a 0.20 CPUs, e a capacidade dos links, de acordo com os dados reais da base de dados.

Como não foi informada a capacidade de CPU na base de dados, foi utilizado, para os nós de substrato secundários, o servidor padrão de datacenter, Figura 27, PowerEdge R640 com Intel® Xeon® Silver, com 12 Núcleos, 16,5 MB de Cache, DDR4-2400, 32GB RDIMM, Dual e 4 interfaces de rede de 10 GbE, cascateados com 2 servidores por nó da rede substrato.

Figura 27 – Servidor Rack PowerEdge R640

Fonte: Dell.

Para os nós de substrato primários, centrais, aqueles que na rede *Brain* são nomeados sem número, foi utilizado o servidor padrão de datacenter, Figura 28, E900h-4 com Huawei® Kunpeng® 920, 64 Núcleos, descrição abaixo.

Figura 28 – Servidor Rack E9000H-4

Tipo	Servidor Blade Telecom
Produto	ES9000H-4 (4U)
Modelo do Processador	2 * Kunpeng 920 (por blade)
Slot DIMM	slot 16 * 32 GB DDR4 (por blade)
Armazenamento Local	SSD 2*3,2 TB (por blade)
Porta de Rede	2 * interface de rede (4*25GE/10HE/GE)(por blade)
Consumo de Energia (Tipico/Máximo)	2100W/2300W
Entrada de eneria	2 + 2 DC: -48 V AC: 200 V a 240 V
Fan	7 + 1
Temperatura de Operação	0°C a 45°C
Vetilação	ventilação a ar com tomada de ar na frente e exaustao na parte de trás
Dimensões (A x L x P)	175 mm * 442 mm * 675 mm
Peso	60 Kg

Fonte: Huawei.

As interligações entre os nós de substrato da rede foram realizadas considerando os transmissores ópticos, Figura 29, interface *gigabit – Gbic*, *Cisco SFP-10G-LR-X module*, Cisco *SFP-10G-ER-S module e Cisco SFP-10G-ZR-S* com especificação, pelo fabricante, das distâncias máximas sem retransmissão, sendo 10, 40 e 80 quilômetros respectivamente.

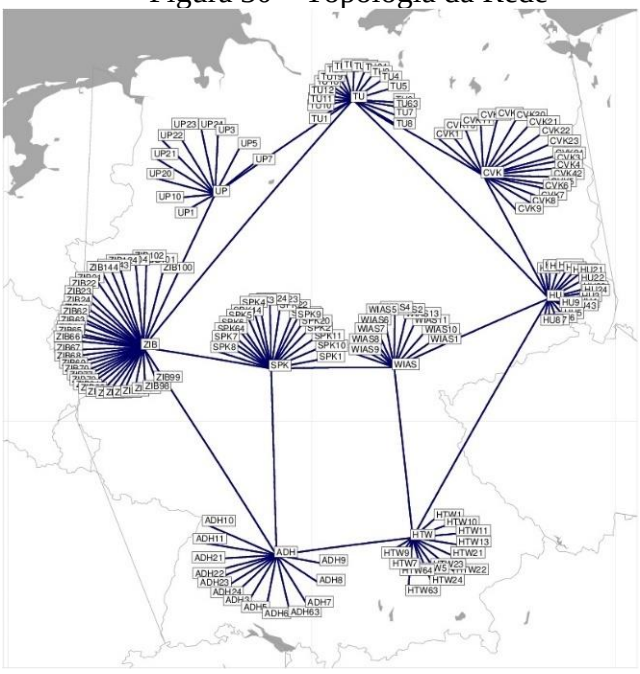
Figura 29 – Transmissor óptico – *Gbic*

Fonte: Cisco.

Foi utilizada a distância entre os nós da rede *Brain* para calcular o tipo e a quantidade de *Gbics* necessárias, sendo escolhida a interface com a especificação mais próxima da distância entre os nós de substrato para, na sequência, calcular latência média nos enlaces entre os nós de substrato através da quantidade de saltos para retransmissão no caminho entre os nós de substrato, considerando 2 ms por salto (*holp*).

A topologia da rede de substrato foi a da rede alemã Brain, vide Figura 30, provida pela biblioteca SNDlib (ORLOWSKI et al. 2010), com as redes virtuais sendo geradas de acordo com os dados de tráfego real da mesma biblioteca, com a interconexão entre os nós centrais realizada com enlaces de 100Mbps e entre os nós regionais com enlaces de 1Gbps, adicionado dois enlaces de saída para a internet, ISPs, localizados nos nós de substrato ZIB e HUB, com 1Gbps de capacidade cada.

Figura 30 – Topologia da Rede



Fonte: SNDlib (ORLOWSKI et al. 2010). Rede Brain.

Os ensaios das requisições foram de quinze minutos, equivalente a 138599 requisições, de alocação de VNs distintas, de acordo com as informações da base de dados SNDlib, com os experimentos iniciando na requisição das 15:00 de 08 de Março de 2003, com as VNRs sendo tratadas 1 a 1 como um processo de Poisson com taxa de chegada de 1 requisição por instante, e as requisições podendo perdurar, após a alocação do requerimento, em até 5 minutos seguindo uma distribuição normal entre 1 e 5 minutos para os tempos de retenção da infraestrutura alocada .

5.3 ATIVIDADES, INPLEMENTAÇÕES E EXPERIMENTOS REALIZADOS

As atividades deste trabalho foram divididas em: modelagem matemática preliminar, implementação dos métodos heurísticos convencionais VNE não coordenado (*Edge Base*) e VNE_GRC (com seleção de elementos pela disponibilidade), e desenvolvimento do método VNE com orquestração por um algoritmo de aprendizado por reforço. Dessas atividades, a primeira foi realizada no processo de elaboração do trabalho para a qualificação, encontrada em 2.4.2, e nos subitens abaixo estão as atividades relacionadas a este trabalho.

5.3.1 Implementação do método heurístico VNE não coordenado e VNE_GRC

Como critério de avaliação dos resultados alcançados por um novo algoritmo para orquestração de algoritmos heurísticos VNE, foi realizada a implementação do algoritmo convencional de alocação de recursos em uma rede substrato e do algoritmo GRC, utilizados na alocação de requisições de recurso da rede alemã *Brain*, de acordo com o descrito na metodologia, item 5.1. Para tal, foi necessário tratar a base de dados com a informação da rede e suas requisições, como também implementar os algoritmos para a comparação.

5.3.1.1 Tratamento da base de dados sndlib.zib.de

A SNDlib (ORLOWSKI et al. 2010) é uma biblioteca com informações dinâmicas de tráfego de redes, em formato xml, com a estrutura da rede (<networkStructure>) contendo os dados dos nós (<node>), com sua identificação e coordenada geográfica, e dos enlaces (<links>), com sua identificação, relação de origem e destino, e a capacidade do enlace (BW); e a demanda (<demands>) de recursos contendo a identificação da requisição, a origem e o destino dessa demanda, e a capacidade requerida.

As informações da rede *Brain* são compostas por 9.723 arquivos totalizando 8.8 Gb de informação, sendo impraticável a carga completa desta informação na memória. Assim, a atividade de tratamento da base de dados criou subarquivos com objetos *python* pelo uso da

biblioteca *pickle*, de armazenamento de objetos em formato de arquivo, já enriquecidos com a informação de latência dos meios calculados utilizando a distância geográfica dos nós de substrato e as especificações dos fornecedores de *Gbic* já referenciados na metodologia, item 5.2.

5.3.1.2 Implementação da VNE não coordenado e VNE GRC

Nesta atividade foi desenvolvido e implementado um módulo para leitura dos requerimentos e da infraestrutura, gerada na atividade anterior, e tratamento pelo algoritmo básico de alocação de recursos em uma rede substrato tratado no item 2.4.1, Algoritmo 01, e do algoritmo GRC com a fase de ranqueamento proposto Gong et al (2014), tratado no capítulo 3 no Algoritmo 06.

5.3.1.3 Simulação VNE convencional x VNE CGR

Após a implementação do item 5.3.1.2 , foram realizadas simulações com os dados de tráfego dinâmico da biblioteca SNDlib (ORLOWSKI et al. 2010) para gerar uma base de comparação dos índices mencionados na metodologia, item 5.4, antes da aplicação da orquestração pelo RL, com e sem o perfil de tráfego 5QI, nas abordagens CSP (*Communications Service Providers*) Convencional, Nuvem Pública, Nuvem Privada, computação de borda e algoritmo GRC; de forma que foi possível visualizar graficamente o problema da alocação de recursos em uma rede substrato e o problema adicional gerado pela inserção de um perfil de tráfego novo, 5G heterógeno, nas redes atuais.

5.3.2 Desenvolvimento e implementação e simulação do VNE_CRS

O desenvolvimento e a implementação do algoritmo de orquestração VNE_CRS foi realizado dividido em etapas: na modelagem da arquitetura com aprofundamento e adaptação das equações de custo, na modelagem matemática do ambiente e na modelagem matemática da recompensa realizados na sessão 4.1; definição da estratégia de aprendizado do algoritmo e da forma de treinamento, sendo utilizada *off-line* e mista, *on-line + off-line*, e quantos episódios de treinamento; e desenvolvimento do algoritmo com sua implementação e simulação utilizando os mesmos dados e forma da atividade 5.3.1.3 para comparações dos resultados obtidos na simulação sem RL com o resultados obtidos na simulação com o VNE_CRS.

5.4 CRITÉRIOS DE AVALIAÇÃO DOS RESULTADOS

Otimizar é criar melhores condições para realizar uma tarefa ou tirar o melhor benefício de algo. Este trabalho tem como objetivo otimizar os recursos de uma rede substrato para melhor atender seus usuários de forma a reaproveitar a infraestrutura e garantir maiores retornos aos provedores de infraestrutura. Os critérios de avaliação de desempenho VNE são amplamente mapeados na literatura e foram a bússola para avaliar os resultados obtidos com este trabalho.

Nessa ótica, o objetivo foi diretamente proporcional a maximizar o número de VNRs alocados (taxa de aceitação de VNE), representados pela equação (46), que é o parâmetro mais utilizado entre todos os trabalhos relacionados nesta área.

Outra forma convencional de validação da eficiência dos VNEs é pelo acompanhamento do incremento de receita com a alocação da infraestrutura definida nas equações (39), (40) e (41) da sessão 4.1Modelagem e Abstrações da VNE.

6 RESULTADOS E CONSIDERAÇÕES

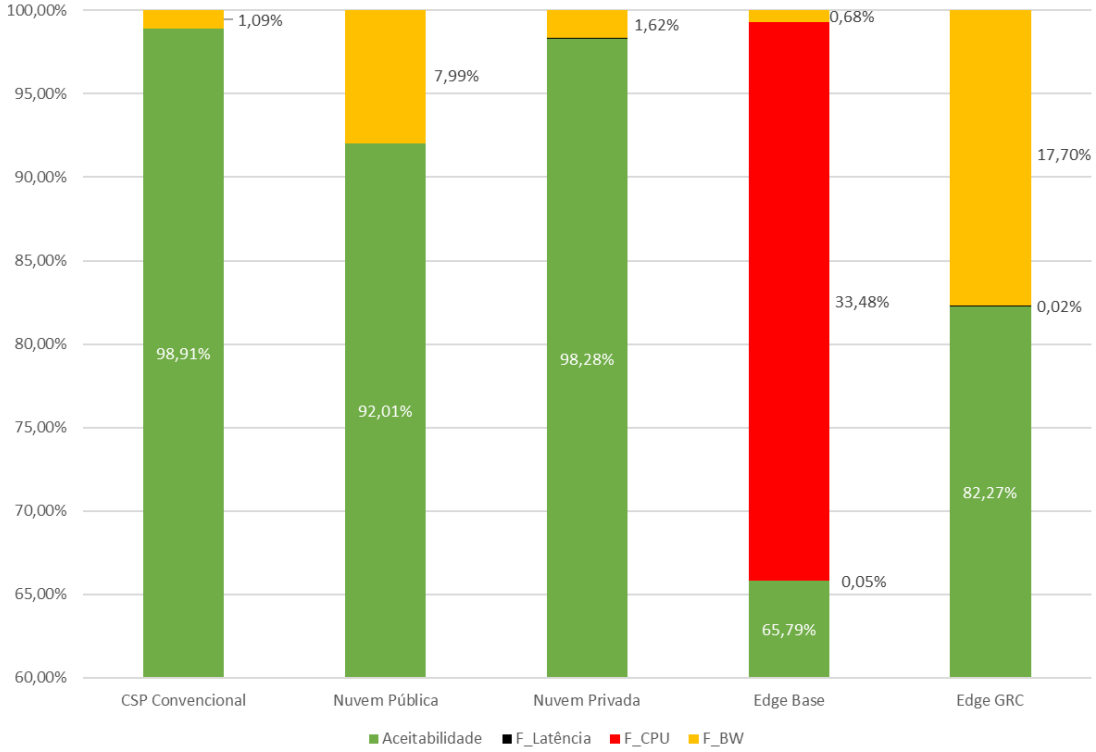
Este capítulo versa sobre: a comparação dos modelos de alocação de recursos de uma infraestrutura de rede 5G sem a orquestração pelo VNE_CRS, sessão 6.1; os resultados da alocação de recursos da rede 5G com a atuação do VNE_CRS em modo *off-line* e misto, sessão 6.2.

6.1 ABORDAGENS DE ALOCAÇÃO DE RECURSOS VNE

Foram concluídas as atividades de tratamento da base de dados e implementação dos algoritmos VNE não coordenado e VNE_GRC das subseções 5.3.1.1 e 5.3.1.2 e realizada uma experimentação via simulação, subseção 5.3.1.3, para comparar 5 cenários de evolução de redes e infraestruturas computacionais com suas capacidades informadas na metodologia de experimentação na subseção 5.2:

a) o cenário do uso da operadora como simples provedor de conectividade, CSP convencional, onde cada empresa tem sua própria infraestrutura de computação local e utiliza a rede das operadoras de telecomunicações (CSP – *Communications Service Providers*) para interconectar seus escritórios utilizando uma rede WAN;
b) o cenário de transferência da necessidade computacional para a Nuvem Pública em que a capacidade de CPU é considerada infinita e é transferida para provedores de infraestrutura em nuvem (InPs – *Infrastructure Providers*), onde as empresas compartilham recursos computacionais dos InPs e usam as CSPs para interconectar escritórios, empregadores e clientes em uma infraestrutura de rede WAN programável de forma flexível;
c) o cenário de CSP conectando uma Nuvem Privada onde a necessidade de CPU fica em uma infraestrutura de nuvem privada visando reutilizar os recursos computacionais dentro da mesma empresa e reduzir a latência e os problemas de rede;
d) o cenário de computação de borda, CSP *Edge Base* (algoritmo VNE não coordenado da subseção 5.3.1.2), onde as empresas compartilham uma capacidade computacional fisicamente próxima às necessidades dos aplicativos com o objetivo de reduzir a latência entre a aplicação e os terminai clientes;
e) e, finalmente, com cenário utilizando o algoritmo GRC (algoritmo VNE_GRC da subseção 5.3.1.2, ordenado pela quantidade de recursos disponíveis), para otimizar o compartilhamento de infraestrutura do modelo *Edge Base*.

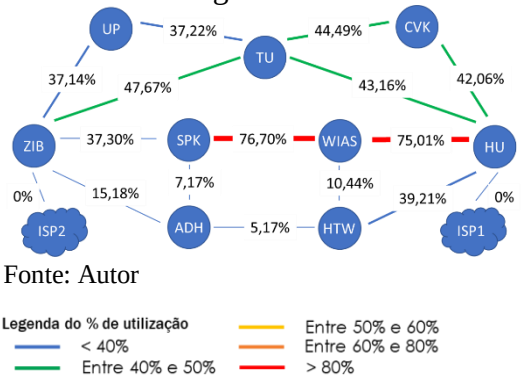
Figura 31 – Avaliação do problema de alocação de recursos e comparação dos cenários



Fonte: Autor

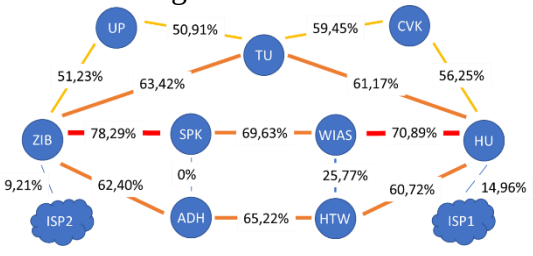
Os resultados apresentados no gráfico da Figura 31 são baseados em 15 minutos de requisitos de VNE, ou 138.599 solicitações de VNR do conjunto de dados da biblioteca SBLIB detalhada na sessão 5.1. Esses resultados mostram que a movimentação de infraestrutura computacional do senário CSP convencional para a Nuvem Pública aumentou a falha de BW da rede de 1,09 pontos percentuais para 7,99 pontos percentuais, ou 633% de aumento de falha de BW. Esse aumento das falhas ocorre porque uma das características das Nuvens Públicas é concentrar os recursos a serem compartilhados, essa característica gera gargalos nos caminhos até esses recursos, que pode ser observada na movimentação de utilização das interconexões da Figura 32 para a Figura 33, em que as linhas que interconectam os nós representam o percentual de utilização do enlace que interligam esses nós.

