



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**AVALIAÇÃO DA CONECTIVIDADE: FORÇA, TENDÊNCIA E
REGULARIDADE EM REDES OPORTUNISTAS**

WILLIAM CASTRO DA ROSA

MANAUS – AMAZONAS
AGOSTO – 2026



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WILLIAM CASTRO DA ROSA

**AVALIAÇÃO DA CONECTIVIDADE: FORÇA, TENDÊNCIA E
REGULARIDADE EM REDES OPORTUNISTAS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Controle e Automação de Sistemas

Orientador: Prof. Dr. Celso Barbosa Carvalho

MANAUS – AMAZONAS
AGOSTO – 2026

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

R788a Rosa, William Castro da
 Avaliação da conectividade: força, tendência e regularidade em
 redes oportunistas / William Castro da Rosa. - 2026.
 142 f. : il., color. ; 31 cm.

 Orientador(a): Celso Barbosa Carvalho.
 Tese (doutorado) - Universidade Federal do Amazonas,
 Programa de Pós-Graduação em Engenharia Elétrica, Manaus,
 Amazonas, 2026.

 1. Redes oportunistas. 2. Redes tolerantes a atrasos e
 desconexões. 3. Roteamento social. 4. Internet das coisas. 5.
 Aprendizado por reforço. I. Carvalho, Celso Barbosa. II.
 Universidade Federal do Amazonas. Programa de Pós-Graduação
 em Engenharia Elétrica. III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Engenharia Elétrica e de Computação
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

WILLIAM CASTRO DA ROSA

AVALIAÇÃO DA CONECTIVIDADE: FORÇA, TENDÊNCIA E REGULARIDADE EM REDES OPORTUNISTAS

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 20 de julho de 2026.

BANCA EXAMINADORA

Prof. Dr. Celso Barbosa Carvalho - Presidente
Prof. Dr. Gabriel Matos Araújo - Membro Titular 1 - Externo
Prof. Dr. Edjair de Souza Mota - Membro Titular 2 - Externo
Prof. Dr. Waldir Sabino da Silva Júnior - Membro Titular 3 - Interno
Prof. Dr. Florindo Antonio de Carvalho Ayres Júnior - Membro Titular 4 - Interno

Documento assinado eletronicamente

Manaus, 17 de julho de 2026.



Documento assinado eletronicamente por **Celso Barbosa Carvalho, Professor do Magistério Superior**, em 27/07/2026, às 08:59, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Florindo Antonio de Carvalho Ayres Júnior, Professor do Magistério Superior**, em 27/07/2026, às 15:14, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Waldir Sabino da Silva Júnior, Professor do Magistério Superior**, em 31/07/2026, às 10:02, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Edjair de Souza Mota, Professor do Magistério Superior**, em 31/07/2026, às 10:27, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Gabriel Matos Araujo, Usuário Externo**, em 03/08/2026, às 09:56, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3264332** e o código CRC **04411D33**.

Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Bairro Coroado Campus Universitário, Setor Norte
- Telefone: 99271-8954
CEP 69080-900 Manaus/AM - Pavilhão do CETELI. E-mail: ppgee@ufam.edu.br

Referência: Processo nº 23105.024930/2026-23

SEI nº 3264332

AGRADECIMENTOS

A realização deste doutorado representa a conclusão de uma longa jornada, que não teria sido possível sem o apoio, a paciência e a orientação de muitas pessoas e instituições.

Em primeiro lugar, expresso gratidão à minha família – Estrela, César e Allan – pelo suporte em diversos momentos de dificuldade e por terem me ensinado a valorizar o trabalho e o estudo. Agradeço-lhes por serem minhas principais referências morais. Faço uma menção especial a César R. N. da Rosa (1954 – 2025). Seu legado será honrado.

Ao meu orientador, Prof. Dr. Celso B. Carvalho, pela confiança durante estes anos de pesquisa. Suas recomendações foram importantes para a consolidação desta tese e para a minha formação como pesquisador.

Aos professores membros da banca examinadora — Dr. Edjair S. Mota, Dr. Waldir S. da Silva Jr., Dr. Florindo A. C. Ayres Jr. e Dr. Gabriel M. Araujo —, agradeço pelas avaliações e sugestões criteriosas, fornecidas durante o processo de defesa. As ponderações de cada um contribuíram para elevar a qualidade e a solidez deste trabalho.

Aos coautores e parceiros de pesquisa, em especial aos professores Dr. Marcel W. R. da Silva, Dr. Raphael M. Guedes e Dr. André C. Mendes, pelas valiosas discussões e colaborações científicas.

À Universidade Federal do Amazonas (UFAM), em especial à Faculdade de Engenharia Elétrica e de Computação (FEEC) e ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), pela infraestrutura oferecida e pelo suporte institucional.

À Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), pelo apoio financeiro e pela concessão de bolsa de estudo, fundamentais para a dedicação exclusiva e a realização desta pesquisa.

Por fim, a todos que, direta ou indiretamente, estiveram comigo nesta jornada acadêmica, fornecendo uma palavra de incentivo, um conselho ou um gesto de amizade: muito obrigado.

RESUMO

Esta tese apresenta o desenvolvimento e a validação de novos protocolos de roteamento para Redes Oportunistas (OppNets). O roteamento nessas redes é desafiador devido à ausência de conectividade fim-a-fim, exigindo o paradigma *store-carry-forward*, onde os nós armazenam e transportam mensagens fisicamente até que surja uma oportunidade de contato. O problema central consiste em maximizar as taxas de entrega, buscando reduzir o consumo de recursos (bateria e *buffer*) em cenários de alta mobilidade. A primeira contribuição é o Protocolo CASTRO (*Connectivity Assessment: Strength, Trend and Regularity in Opportunistic networks*). Diferente de abordagens baseadas apenas na frequência de encontros, o CASTRO fundamenta-se na análise do histórico integral de interações, segmentando-o em intervalos de conexão direta e de desconexão. A partir desses dados, o algoritmo extrai três indicadores sociais – Força (S), Tendência (T) e Regularidade (R) – que juntos definem a "Consistência" e os "Scores" dos nós, orientando uma difusão controlada de mensagens. A segunda contribuição é o Protocolo QL-CASTRO, que integra o Aprendizado por Reforço (*Q-Learning*) à arquitetura do CASTRO. Este modelo visa mitigar as limitações de latência do protocolo base ao utilizar estimativas dinâmicas de atraso de entrega em vez de apenas métricas sociais. A metodologia de avaliação utilizou o simulador ONE em dois cenários distintos: uma rede densa de pedestres baseada no mapa de Helsinki e uma rede esparsa veicular baseada na malha viária de Manaus (AM), modelada via software QGIS. Os resultados indicam que ambos os protocolos apresentaram desempenho superior quando comparados a protocolos clássicos (PRoPHET e Spray and Wait), bem como a propostas recentes de destaque (HESnW e QoN-ASW), alcançando taxas de entrega próximas a 90% com consumo energético moderado.

Palavras-chave: Redes Oportunistas, Redes Tolerantes a Atrasos e Desconexões, Roteamento Social, Aprendizado por Reforço, Protocolo CASTRO.

ABSTRACT

This thesis presents the development and validation of new routing protocols for Opportunistic Networks (OppNets). Routing in these networks is challenging due to the absence of end-to-end connectivity, requiring the store-carry-forward paradigm, where nodes store and physically transport messages until a contact opportunity arises. The core problem consists of maximizing delivery rates while seeking to reduce resource consumption (battery and buffer) in high-mobility scenarios. The first contribution is the CASTRO Protocol (Connectivity Assessment: Strength, Trend, and Regularity in Opportunistic Networks). Unlike approaches based solely on the frequency of encounters, CASTRO is based on the analysis of the complete interaction history, segmenting it into direct connection and disconnection intervals. From these data, the algorithm extracts three social indicators – Strength (S), Trend (T), and Regularity (R) – which together define node "Consistency" and "Scores", guiding controlled message diffusion. The second contribution is the QL-CASTRO Protocol, which integrates Reinforcement Learning (Q-Learning) into the CASTRO architecture. This model aims to mitigate the latency limitations of the base protocol by using dynamic delivery delay estimates instead of only social metrics. The evaluation methodology used the ONE simulator in two distinct scenarios: a dense pedestrian network based on the Helsinki map and a sparse vehicular network based on the road network of Manaus (AM), modeled using QGIS software. The results indicate that both protocols showed superior performance when compared to well-established and consolidated proposals (PRoPHET and Spray and Wait), as well as notable recent proposals (HESnW and QoN-ASW), reaching delivery rates close to 90% with moderate energy consumption.

Keywords: Opportunistic Networks, Delay and Disruption Tolerant Networks, Social Routing, Reinforcement Learning, CASTRO Protocol.