Figura 32 – Distribuição de carga abordagem CSP convencional



Fonte: Autor

Figura 33 – Distribuição de carga abordagem CSP + Nuvem Pública



Fonte: Autor

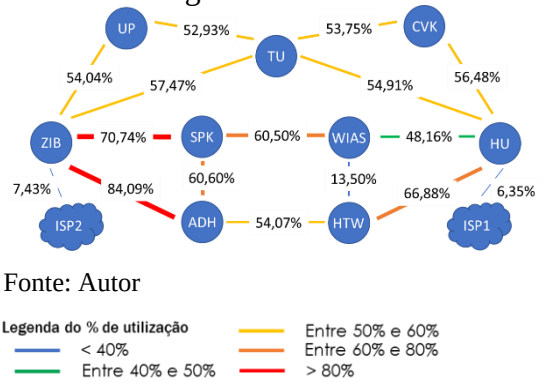
Nas representações gráficas de ocupação dos enlaces de interconexão dos nós de substratos (Figura 32, Figura 33, Figura 34 e Figura 35) foi utilizada uma paleta de cores para facilitar a observação da utilização dos enlaces seguindo a seguinte colocação: azul para ocupações inferiores a 40%; verde entre 40% e 50%, amarelo entre 50% e 60% de ocupação, laranja para as interconexões com ocupação entre 60% e 80% e vermelho para aquelas interconexões com mais de 80% de ocupação.

A redução da aceitabilidade de VNR observada no caso da Cloud Pública pode, em parte, ser contornada pelo aumento da capacidade de conexão em todo caminho da infraestrutura dos CSPs até os provedores de nuvem (concentrados nos ISPs). Porém aumentar a infraestrutura de rede com o objetivo de atenuar o problema com a concentração gerada pela Nuvem Pública cria dois desafios, o primeiro é o investimento elevado com a ampliação dessa infraestrutura de interconexão e o segundo é que apenas aumentar essa infraestrutura de rede não garante o QoS para os pontos nas extremidades dos Provedores de Nuvem, principalmente quando as aplicações são sensíveis à latência como abordado na subseção 2.2.1, sobre qualidade da rede, e por Nayak, Pandey e Knightky (2019), McClellan (2019) e Jordan (2019).

Ainda no gráfico da Figura 31, observa-se que quando o poder computacional e, consequentemente, o tráfego se movem para uma Nuvem Privada o total de falhas na rede, comparada com o modelo de Nuvem Pública, diminui para 1,62 pontos percentuais. Isso ocorre porque a concentração de recursos nos ISPs foi removida no modelo de Nuvem Privada e distribuída por toda a infraestrutura, vide Figura 34.

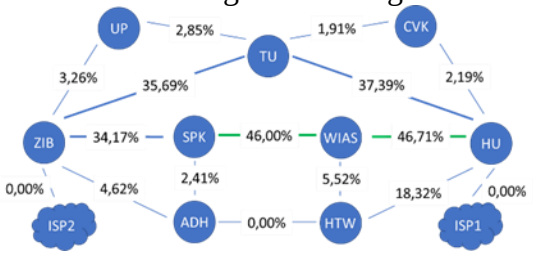
O cenário de Nuvem Privada, inicialmente, parece ser uma boa opção do ponto de vista de falhas de incorporação, mas é necessário investimentos na infraestrutura computacional em Nuvem Privada e, nesse modelo, não há compartilhamento e nem uso elástico dos recursos entre diversos usuários, de empresas e origens distintas, como ocorre nos modelos de Nuvem Pública, ou seja, não há otimização desses recursos.

Figura 34 – Distribuição de carga abordagem CSP + Nuvem Privada



Fonte: Autor

Figura 35 – Distribuição de carga abordagem CSP Edge Base



Fonte: Autor

Já no cenário da infraestrutura computacional de borda, Edge Base, apesar da distribuição da utilização dos enlaces de interconexão da Figura 35 estarem descongestionados, o resultado da alocação de recursos da infraestrutura mostrado no gráfico da Figura 31 obteve 34,21 pontos percentuais de falhas totais, sendo 0,68 pontos percentuais de falha de BW e 33,48 pontos percentuais de falhas de alocação de CPU, ou 3.038% de aumento total de falhas comparado com a abordagem CSP Convencional. Esse cenário unifica as características de compartilhamento e elasticidade das Nuvens Públicas com as vantagens de QoS das Nuvens privadas; por esse motivo, a abordagem Edge Base torna-se importante e conveniente, sendo necessário buscar mecanismos para otimizar essa infraestrutura computacional de borda.

Para reduzir os problemas de falhas de aceitabilidade da abordagem Edge Base, a primeira abordagem analisada foi o Edge GRC, que realiza uma troca, *trade-off*, entre as necessidades de CPU e a as necessidades de BW com o objetivo de homogeneizar o uso de CPU entre os nós de substrato e manter sempre algum recurso disponível em todos os nós para eventuais necessidades, como pode ser observado nos resultados da Ao avaliar os outros indicadores de resultados, na Figura 44, apesar das abordagens CSP convencional e nuvem privada atenderem um número maior de requisições simultâneas, essas abordagens não remuneram a infraestrutura computacional para os CSPs e não possibilitam o compartilhamento da infraestrutura entre diversas empresas, logo não conseguem atender a conveniência definida por Mell et al.(2011), sessão 2.3. e não otimizam os investimentos com a capacidade computacional.

Ainda na Figura 44, é possível observar que as abordagens de compartilhamento da infraestrutura computacional *edge* (Edge Base, Edge GRC, VNE_Hlog, VNE_CRS *On-line* e VNE_CRS *Off-line* + *On-line*) atendem a necessidade da criação de uma nova fonte de remuneração para as operadoras de telecomunicações, o que reverte a tendencia de redução do ARPU com a transformação das aplicações consolidadas que se tornam apenas *commodities*.

Por fim, o resultado financeiro e a média de atendimento de requisições de rede (G^vs) ativas simultaneamente observados na Figura 44 mostram que a abordagem VNE_CRS (*Off-line* + *On-line*) além de atender mais requisições simultânea que a abordagem em nuvem pública ela também foi a abordagem que teve uma melhor relação de retorno, tanto em receita absoluta quanto em resultado líquido financeiro.

Tabela 13. O Edge GRC move o requisito de CPU do nó de substrato da requisição original para outro nó de substrato, respeitando as requisições de latência da VNR nos caminhos

de ida e retorno ao nó original, utilizando a capacidade BW entre os enlaces que interconectam esses nós de substrato.

Como pode ser visto no gráfico da Figura 31, o algoritmo Edge GRC consegue, em comparação com a aceitabilidade do algoritmo convencional Edge Base, reduzir as falhas de aceitabilidade das VNRs de 34,21 para 17,73 pontos percentuais do total de falhas. Observa-se também o *trade-off* entre a necessidade de CPU e o uso de BW, com a redução das falhas de CPU, de 33,48 pontos percentuais (*Edge Base*) para 0,02 pontos percentuais, e o crescimento das falhas de BW, de 0,68 pontos percentuais (Edge Base) para 17,70 pontos percentuais. Porém, a abordagem Edge GRC, quando comparada com a abordagem MPS, tem 16,64 pontos percentuais mais falhas totais na aceitabilidade de alocação de VNRs e ainda é 9,74 pontos percentuais menos eficiente que a Nuvem Pública na alocação de recursos VNR.

Esses resultados mostram o panorama do problema VNE, a necessidade e oportunidade de otimização dos recursos em uma infraestrutura de computação de borda (*Edge*) que são o tema e o objetivo deste trabalho. O resultado do Edge_GRC é o valor de referência (*Baseline*) utilizado neste trabalho que tem como objetivo melhorar a taxa de aceitabilidade total da infraestrutura 5G em arquitetura computacional *Edge* e, consequentemente, melhorar a receita com essa infraestrutura.

6.2 AVALIAÇÃO DOS RESULTADOS DO VNE_CRS NA ALOCAÇÃO VNE

Nessa sessão será abordada a técnica utilizada para o treinamento do VNE_CRS seguido da comparação dos resultados dos algoritmos de alocação apresentados na sessão anterior com a atuação do algoritmo VNE_CRS, nas abordagens *on-line* e *mista*, coordenando os multidomínios do algoritmo VNE_GRC.

6.2.1 Abordagem de treinamento VNE_CRS

A estratégia utilizada para o treinamento do VNE_CRS foi pela definição dos meta-parâmetros do algoritmo de aprendizado por reforço, a taxa de aprendizado α e o fator de desconto γ (Algoritmo 09, linha 10 e linha 11 respectivamente), através da experimentação do VNE_CRS *Off-line* pela execução com dados (treinamento) de 2 minutos, ou 18914 solicitações VNR, das solicitações de recursos de rede virtual da base de dados SNDlib, iniciando no registro da requisição das 15:00 de 08 de março, sendo os resultados da experimentação

apresentados na Tabela 10, seguida do acréscimo do ciclo de treinamento até alcançar 5 minutos, contemplando o clico máximo de uma solicitação conforme explicado na sessão 5.1, e concluindo com mais 2 minutos de treinamento para o agente conhecer o ambiente após o ciclo máximo de retenção de recursos da infraestrutura.

Tabela 10 – Treinamento 2 Minutos para definição de estratégia

Taxa de Aprendizagem [α]	Fator de Desconto [γ]	Qt Episódios para convergir	Taxa de Aceitabilidade de VNRs	Tamanho do Espaço de Estados
10%	70%	95	98,69%	3693
10%	80%	154	97,78%	6950
10%	90%	33	99,07%	2882
20%	70%	61	99,02%	3227
20%	80%	78	98,44%	2923
20%	90%	157	98,89%	2261
30%	70%	62	99,16%	4270
30%	80%	64	98,91%	3736
30%	90%	109	98,65%	3816

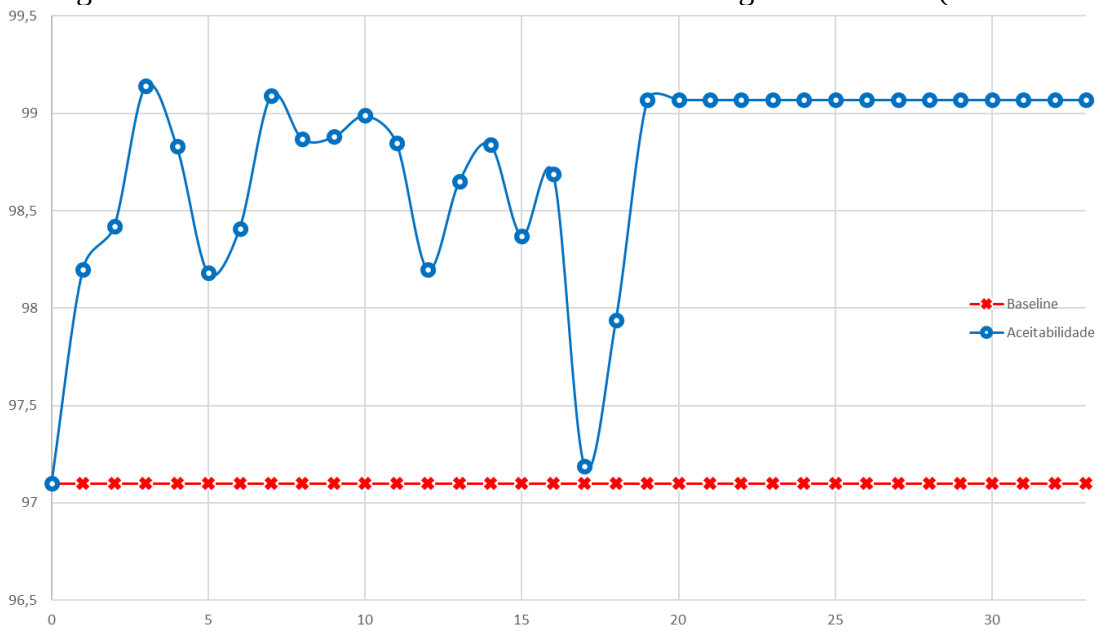
Fonte: Autor

Apesar de a taxa de Aceitabilidade de VNR na etapa de definição dos meta-parâmetros ter obtido um resultado maior (99,16%) para uma taxa de aprendizagem de 30% e um fator de desconto de 70%, foi selecionado a taxa de aprendizagem de 10% com fator de desconto de 90% por ter convergido com uma quantidade menor de episódios de treinamento (33), sendo esse fator decisivo em casos com muitos estados de espaços, como no caso deste trabalho.

O treinamento inicial para o aprendizado da política de atuação e do estado de espaços do ambiente foi realizado através de simulações utilizando o Algoritmo 10, VNE_CRS *Off-line*, durante a subida do tráfego (*ramp-up*) com toda infraestrutura inicialmente disponível, ou seja, sem nenhum recurso alocado.

Os resultados na Figura 36 mostram que o VNE_CRS converge, no cenário de treinamento com 2 minutos de requerimentos, após 20 episódios de treinamento, com uma aceitabilidade de VNRs de 99,07%. Esse resultado não pode ser considerado como um acréscimo de aceitabilidade em comparação com o *baseline* por conter somente informações dos dados de treinamento.

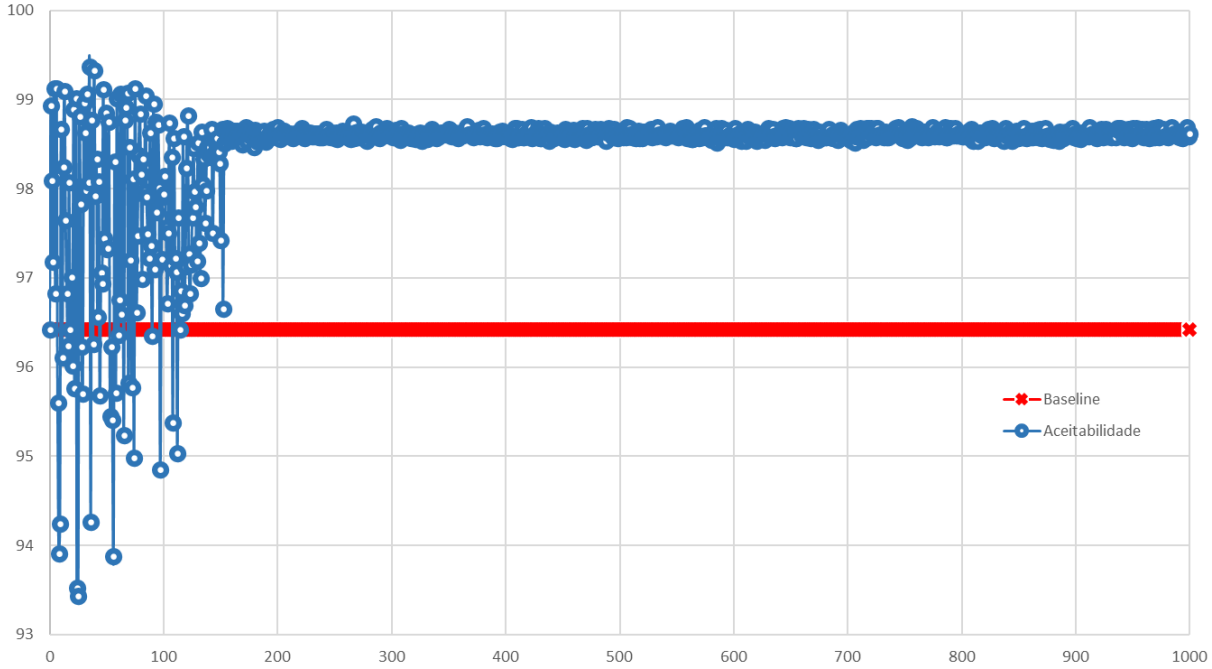
Figura 36 – Taxa de Aceitabilidade na subida do tráfego – 2 minutos (treinamento)



Fonte: Autor

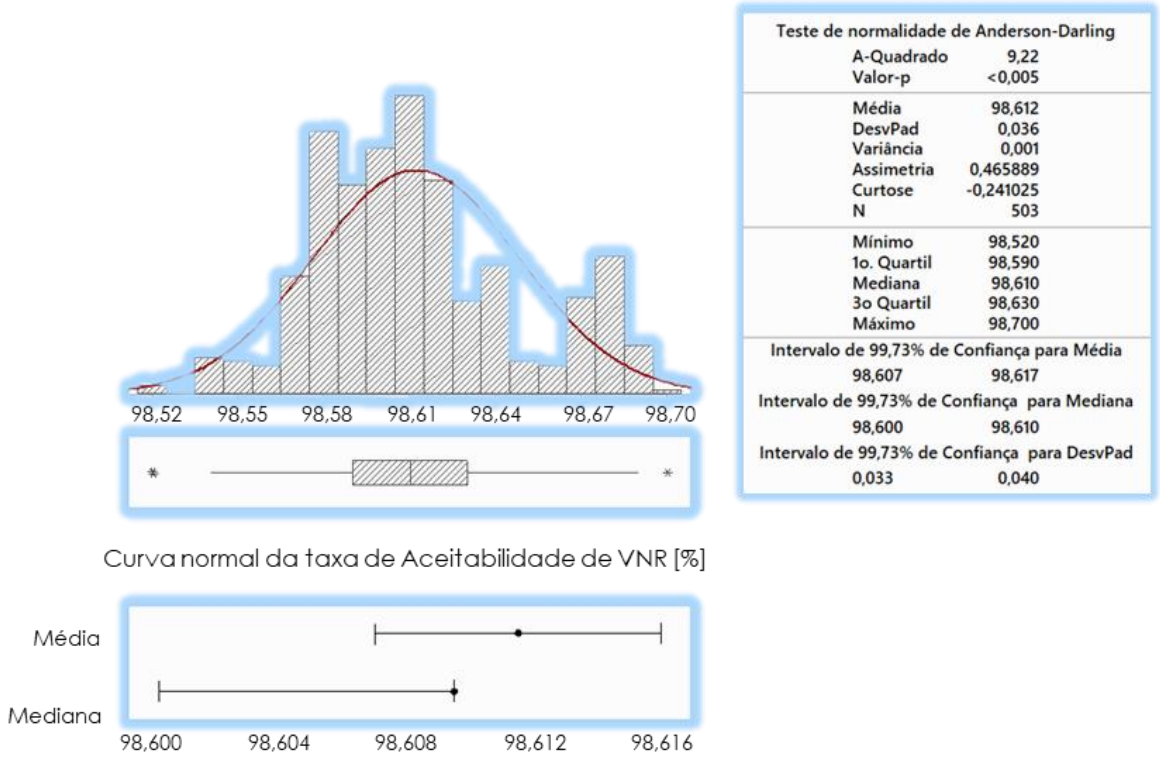
Seguindo na estratégia de treinamento, foram acrescidos 3 minutos da sequência de requisições de VNR à base de treinamento do experimento anterior, de 2 minutos de treinamento, consolidando um total de 5 minutos de treinamento, ou 46.371 solicitações de VNR. Nessa experiência, mesmo após 1.000 episódios de treinamento, o VNE_CRS não converge completamente um único resultado, como pode ser observado no gráfico da Figura 37. Mas, ao verificar a dispersão e distribuição dos resultados após 500 episódios, mostrado na Figura 38, o desvio padrão (0,036), a variância (0,001), o valor p (menor que 0,005) e o Coeficiente de Variação (DevPad/Média) = 0,00000037 são baixos e aceitáveis, sendo as referências dos resultados, com 3 desvios padrão (99,73% das amostras), para a mediana da taxa de aceitabilidade entre 98,600% e 98,610%, e para a média dos valores da taxa de aceitabilidade entre 98,607% e 98,617%, ambas referências com valores da taxa de aceitabilidade de VNR superiores à linha do *baseline* de 97,10%.

Figura 37 – Taxa de Aceitabilidade na subida do tráfego – 5 minutos (treinamento)



Fonte: Autor

Figura 38 – Resumo estatístico da taxa de aceitabilidade – 5 minutos (treinamento)



Teste de normalidade de Anderson-Darling	
A-Quadrado	9,22
Valor-p	<0,005
Média	98,612
DesvPad	0,036
Variância	0,001
Assimetria	0,465889
Curtose	-0,241025
N	503
Mínimo	98,520
1o. Quartil	98,590
Mediana	98,610
3o Quartil	98,630
Máximo	98,700
Intervalo de 99,73% de Confiança para Média	
98,607	98,617
Intervalo de 99,73% de Confiança para Mediana	
98,600	98,610
Intervalo de 99,73% de Confiança para DesvPad	
0,033	0,040

Fonte: Autor

Conforme explicado na sessão 5.1, sobre o banco de dados e ambiente, uma solicitação pode reter os recursos da infraestrutura por até 5 minutos, o que significa que após 5 minutos de amostras todas as solicitações dos primeiros requerimentos de VNRs foram liberados e, em

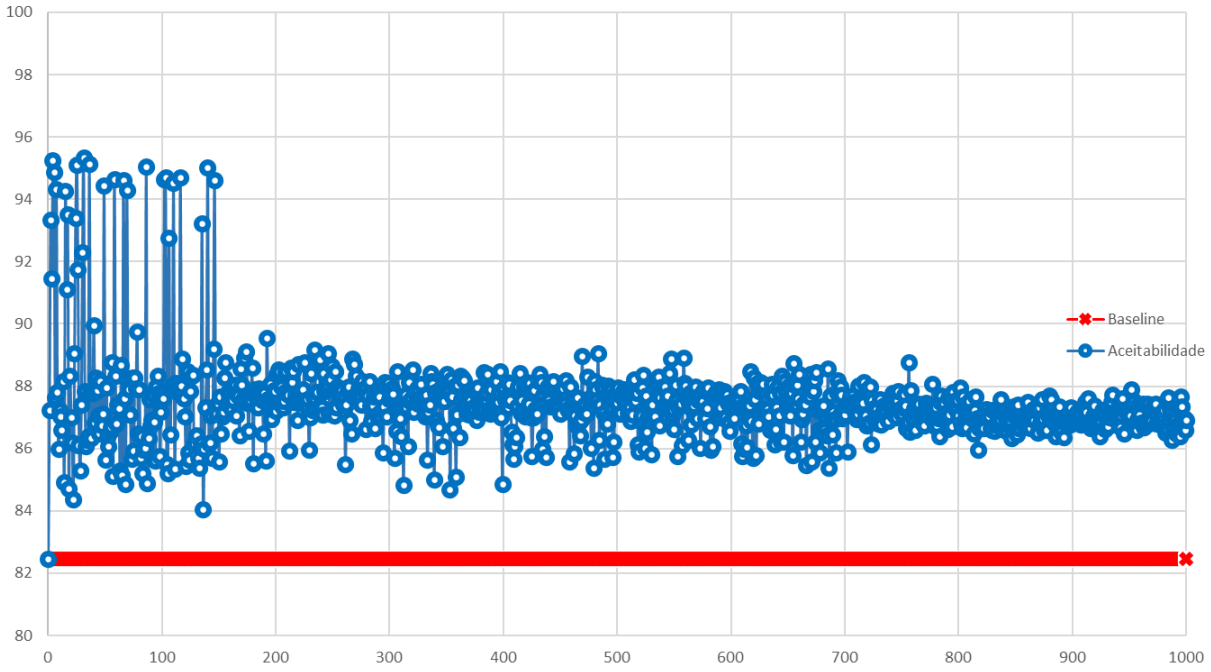
seguida, haverá um novo ciclo completo de solicitações e liberações, o que faz o treinamento de 5 minutos importante para este experimento.

Assim sendo, para concluir o processo de treinamento deste experimento e gerar a base de conhecimento do espaço de estados e da política π inicial, foi realizado, após a conclusão do ciclo máximo de retenção das VNRs e de carga da infraestrutura (5 minutos treinamento), um treinamento adicional de 2 minutos, que adiciona 18.780 solicitações VNR, com um total de 7 minutos treinamento, ou 65.151 solicitações VNR, sendo 1.000 episódios de 5 minutos de treinamento na subida tráfego e mais 1.000 episódios de treinamento com a infraestrutura carregada com 5 minutos, acrescida com o trafego de 2 minutos de solicitações.

Tal como na abordagem de treinamento na subida do tráfego com 5 minutos, mesmo após 1.000 episódios de treinamento o VNE_CRS não conseguiu convergir completamente para um único resultado, como pode ser observado na Figura 39, mas, assim como a fase de treinamento anterior, ao verificar a dispersão e distribuição dos resultados após 500 episódios, mostrado na Figura 40, o desvio padrão (0,558), a variância (0,311), o valor p (menor que 0,005) e o Coeficiente de Variação (DevPad/Média) = 0,00064 são baixos e aceitáveis, sendo as referências dos resultados, com 3 desvios padrão (99,73% das amostras), para a mediana da taxa de aceitabilidade entre 87,060% e 87,240%, e para a média dos valores da taxa de aceitabilidade entre 87,104% e 87,254%, ambas referências com valores da taxa de aceitabilidade de VNR superiores à linha do *baseline* de 82,42% para 5 + 2 minutos de requerimentos de VNR.

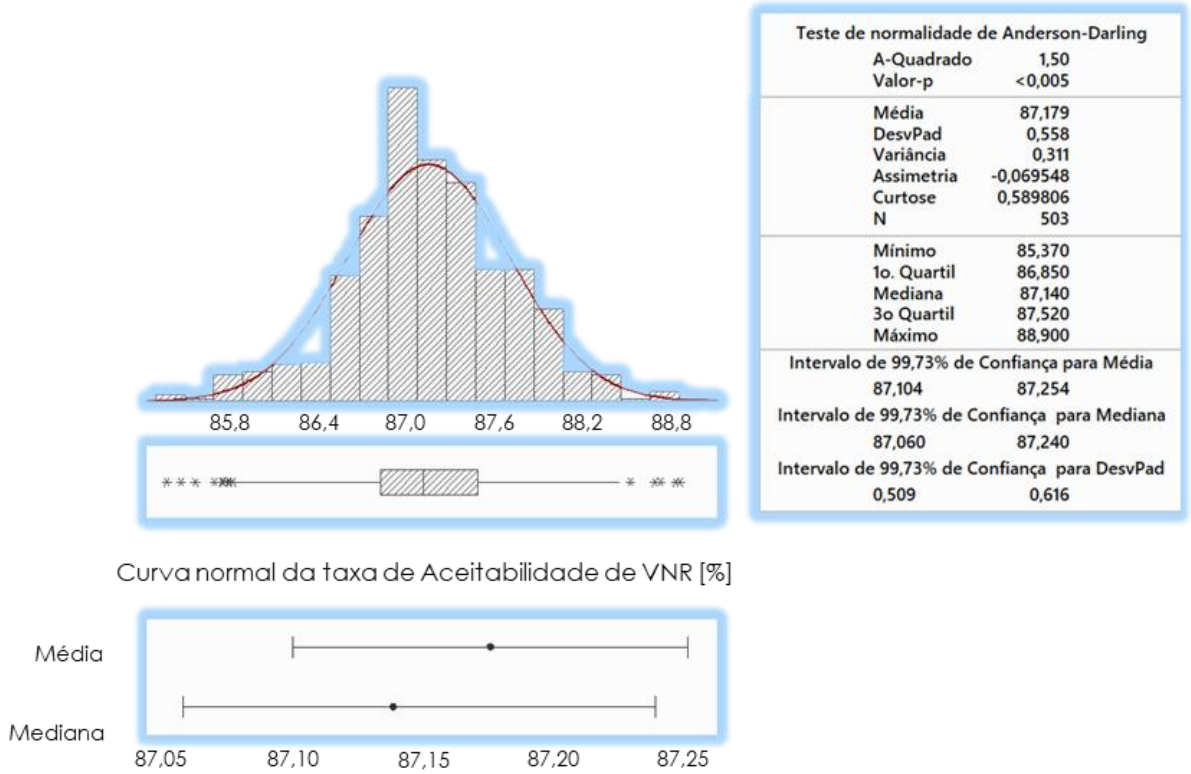
Ao final da etapa de treinamento, o Algoritmo 10 armazena as informações do valor utilidade Q^{π}, da política π e do estado de espaços S_t em arquivos binários, através do uso da biblioteca *pickle* do Python. Esses arquivos são utilizados nas etapas seguintes deste experimento para verificar o desempenho do algoritmo VNE_CRS *Off-line* e *Off-line* + *On-line* (misto) com requerimentos VNR que não fazem parte dos dados de treinamento.

Figura 39 – Taxa de Aceitabilidade em carga de tráfego + 2 minutos (treinamento)



Fonte: Autor

Figura 40 – Resumo estatístico da taxa de aceitabilidade – 5 minutos (treinamento)

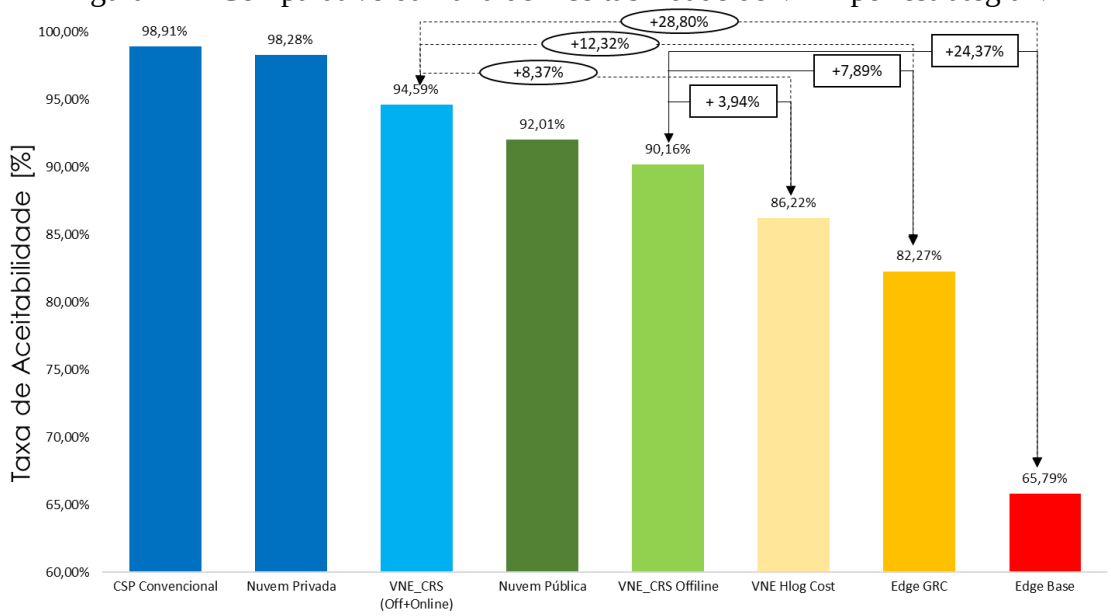


Fonte: Autor

6.2.2 Validação VNE_CRS e comparação de resultados

Para validar o VNE_CRS, foram comparados os resultados dos 5 cenários de evolução de redes e infraestruturas computacionais – o CSP Concencional, a Nuvem Pública, a Nuvem Privada, o CSP Edge Base e o Edge GRC – com os cenários utilizando a proposta de heurística pela otimização das equações de custo apresentadas na sessão 4.1 e com a atuação do VNE_CRS nas abordagens *off-line* e *off-line* + *on-line*. Essa validação foi realizada com 15 minutos de solicitações VNR, ou 138.599 solicitações, iniciando no registro da requisição das 15:00 de 08 de março da base SNDlib, com seus resultados consolidados na Figura 41 e detalhamentos na Tabela 11.