LISTA DE ACRÔNIMOS

ACK	<i>Acknowledgement</i>
BLE	<i>Bluetooth Low Energy</i>
BO	<i>Buffer Occupancy</i>
CASTRO	<i>Connectivity Assessment: Strength, Trend and Regularity in Opportunistic networks</i>
CSV	<i>Comma Separated Values</i>
DQN	<i>Deep Q-Networks</i>
DRL	<i>Deep Reinforcement Learning</i>
DTNs	<i>Delay/Disruption Tolerant Networks</i>
EEV	<i>Expected Encounter Value</i>
FIFO	<i>First-In, First-Out</i>
GIS	<i>Geographic Information Systems</i>
GUI	<i>Graphical User Interface</i>
HESnW	<i>History of Encounters-Based Spray and Wait</i>
HSA	<i>Harmonic Search Algorithm</i>
IoT	<i>Internet of Things</i>
IPN	<i>Interplanetary Network</i>
MANETs	<i>Mobile Ad Hoc Networks</i>
MDP	<i>Markov Decision Processes</i>
MEMD	<i>Minimum Expected Meeting Delay</i>
MHC	<i>Message Handling Capacity</i>
ONE	<i>Opportunistic Network Environment</i>
OppNets	<i>Opportunistic Networks</i>
OSGeo	<i>Open Source Geospatial Foundation</i>
OSM	<i>OpenStreetMap</i>

POMDP *Partially Observable Markov Decision Process*

PRoPHET *Probabilistic Routing Protocol using History of Encounters and Transitivity*

QGIS *Quantum GIS*

QL-CASTRO *Q-Learning Applied to CASTRO Protocol*

QoN *Quality of Node*

QoN-ASW *Adaptive Spray and Wait based on Quality of Node*

QoS *Quality of Service*

RWP *Random Waypoint*

SIG *Sistema de Informação Geográfica*

SnW *Spray and Wait*

SRC *Sistema de Referência de Coordenadas*

TCP/IP *Transmission Control Protocol / Internet Protocol*

TES *Triple Exponential Smoothing*

TTL *Time-to-Live*

UAVs *Unmanned Aerial Vehicles*

UTM *Universal Transversa de Mercator*

VANETs *Vehicular Ad-hoc Networks*

VDTNs *Vehicular Delay Tolerant Networks*

WDM *Working Day Movement*

WKT *Well Known Text*

WSN *Wireless Sensor Networks*

LISTA DE FIGURAS

Figura 2.1 – Comparação das pilhas de protocolos (internet tradicional vs. DTN)	23
Figura 2.2 – Representação esquemática do mecanismo <i>store-carry-forward</i>	24
Figura 2.3 – Diagrama Comparativo de Topologia: MANET vs OppNet	26
Figura 2.4 – Taxonomia de protocolos de roteamento em redes oportunistas	28
Figura 2.5 – Taxonomia de protocolos com técnicas de roteamento	29
Figura 2.6 – Comparação de encaminhamento: Centralidade global vs parcial	32
Figura 2.7 – Fluxograma gerenciamento de buffer baseado em função de utilidade... 35	
Figura 2.8 – Principais fontes de consumo energético em nós móveis de DTNs	37
Figura 2.9 – Taxonomia geral dos modelos de mobilidade	39
Figura 2.10 – Arquitetura do simulador ONE	42
Figura 3.1 – Transmissão epidêmica de mensagens	45
Figura 3.2 – Modelos <i>fonte</i> e <i>binário</i> da fase <i>Spray</i> – protocolo <i>Spray and Wait</i>	46
Figura 3.3 – Atualização da previsibilidade de entrega no protocolo PРоPHET	48
Figura 4.1 - Exemplo ilustrativo de intervalos de conexão	62
Figura 4.2 - Exemplo ilustrativo de intervalos de desconexão	62
Figura 6.1 – Interação agente-ambiente em um processo de decisão de Markov	83
Figura 8.1 – Mapa do centro da cidade de Helsinki no simulador ONE	101
Figura 8.2 – Mapa da cidade de Manaus no simulador ONE	102
Figura A.1 – Consulta de malha viária pela base OSM.	137
Figura A.2 – Mapeamento das vias da cidade	138
Figura A.3 – Reprojeção de Camada Vetorial	139
Figura A.4 – Mapa recortado e ferramenta <i>Disconnected Islands</i>	140
Figura A.5 – Detecção de ilhas topológicas	140
Figura A.6 – Remoção de ilhas topológicas	141
Figura A.7 – Grafo conectado e explosão de linhas	141
Figura A.8 – Arquivo .wkt para o mapa de Manaus	142

LISTA DE ALGORITMOS

Algoritmo 5.1 – Método <i>updParameters</i>	73
Algoritmo 5.2 – Método <i>updDirScore</i>	73
Algoritmo 5.3 – Método <i>updTrScores</i>	74
Algoritmo 5.4 – Método <i>factor</i>	75
Algoritmo 5.5 – Método <i>renewTTL</i> (Protocolo CASTRO)	76
Algoritmo 5.6 – Método <i>bufferManagement</i>	76
Algoritmo 5.7 – Método <i>Main</i> (Protocolo CASTRO)	78
Algoritmo 5.8 – Método <i>sendMessages</i> (Protocolo CASTRO)	79
Algoritmo 7.1 – Método <i>getEstimatedDelay</i>	93
Algoritmo 7.2 – Método <i>updateTransitiveDelays</i>	94
Algoritmo 7.3 – Método <i>renewTTL</i> (Protocolo QL-CASTRO)	95
Algoritmo 7.4 – Método <i>Main</i> (Protocolo QL-CASTRO)	96
Algoritmo 7.5 – Método <i>sendMessages</i> (Protocolo QL-CASTRO)	98

LISTA DE TABELAS

Tabela 3.1 – Tabela comparativa entre protocolos de roteamento para OppNets.....	60
Tabela 4.1 – Exemplo de cálculo de Tendência à Conectividade	64
Tabela 8.1 – Configurações dos cenários de teste	103
Tabela 8.2 – Padrões adotados para variáveis de análise	103

LISTA DE GRÁFICOS

Gráficos 8.1 – Desempenho para número de nós variável – rede densa	107
Gráficos 8.2 – Desempenho para tamanho de <i>buffer</i> variável – rede densa	109
Gráficos 8.3 – Desempenho para TTL variável – rede densa	111
Gráficos 8.4 – Desempenho para intervalo de geração variável – rede densa.....	113
Gráficos 8.5 – Desempenho para número de nós variável – rede esparsa	116
Gráficos 8.6 – Desempenho para tamanho de <i>buffer</i> variável – rede esparsa	118
Gráficos 8.7 – Desempenho para TTL variável – rede esparsa	120
Gráficos 8.8 – Desempenho para intervalo de geração variável – rede esparsa.....	122

SUMÁRIO

1. INTRODUÇÃO.....	17
1.1 Descrição do Problema	19
1.2 Motivação.....	19
1.3 Objetivos	20
1.3.1 Geral	20
1.3.2 Específicos	20
1.4 Estrutura do Trabalho.....	21
2. FUNDAMENTAÇÃO TEÓRICA	22
2.1 Redes tolerantes a atrasos e desconexões (DTNs).....	22
2.1.1 Origem e evolução	22
2.1.2 Arquitetura em camadas e o protocolo <i>Bundle</i>	23
2.1.3 O paradigma <i>store-carry-forward</i>	24
2.2 Redes Oportunistas.....	25
2.2.1 Diferença entre MANET e OppNet.....	25
2.2.2 Integração com IoT.....	27
2.2.3 Taxonomia e estratégias de encaminhamento	28
2.3 Análise de redes sociais e métricas de grafo.....	30
2.3.1 Fenômeno do Mundo Pequeno	30
2.3.2 Centralidade	31
2.3.3 Previsibilidade e regularidade de contatos.....	33
2.4 Gerenciamento de recursos em nós móveis.....	34
2.4.1 Gerenciamento de <i>buffer</i>	34
2.4.2 Gerenciamento de energia	37
2.5 Simulação de redes móveis.....	39
2.5.1 Modelos de mobilidade.....	39
2.5.2 Simulador ONE.....	41
2.5.3 Software QGIS	43
3. TRABALHOS RELACIONADOS.....	44
3.1 Introdução	44
3.2 Protocolos baseados em inundação e replicação controlada	44
3.3 Protocolos baseados em padrões de contato	47
3.4 Estratégias adaptativas e híbridas.....	50
3.4.1 Protocolo HESnW.....	52
3.4.2 Protocolo QoN-ASW.....	53

3.5	Protocolos baseados em aprendizado por reforço.....	56
3.6	Considerações	58
4.	CONECTIVIDADE E SCORES SOCIAIS.....	61
4.1	Força, tendência e regularidade entre conexões	61
4.1.1	Força de conexão	63
4.1.2	Tendência à conexão	63
4.1.3	Regularidade entre conexões	65
4.2	Scores	69
4.2.1	Scores diretos	69
4.2.2	Scores transitivos	70
5.	PROTOCOLO CASTRO	72
5.1	Introdução	72
5.2	Métodos de suporte.....	72
5.3	Método <i>Main</i>	77
5.4	Análise de complexidade computacional	80
6.	APRENDIZADO POR REFORÇO EM REDES OPORTUNISTAS	82
6.1	Aprendizado de máquina no roteamento de dados	82
6.2	Processos de decisão de Markov	83
6.2.1	Roteamento oportunista sob a quintupla do MDP.....	85
6.3	O algoritmo <i>Q-Learning</i>	86
6.4	Aplicações de <i>Q-Learning</i> no roteamento oportunista	88
6.5	Considerações	89
7.	PROTOCOLO QL-CASTRO	91
7.1	Introdução	91
7.2	Métodos de suporte.....	91
7.3	Métodos principais	95
7.4	Análise de complexidade computacional do QL-CASTRO	98
8.	AVALIAÇÃO EXPERIMENTAL E RESULTADOS.....	100
8.1	Cenários e configurações de teste	100
8.1.1	Cenário 1: Rede densa de pedestres (Helsinki).....	100
8.1.2	Cenário 2: Rede esparsa veicular (Manaus).....	101
8.1.3	Configurações	102
8.2	Critérios de Avaliação.....	104
8.3	Análise de resultados: Rede densa de pedestres.....	105
8.3.1	Variação do número de nós.....	106
8.3.2	Variação do tamanho de <i>buffer</i>	108