Figura 41 – Comparativo da Taxa de Aceitabilidade de VNR por estratégia VNE



Fonte: Autor

Pela Tabela 11, observa-se que, nas abordagens por CSP Convencional e de Nuvem Privada, o valor da taxa de aceitabilidade de VNRs e o valor da média das redes virtuais ativas simultaneamente são sempre maiores. Esse fato ocorre porque nos modelos de CSP Convencional e Nuvem Privada não existe a necessidade de uso de CPU da infraestrutura da rede compartilhada.

Os modelos CSP Convencional e Nuvem Privada utilizam recurso de CPU próprio, o que possibilita menores pontos de falhas e consequentemente um maior atendimento às requisições VNR. Nessas abordagens a alocação de recursos pelo VNE é simplesmente a alocação de recursos de rede de comunicação, o que representa somente as necessidades das camadas mais baixas da pirâmide de Maslow (Li, 2018), sessão 2.1, elas não compartilham a capacidade computacional, e não são alocadas de forma elástica e dinâmica, logo, essas abordagens não conseguem atender a conveniência definida por Mell et al.(2011), sessão 2.3.

Tabela 11 – Resultados dos algoritmos de VNE – 15 minutos de requisições VNR

Tipo	Geral	Dia	Hora	Min	Alocados	Requisitados	Simultaneos	%Aceitabilidade	Receita Total	Custo Total
CSP Convencional	98,91%	20130308	15	0	9497	9497	9497	100,00%	202391,5	27211,24
		20130308	15	1	9417	9417	16997	100,00%	393898,68	52899,88
		20130308	15	2	9206	9279	22410	99,74%	564341,17	76428,1
		20130308	15	3	9033	9045	25840	99,77%	743144,21	104091,54
		20130308	15	4	9070	9133	27462	99,68%	883174,46	125482,3
		20130308	15	5	9223	9463	27483	99,31%	1018682,05	149688,03
		20130308	15	6	9260	9317	27529	99,32%	1136472,84	166963,98
		20130308	15	7	9342	9451	27689	99,26%	1287995,04	189126,64
		20130308	15	8	9292	9321	27766	99,31%	1447595,73	214325,57
		20130308	15	9	9023	9074	27497	99,32%	1561498,01	230789,64
		20130308	15	10	8937	9009	27188	99,31%	1704410,88	253348,53
		20130308	15	11	8959	9066	27196	99,27%	1824883,54	275671,62
		20130308	15	12	9003	9003	27150	99,32%	1971396,01	300056,77
		20130308	15	13	8623	9173	26738	98,95%	2092525,48	319220,79
		20130308	15	14	9203	9351	26997	98,91%	2201600,92	335698,79
Nuvem Publica	92,01%	20130308	15	0	9470	9497	9470	99,72%	190978,66	32740,33
		20130308	15	1	9361	9417	16920	99,56%	370245,61	67211
		20130308	15	2	9004	9279	22146	98,73%	493631,44	93397,04
		20130308	15	3	6906	9045	23503	93,29%	614862,64	119613,05
		20130308	15	4	8835	9133	25388	93,97%	720331,74	146087,84
		20130308	15	5	6656	9463	23369	89,97%	781588,03	160895,46
		20130308	15	6	8810	9317	23977	90,62%	867460,02	179939,52
		20130308	15	7	8694	9451	24626	90,80%	955891,49	200911,94
		20130308	15	8	8342	9321	25000	90,65%	1054573,87	222350,29
		20130308	15	9	8813	9074	25478	91,28%	1159591,23	246908,73
		20130308	15	10	8795	9009	25979	91,84%	1240133,97	269800,62
		20130308	15	11	8841	9066	26329	92,31%	1332197,48	290647,05
		20130308	15	12	8612	9003	26315	92,56%	1433937,41	315064,13
		20130308	15	13	7874	9173	25497	92,08%	1528326,64	336656,75
		20130308	15	14	8513	9351	25359	92,01%	1617916,92	355628,4
Nuvem Privada	98,28%	20130308	15	0	9496	9497	9496	99,99%	202386,83	30103,7
		20130308	15	1	9413	9417	16992	99,97%	393838,46	60923,57
		20130308	15	2	9184	9279	22385	99,65%	564152,64	88594,81
		20130308	15	3	8962	9045	25750	99,51%	677208,77	114131,68
		20130308	15	4	9018	9133	27341	99,36%	789104,23	133086,46
		20130308	15	5	9373	9463	27542	99,31%	930961,94	157933,61
		20130308	15	6	9231	9317	27546	99,27%	1090146,27	187049,26
		20130308	15	7	9354	9451	27732	99,23%	1230116,85	211583,48
		20130308	15	8	9179	9321	27704	99,15%	1349369,39	231145,33
		20130308	15	9	8878	9074	27300	99,02%	1485715,64	261138,86
		20130308	15	10	8695	9009	26783	98,80%	1591377,03	279425,39
		20130308	15	11	8562	9066	26494	98,45%	1738243,81	311515,55
		20130308	15	12	8805	9003	26413	98,40%	1833139,05	330352,95
		20130308	15	13	8971	9173	26531	98,35%	1959463,71	354743,84
		20130308	15	14	9098	9351	26787	98,28%	2057545,7	371898,1
Edge Base	65,79%	20130308	15	0	9496	9497	9496	99,99%	1444345,4	207526,35
		20130308	15	1	8171	9417	15750	93,41%	2665356,98	1428537,93
		20130308	15	2	4682	9279	16884	79,27%	3314821,76	2078002,71
		20130308	15	3	4698	9045	17183	72,63%	3950642,87	2713823,82
		20130308	15	4	5633	9133	17350	70,48%	4743463,05	3506644
		20130308	15	5	6478	9463	17302	70,13%	5670048,88	4433229,83
		20130308	15	6	5998	9317	17315	69,31%	6513148,02	5276328,97
		20130308	15	7	5690	9451	17527	68,16%	7311589,37	6074770,32
		20130308	15	8	5734	9321	17569	67,42%	8142411,73	6905592,68
		20130308	15	9	5806	9074	17548	67,08%	8941793,86	7704974,81
		20130308	15	10	5984	9009	17447	67,03%	9770562,35	8533743,3
		20130308	15	11	5653	9066	17410	66,64%	10558375,32	9321556,27
		20130308	15	12	5677	9003	17396	66,38%	11332387,09	10095568,04
		20130308	15	13	5611	9173	17200	66,01%	12131561,02	10894741,97
		20130308	15	14	5879	9351	17253	65,79%	12938524,39	11701705,34

Edge GRC	82,27%	20130308	15	0	9496	9497	9496	99,99%	1444345,4	219169,46
		20130308	15	1	9179	9417	16758	98,74%	2843966,93	434099,3
		20130308	15	2	5881	9279	18897	87,10%	3771654,33	572727,91
		20130308	15	3	6484	9045	20510	83,36%	4778855,53	724018,85
		20130308	15	4	4889	9133	19191	77,48%	5597206,83	845099,12
		20130308	15	5	7673	9463	19703	78,09%	6779717,23	1022099,5
		20130308	15	6	8671	9317	21541	80,23%	8093758,96	1221012,64
		20130308	15	7	7155	9451	22019	79,66%	9175283,72	1381941,31
		20130308	15	8	7651	9321	22659	79,93%	10318140,11	1551170,96
		20130308	15	9	7677	9074	23029	80,39%	11489018,5	1727742,36
		20130308	15	10	7946	9009	23188	81,08%	12656781,78	1903122,33
		20130308	15	11	7425	9066	22979	81,14%	13813374,97	2079930,42
		20130308	15	12	7882	9003	23346	81,62%	14948044,41	2247776,09
		20130308	15	13	7346	9173	22921	81,51%	16054912,79	2413521,38
		20130308	15	14	8675	9351	23904	82,27%	17318948,15	2601719,07
VNE_Hlog	86,22%	20130308	15	0	9245	9497	9245	97,35%	1395756,28	43223,01
		20130308	15	1	9120	9417	16509	97,10%	2777272,86	162422,06
		20130308	15	2	9277	9279	22101	98,05%	4172532	346961,26
		20130308	15	3	8874	9045	25460	98,06%	5507181,67	611098,15
		20130308	15	4	7996	9133	26149	95,99%	6670268,81	876761,33
		20130308	15	5	8485	9463	25757	94,92%	7912701,14	1147386,9
		20130308	15	6	7627	9317	24611	93,05%	9067701,07	1379341,13
		20130308	15	7	8147	9451	24294	92,18%	10276510,44	1606867,97
		20130308	15	8	7726	9321	23781	91,15%	11386529,05	1808372,69
		20130308	15	9	6648	9074	22379	89,41%	12361478,65	1973896,64
		20130308	15	10	7464	9009	22093	88,83%	13469225,35	2141810,24
		20130308	15	11	7803	9066	22575	88,60%	14583559,72	2318132,99
		20130308	15	12	6917	9003	21998	87,72%	15578388,58	2475817,91
		20130308	15	13	7025	9173	21683	86,93%	16612123,25	2640406,87
		20130308	15	14	7148	9351	21611	86,22%	17667818,09	2803513,78
VNE_CRS - Offiline	90,16%	20130308	15	0	9363	9497	9363	98,59%	1425647,06	44786,86
		20130308	15	1	9244	9417	16728	98,38%	2802645,18	164020,17
		20130308	15	2	9275	9279	22276	98,90%	4197564,64	349258,77
		20130308	15	3	8910	9045	25615	98,80%	5543523,65	618774,73
		20130308	15	4	8896	9133	27133	98,53%	6824204,08	938533,33
		20130308	15	5	7290	9463	25325	94,88%	7924555,87	1181108,44
		20130308	15	6	8362	9317	24946	94,15%	9151308,99	1428718,96
		20130308	15	7	7514	9451	23891	92,30%	10259577,92	1631232,96
		20130308	15	8	7866	9321	23533	91,42%	11435416,81	1840632,12
		20130308	15	9	7643	9074	23151	90,72%	12549097,03	2029591,8
		20130308	15	10	8349	9009	23733	90,89%	13752972,42	2237593,44
		20130308	15	11	7837	9066	23826	90,53%	14885023,77	2436870,86
		20130308	15	12	7911	9003	23985	90,33%	16047445,31	2641929,42
		20130308	15	13	8414	9173	24454	90,43%	17280313,68	2868754,37
		20130308	15	14	8091	9351	24473	90,16%	18445058,59	3087654,29
VNE_CRS - Offline+Online aprendizado continuo	94,59%	20130308	15	0	9370	9497	9370	98,66%	1426699,58	44604,14
		20130308	15	1	9365	9417	16852	99,05%	2858590,38	168947,39
		20130308	15	2	9276	9279	22375	99,35%	4253717,46	360201,03
		20130308	15	3	8404	9045	25181	97,79%	5532753,71	614028,57
		20130308	15	4	8726	9133	26592	97,35%	6810360,64	916723,76
		20130308	15	5	8804	9463	26416	96,62%	8079197,86	1215974,45
		20130308	15	6	8761	9317	26217	96,25%	9377564,48	1507352,69
		20130308	15	7	8915	9451	26363	96,00%	10673419,57	1806624,46
		20130308	15	8	8557	9321	26222	95,54%	11912202,68	2084549,19
		20130308	15	9	8496	9074	25895	95,35%	13165750,42	2351432,29
		20130308	15	10	8504	9009	25665	95,27%	14417261,65	2613509,88
		20130308	15	11	8529	9066	25759	95,17%	15640293,66	2875645,02
		20130308	15	12	8328	9003	25527	94,97%	16846943,46	3126472,49
		20130308	15	13	8325	9173	25315	94,67%	18056245,48	3375360,68
		20130308	15	14	8745	9351	25587	94,59%	19313097,84	3642454,98

Fonte: Autor

A conveniência e o compartilhamento de toda a infraestrutura das redes de quinta geração são estruturados por domínios segmentados (XU et al., 2020) em rede de acesso, transporte e core, esse último compartilha os recursos computacionais com as aplicações em uma topologia de borda, *Edge*. Essa conveniência aumenta a quantidade de elementos e de domínios que devem ser orquestrados durante a alocação dos recursos, o que aumenta sua complexidade e consequentemente aumente a possiblidade de falhas nesse processo. Neste trabalho o VNE_Edge (*Edge Base – VNE não coordenado*), VNE_GRC (*Edge GRC*), VNE_Hlog e VNE_CRS na abordagem *off-line* e mista, listados na Tabela 11, são os algoritmos que buscam atender essa conveniência e compartilhamento.

Pela proposta de equações de custo pela razão da disponibilidade logarítmica na modelagem na sessão 4.1, Equação (44) e Equação (45), a abordagem pelo algoritmo VNE_Hlog gerencia a alocação dos recursos da infraestrutura com um logaritmo do inverso da disponibilidade de cada recurso, o que possibilita a criação de uma distribuição normalizada pela alocação dos recursos com base na porcentagem de disponibilidade de cada recurso. Essa abordagem tem uma ênfase maior, logarítmica, no rápido crescimento dos custos ao iniciar a ocupação dos recursos e uma suavização na relação de crescimento desses custos quando a alocação se aproxima de sua capacidade máxima.

Seguindo para a atividade da subseção 5.3.2, pela avaliação dos dados na Tabela 11, a distribuição normalizada VNE_Hlog possibilita o aumento da taxa de aceitabilidade em comparação com o Edge Base (VNE não coordenado) e com o Edge GRC (*baseline* – VNE_GRC – seleção pela quantidade de recursos disponíveis), alcançando 20,43 pontos percentuais acima da abordagem Edge Base e 3,95 pontos percentuais acima da abordagem de Edge GRC. O VNE_HLog possibilita também o aumento da média de redes virtuais ativas simultaneamente de 20.676 (*baseline*) para 22.016 redes virtuais atendidas simultaneamente.

Essa melhora nos resultados observados com o VNE_Hlog ocorre porque, quando um nó de substrato com pouca capacidade percentual disponível é solicitado por um VNR, o custo percebido desse nó de substrato será maior do que outros nós de substrato que tenham uma capacidade disponível absoluta menor, porém, uma capacidade disponível percentual maior, o que leva o VNE_Hlog tentar alocar o VNR nesses nós de substrato percentualmente mais livres e equilibrar a taxa de disponibilidade dos nós.

Com uma modelagem semelhante à do GRC, porém distinto na avaliação da disponibilidade realizada pelo algoritmo que ordena a seleção pela quantidade absoluta de recursos disponíveis, o VNE_Hlog gera uma distribuição percentual de alocação mais ampla, observada comparando-se a Figura 42 e a Figura 43 (uma compilação dos dados das tabelas

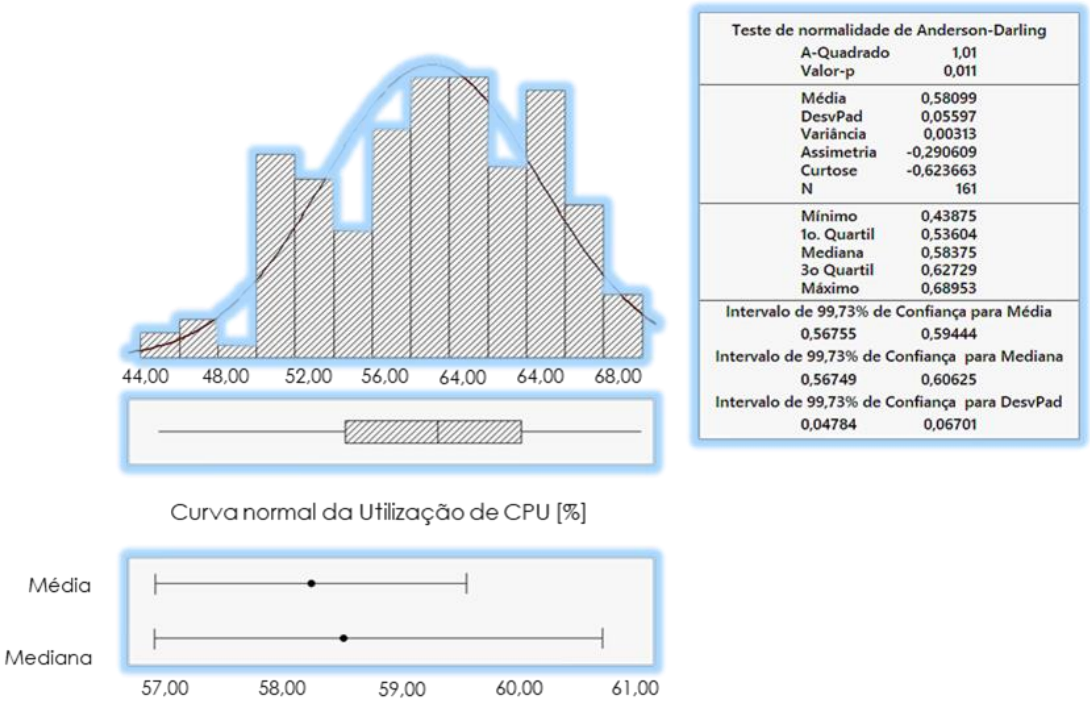
Tabela 12 e Ao avaliar os outros indicadores de resultados, na Figura 44, apesar das abordagens CSP convencional e nuvem privada atenderem um número maior de requisições simultâneas, essas abordagens não remuneram a infraestrutura computacional para os CSPs e não possibilitam o compartilhamento da infraestrutura entre diversas empresas, logo não conseguem atender a conveniência definida por Mell et al.(2011), sessão 2.3. e não otimizam os investimentos com a capacidade computacional.

Ainda na Figura 44, é possível observar que as abordagens de compartilhamento da infraestrutura computacional *edge* (Edge Base, Edge GRC, VNE_Hlog, VNE_CRS *On-line* e VNE_CRS *Off-line* + *On-line*) atendem a necessidade da criação de uma nova fonte de remuneração para as operadoras de telecomunicações, o que reverte a tendencia de redução do ARPU com a transformação das aplicações consolidadas que se tornam apenas *commodities*.

Por fim, o resultado financeiro e a média de atendimento de requisições de rede (G^vs) ativas simultaneamente observados na Figura 44 mostram que a abordagem VNE_CRS (*Off-line* + *On-line*) além de atender mais requisições simultânea que a abordagem em nuvem pública ela também foi a abordagem que teve uma melhor relação de retorno, tanto em receita absoluta quanto em resultado líquido financeiro.

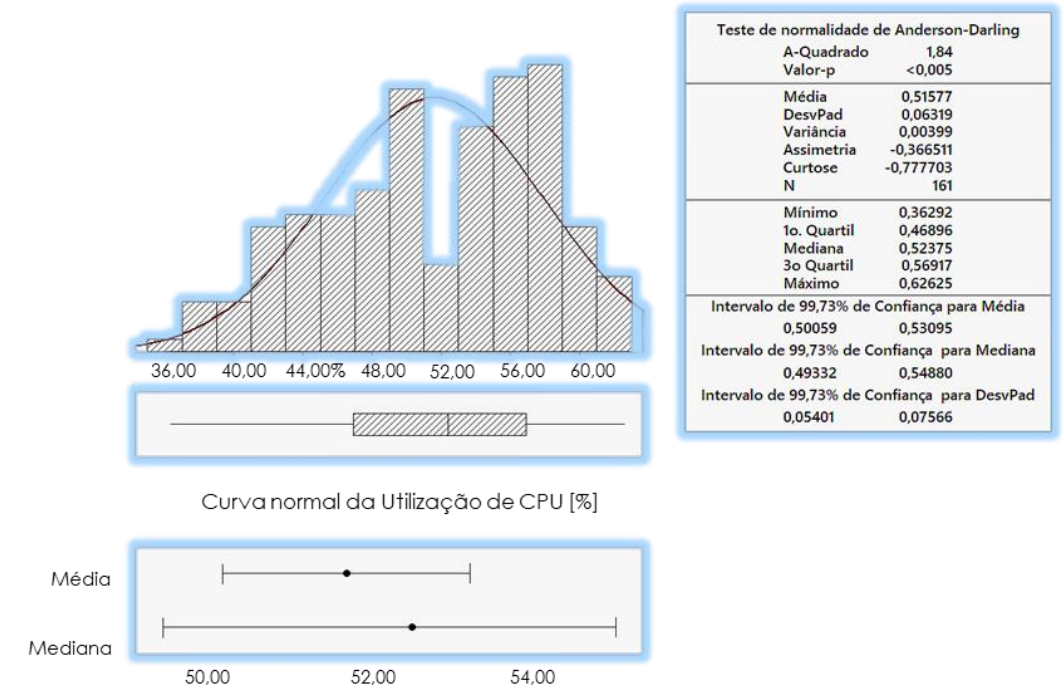
Tabela 13) de alocação percentual dos principais recursos de CPU, onde é possível observar que o VNE_Hlog tem uma média de ocupação em 51,57%, sendo 6,5% mais disponível que a abordagem VNE_GRC que, em média, tem sua ocupação em 58,09%.

Figura 42 – Distribuição Carga CPU GCR



Fonte: Autor

Figura 43 – Distribuição de carga CPU Hlog



Fonte: Autor

Tabela 12 – Utilização de CPU nos nós primários pelas abordagens de alocação

Nós Substrato	EdgeBase	Edge GRC	VNE_Hlog	VNE_CRS_Offline	VNE_CRS_Misto
Node: ADH	0,00%	65,86%	54,00%	58,13%	60,9%
Node: CVK	0,00%	62,52%	58,20%	56,59%	58,7%
Node: HTW	0,00%	68,95%	55,55%	60,56%	63,9%
Node: HU	0,00%	66,31%	59,48%	59,42%	62,2%
Node: SPK	0,00%	62,59%	47,39%	57,78%	62,6%
Node: TU	0,00%	63,55%	57,03%	60,20%	61,3%
Node: UP	0,00%	65,50%	56,50%	55,98%	63,6%
Node: WIAS	0,00%	64,94%	54,52%	60,73%	64,5%
Node: ZIB	0,00%	65,17%	58,25%	57,53%	63,7%

Fonte: Autor

Ainda na atividade da subseção 5.3.2, pela observação dos resultados dos experimentos realizados na Figura 41, a abordagem VNE_CRS *Off-line*, com treinamento durante os 5 minutos de carregamento do tráfego mais 2 minutos com a infraestrutura carregada com tráfego e realizando seus ciclos de retenção de recursos da infraestrutura pela VNR, obteve melhores resultados de taxa de aceitabilidade de VNR quando comparado com a abordagem Edge GRC e VNE_Hlog alcançando 7,89 pontos percentuais acima da abordagem Edge GRC e 3,94 pontos percentuais acima da abordagem de VNE_Hlog.

Além da melhora no índice de aceitabilidade de requisições de rede virtual pela abordagem VNE_CRS *Off-line*, é possível observar, pela Tabela 11, que a receita a longo prazo e a da média de redes virtuais ativas simultaneamente aumentaram, de 17.318.948,15 (Edge GRC) e 17.667.818,09 (VNE_Hlog) pontos de receita para 18.445.058,59 (VNE_CRS *Off-line*) pontos de receita, e a média de redes virtuais ativas no Edge GRC com 20.676 redes virtuais ativas simultaneamente indo para 22.828 com a abordagem VNE_CRS *Off-line*. A abordagem VNE_CRS *Off-line* consegue otimizar o uso da infraestrutura e permitir um aumento de 10.4% de redes virtuais ativas simultaneamente, utilizando os mesmos recursos da infraestrutura.

Com as mesmas equações de custos e lógica de alocação dos recursos utilizados no VNE_Hlog, porém com uma abordagem de uso de aprendizado de máquina com técnicas de aprendizado por reforço, o VNE_CRS *Off-line* consegue melhorar os resultados do VNE_HLog por utilizar os conhecimentos da infraestrutura e do comportamento de uso da rede Brain, obtidos durante a fase de treinamento (subseção 6.2.1).

O aprendizado do comportamento da infraestrutura auxilia o VNE_CRS *Off-line* a prever qual nó da rede substrato é mais solicitado durante o ciclo de uso da rede e, com essa informação, gerar uma estratégia de custo de recursos (CRS – *Cost Resource System*) para interferir na alocação de recursos da infraestrutura, realizada pelos algoritmos de domínio (Algoritmo 11 e Algoritmo 12) no VNE_Hlog, através da inserção de um viés α nas equações

(44) e (45), com o objetivo de aumentar de taxa de aceitação de VNR e a aumentar a receita total com o uso da infraestrutura ao longo prazo. Essa abordagem do VNE_CRS *Off-line* confirma a motivação para utilizar o Aprendizado por Reforço por Diferença Temporal descrita na subseção 2.5.2.3, onde a solução deve maximizar ganhos no futuro, mas a tomada de decisão ocorre quando o VNE recebe solicitações no presente, *on-line*.

Como a última etapa deste experimento, na abordagem VNE_CRS *Off-line* + *On-line* a taxa de aceitabilidade tornou-se 12,32 pontos percentuais maior que a abordagem GRC, 4,43 pontos percentuais acima da abordagem VNE_CRS *Off-line* e ainda atingiu uma taxa de aceitabilidade de VNR com 2,58 pontos percentuais acima da abordagem com nuvem pública.

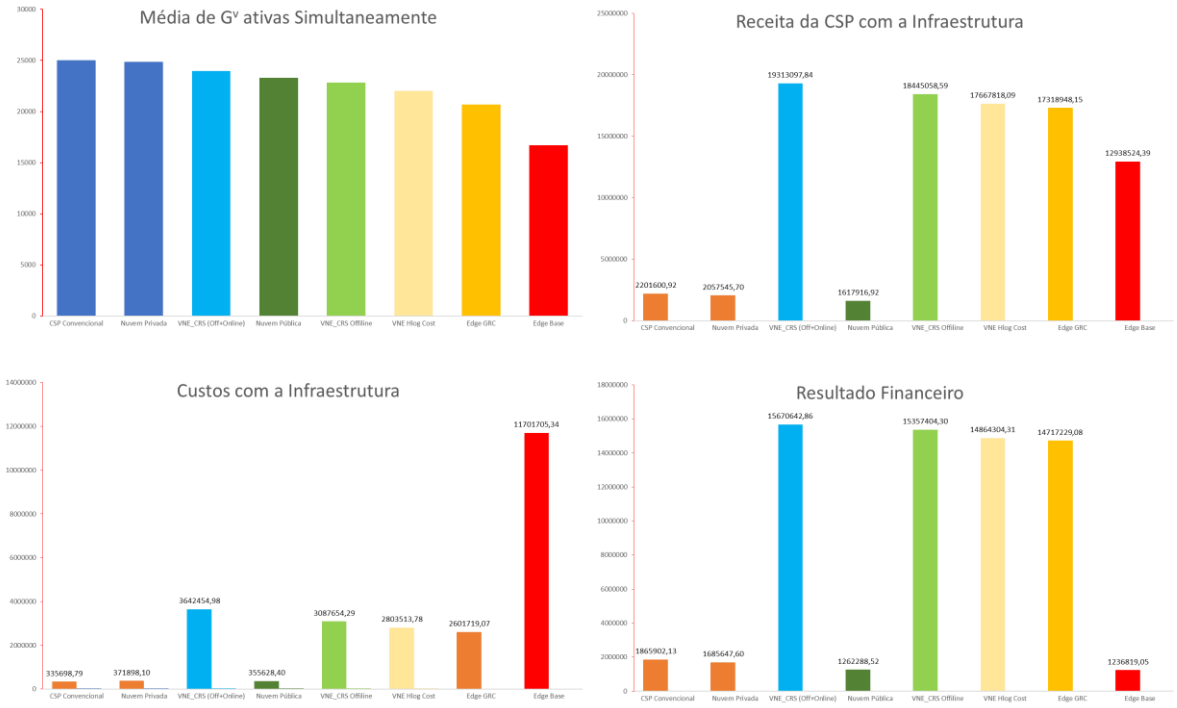
Além da taxa aceitabilidade de VNR, pelos resultados, observa-se que que a abordagem mista possibilitou melhorias na receita e na média de requisitos ativos simultâneos em comparação com o GRC (*baseline*), de 20676 requisições simultâneas para 23955 usando VNE_CRS. Esse resultado mostra que a abordagem VNE pôde otimizar o uso da infraestrutura e ainda aumentar em 15,85% a quantidade requisições ativas simultaneamente.

O VNE_CRS (*Off-line* + *On-line*) usou o conhecimento e os benefícios do VNE_CRS *Off-line* e aumentou os resultados por aprender mais sobre o comportamento da rede Brain da Alemanha durante todas as novas solicitações, *on-line*, de rede carregada iniciando com 1.116.992 estados de espaço e concluindo com 1.123.739 estados de espaço na abordagem mista, ou seja, a abordagem mista aprendeu, durante a interação, 6.747 novos estados de espaço que auxiliaram na sua tomada de decisão.

À primeira vista, comparando a abordagem *off-line* e *on-line*, o novo conhecimento de rede é o único ponto que possibilitou melhores previsões durante o processo de alocação. Mas além desta melhor previsão, o modelo do VNE_CRS utiliza um estado de espaço *S* baseado na taxa de disponibilidade de elementos por uma estratificação da disponibilidade do ambiente, de acordo com a proposta de Mijumbi et al. (2014), que reduz o tamanho do estado do espaço, mas permite que diferentes alocações de recursos gerem o mesmo estado do espaço, tornando a política, a tabela de estado-ação, mais dinâmica o que torna a atualização do espaço de estado e da política constante e mais precisa ao comportamento da rede, porém esse mesmo comportamento também pode gerar imprevisibilidades nos resultados ao longo prazo.

Esta última abordagem oferece o benefício do compartilhamento de recursos do Edge e realiza uma troca entre as necessidades de recursos de CPU e BW que evita investimentos prematuros na infraestrutura computacional e em enlaces de rede para as aplicações que demandam 5QI de ultrabaixa latência.

Figura 44 – Comparativo de receita, conexões e resultado financeiro



Fonte: Autor

Ao avaliar os outros indicadores de resultados, na Figura 44, apesar das abordagens CSP convencional e nuvem privada atenderem um número maior de requisições simultâneas, essas abordagens não remuneram a infraestrutura computacional para os CSPs e não possibilitam o compartilhamento da infraestrutura entre diversas empresas, logo não conseguem atender a conveniência definida por Mell et al.(2011), sessão 2.3. e não otimizam os investimentos com a capacidade computacional.

Ainda na Figura 44, é possível observar que as abordagens de compartilhamento da infraestrutura computacional *edge* (Edge Base, Edge GRC, VNE_Hlog, VNE_CRS *On-line* e VNE_CRS *Off-line* + *On-line*) atendem a necessidade da criação de uma nova fonte de remuneração para as operadoras de telecomunicações, o que reverte a tendencia de redução do ARPU com a transformação das aplicações consolidadas que se tornam apenas *commodities*.

Por fim, o resultado financeiro e a média de atendimento de requisições de rede (Gᵛs) ativas simultaneamente observados na Figura 44 mostram que a abordagem VNE_CRS (*Off-line* + *On-line*) além de atender mais requisições simultânea que a abordagem em nuvem pública ela também foi a abordagem que teve uma melhor relação de retorno, tanto em receita absoluta quanto em resultado líquido financeiro.