8.3.3	Variação do tempo de vida das mensagens (TTL).....	110
8.3.4	Variação do intervalo de geração de mensagens	112
8.4	Análise de resultados: Rede esparsa veicular	115
8.4.1	Variação do número de nós.....	115
8.4.2	Variação do tamanho do <i>buffer</i>	117
8.4.3	Variação do tempo de vida das mensagens (TTL).....	119
8.4.4	Variação do intervalo de geração de mensagens	121
8.5	Síntese do desempenho.....	123
9.	CONCLUSÕES.....	125
	PRODUÇÃO CIENTÍFICA DA TESE	127
	REFERÊNCIAS BIBLIOGRÁFICAS.....	128
	APÊNDICE	137

1. INTRODUÇÃO

A evolução tecnológica dos sistemas embarcados e a onipresença dos dispositivos móveis impulsionaram o desenvolvimento da Internet das Coisas (IoT), caracterizada por uma malha de dispositivos eletrônicos interconectados e dotados de sensores. Tais dispositivos são fundamentais para a coleta, o processamento e o compartilhamento de dados em diversos cenários da sociedade moderna. Para maximizar a eficiência no sensoriamento e na transmissão, empregam-se variadas tecnologias de gerenciamento de recursos de rede, análise em tempo real e automatização de processos.

Nesse contexto, muitos dispositivos IoT possuem mobilidade intrínseca, comunicando-se através de redes cuja conectividade é, por natureza, intermitente. As chamadas Redes Oportunistas (OppNets) operam em ambientes dinâmicos onde os enlaces entre dispositivos são frequentemente rompidos devido a falhas de comunicação, infraestrutura precária, interferência de sinal ou, simplesmente, pela livre movimentação dos nós.

As redes oportunistas são uma evolução das Redes Tolerantes a Atrasos e Desconexões (DTNs) (Boldrini *et al.*, 2014; Gautam e Dev, 2019; Baudic *et al.*, 2020). Elas permitem a comunicação direta e descentralizada entre dispositivos, reduzindo a dependência de infraestruturas centralizadas (como antenas de telefonia celular ou roteadores Wi-Fi) e adequando-se a cenários de alta mobilidade. Nestes ambientes, os nós aproveitam oportunidades esporádicas de contato para a troca de informações. As aplicações práticas são vastas, abrangendo desde a comunicação entre veículos aéreos não tripulados (UAVs) (Bacanli e Turgut, 2020) e cenários de emergência (Xian *et al.*, 2022; Wang *et al.*, 2016), até o sensoriamento em áreas remotas (Azabal *et al.*, 2021; Wu *et al.*, 2022), redes veiculares (VANETs) (Sohail *et al.*, 2023; Vegni *et al.*, 2021; Abbas *et al.*, 2021; Saleem *et al.*, 2021) e operações militares e de segurança (Arafath, Khan e Sunitha, 2017; Deebak e Hwang, 2023).

Para contornar a instabilidade das conexões, a abordagem predominante é o paradigma *store-carry-forward* (armazenar, carregar e encaminhar). Nessa estratégia, os nós armazenam os dados em seus *buffers* locais e os transportam fisicamente até que uma oportunidade de repasse ocorra.

A eficácia do paradigma *store-carry-forward* depende das políticas de armazenamento e repasse de mensagens adotadas pelos protocolos de roteamento. Na literatura especializada, diversos protocolos têm sido amplamente estudados, sendo os modelos Epidêmico (Vahdat e Becker, 2000), PRoPHET (Lindgren, Doria e Schelén, 2003) e *Spray and Wait* (SnW) (Spyropoulos, Psounis e Raghavendra, 2005) alguns dos mais conhecidos. Contudo, abordagens de inundação causam rápida exaustão de recursos, enquanto modelos estritamente probabilísticos apresentam limitações na detecção de nuances da mobilidade.

Recentemente, novos modelos de roteamento foram propostos como adaptações ou fusões de protocolos mais antigos, sendo dotados de funções utilidade, como o protocolo baseado em histórico de encontros HESnW (Gan *et al.*, 2017) e o protocolo baseado na qualidade dos nós QoN-ASW (Cui *et al.*, 2019), que compõem a base comparativa desta tese.

O presente trabalho introduz uma nova e robusta arquitetura de roteamento baseada em padrões de contato. A primeira contribuição central é o Protocolo CASTRO (*Connectivity Assessment: Strength, Trend and Regularity in Opportunistic networks*), que se fundamenta na análise do histórico integral de interações sociais entre os nós da rede. O histórico é segmentado em dois tipos de intervalos temporais: os intervalos de conexão direta e os de desconexão. A partir destes dados, o algoritmo extrai indicadores sociais de Força, Tendência e Regularidade, que determinam a estratégia de encaminhamento.

Buscando mitigar as limitações de latência inerentes ao Protocolo CASTRO, a tese propõe ainda uma segunda contribuição: o Protocolo QL-CASTRO. Este modelo integra o aprendizado de máquina — especificamente o Aprendizado por Reforço (*Q-Learning*) — à topologia oportunista, focando na redução do tempo de atraso das mensagens, enquanto preserva as taxas de entrega do Protocolo CASTRO.

A validação destas propostas foi conduzida de forma analítica, através do simulador ONE (*Opportunistic Network Environment*) (Keränen *et al.*, 2010) em dois cenários distintos: uma rede densa de pedestres (Helsinki) e uma rede veicular esparsa (Manaus).

1.1 Descrição do Problema

Em redes oportunistas, os nós dependem de contatos momentâneos para transmitir mensagens. A mobilidade resulta em conectividade intermitente, tornando incerta a entrega de dados aos destinos antes que seus tempos de vida expirem. Além disso, a replicação indiscriminada de mensagens para garantir a entrega contribui significativamente para o esgotamento de baterias, aumento do tráfego redundante e saturação da ocupação de *buffer*.

A revisão da literatura aponta que a maioria das estratégias foca na duração e frequência dos contatos. Entretanto, existe uma lacuna no que tange à análise sistemática dos intervalos de desconexão. A identificação de padrões de tendência e regularidade nestes períodos de ausência de contato, conforme proposto nesta tese, permite uma predição mais fundamentada para o encaminhamento.

Diante da dificuldade em conciliar altas taxas de entrega e uso eficiente de recursos limitados, emergem as seguintes questões de pesquisa que guiam este trabalho:

1. É viável desenvolver algoritmos que elevem as taxas de entrega de mensagens mantendo um baixo custo energético e de sobrecarga de rede, baseando-se exclusivamente no histórico local de encontros prévios?
2. A integração de técnicas de Aprendizado por Reforço é capaz de reduzir as latências de entrega em ambientes de topologia intermitente e imprevisível?

1.2 Motivação

A definição de rotas em redes oportunistas não pode depender de um conhecimento prévio ou global da topologia, uma vez que a constante movimentação dos nós inviabiliza o uso de mapas estáticos ou tabelas de roteamento tradicionais. Contudo, observa-se que a mobilidade humana e de sistemas automatizados (como frotas de veículos) tende a seguir padrões de deslocamento e relacionamento recorrentes.

A transmissão de dados em cenários de incerteza demanda o consumo de recursos escassos, muitas vezes sem garantia de sucesso. A motivação primária desta tese reside na hipótese – embasada na análise de trabalhos correlatos – de que a exploração analítica de padrões temporais de conexão e, fundamentalmente, de desconexão pode otimizar o processo de roteamento. Adicionalmente, a necessidade de reduzir as altas latências inerentes aos modelos estritamente baseados em histórico motiva a evolução desta arquitetura através do Aprendizado por Reforço (*Q-Learning*). Ao dotar os nós móveis de adaptabilidade dinâmica para a tomada de decisão, esta pesquisa visa conciliar a alta eficiência na entrega de dados com um consumo de recursos de rede mais viável, impulsionando a adoção das redes tolerantes a atrasos na Internet das Coisas e em cidades inteligentes.

1.3 Objetivos

1.3.1 Geral

Propor, avaliar e validar novos protocolos de roteamento para Redes Oportunistas (CASTRO e QL-CASTRO), baseados na extração de padrões de conectividade e em aprendizado por reforço, visando maximizar as taxas de entrega de mensagens com baixo consumo de recursos de rede.