Tabela 13 – Utilização de CPU nos nós secundários pelas abordagens de alocação

Nós Substrato	EdgeBase	Edge GRC	VNE_Hlog	VNE_CRS_Offline	VNE_CRS_Misto
Node: ADH10	3,00%	60,96%	51,63%	60,13%	68,79%
Node: ADH11	71,46%	63,21%	54,04%	58,25%	63,37%
Node: ADH21	70,67%	57,29%	49,08%	57,71%	65,13%
Node: ADH22	67,79%	50,38%	47,50%	53,75%	52,08%
Node: ADH23	61,58%	56,08%	48,37%	62,33%	63,46%
Node: ADH24	75,13%	57,08%	46,00%	60,83%	57,75%
Node: ADH3	6,25%	58,62%	46,50%	61,75%	54,42%
Node: ADH5	3,46%	56,33%	48,42%	59,83%	59,79%
Node: ADH6	0,00%	56,13%	49,25%	59,75%	60,50%
Node: ADH63	35,96%	58,92%	49,13%	61,96%	57,67%
Node: ADH7	70,00%	60,50%	51,21%	65,08%	63,42%
Node: ADH8	4,75%	53,92%	43,75%	53,96%	57,21%
Node: ADH9	4,33%	54,71%	49,29%	54,67%	62,42%
Node: CVK1	0,00%	60,46%	54,13%	57,08%	63,75%
Node: CVK10	0,00%	63,17%	50,54%	51,42%	64,63%
Node: CVK11	0,00%	59,29%	61,08%	57,04%	62,79%
Node: CVK12	64,92%	58,08%	58,38%	58,25%	64,29%
Node: CVK2	57,62%	57,54%	57,54%	59,96%	64,21%
Node: CVK20	70,67%	64,92%	61,63%	64,92%	67,96%
Node: CVK21	66,38%	58,38%	59,75%	60,96%	66,92%
Node: CVK22	66,46%	57,25%	59,67%	55,71%	61,33%
Node: CVK23	71,58%	65,58%	57,67%	59,88%	62,42%
Node: CVK24	63,96%	61,38%	54,75%	59,29%	60,75%
Node: CVK3	18,00%	59,04%	54,87%	54,46%	63,21%
Node: CVK4	66,08%	55,17%	54,88%	52,54%	58,13%
Node: CVK42	0,00%	57,67%	54,33%	61,17%	63,13%
Node: CVK5	68,54%	59,63%	56,08%	55,54%	62,54%
Node: CVK6	68,58%	65,71%	61,71%	62,83%	70,42%
Node: CVK7	65,46%	56,17%	57,29%	58,63%	59,33%
Node: CVK8	65,46%	59,92%	58,58%	61,08%	66,83%
Node: CVK9	61,25%	58,29%	56,46%	58,50%	63,88%
Node: HTW1	64,96%	57,42%	56,21%	55,29%	63,08%
Node: HTW10	5,50%	58,46%	49,83%	59,38%	63,13%
Node: HTW11	63,38%	68,67%	57,50%	59,17%	65,96%
Node: HTW13	72,88%	63,42%	57,21%	56,00%	66,46%
Node: HTW21	67,00%	62,79%	50,46%	59,50%	61,67%
Node: HTW22	66,83%	59,96%	52,38%	56,50%	58,46%
Node: HTW23	66,54%	64,71%	56,96%	60,46%	63,58%
Node: HTW24	65,71%	64,92%	56,46%	61,71%	62,58%
Node: HTW5	13,13%	60,83%	49,46%	50,83%	61,08%
Node: HTW63	68,25%	60,75%	52,71%	56,33%	61,88%
Node: HTW64	63,04%	57,50%	57,83%	64,46%	58,88%
Node: HTW7	67,63%	61,79%	56,46%	61,04%	62,29%
Node: HTW9	0,00%	65,58%	57,67%	59,54%	53,33%
Node: HU1	76,21%	52,79%	60,58%	64,79%	65,54%
Node: HU10	4,71%	50,58%	58,96%	64,54%	60,21%
Node: HU17	29,13%	43,88%	55,08%	50,54%	55,58%
Node: HU19	42,67%	52,92%	54,00%	57,63%	60,96%
Node: HU2	69,96%	50,88%	58,75%	61,13%	65,67%
Node: HU20	0,00%	52,38%	47,63%	53,29%	60,00%
Node: HU21	30,58%	45,04%	56,29%	56,67%	61,63%
Node: HU22	68,54%	50,92%	50,63%	54,50%	58,83%
Node: HU23	10,25%	50,46%	57,17%	56,25%	61,75%
Node: HU24	8,75%	53,87%	53,12%	64,54%	61,42%

Nós Substrato	EdgeBase	Edge GRC	VNE_Hlog	VNE_CRS_Offline	VNE_CRS_Misto
Node: HU3	58,83%	44,46%	53,75%	47,71%	56,96%
Node: HU4	72,46%	50,92%	55,75%	62,71%	68,17%
Node: HU43	60,38%	49,17%	53,00%	55,25%	59,63%
Node: HU5	68,96%	61,96%	56,88%	55,46%	58,29%
Node: HU6	0,00%	50,38%	46,42%	59,88%	56,25%
Node: HU7	68,13%	51,04%	55,08%	60,54%	61,38%
Node: HU8	23,71%	46,79%	55,54%	55,08%	59,38%
Node: HU9	17,17%	45,75%	54,46%	56,50%	60,71%
Node: SPK1	64,58%	56,75%	38,04%	56,17%	62,92%
Node: SPK10	0,79%	58,00%	37,25%	55,54%	58,92%
Node: SPK11	71,33%	63,79%	44,17%	63,37%	65,21%
Node: SPK2	70,58%	63,08%	41,83%	55,92%	67,29%
Node: SPK20	0,00%	60,96%	40,88%	61,46%	57,88%
Node: SPK21	60,67%	61,54%	44,50%	59,00%	59,83%
Node: SPK22	63,96%	57,13%	41,25%	58,96%	62,21%
Node: SPK23	68,42%	64,33%	38,42%	59,83%	61,29%
Node: SPK24	62,92%	63,46%	42,00%	60,75%	57,33%
Node: SPK3	65,00%	56,46%	42,17%	54,75%	63,33%
Node: SPK4	0,00%	66,42%	42,46%	57,83%	69,71%
Node: SPK44	0,00%	61,50%	44,79%	54,54%	61,50%
Node: SPK5	68,92%	64,42%	43,08%	65,92%	68,54%
Node: SPK6	57,38%	55,08%	36,29%	54,08%	50,00%
Node: SPK64	0,00%	63,42%	42,38%	59,33%	60,79%
Node: SPK7	64,46%	60,04%	39,17%	56,38%	61,92%
Node: SPK8	68,17%	65,50%	44,83%	57,83%	62,42%
Node: SPK9	0,00%	67,83%	37,13%	51,58%	62,17%
Node: TU1	61,88%	61,29%	60,50%	61,58%	63,75%
Node: TU10	65,67%	60,38%	58,79%	62,00%	63,00%
Node: TU11	67,83%	58,38%	57,21%	58,63%	64,42%
Node: TU12	15,25%	59,33%	53,67%	59,21%	59,63%
Node: TU18	54,58%	53,33%	56,00%	59,58%	60,00%
Node: TU19	32,29%	60,67%	60,42%	58,79%	62,79%
Node: TU2	67,17%	63,42%	59,42%	63,83%	71,00%
Node: TU21	67,46%	60,17%	61,38%	61,67%	59,63%
Node: TU22	74,38%	60,63%	60,92%	61,33%	65,88%
Node: TU23	69,33%	62,13%	62,63%	63,50%	70,71%
Node: TU24	68,17%	60,92%	58,25%	63,00%	63,88%
Node: TU3	67,92%	58,83%	57,96%	60,75%	60,29%
Node: TU4	64,67%	52,08%	54,50%	51,67%	62,46%
Node: TU5	66,08%	60,63%	54,83%	61,00%	64,38%
Node: TU6	62,83%	56,50%	55,13%	53,42%	63,04%
Node: TU63	34,79%	55,25%	58,38%	62,88%	64,50%
Node: TU7	63,50%	57,54%	49,92%	50,54%	59,42%
Node: TU8	0,00%	57,88%	55,33%	61,21%	66,21%
Node: UP1	63,54%	59,92%	50,58%	53,46%	62,46%
Node: UP10	7,04%	62,67%	55,42%	61,46%	66,38%
Node: UP20	0,00%	68,46%	56,63%	63,46%	64,63%
Node: UP21	75,04%	59,88%	62,29%	60,54%	65,79%
Node: UP22	65,25%	63,21%	59,54%	59,87%	61,71%
Node: UP23	65,67%	54,71%	55,88%	53,21%	58,21%
Node: UP24	66,29%	51,63%	54,13%	49,71%	60,75%
Node: UP3	50,63%	63,25%	58,46%	60,71%	63,75%
Node: UP5	25,75%	62,88%	59,38%	58,42%	57,58%
Node: UP7	65,08%	63,04%	57,63%	61,17%	63,13%

Nós Substrato	EdgeBase	Edge GRC	VNE_Hlog	VNE_CRS_Offline	VNE_CRS_Misto
Node: WIAS1	68,37%	58,25%	44,54%	59,83%	63,96%
Node: WIAS10	3,83%	64,46%	57,63%	54,12%	64,71%
Node: WIAS11	24,21%	61,25%	47,67%	65,21%	64,29%
Node: WIAS13	31,54%	65,92%	54,42%	64,83%	66,13%
Node: WIAS2	69,75%	67,21%	44,83%	57,58%	63,54%
Node: WIAS4	14,42%	65,79%	47,29%	61,17%	65,96%
Node: WIAS5	68,08%	60,71%	46,17%	55,58%	64,29%
Node: WIAS6	42,42%	66,42%	49,33%	55,67%	65,58%
Node: WIAS7	70,79%	56,92%	49,50%	56,08%	57,71%
Node: WIAS8	72,75%	64,46%	47,96%	62,46%	63,92%
Node: WIAS9	63,21%	62,08%	48,42%	57,63%	65,29%
Node: ZIB100	66,83%	54,54%	47,50%	56,38%	65,75%
Node: ZIB101	66,79%	52,83%	45,50%	52,88%	58,08%
Node: ZIB102	66,17%	56,54%	51,38%	62,58%	66,58%
Node: ZIB104	0,00%	54,04%	48,29%	59,63%	66,13%
Node: ZIB124	0,00%	50,58%	45,71%	57,38%	65,67%
Node: ZIB143	64,33%	50,88%	44,54%	54,50%	56,38%
Node: ZIB144	0,00%	48,67%	50,50%	59,17%	60,67%
Node: ZIB21	67,42%	49,96%	48,83%	56,75%	63,29%
Node: ZIB22	68,67%	53,25%	44,88%	58,96%	61,25%
Node: ZIB23	74,71%	57,04%	49,63%	60,00%	67,88%
Node: ZIB24	70,38%	50,08%	46,88%	57,33%	63,88%
Node: ZIB61	69,21%	61,79%	49,67%	60,71%	66,67%
Node: ZIB62	7,08%	51,54%	52,08%	53,21%	58,88%
Node: ZIB63	0,00%	52,29%	40,50%	57,67%	59,17%
Node: ZIB64	63,00%	51,50%	45,58%	57,33%	55,17%
Node: ZIB65	11,42%	49,88%	49,04%	55,42%	62,29%
Node: ZIB66	69,17%	63,42%	55,63%	60,12%	60,75%
Node: ZIB67	70,88%	55,75%	55,71%	65,79%	68,42%
Node: ZIB68	14,54%	53,25%	51,08%	51,71%	68,17%
Node: ZIB69	63,50%	53,13%	45,42%	61,83%	63,96%
Node: ZIB70	60,83%	51,00%	42,96%	53,67%	61,92%
Node: ZIB77	0,00%	51,04%	42,63%	58,71%	63,50%
Node: ZIB79	0,00%	56,50%	50,13%	58,13%	60,83%
Node: ZIB81	64,75%	51,88%	47,08%	59,83%	59,71%
Node: ZIB82	0,00%	55,00%	49,17%	64,54%	64,13%
Node: ZIB83	68,25%	49,00%	46,58%	56,00%	62,67%
Node: ZIB85	17,63%	56,71%	40,67%	61,92%	67,71%
Node: ZIB87	28,58%	51,29%	42,54%	64,33%	68,79%
Node: ZIB95	0,00%	49,17%	49,75%	59,13%	62,13%
Node: ZIB96	0,00%	55,58%	46,92%	55,92%	57,46%
Node: ZIB97	0,00%	57,92%	41,13%	51,92%	61,17%
Node: ZIB98	61,71%	50,79%	44,83%	59,54%	57,38%
Node: ZIB99	67,63%	56,96%	49,04%	58,83%	70,58%

Fonte: Autor

7 CONSIDERAÇÕES FINAIS

À medida em que o uso dos sistemas de telefonia móvel aumentou e possibilitou a troca de informações entre pessoas durante a mobilidade, surgiram novas aplicações para se usar os terminais móveis de telecomunicação, como baixar informações da Internet, enviar vídeos e interagir em jogos *on-line*.

Cada evolução tecnológica trouxe de fato muito mais do que apenas aumentos de capacidade ou velocidade de dados; elas são caracterizadas pela entrada de novos serviços e aplicações que impactam, cada uma com suas características, na qualidade percebida pela experiência do usuário (QoE) e, consequentemente, nos investimentos e negócios dos provedores de serviços.

No 5G as novas aplicações necessitam de uma infraestrutura compartilhada de forma conveniente e integrada com elementos de rede e elementos computacionais que tornam o *Network Slicing*, a segurança, a estabilidade e a baixa latência características mais importantes do que a velocidade da transmissão (*throughput*) para as novas aplicações como cirurgias remota ou, ainda, carros conectados e autônomos.

A abrangência de cobertura e conectividade massiva para IOT e para a indústria 4.0 também têm pouca relação com a velocidade de transmissão de dados (*throughput*), mas têm, principalmente, relação com a possibilidade de abrangência de cobertura, pública ou hibrida, e de acesso a um poder computacional distribuído, compartilhado e com acesso por baixa latência, uma infraestrutura habilitadora integrada com *edge computing*, conectividade e *software* em uma arquitetura distribuída e heterogênea.

Como já observado nos documentos das serie “E” da ITU, a percepção de qualidade da infraestrutura de telecomunicações não é uma razão objetiva de um único indicador de QoS de rede, muito menos de apenas uma única característica dentre as 94 indicadas pela ITU (APÊNDICE A) ou de somente um dos serviços prestados pela rede de telecomunicações, como a velocidade de dados em somente um dos domínios da infraestrutura; é sim a união de vários parâmetros em diversos ângulos em todas as camadas (domínios) da rede, acesso, transporte, core e interconexões; e em cada uma de suas aplicações, que devem ser orquestradas em conjunto, de forma a alcançar os requisitos específicos do serviço acordado (SLA) e otimizar os recursos da infraestrutura do provedor de serviços.

Às vistas das aplicações de quinta geração, essa orquestração dos recursos da infraestrutura deve observar tanto as necessidades básicas das aplicações das redes antigas (necessidade de comunicação para segurança, de comunicação para atender necessidades

sociais e de consumo de informação) como também as aplicações que atendem às necessidades das camadas superiores (extensão sensorial e autoliberação), com características de latência, velocidade, distância e continuidade, considerando os diversos domínios de uma arquitetura heterogênea.

A complexidade adicionada por múltiplos domínios e a possibilidade de redes heterogêneas e compartilhadas de forma conveniente faz a orquestração dessa infraestrutura habilitadora tornar-se um desafio, um problema *NP-hard*, onde MIP é ineficiente na otimização da alocação de recursos da rede substrato, o que nos leva a buscar nos mecanismos inteligência artificial uma opção de otimização dos recursos dessa infraestrutura, que é o objeto de experimento deste trabalho.

O experimento realizado nas simulações comparando o VNE_CRS (o algoritmo de aprendizado por reforço proposto neste trabalho) com o algoritmo Edge GRC (*baseline*) apresentou um ganho de 12,32 pontos percentuais na taxa de aceitabilidade de VNRs e 28,80 pontos percentuais quando comparado com a abordagem Edge Base, apresentando uma taxa de aceitabilidade mais próxima à abordagem nuvem pública, mantendo os benefícios de compartilhamento e conveniência de uma arquitetura *Edge*.

Com os resultados desse experimento é possível responder à pergunta de pesquisa deste trabalho e pode-se concluir que, para tornar possível um algoritmo de aprendizado por reforço coordenar múltiplos algoritmos de uma VNE, é necessário que a modelagem matemática de cada domínio contenha um parâmetro único de controle, no caso deste trabalho o custo dos elementos da infraestrutura, que possibilite o RL influenciar na tomada de decisão de cada algoritmo.

Conclui-se também que nesse experimento o uso de técnicas de *Reinforcement Learning* para alocação de recursos VNE em uma infraestrutura multidomínio melhora e simplifica a arquitetura de orquestração da infraestrutura, uma vez que permite reutilizar os algoritmos especializados em cada domínio e atua como um sistema de orquestração completo que visa os resultados estratégicos de longo prazo, observando toda a infraestrutura disponível e as previsões de uso dessa infraestrutura. Conclusão essa que confirma a hipótese proposta neste trabalho.

7.1 RESUMO DAS CONTRIBUIIÇÕES

Dentre as contribuições deste trabalho, as principais foram: uma modelagem dos domínios da rede para uso em aprendizado por reforço, otimização dos recursos de

infraestrutura computacional e de rede em uma arquitetura com distribuição das aplicações e recursos na borda da rede, a formalização de um algoritmo de orquestração dos diversos domínios da infraestrutura computacional e de rede com reaproveitamento dos algoritmos especializados em cada subdomínio da rede e a utilização dos critérios de 5QI no processo de alocação de recursos das redes de quinta geração 5G.

7.1.1 MODELAGEM DOS DOMÍNIOS DE REDE PARA USO EM RL

A primeira contribuição apresentada neste trabalho foi uma modelagem matemática que considerou os diversos domínios da infraestrutura compartilhada e integrada das redes de quinta geração, com uso de três tipos de 5QIs (redes heterogêneas) em arquitetura distribuída com *edge computing*, com o objetivo de maximizar os ganhos no futuro com a tomada de decisão ocorrendo no recebimento das solicitações (*on-line*).

A modelagem inseriu em cada domínio o índice de ações estratégias de custo, um parâmetro balizador α que permite influenciar na percepção de custo de cada algoritmo especializado em seus domínios.

Ainda na modelagem de custos, foram desenvolvidos dois novos modelos de equações de custo que diferem do modelo de custo utilizado pelos algoritmos da bibliografia, um modelo que observa e ordenam a seleção de elementos computacionais e de rede pela quantidade absoluta de recursos disponíveis.

Os dois novos modelos de equações de custos propostos possibilitaram ao agente de RL observar a capacidade percentual disponível, no primeiro modelo, ou a capacidade logarítmica do inverso da disponibilidade percentual de cada recurso de cada elemento computacional e de rede, no segundo modelo.

Essa modelagem nova das equações de custo provou-se mais eficiente fazendo com que a ocupação dos elementos ocorra de forma mais normalizada evitando que elementos críticos sejam completamente ocupados sem necessidade e, consequentemente, aumentando as taxas de aceitabilidade de requisições de rede e a receita total com a utilização da infraestrutura computacional e de rede.

Uma outra contribuição deste trabalho foi o relaxamento do espaço contínuo da ocupação dos elementos da infraestrutura através de uma modelagem baseada na proposta de Wang et al.(2019), que usa a relação de estados da capacidade disponível dos elementos, confluída com a proposta de estratificação de Mijumbi et al. (2014) em 3 classes (espaço inteiro) criando uma abstração do estado do ambiente por uma matriz agrupamento do estado de espaços

pela classificação da ocupação dos elementos, estratificada em três estados: o estado inferior a 30%, o estado intermediário entre 30% e 80% e o estado superior a 80%.

Mesmo com a segmentação do espaço contínuo em 3 classes, para redes com uma quantidade expressiva de elementos – como é o caso das redes celulares comercialmente ativas em 2020 e o caso da rede Brain, (161 nós e 163 enlaces) – a relação de possibilidades do estado de espaço continua exponencial (3^n), o que torna o tempo de execução inviável, ou seja, MIP é ineficiente na otimização da alocação de recursos da infraestrutura.

Concluindo as contribuições pela modelagem, a forma de exploração do ambiente pela interação com o agente de aprendizado por reforço, que pode ser observada na quantidade de 1.123.739 estados de espaço explorados, uma quantidade consideravelmente inferior aos $3^{(324)}$ da exploração completa do ambiente, também se mostrou uma contribuição deste trabalho. Essa interação com o algoritmo de aprendizado por reforço introduz uma complexidade de tempo de execução como sendo O(n), uma vez que o aprendizado por reforço, após sua fase de treinamento, somente realiza uma interação com uma tabela de estado -> ação.

7.1.2 OTIMIZAÇÃO DOS RECURSOS DA INFRAESTRTURA

As novas equações de custos desenvolvidas neste trabalho já proporcionaram uma taxa de aceitabilidade superior às abordagens VNE_GRC e VNE Convencional com uma melhora, respectivamente, de 3,95 pontos percentuais e 20,43 pontos percentuais. Essas equações já representam uma contribuição e um ganho na otimização da utilização dos recursos da infraestrutura computacional e de rede como também são elas que criaram a base do índice de influência (α) que possibilitou a utilização das técnicas de aprendizado por reforço.

Como uma outra contribuição na otimização dos recursos da infraestrutura, o VNE_CRS (*on-line* + *off-line*) permitiu um método de aprendizado continuo com as interações com o ambiente que resultou em um ganho na otimização dos recursos da infraestrutura que pode ser observado pelas melhora, em relação com o Edge GRC (*baseline*), na taxa de aceitabilidade de VNEs (12,32 pontos percentuais), no aumento na receita (11.51%), na resultado financeiro (6.48%) e no aumento da quantidade requisições ativas simultaneamente (15,85%), utilizando os mesmos recursos da infraestrutura.

7.1.3 REAPROVEITAMENTO DOS ALGORITMOS ESPECIALIZADOS

Neste trabalho, foi formalizado um algoritmo orquestração para incorporação de redes virtuais (VNE), o VNE_CRS, inspirado nos ganhos obtidos com as metodologias de gestão por coordenação estratégica das empresas, usando uma estratégia de influência no custo dos recursos de forma a possibilitar o reuso dos algoritmos especializados em cada parte da infraestrutura, visando resultados conjuntos, fim a fim, de longo prazo para maximizar a taxa de aceitação de solicitação de rede e maximizar a receita gerada com a infraestrutura, utilizando uma técnica de Inteligência Artificial chamada Aprendizado por Reforço por Diferença Temporal.

O algoritmo VNE_CRS possibilitou que os trabalhos realizados na otimização dos diversos algoritmos dos domínios da infraestrutura fossem reaproveitados, de forma a não perder os benefícios da especialização desses algoritmos de domínio como também a possibilitar que a entrada de novos domínios ou de novos algoritmos especializados continuem sendo desenvolvidos, sem que haja a necessidade de recriar o orquestrador.

Uma das principais contribuições deste trabalho é a possibilidade de reaproveitamento dos algoritmos especializados em cada domínio. Isso ocorreu porque a nova modelagem das equações de custo, que inseriu um índice (α) que influencia os algoritmos especializados, somada à característica de o aprendizado por reforço ter visão fim a fim de toda a cadeia de alocação de recursos, possibilitou ao VNE_CRS coordenar os algoritmos especializados.

Assim, o algoritmo de orquestração formalizado neste trabalho atua como um influenciador baseado na estrutura organizacional de Mintzberg (LUNENBURG, 2012) por uma coordenação indireta com direcionamento estratégico, em que o sistema supervisiona as ações do agente e influência na condução do resultado de longo prazo, buscando mitigar os conflitos de interesse gerados nas especializações de cada algoritmo de domínio.

7.1.4 UTILIZAÇÃO DOS CRITÉRIOS DE 5QI – PARA REDES 5G

Apesar da normatização dos 5QIs no 3GPP TS 23.501 (2018), os algoritmos VNE apresentados na bibliografia deste trabalho não consideram esse parâmetro na otimização dos recursos da infraestrutura computacional e de rede, e aqueles que consideram os parâmetros de velocidade continuidade e latência referenciados no 5QI fazem essa abordagem somente como restrições em cada domínio da rede e não como um método de otimização com visibilidade a longo prazo.

O algoritmo VNE_CRS formalizado neste trabalho abrangeu uma modelagem matemática que considerou os diversos domínios da infraestrutura compartilhada e integrada das redes de quinta geração, com uma experimentação de três tipos de 5QIs (redes heterogêneas) em arquitetura 5G distribuída com *edge computing*, com o objetivo de maximizar os ganhos no futuro com a tomada de decisão ocorrendo no recebimento das solicitações (*on-line*).

Na modelagem da equação de reforço do RL foi adicionada, em uma forma exponencial com pesos maiores às solicitações de maior criticidade, uma relação de recompensa que inclui os 5QI das redes 5G, o tp5QI (Tabela 8).

A modelagem de recompensa com os 5QIs inserida como reforço do RL possibilitou uma abrangência de senários em que o agente RL aprendeu e priorizou as aplicações de maior valor e criticidade de forma a atender a diferenciação de serviços proposta nas redes de 5G e, por exemplo, reservar recursos estratégicos para atender as aplicações de baixa latência.

7.2 TRABALHOS FUTUROS

Como já demostrado por Erlang (1909), o comportamento de uso da infraestrutura das redes é, de certa forma, previsível, principalmente quando se lida com grandes populações. Esse comportamento é notoriamente sazonal e regionalizado, o que pode possibilitar a especialização de bases de conhecimento comportamental das redes em cada período.

Como trabalho futuro, pode ser desenvolvida uma Rede Neural Artificial (RNA) atuando como direcionador de tráfego para determinar um algoritmo VNE_CRS especializado de acordo com a classificação do comportamento da rede por sua localização e sazonalidade, para cada hora ou mesmo para cada dia da semana. Essa abordagem pode reduzir o tamanho do espaço de estado e pode gerar políticas mais precisas para aumentar a aceitabilidade.

Outros trabalhos futuros também podem ser realizados, como a implementação de um laboratório de redes virtualizadas de quinta geração em um campus, para possibilitar a orquestração não apenas da infraestrutura, mas de todas as aplicações em ambiente controlado, adicionando critérios de qualidade percebida pelos usuários para realimentar o orquestrador com as informações da percepção humana como mais um parâmetro nas equações de recompensa do VNE, de forma a tornar o orquestrador mais alinhado com as expectativas dos usuários.

REFERÊNCIAS

3RD GENERATION PARTNERSHIP PROJECT. **3GPP TR 28.801 V15.1.0: Telecommunication management; Study on management and orchestration of network slicing for next generation network**. 2018.

3RD GENERATION PARTNERSHIP PROJECT. **3GPP TS 22.261 V17.3.0: Technical Specification Group Services and System; Aspects; Service requirements for the 5G system Release 17**. 2020.

3RD GENERATION PARTNERSHIP PROJECT. **3GPP TS 28.530 V16.2.0: Technical Specification Group Services and System; Aspects; Management and orchestration; Concepts, use cases and requirements Release 16**. 2020

3RD GENERATION PARTNERSHIP PROJECT. **3GPP TS 28.541 V16.5.0: Technical Specification Group Services and System; Aspects; Management and orchestration; 5G Network Resource Model (NRM)**. 2020.

ASTUDILLO, Carlos A.; DA FONSECA, Nelson LS; BORIN, Juliana Freitag. Quality of Service Provisioning in LTE-EPON Integrated Networks. In: **Anais do XXIX Concurso de Teses e Dissertações**. SBC, 2016. p. 55-60.

BENKACEM, Ilias et al. Optimal VNFs placement in CDN slicing over multi-cloud environment. **IEEE Journal on Selected Areas in Communications**, v. 36, n. 3, p. 616-627, 2018.

Callet, P. L., Möller, S. and Perkis, A. (2013) "**Qualinet White Paper on Definitions of Quality of Experience (2012)**", European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Jun. 2013. Disponível em: < http://www.qualinet.eu/images/stories/QoE_whitepaper_v1.2.pdf >. Acesso em: 5 fev. 2020.

CALVERT, Ken. Reflections on network architecture: an active networking perspective. **ACM SIGCOMM Computer Communication** Review, v. 36, n. 2, p. 27-30, 2006.

CAO, Haotong et al. Efficient Virtualized Resources Allocation in Network Virtualization Environment: A Service Oriented Perspective. In: **2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)**. IEEE, 2019. p. 1-6

CAO, Haotong et al. On Virtual Resource Allocation of Heterogeneous Networks in Virtualization Environment: A Service Oriented Perspective. **IEEE Transactions on Network Science and Engineering**, 2020.

CHELLAPPA, Ramnath. Intermediaries in cloud-computing: A new computing paradigm. In: **INFORMS Annual Meeting, Dallas**. 1997. p. 26-29.

CHELLAPPA, Ramnath; BARUA, Anitesh; WHINSTON, Andrew B. An electronic infrastructure for a virtual university. **Communications of the ACM**, v. 40, n. 9, p. 56-58, 1997.

CHENG, Xiang et al. Virtual network embedding through topology-aware node ranking. **ACM SIGCOMM Computer Communication Review**, v. 41, n. 2, p. 38-47, 2011.

CHOWDHURY, Mosharaf; RAHMAN, Muntasir Raihan; BOUTABA, Raouf. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. **IEEE/ACM Transactions on networking**, v. 20, n. 1, p. 206-219, 2011.

COSTA, Paolo et al. NaaS: Network-as-a-Service in the Cloud. In: **2nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 12)**. 2012.

DEHURY, Chinmaya Kumar; SAHOO, Prasan Kumar. DYVINE: Fitness-based dynamic virtual network embedding in cloud computing. IEEE Journal on **Selected Areas in Communications**, v. 37, n. 5, p. 1029-1045, 2019.

DI, Hao et al. Efficient online virtual network mapping using resource evaluation. **Journal of Network and Systems Management**, v. 20, n. 4, p. 468-488, 2012.

DOLATI, Mahdi et al. DeepViNE: Virtual network embedding with deep reinforcement learning. In: **IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. IEEE, 2019. p. 879-885.

ERLANG, Agner Krarup. The theory of probabilities and telephone conversations. **Nyt. Tidsskr**. Mat. Ser. B, v. 20, p. 33-39, 1909.

FAJJARI, Ilhem et al. VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic. In: **2011 IEEE international conference on communications (ICC)**. IEEE, 2011. p. 1-6.

GONG, Long et al. Toward profit-seeking virtual network embedding algorithm via global resource capacity. In: **IEEE INFOCOM 2014-IEEE Conference on Computer Communications**. IEEE, 2014. p. 1-9.

GUO, Dongyu et al. QoE-Oriented Resource Optimization for Mobile Cloud Gaming: A Potential Game Approach. In: **ICC 2019-2019 IEEE International Conference on Communications (ICC)**. IEEE, 2019. p. 1-6.

HAERI, Soroush; TRAJKOVIĆ, Ljiljana. Virtual network embedding via Monte Carlo tree search. **IEEE transactions on cybernetics**, v. 48, n. 2, p. 510-521, 2018.

HAYAT, Samira et al. Edge Computing in 5G for Drone Navigation: What to Offload. **IEEE Robotics and Automation Letters**, 2021.

HAYKIN, Simon. **Neural Networks and Learning Machines. 3. ed**. Hamilton, Ontario, Canada: Pearson Prentice Hall, 2009.

HE, Wenchen et al. Markov approximation method for optimal service orchestration in IoT network. **IEEE Access**, v. 7, p. 49538-49548, 2019.

HOUAISS, A.; VILLAR, M. S. **Dicionário Houaiss da Língua Portuguesa**. Rio de Janeiro: Objetiva, 2009 Disponível em: <https://houaiss.uol.com.br> Acesso em 07 de setembro de 2020>. Acesso em: 5 fev. 2020

HOUIDI, Ines; LOUATI, Wajdi; ZEGHLACHE, Djamal. A distributed virtual network mapping algorithm. In: **2008 IEEE International Conference on Communications**. IEEE, 2008. p. 5634-5640.

INTERNATIONAL TELECOMMUNICATION UNION, 2020. **Overview of ITU Workshops on QoS and QoE**. Disponível em: <https://www.itu.int/en/ITU-T/Workshops-and-Seminars/qos/Pages/default.aspx>. Acesso em: 20 Jun. 2020.

IoT Analytics, **State of the IoT 2018**: Number of IoT devices now at 7B – Market accelerating, 2018. Disponível em: <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>. Acesso em: 21 set. 2020

ITU. (2008) "**E.800: Definitions of Terms Related to Quality of Service**", ITU Recommendation E. 800, Sep. 2008. [On-line]. Disponível em: <http://www.itu.int/rec/T-REC-E.800-200809-I> . Acesso em: 2 jan. 2020.

JMILA, Houda. **Dynamic resource allocation and management in virtual networks and Clouds**. 2015. Tese de Doutorado. Evry, Institut national des télécommunications.
JORDAN, Scott. Examining Mobile Broadband Transparency. In: **TPRC Conferences, TPRC**. 2019.

KIBALYA, Godfrey et al. A Reinforcement Learning Based Approach for 5G Network Slicing Across Multiple Domains. In: **2019 15th International Conference on Network and Service Management (CNSM)**. IEEE, 2019. p. 1-5.

KLYMASH, Mykhailo et al. Method for optimal use of 4G/5G heterogeneous network resourses under M2M/IoT traffic growth conditions. In: **2017 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)**. IEEE, 2017. p. 1-5.

KRAUTER, Klaus; BUYYA, Rajkumar; MAHESWARAN, Muthucumaru. A taxonomy and survey of grid resource management systems for distributed computing. **Software: Practice and Experience**, v. 32, n. 2, p. 135-164, 2002.

LACY, A. M. et al. 5G-assisted telementored surgery. **British Journal of Surgery**, 2019.
LI, Xu et al. End-to-End Network Slicing in Radio Access Network, Transport Network and Core Network Domains. **IEEE Access**, v. 8, p. 29525-29537, 2020.