1.3.2 Específicos

- Modelar matematicamente o histórico de contatos locais extraindo as métricas de Força, Tendência e Regularidade para a formulação do Protocolo CASTRO.
- Integrar o algoritmo *Q-Learning* e políticas avançadas de gerenciamento de *buffer* (renovação condicional de TTL, aposentadoria ativa de mensagens e escalonamento *Newest-First*) para a concepção do Protocolo QL-CASTRO, focando na redução de latência.
- Simular e comparar o desempenho dos protocolos propostos frente a modelos clássicos da literatura (PRoPHET, *Spray and Wait*) e propostas

recentes (HESnW e QoN-ASW) sob os critérios de taxa de entrega, latência, sobrecarga em rede e consumo energético.

1.4 Estrutura do Trabalho

Este documento está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica sobre Redes Tolerantes a Atrasos e Desconexões (DTNs) e Redes Oportunistas (OppNets). O Capítulo 3 discute os trabalhos correlatos e o estado da arte em roteamento. O Capítulo 4 apresenta as modelagens matemáticas das métricas sociais (Força, Tendência e Regularidade), o teorema de máxima variabilidade e o cálculo de *Scores* entre nós da rede. O Capítulo 5 detalha a concepção do Protocolo CASTRO.

No Capítulo 6, são explorados os fundamentos de Aprendizado por Reforço e Processos de Decisão de Markov (MDP), que embasam a criação do Protocolo QL-CASTRO, detalhado no Capítulo 7. O Capítulo 8 descreve a metodologia de simulação no software ONE e analisa os resultados experimentais. Por fim, o Capítulo 9 sintetiza as conclusões alcançadas e projeta diretrizes para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Redes tolerantes a atrasos e desconexões (DTNs)

A arquitetura TCP/IP pressupõe a existência de caminhos fim-a-fim completos, baixas taxas de erro e latência reduzida. No entanto, diversos cenários operacionais, caracterizados por conectividade intermitente e topologias dinâmicas, comprometem a eficiência dessas premissas. Para operar nestes ambientes, consolidou-se o paradigma das *redes tolerantes a atrasos e desconexões* (DTNs — *delay/disruption tolerant networks*) (Silva et al., 2016).

Ao contrário das redes *ad hoc* móveis (MANETs), que dependem da existência de um caminho fim-a-fim instantâneo entre os nós, as DTNs não exigem uma rota previamente estabelecida entre origem e destino (Castillo; Juiz; Bermejo, 2024). A comunicação entre os nós da rede utiliza oportunidades de contato para disseminar dados, operando sob restrições de energia, armazenamento e largura de banda (Tekouabou; Alaoui; Gallais, 2021).

2.1.1 Origem e evolução

O conceito de DTN originou-se no desenvolvimento da *internet interplanetária* (IPN), visando mitigar os longos atrasos de propagação e as interrupções orbitais inerentes à comunicação espacial (Silva et al., 2016). Posteriormente, a aplicação das DTNs estendeu-se a cenários terrestres, como as redes veiculares (VDTNs) com tráfego esparso e em regiões rurais desprovidas de infraestrutura fixa (Benamar et al., 2014).

No contexto atual, a integração de DTNs com a Internet das Coisas (IoT) é fundamental para garantir uma operação energeticamente eficiente e sustentável dos nós da rede (Spaho et al., 2024). Por meio da persistência de dados, essa arquitetura permite que os dispositivos operem em ciclos de economia de energia (*sleep states*), contornando a inviabilidade técnica de manter conectividade contínua em hardwares de baixa potência (Ward, Martinez e Chown, 2015).

2.1.2 Arquitetura em camadas e o protocolo *Bundle*

Para gerenciar a heterogeneidade das redes e superar as interrupções de conectividade, a arquitetura DTN introduz a camada de *bundle* (feixe), situada entre as camadas de transporte e de aplicação, de acordo com a Figura 2.1. O *bundle* é uma unidade de dados que encapsula as informações necessárias para processar uma transação completa, tais como identificadores de origem e destino, tempo de vida da mensagem, políticas de prioridade, dados de autenticação e o próprio *payload* da aplicação (Benamar *et al.*, 2014).

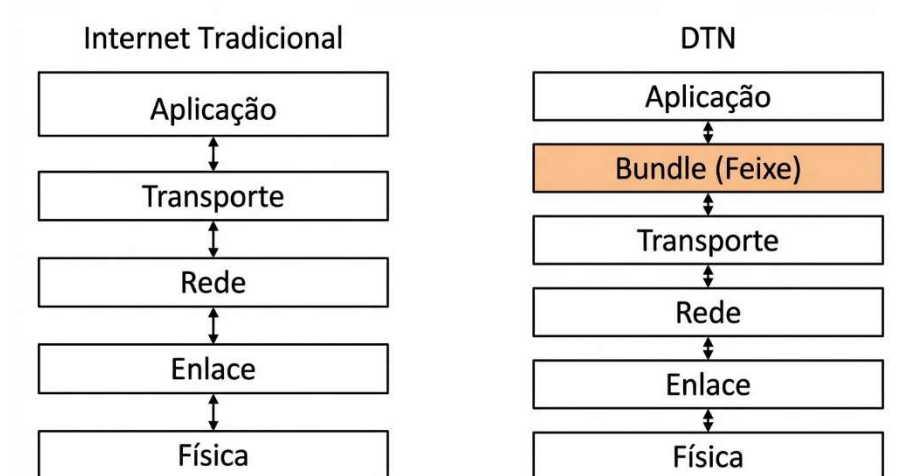


Figura 2.1 – Comparação das pilhas de protocolos (internet tradicional vs. DTN) e a inserção da camada *bundle*. Fonte: Adaptado de Warthman (2015, p.15)

A arquitetura DTN modifica o papel da camada de transporte. Enquanto protocolos como o TCP estabelecem uma conexão contínua fim-a-fim na internet tradicional, a camada de *bundle* segmenta essa comunicação na DTN. Ela utiliza os protocolos das camadas inferiores apenas para transferências salto-a-salto entre nós adjacentes, encerrando essas sessões locais em cada nó intermediário. Esse mecanismo isola os atrasos de cada enlace, permitindo o tráfego assíncrono de dados através de redes distintas (Warthman, 2015).

2.1.3 O paradigma *store-carry-forward*

O mecanismo central que viabiliza a operação das DTNs é o paradigma *store-carry-forward* (armazenar-carregar-encaminhar), segundo o qual os nós da rede aproveitam suas mobilidades para transportar dados fisicamente.

O processo ocorre em três etapas (Lee, Seo e Chung, 2018):

1. *Store* (armazenar): Quando um nó gera ou recebe uma mensagem e não há conexão disponível com o destino ou um próximo salto adequado, a mensagem é armazenada em um *buffer* de memória não volátil.

2. *Carry* (carregar): O nó transporta a mensagem enquanto se move pelo ambiente. A mobilidade do nó é explorada para cobrir a distância física entre ilhas de conectividade desconectadas.

3. *Forward* (encaminhar): Quando uma oportunidade de contato ocorre com outro nó, a mensagem pode ser transferida ou replicada, em um processo de tomada de decisão baseado em métricas de predição ou de utilidade.

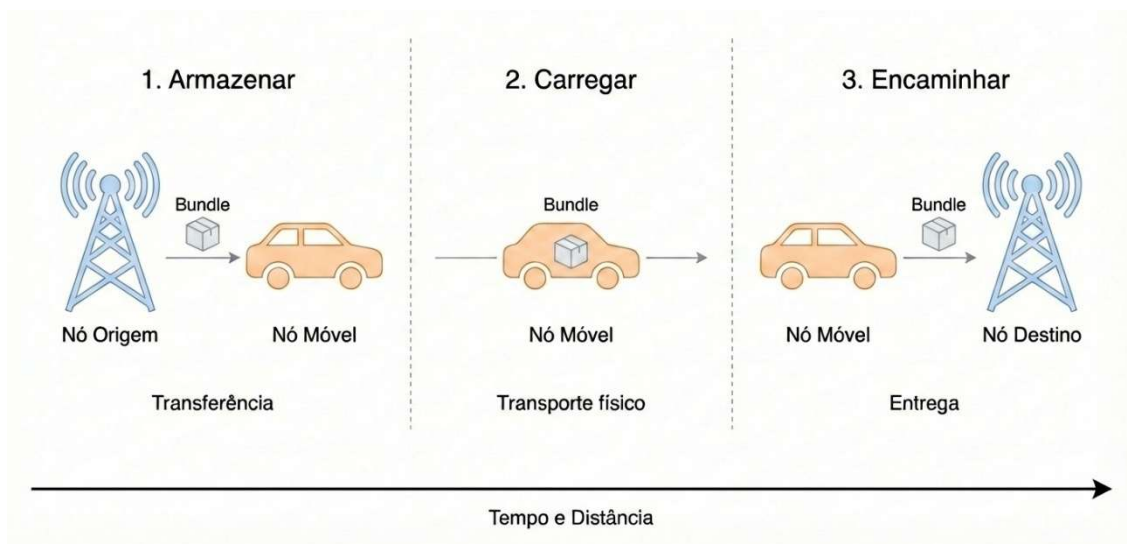


Figura 2.2 – Representação esquemática do mecanismo *store-carry-forward*

Na Figura 2.2, o diagrama ilustra o processo de comunicação em três etapas: (1) o nó móvel recebe e armazena o *bundle* do nó de origem; (2) o nó móvel transporta fisicamente os dados por uma região sem conectividade; e (3) o *bundle* é encaminhado ao nó de destino.

A eficácia deste paradigma depende da capacidade de armazenamento dos nós e das estratégias de roteamento, as quais devem ser projetadas para evitar o esgotamento do *buffer* e o desperdício de recursos de rede (Huang *et al.*, 2024; Chen, *et al.*, 2024b).

2.2 Redes Oportunistas

As *redes oportunistas* (OppNets) constituem uma evolução das redes sem fio tradicionais e uma subclasse das DTNs, caracterizando-se por não assumirem a existência de um caminho completo entre dois nós que desejam se comunicar (Cuka *et al.*, 2018). Essas redes são definidas por Bharamagoudar e Saboji (2017) como arquiteturas sem fio únicas que permitem a comunicação entre nós móveis em ambientes onde não existe rota contínua entre a origem e o destino, sofrendo frequentemente de topologias instáveis e latências prolongadas.

A operação das redes oportunistas fundamenta-se na exploração de contatos espontâneos. Em redes sem fio esparsas, a ausência de uma rota completa obriga o sistema a adotar o paradigma de *store-carry-forward* (Gan, Zhou e Wei, 2017), onde a entrega bem-sucedida ocorre apenas quando um dos nós portadores da mensagem encontra o nó de destino.

2.2.1 Diferença entre MANET e OppNet

Embora as redes oportunistas compartilhem similaridades com as redes *ad hoc* móveis (MANETs), como a ausência de infraestrutura fixa, suas premissas operacionais divergem quanto ao tratamento da conectividade. Conforme observado por Sakthivel e Balaram (2021), os protocolos tradicionalmente empregados em MANETs consideram, em geral, a existência de conectividade suficiente para a descoberta e a manutenção de rotas multissaltos (*hop-by-hop*), ainda que a mobilidade dos nós provoque mudanças frequentes na topologia. Em redes sociais móveis e oportunistas, por outro lado, a elevada dinamicidade da mobilidade pode resultar em contatos intermitentes e na ausência de caminhos fim a fim continuamente disponíveis. Nesse contexto, apesar da

instabilidade da conectividade física, os contatos entre os nós podem apresentar padrões de recorrência e previsibilidade (Sakthivel e Balaram, 2021). Essas características permitem que o roteamento seja orientado por informações de contexto extraídas do histórico de interações, como a análise sistemática de intervalos de conexão e desconexão.

Enquanto as MANETs tentam manter uma conexão fim-a-fim, as redes oportunistas criam uma topologia de roteamento dinâmica ao identificar *nós oportunistas* que podem servir como o próximo salto, caso sejam capazes de aproximar a mensagem do destino (Lohachab e Jangra, 2019). Devido ao comportamento dinâmico de dispositivos móveis e objetos inteligentes, a comunicação ocorre em tempo de execução (Kosmides e Lambrinos, 2018). Isso a torna dependente da disponibilidade momentânea de links de comunicação, sem a garantia de caminhos pré-estabelecidos.

A Figura 2.3 ilustra a diferença na abordagem de conectividade e roteamento entre redes *ad hoc* móveis e redes oportunistas.

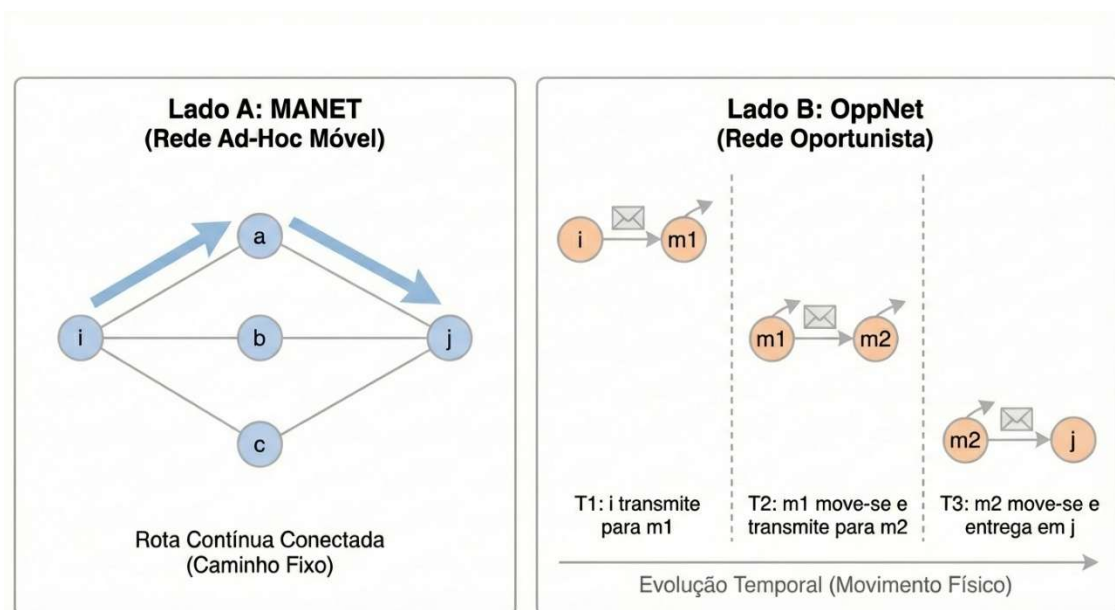


Figura 2.3 – Diagrama Comparativo de Topologia: MANET vs OppNet

O Lado A representa uma MANET tradicional. Neste modelo, a premissa básica é que, para haver comunicação entre um nó fonte (*i*) e um nó de destino (*j*), deve existir uma rota completa formada por enlaces ativos no momento da

transmissão. A seta azul destaca uma rota ($i \rightarrow a \rightarrow j$) que está disponível no instante de partida. Se a conectividade fosse quebrada em qualquer ponto desse caminho, a comunicação falharia até que uma nova rota completa fosse descoberta.

O Lado B demonstra o funcionamento de uma rede oportunista, que opera sob a premissa de conectividade intermitente. Não há um caminho direto de i para j . A figura mostra a "evolução temporal" da entrega da mensagem em três etapas (T1, T2, T3), utilizando o paradigma *store-carry-forward*.

Portanto, enquanto a MANET depende de conectividade fim-a-fim instantânea no domínio espacial, a rede oportunista explora a mobilidade dos nós e a dimensão temporal para estabelecer um caminho espaço-temporal, superando a ausência de links contínuos.

2.2.2 Integração com IoT

A convergência das redes oportunistas com a Internet das Coisas deu origem a novos paradigmas de computação descentralizada. A *IoT oportunista* é descrita por Lohachab e Jangra (2019), na qual a disseminação de dados coletados representa um dilema, visto que provedores e consumidores de dados desconhecem as posições em tempo real uns dos outros. Os autores destacam que a infraestrutura escassa e a conectividade esporádica complicam a tarefa de roteamento, exigindo mecanismos que não dependam de conhecimento prévio da topologia global. Nesse sentido, o desempenho do sistema é pautado por aspectos como:

Heterogeneidade e restrições de recursos: As redes IoT diferem das redes empresariais tradicionais por consistirem em coleções heterogêneas de nós com recursos limitados que interagem estreitamente com seu ambiente, exigindo protocolos leves (Cuka *et al.*, 2018).

Aplicação em monitoramento remoto: As redes oportunistas podem ser aplicadas em sistemas de monitoramento de vida selvagem (Ayele, Meratnia e Havinga, 2018). Em cenários onde a infraestrutura celular é inexistente ou custosa, a arquitetura proposta utiliza nós sensores com rádio de curto alcance

(*Bluetooth Low Energy*) para comunicação par-a-par e rádio de longo alcance (LoRa) para *gateways*, indicando que a rede oportunista tem potencial para preencher lacunas de cobertura em ambientes hostis e remotos.

Influência social no roteamento: Estruturas sociais afetam diretamente os algoritmos de roteamento (Kosmides e Lambrinos, 2018). A localização do usuário e os padrões de mobilidade estão associados a relacionamentos sociais, tornando este um fator importante para a predição de contatos futuros e a eficiência da entrega.

2.2.3 Taxonomia e estratégias de encaminhamento

A Figura 2.4 apresenta uma taxonomia hierárquica para classificar os protocolos de roteamento utilizados em redes oportunistas. No primeiro nível, a classificação divide os protocolos entre os dependentes de algum tipo de suporte de rede (com infraestrutura, fixa ou móvel) e os que operam de forma totalmente *ad hoc* (sem infraestrutura centralizada).