LI, Zhengmao. **Telecommiunication 4.0:** Reinvention of the Communication Network. Singapore: Springer, 2018.

LUNENBURG, Fred C. Organizational structure: Mintzberg's framework. **International journal of scholarly, academic, intellectual diversity**, v. 14, n. 1, p. 1-8, 2012.

MASLOW, Abraham Harold. A theory of human motivation. **Psychological review**, v. 50, n. 4, p. 370, 1943.

McCLELLAN, Miranda. **WebRTC based network performance measurements.** 2019. Tese de Doutorado. Massachusetts Institute of Technology.

MELL, Peter et al. The NIST definition of cloud computing. 2011.

MIJUMBI, Rashid et al. Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In: **2014 IEEE network operations and management symposium (NOMS)**. IEEE, 2014. p. 1-9.

MIJUMBI, Rashid et al. Neural network-based autonomous allocation of resources in virtual networks. In: **2014 European Conference on Networks and Communications (EuCNC)**. IEEE, 2014. p. 1-6.

MINSKY, Marvin Lee; PAPERT, Seymour A. **Perceptrons: an introduction to computational geometry**. Cambridge, Massachussets: Mit Press, 1969.

NAYAK, Peshal; PANDEY, Santosh; KNIGHTLY, Edward W. Virtual Speed Test: an AP Tool for Passive Analysis of Wireless LANs. In: **IEEE INFOCOM 2019-IEEE Conference on Computer Communications**. IEEE, 2019. p. 2305-2313.

NUNNA, Swaroop et al. Enabling real-time context-aware collaboration through 5G and mobile edge computing. In: **2015 12th International Conference on Information Technology-New Generations**. IEEE, 2015. p. 601-605.

ORLOWSKI, Sebastian et al. SNDlib 1.0—Survivable network design library. **Networks: An International Journal**, v. 55, n. 3, p. 276-286, 2010.

OSSEIRAN, Afif; MONSERRAT, Jose F.; MARSCH, Patrick (Ed.). **5G mobile and wireless communications technology**. Cambridge University Press, 2016.

PATRIK CERWALL (Sweden). **Ericsson Mobility Report**. Stockholm, 2019. Disponível em: <https://www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf>. Acesso em: 5 fev. 2020.

Qualcomm Technologies, Inc: **Making 5G NR a reality**. Disponível em: <https://www.qualcomm.com/media/documents/files/whitepaper-making-5g-nr-a-reality.pdf>. Acesso em: 5 fev. 2020.

QUIN, M. parallel programming in Cwith MPI and OpenMP. **Tata McGraw Hills edition**, 2000.

RAHMAN, Waqas Ur; HONG, Choong Seon; HUH, Eui-Nam. Edge Computing Assisted Joint Quality Adaptation for Mobile Video Streaming. **IEEE Access**, v. 7, p. 129082-129094, 2019.

RAISSI, Fatma; YANGUI, Sami; CAMPS, Frederic. Autonomous Cars, 5G Mobile Networks and Smart Cities: Beyond the Hype. In: **2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)**. IEEE, 2019. p. 180-185.

ROBERTAZZI, Thomas G.; SHI, Li. **Networking and Computation. Springer International Publishing**, 2020.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, v. 323, n. 6088, p. 533-536, 1986.
RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. River, Nj, Usa: Pearson, 2010.

SARACCO, Roberto; HARROW, Jeffrey R.; WEIHMAYER, Robert. **The Disappearance of Telecommunications**. New York: Ieee Press, 2000.

SCIANCALEPORE, Vincenzo; YOUSAF, Faqir Zarrar; COSTA-PEREZ, Xavier. z-TORCH: An automated NFV orchestration and monitoring solution. **IEEE Transactions on Network and Service Management**, v. 15, n. 4, p. 1292-1306, 2018.

SHAHRIAR, Nashid et al. Virtual network survivability through joint spare capacity allocation and embedding. **IEEE Journal on Selected Areas in Communications**, v. 36, n. 3, p. 502-518, 2018.

SIMSEK, Meryem et al. 5G-enabled tactile internet. **IEEE Journal on Selected Areas in Communications**, v. 34, n. 3, p. 460-473, 2016.

SOUALAH, Oussama et al. A reliable virtual network embedding algorithm based on game theory within cloud's backbone. In: **2014 IEEE International Conference on Communications (ICC)**. IEEE, 2014. p. 2975-2981.

STATISTA (Org.). **Monthly average revenue per user (ARPU) of mobile voice and data services in Brazil from 2007 to 2018. 2020**. Disponível em: <https://www.statista.com/statistics/524011/mobile-voice-and-data-arpu-in-brazil/>. Acesso em: 7 fev. 2020.

STRATEGY. **2017 Telecommunications Trends: Aspiring to digital simplicity and clarity in strategic identity**. New York City, 2017. Subsidiary of PricewaterhouseCoopers (PwC). Disponível em: <https://www.strategyand.pwc.com/gx/en/trends/2017>. Acesso em: 5 fev. 2020.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement learning: An introduction**. MIT press, 2018.

TANENBAUM, Andrew S. **Redes de computadores quarta edição**. Editora Campus, 2003
VIGNERI, Luigi; SPYROPOULOS, Thrasyvoulos; BARAKAT, Chadi. Quality of experience-aware mobile edge caching through a vehicular cloud. **IEEE Transactions on Mobile Computing**, 2019.

TANWAR, Sudeep; TANWAR. **Fog computing for healthcare 4.0 environments**. Springer International Publishing, 2021.

WANG, Haozhe et al. Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach. **Information Sciences**, v. 498, p. 106-116, 2019.

WANG, Sen et al. VNE-TD: A virtual network embedding algorithm based on temporal-difference learning. **Computer Networks**, v. 161, p. 251-263, 2019.

WATKINS, C. J. C. H. Learning from Delayed Rewards. 1989. **Tese (Doutorado) – Universidade de Cambridge**, Cambridge.

WEI, Fang. TEDI: efficient shortest path query answering on graphs. In: **Graph Data Management: Techniques and Applications**. IGI Global, 2012. p. 214-238.

YAN, Zhongxia et al. Automatic Virtual Network Embedding: A Deep Reinforcement Learning Approach with Graph Convolutional Networks. **IEEE Journal on Selected Areas in Communications**, v. 38, n. 6, p. 1040-1057, 2020.

YAO, Haipeng et al. RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding. **IEEE Transactions on Emerging Topics in Computing**, 2018.

YU, Chunyan et al. PAME: Evolutionary membrane computing for virtual network embedding. **Journal of Parallel and Distributed Computing,** v. 111, p. 136-151, 2018.

YUAN, Ningcheng et al. Delay-Aware NFV Resource Allocation with Deep Reinforcement Learning. In: **NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium**. IEEE, 2020. p. 1-7.

YUAN, Ying et al. A Q-learning-based approach for virtual network embedding in data center. **Neural Computing and Applications**, v. 32, n. 7, p. 1995-2004, 2020.

ZHANG, Qi; CHENG, Lu; BOUTABA, Raouf. Cloud computing: state-of-the-art and research challenges. **Journal of Internet services and applications**, v. 1, n. 1, p. 7-18, 2010.

ZHENG, Guangcong et al. A single-player Monte Carlo tree search method combined with node importance for virtual network embedding. **Annals of Telecommunications**, p. 1-16, 2020.

ZHU, Yong; AMMAR, Mostafa H. Algorithms for assigning substrate network resources to virtual network components. In: **INFOCOM**. 2006. p. 1-12.

APÊNDICE A – RECOMENDAÇÕES ITU E. 800

ITU Recommendation E. 800 - Related to Quality of Service:

I. Service provision
    a. Call set-up
        i. Call set-up time
        ii. Service accessibility performance
        iii. Mean service access delay
        iv. Service user mistake probability
        v. Dialling mistake probability
        vi. Call abandonment probability
    b. Call progress
        i. Interruption; break (of service)
        ii. QoS variable
        iii. Relative QoS
    c. Service restoration/repair
    d. Service quality characteristics
        i. Speed
        ii. Accuracy
        iii. Dependability
        iv. Availability
        v. Reliability
        vi. Simplicity
    e. Charging and billing
        i. Incorrect charging or accounting probability
        ii. Undercharging probability
        iii. Overcharging probability
    f. Billing
        i. Billing error probability
        ii. Billing integrity
    g. Service-specific terms
        i. Mean service provisioning time
        ii. Service integrity performance

iii. voice
    1. Speech quality
    2. Speech transmission quality
    3. Conversational quality
    4. Conversational speech quality
    5. One-way voice transmission quality
    6. Opinion score (in telephony)
    7. Mean opinion score (MOS)

iv. data
    1. Multimedia
    2. Multimedia services
    3. Streaming (in multimedia services)
    4. Download

h. Security-specific terms
    i. Information security
    ii. Data security
    iii. Privacy
    iv. Password
    v. Confidentiality
    vi. Data confidentiality
    vii. Integrity
    viii. Data integrity
    ix. Malware
    x. Hacking
    xi. Phishing
    xii. Virus
    xiii. Worms
    xiv. Trojan horse
    xv. Fraud
    xvi. Forgery
    xvii. Spam

II. Network-related terms
    a. route
        i. Network/user interface

ii. Interconnection
iii. Network accessibility
iv. Connection accessibility
v. Connection establishment error probability
vi. Connection establishment failure probability
vii. Unacceptable transmission probability
viii. No tone probability
ix. Misrouting probability

b. application
i. Trafficability performance
ii. Transmission performance
iii. Bit transparency
iv. Bit error ratio (BER)
v. Error free seconds ratio
vi. Propagation performance
vii. End-to-end IP network
viii. End-to-end IP network performance
ix. Percent IP service availability (PIA)
x. Release failure probability

c. Connection clearing failure probability
d. Connection establishment success ratio (CESR)

III. QoS resource management
a. Class of service
b. Customer relationship management (CRM)
c. Benchmark
d. Compensation schemes
e. Service level agreement (SLA)
f. Service quality agreement (SQA)
g. Time between interruptions
h. Interruption duration
i. Mean time between interruptions (MTBI)
j. Mean time to restoration (MTTR); Mean time to recovery; Mean time to repair
k. Fault coverage

l. Repair coverage
m. Fault
n. Corrective maintenance; repair
o. Reliability performance
p. Mean time to first failure (MTTFF)
q. Mean time to failure (MTTF)
r. Mean time between failures (MTBF)
s. Failure rate acceleration factor
t. Disaster recovery, Business continuity
u. Complaint
v. Directory services

APÊNDICE B – TABELA DE DESCRIÇAO DE ATRIBUTOS PARA SLA

Tabela 14 – Descrição dos principais atributos para SLA da NS

Nome do Atributo	Descrição
availability	Este parâmetro especifica o requisito de disponibilidade do serviço de comunicação, expresso como uma porcentagem. A disponibilidade do serviço de comunicação é definida na cláusula 3.1 do TS 22.261
perfReq	Este parâmetro especifica os requisitos para a sub-rede de fatia de rede em termos dos cenários definidos no TS 22.261 ou seja, os requisitos de desempenho para cenários de alta taxa de dados e densidade de tráfego Estruturado da seguinte forma: - lista do perfReq De acordo com o valor do tipo da NSI, SST (*Slice/Service type*) perfReq pode ser: - lista de requerimentos *eMBB* ou - lista de requerimentos *uRLLC* ou - lista de requerimentos *eMTC*
maxNumberofUEs	Especifica o número máximo de UEs que podem acessar simultaneamente a instância de NSI (*Network Slice Instance* – fatia de rede).
coverageAreaTAList	Especifica uma lista de *TrackingAreas* onde o NSI pode ser selecionado.
latency	Especifica a latência de transmissão do pacote (milissegundo) através da parte RAN, CN e TN da rede 5G e é usado para avaliar o desempenho de utilização da instância de NSI de ponta a ponta.
dLThptPerSlice	Define a taxa de dados (velocidade/*throughput*) alcançável da NSI no *downlink.*
dLThptPerUE	Define a taxa de dados no *downlink* (velocidade/*throughput*) suportada pela NSI por UE.
guaThpt	Define a taxa de dados (velocidade/*throughput*) garantida
maxThpt	Define a taxa de dados (velocidade/*throughput*) máxima
uLThptPerSlice	Define a taxa de dados (velocidade/*throughput*) alcançável da NSI no *uplink.*
uLThptPerUE	Define a taxa de dados no *uplink* (velocidade/*throughput*) suportada NSI por UE.
maxPktSize	Especifica o tamanho máximo do pacote de dados suportado pela NSI.
maxNumberofConns	Especifica o tamanho máximo sessões suportada pela NSI.
MaxNumberofConns.nOofConn	Especifica o tamanho máximo sessões concorrente suportada pela NSI.
supportedAccessTech	Especifica o quais redes de acesso são suportadas pela NSI.
SupportedAccessTech.accTechList	Especifica o quais tecnologias de acesso são suportadas pela NSI. 1: NR (*New Radio*) 2: NB-IoT (*Narrowband* IoT) 3: WI-Fi 4: *Fixed access* (e.g. DSL, Fibra)
coverageArea	Especifica a área de cobertura geográfica que os serviços da NSI estão disponíveis.
termDensity	Especifica a densidade geral do usuário na área de cobertura da NSI. Consulte a Tabela 7.1-1 no TS 22.261
activityFactor	Especifica a quantidade de usuários ativos (transmitindo ou recebendo dados) simultaneamente pela quantidade total de US de uma NSI Table 7.1-1 no TS 22.261.
survivalTime	Especifica o tempo que um aplicativo que consome um serviço de comunicação pode continuar ativo sem uma mensagem de nova solicitação.
jitter	Especifica o desvio do valor aceitável no valor da latência
reliability	Especifica, no contexto das transmissões de pacotes da camada de rede, o valor percentual da quantidade de pacotes da camada de rede enviados e entregues com sucesso TS 22.261.
ipAddress	Este parâmetro especifica o endereço IP atribuído a uma interface / terminal de transporte lógico.

Fonte: Autor “adaptado de” 3GPP TS 28.541 V16.5.0, 2020. p. 159-166

APÊNDICE C – TABELA DE PADRONIZAÇÃO 5QI

Tabela 15 – Padronização do mapeamento de 5QI para características de QoS

5QI	Resource Type	Default Priority Level	Packet Delay Budget [ms]	Packet Error Rate	Default Maximum Data Burst Volume [bytes]	Default Averaging Windows [ms]	Example Service
1	GBR	20	100	0,01		2000	Conversational Voice
2	GBR	40	150	0,001		2000	Conversational Video (Live Streaming)
3	GBR	30	50	0,001		2000	Real Time Gaming, V2X messages Eletricity distribution - medium voltage, Process automation - monitoring
4	GBR	50	300	0,000001		2000	Non-Conversational Video (Buffered Streaming)
65	GBR	7	75	0,01		2000	Mission Critical user plane Push to Talk voice (e.g., MCPTT)
66	GBR	20	100	0,01		2000	Non-Mission-Critical user plane Push To Talk voice
67	GBR	15	100	0,001		2000	Mission Critical Video user plane
5	Non-GBR	10	100	0,000001			IMS Signalling
6	Non-GBR	60	300	0,000001			Video (Buffered Streaming) TCP-based (e.q.., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7	Non-GBR	70	100	0,001			Voice Video (Lve Streaming) Interactive Gaming
8	Non-GBR	80	300	0,000001			Video (Buffered Streaming) TCP-based (e.q.., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9	Non-GBR	90	300	0,000001			Video (Buffered Streaming) TCP-based (e.q.., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
69	Non-GBR	5	60	0,000001			Mission Critical delay sensitive signalling (e.q. example MC-PTT signalling)
70	Non-GBR	55	200	0,000001			Mission Critical Data Video (Buffered Streaming) TCP-based (e.q.., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
79	Non-GBR	65	50	0,01			V2X messages
80	Non-GBR	68	10	0,000001			Low Latency eMBB applications Augmented Reality
82	Delay Critical GBR	19	10	0,0001	255	2000	Discrete Automation
83	Delay Critical GBR	22	10	0,0001	1354	2000	Discrete Automation
84	Delay Critical GBR	24	30	0,00001	1354	2000	Intelligent transport systems
85	Delay Critical GBR	21	5	0,00001	255	2000	Electricity Distribution high voltage

Fonte: Autor “adaptado de” 3GPP TS 23.501 V16.5.1, 2018. p. 146-148