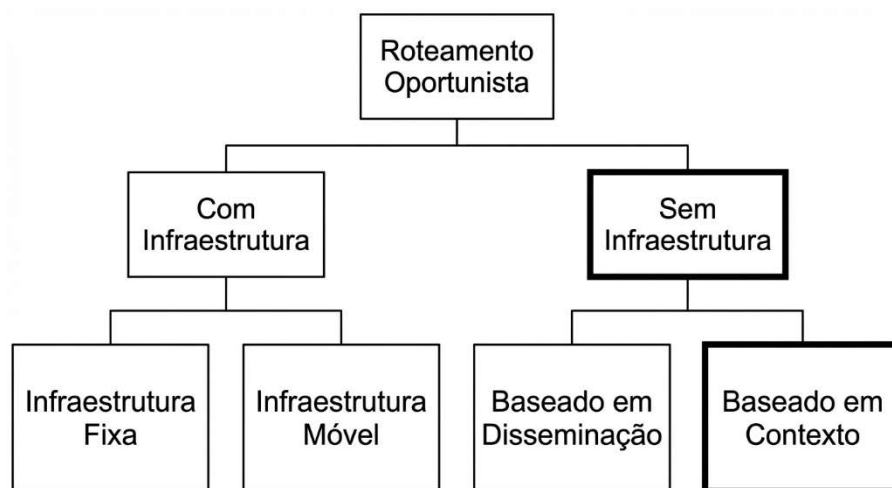


Figura 2.4 – Taxonomia de protocolos de roteamento em redes oportunistas.
Fonte: Adaptado de Pelusi et al (2006), Lohachab e Jangra (2019).

O diagrama destaca, através de bordas mais espessas, o ramo "sem infraestrutura", dividido em duas categorias – disseminação e contexto (Pelusi et al., 2006; Lohachab e Jangra, 2019). Os protocolos baseados em disseminação utilizam técnicas de inundação (*flooding*) ou replicação controlada para entregar

mensagens, enquanto os baseados em contexto utilizam informações como histórico de contatos, localização ou previsibilidade de movimento para tomar decisões de encaminhamento mais inteligentes e eficientes. Esta última categoria também está destacada, indicando o foco da presente pesquisa.

Complementando a estrutura da Figura 2.4, uma segunda estrutura taxonômica é proposta por Avoussoukpo *et al.* (2021) para categorizar os modelos de decisão utilizados no encaminhamento de dados pela rede. Os autores dividem o roteamento em tipos clássicos (cópia única, inundação e probabilístico) e mapeiam uma vasta gama de técnicas de decisão, como predição de links, teoria dos jogos, confiança social, aprendizado de máquina, entre outras (Figura 2.5). Para o escopo desta pesquisa, essa taxonomia é relevante, pois evidencia que as técnicas mais sofisticadas — especialmente a predição de links e a avaliação de experiência social — dependem do histórico de encontros para calcular funções de utilidade e mitigar o custo de roteamento.

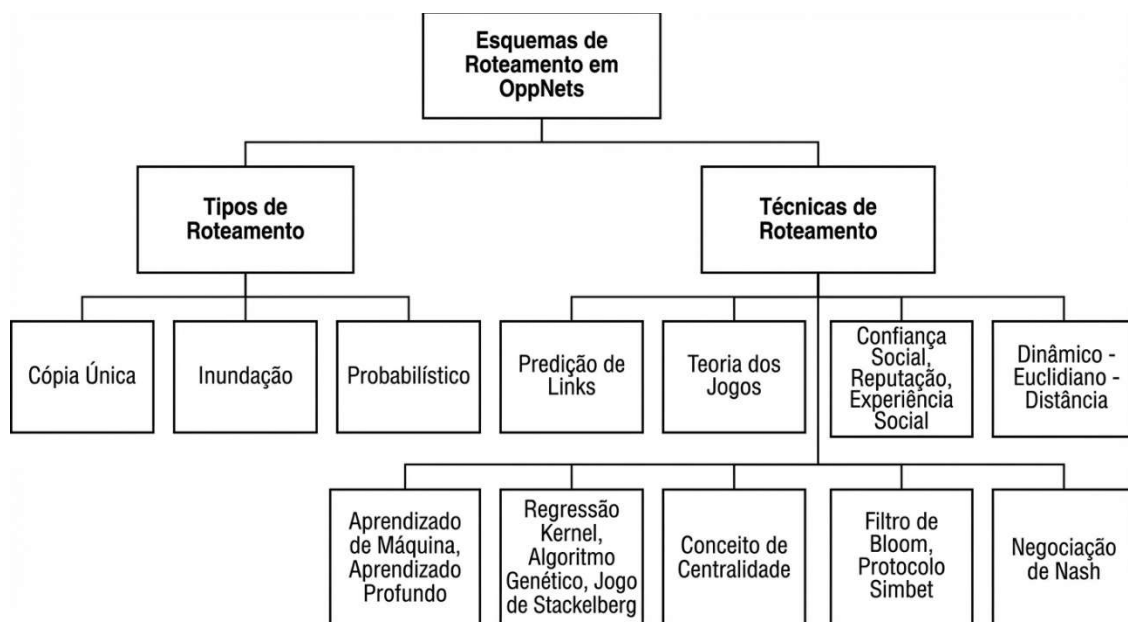


Figura 2.5 – Taxonomia de protocolos com técnicas de roteamento.

Fonte: Adaptado de Avoussoukpo *et al* (2021).

Diversos protocolos oportunistas utilizam o histórico de interações para realizar previsões de conectividade. Yuan *et al.* (2016) categorizam essas propostas como *zero-information* ou *information-rich*; nesta última, a tomada de decisão fundamenta-se em padrões sociais ou no histórico de contatos entre os

nós para otimizar o roteamento. A seleção de nós intermediários é categorizada por Sobin *et al.* (2016) como puramente oportunista ou baseada em métricas sociais. Dentre essas métricas, o histórico modela laços sociais baseados em características de indivíduos, grupos ou locais (Zhang *et al.*, 2021), permitindo decisões mais precisas que a simples inundação.

O encaminhamento de dados ou mensagens é classificado por Zhu e Lu (2007) em função da dependência de conhecimento (local *versus* global). O uso de históricos alinha-se às abordagens de conhecimento local, viabilizando um encaminhamento temporalmente diverso que reage a quebras de rotas sem exigir topologia global. Por fim, para tráfegos exigentes como em *redes de sensores multimídia* (WMSN), o histórico de qualidade dos enlaces deve integrar objetivos de otimização, como *Qualidade do Serviço* (QoS), prevenção de congestionamento, eficiência energética, entre outros (Shen e Bai, 2016).

2.3 Análise de redes sociais e métricas de grafo

2.3.1 Fenômeno do Mundo Pequeno

O *Fenômeno do Mundo Pequeno* estabelece que indivíduos em redes sociais se conectam por cadeias curtas de conhecidos. Sob uma perspectiva algorítmica, os nós de uma rede são capazes de construir tais cadeias de forma descentralizada (Kleinberg, 2000), guiando-se exclusivamente por informações locais.

Em redes oportunistas, essa dinâmica é refletida na mobilidade dos nós, cujo histórico de encontros revela duas propriedades fundamentais para o roteamento: o 'comprimento médio de caminho reduzido' e o alto 'coeficiente de agrupamento' (Wei *et al.*, 2015).

Tais propriedades permitem otimizar o equilíbrio entre a eficiência da entrega e o custo de rede. Especificamente, o comprimento médio de caminho reduzido justifica a imposição de limites de saltos (*hops*), uma vez que o princípio dos 'seis graus de separação' indica que mensagens podem alcançar destinos passando por poucos intermediários. Essa premissa possibilita restringir a propagação e

reduzir o desperdício de recursos – bateria, *buffer* e largura de banda – sem comprometer a eficácia do sistema (Xu et al., 2015).

Paralelamente, o alto agrupamento refina a seleção de intermediários (*relays*). O histórico de contatos permite priorizar nós que, embora não possuam conexão direta com o destino, interagem frequentemente com vizinhos próximos a ele (Wei et al., 2015). Ao expandir essa lógica, a *autoridade de um nó* pode ser inferida a partir de sua centralidade de grau (Xu et al., 2015), transformando decisões puramente locais em uma estratégia eficaz para direcionar mensagens aos elementos mais influentes da topologia.

2.3.2 Centralidade

A centralidade quantifica a importância estrutural de um nó e sua relevância no fluxo de informações da rede. Métricas baseadas em caminhos mínimos — como proximidade (*closeness*), intermediação (*betweenness*) e tensão (*stress*) — fundamentam-se na análise das geodésicas de um grafo (Brandes, 2001). Contudo, o cálculo ou a estimativa precisa desses índices requer o conhecimento global da topologia (Brandes e Pich, 2006). Tal requisito torna-se proibitivo em redes oportunistas, dada a natureza dinâmica e intermitente dessas arquiteturas, que impede a manutenção de uma visão topológica completa em tempo de execução.

No roteamento baseado em contexto, a centralidade pode ser estimada localmente. Islam et al. (2017) apontam que a frequência e a duração de contatos se correlacionam com a centralidade global, servindo como *proxies* para decisões distribuídas. Nesse sentido, Kim e Kim (2016) utilizam a intermediação de vizinhança (*ego-betweenness*) e coordenadas geográficas para identificar potenciais *hubs*.

A mensuração da centralidade admite estratificação e ponderação temporal. Borrego et al. (2019) segmentam essa métrica em duas camadas: eletrônica (baseada no grau de encontros físicos) e virtual (fundamentada na intermediação social). Para refletir a dinamicidade da rede, os autores aplicam uma função de

decaimento exponencial sobre a camada eletrônica, garantindo que contatos antigos tenham menor relevância no cálculo da centralidade atual.

Apesar das recentes soluções propostas na literatura, o roteamento baseado em centralidade enfrenta dois grandes desafios: o consumo de energia e a precisão do destino. Como os nós centrais são frequentemente selecionados como intermediários, suas baterias esgotam-se rapidamente. Para amenizar esse efeito, Socievole e De Rango (2015) propõem modular o valor da centralidade com base na energia residual do nó, equilibrando a carga na rede. Além disso, um nó globalmente central pode não ter relação com o destino da mensagem. Para refinar essa seleção, Yuan *et al.* (2014) introduzem a *centralidade parcial*, que utiliza o histórico de contatos para medir a importância de um nó exclusivamente em relação à comunidade do destinatário, evitando o envio de dados para nós que são populares, mas irrelevantes para aquela rota específica.

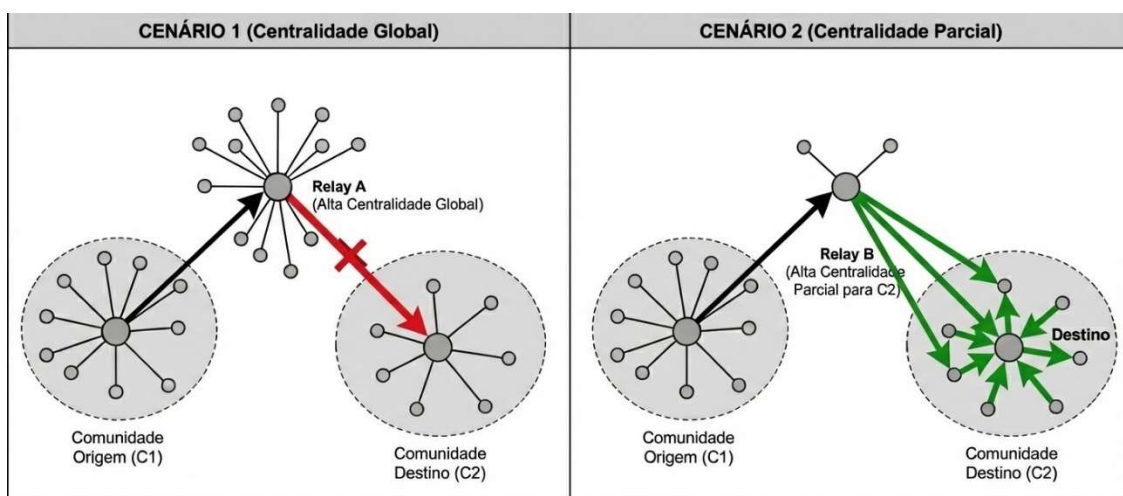


Figura 2.6 – Comparação de encaminhamento: Centralidade global vs parcial. Baseado em Yuan et al. (2014)

A Figura 2.6 ilustra o conceito apresentado por Yuan *et al.* (2014), comparando as centralidades global e parcial em dois cenários. No cenário 1, o intermediário (*relay*) A é dotado de alta centralidade global e possui muitas conexões na rede. No entanto, ele possui pouca ou nenhuma conexão direta com a comunidade de destino C2. Isso resulta em uma falha no encaminhamento da mensagem, pois ele não é um candidato adequado para alcançar o destino específico.

Por outro lado, o cenário 2 mostra um intermediário (*relay*) B que é dotado de menor centralidade global, mas possuidor de alta centralidade parcial em relação à comunidade de destino C2. Isso significa que ele mantém fortes conexões históricas com os membros da referida comunidade, possibilitando um encaminhamento bem-sucedido e eficiente da mensagem.

2.3.3 Previsibilidade e regularidade de contatos

A mobilidade dos nós em redes oportunistas raramente é aleatória; ela é fortemente influenciada por rotinas diárias que geram padrões espaciais e temporais. Segundo Pirozmand *et al.* (2014), os integrantes das redes tendem a visitar um conjunto restrito de locais com alta probabilidade e o comportamento periódico explica a maior parte da mobilidade humana. Essa característica torna os padrões de contato altamente previsíveis. Corroborando essa visão, Tulu *et al.* (2020) apontam que a previsibilidade da mobilidade se baseia em padrões de movimento recorrentes, os quais viabilizam a estimativa dos próximos locais visitados e a duração dos contatos.

Essa previsibilidade transforma o histórico de encontros em uma importante informação para a tomada de decisão no roteamento. Chen e Lou (2016) demonstram que observações de mobilidade permitem previsões de alta precisão. Para isso, os autores propõem métricas como o *valor de encontro esperado* (*expected encounter value* - EEV) e o *atraso mínimo esperado de encontro* (*minimum expected meeting delay* - MEMD), que são calculados diretamente a partir dos intervalos e da frequência dos encontros registrados no histórico dos nós.

A exploração desses padrões regulares molda as estratégias de encaminhamento baseadas em contexto físico. Diversas abordagens preditivas foram analisadas por Kaiser *et al.* (2023), dentre as quais o *Predict-and-Relay* (PER), que utiliza padrões de movimento semideterminísticos para identificar intermediários, e o *Discover-Predict-Deliver* (DPD), onde o histórico de mobilidade individual determina a probabilidade de entrega da mensagem.

Além de otimizar o roteamento, a recorrência dos contatos é fundamental para a descoberta de serviços e objetos. Roopa *et al.* (2019) definem, como um aspecto intrínseco da rede, a descoberta baseada no histórico de encontros e nos padrões de mobilidade. Nesse contexto, trajetórias semelhantes e relações sociais de longo prazo — evidenciadas por encontros frequentes no passado — são utilizadas para garantir que a troca de dados ocorra de forma confiável e com menor latência.

2.4 Gerenciamento de recursos em nós móveis

2.4.1 Gerenciamento de *buffer*

Em DTNs, o roteamento *store-carry-forward* exige que os nós retenham mensagens até que surja uma oportunidade de encaminhamento. As restrições de armazenamento em *buffer* incentivam a adoção de políticas de descarte, que podem impactar a taxa de entrega e o atraso na rede (Prasad *et al.*, 2024). Para sistematizar essas estratégias, Ezife *et al.* (2017) classificam-nas com base na dependência do protocolo de roteamento e na dinamicidade da política de descarte. Ratificando a necessidade de abordagens dinâmicas, Sati *et al.* (2016) analisaram 121 combinações de políticas de fila e descarte, indicando que não existe uma solução única ideal para todos os cenários.

Políticas tradicionais (ex: FIFO) podem causar o fenômeno de *troca cega* (*blind swap*), onde mensagens críticas são descartadas para acomodar dados de baixa utilidade recém-chegados. Para mitigar isso, Le (2026) propõe uma política que compara funções de utilidade, representada pelo fluxograma da Figura 2.7.

Ao receber uma nova mensagem m , o nó verifica a disponibilidade de espaço em seu *buffer*. Caso haja capacidade, a mensagem é armazenada de imediato. Contudo, se o *buffer* estiver cheio, o algoritmo calcula a utilidade da mensagem recém-chegada (U_{new}) e a compara com a utilidade da mensagem menos relevante já armazenada (U_{min}), consultando a lista do *buffer* que se mantém ordenada por esse critério. A substituição só ocorre se houver um ganho líquido

positivo, ou seja, se a utilidade da nova mensagem for estritamente maior que a da mensagem residente ($U_{new} > U_{min}$). Nesse cenário, a mensagem de menor valor é descartada para dar lugar à nova; caso contrário, a nova mensagem é rejeitada. Essa lógica garante que os recursos escassos do nó sejam alocados apenas aos dados que proporcionam o maior benefício para a rede.

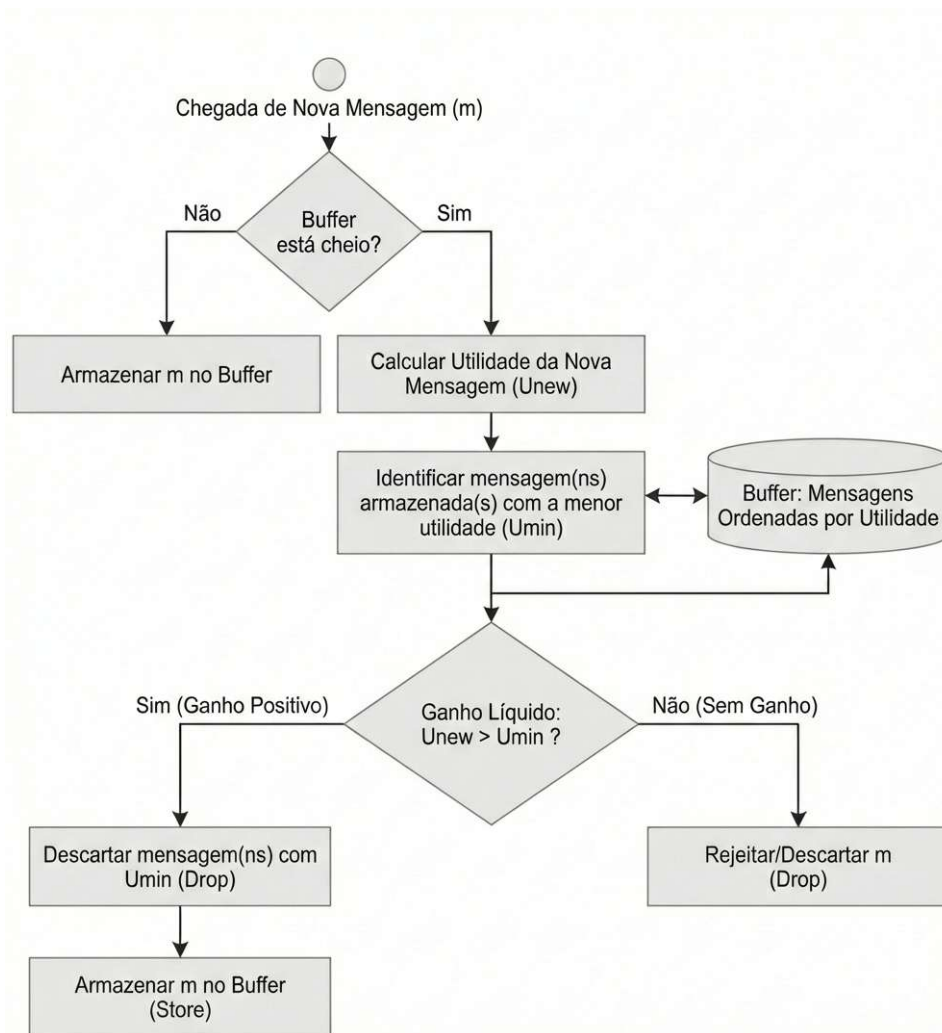


Figura 2.7 – Fluxograma gerenciamento de buffer baseado em função de utilidade. Baseado em Le (2026)

A formulação dessas funções de utilidade pode ser adaptada para atender a diferentes objetivos de rede. Focando em transmissões para múltiplos destinos (*multicast*), Hsu e Hu (2016) propuseram o esquema E-GBSD, que prioriza as mensagens no *buffer* visando maximizar a taxa de entrega global. Por outro lado, Le *et al.* (2016) voltaram-se para o comportamento humano, baseando sua

função de utilidade em contatos com distribuição de lei de potência. Avançando para modelagens matemáticas mais complexas, Rahmouni *et al.* (2014) utilizaram um *processo de decisão de Markov parcialmente observável* (POMDP). Nesse modelo, o nó toma decisões de descarte mesmo sem conhecer o estado global da rede, baseando-se apenas no histórico de transmissões e na probabilidade de cooperação dos nós vizinhos.

Técnicas metaheurísticas e de estruturação de filas têm ganhado destaque por sua eficiência prática. Sonkar e Kumar (2022), por exemplo, aplicaram o *algoritmo de busca harmônica* (HSA) para otimizar o uso da memória. O algoritmo determina o descarte avaliando, de forma simultânea, o tamanho da mensagem, o número de réplicas e o *tempo de vida* (TTL) restante. Sob uma perspectiva de organização interna, Guo *et al.* (2024) propuseram o esquema MQAE, que substitui a abordagem de fila única pela divisão do *buffer* em múltiplas partições. Essa divisão é baseada na prioridade da mensagem e na contagem de saltos, permitindo que a rede forneça níveis básicos de *Qualidade de Serviço* (QoS).

Por fim, a literatura indica que o contexto da aplicação é um dos fatores mais determinantes para a escolha da política de gerenciamento. Em redes sociais tolerantes a atrasos, onde o foco recai sobre o interesse dos usuários, Majeed *et al.* (2018) sugerem a remoção proativa de mensagens que possuem a maior contagem de saltos e os menores valores restantes de TTL, liberando espaço para novas interações. Em cenários de desastres, Prasad *et al.* (2024) ressaltam a urgência de técnicas sensíveis ao contexto e integradas a *sistemas de localização* (GIS) para lidar com topologias que mudam drasticamente. No âmbito das *redes de sensores sem fio* (WSN) assistidas por nós móveis, Biabani *et al.* (2022) criaram o algoritmo EE-MSWSN. Essa solução considera o *buffer* finito dos sensores e suas diferentes taxas de geração de dados para evitar transbordamentos nos pontos de coleta, garantindo a integridade das medições físicas.

2.4.2 Gerenciamento de energia

O gerenciamento de energia é um dos desafios mais críticos em DTNs devido à dependência de baterias com capacidade limitada. A otimização do consumo energético afeta diretamente a sustentabilidade da rede em cenários extremos (Sreya *et al.*, 2024). A literatura aponta diversas causas simultâneas para o rápido esgotamento das baterias dos dispositivos móveis, conforme ilustrado na Figura 2.8:

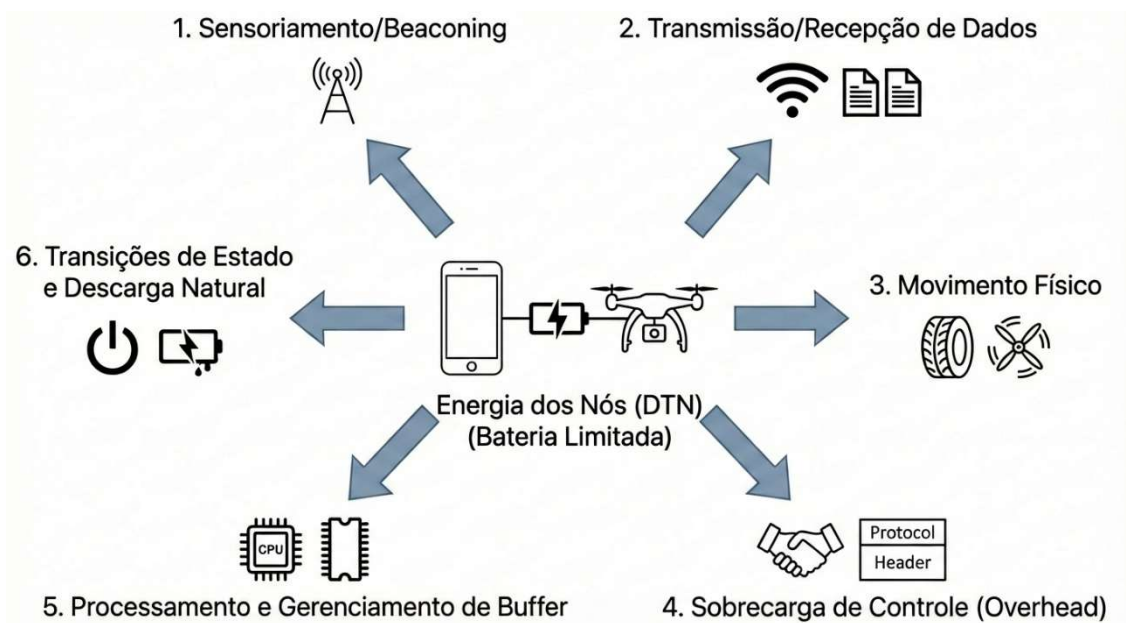


Figura 2.8 – Principais fontes de consumo energético em nós móveis de DTNs

1. Sensoriamento contínuo e *beaconing*: A necessidade de descobrir vizinhos mantém os nós em estado de escuta ociosa (*idle listening*). Wang *et al.* (2017) e Chen *et al.* (2024b) ressaltam que a alta frequência de emissão de sinais de *beaconing* é um importante fator de consumo de energia.

2. Transmissão e recepção de dados: O custo de envio e recepção de pacotes é agravado pela distância entre os nós, como modelado por Lu *et al.* (2016), e pela replicação excessiva de mensagens em protocolos de inundação (Eshghi *et al.*, 2015).

3. Movimento físico: Em redes com mobilidade controlada, o deslocamento mecânico do nó móvel pode incorrer em altos custos energéticos (Vazintari e Cottis, 2016).

4. Sobrecarga de controle: A troca de mensagens de controle para manter topologias ou gerenciar comunidades gera um consumo adicional de energia significativo (Li *et al.*, 2017).

5. Processamento interno e gerenciamento de *buffer*: O processamento de *bundles*, o cálculo de rotas e o gerenciamento das filas de dados exigem poder computacional que drena baterias (Lent, 2024; Niyato *et al.*, 2015).

6. Transições de estado e descarga natural: O tempo e a energia gastos para "acordar" o hardware (*boot up*) ao sair do modo de suspensão (Celebi *et al.*, 2019), somados à descarga natural da bateria mesmo quando ociosa (Lent, 2024), completam o quadro de consumo.

O diagrama da Figura 2.8 ilustra múltiplos fatores que contribuem para o esgotamento das baterias limitadas dos dispositivos, englobando custos de comunicação (sensoriamento, transmissão de dados e sobrecarga de controle), movimentação física, processamento interno e características inerentes ao hardware (transições de estado e descarga natural).

Para reduzir esses múltiplos custos, diversas estratégias têm sido propostas. Wang *et al.* (2017) propõem o *controle dinâmico de beaconing* (DBCEC) usando cadeias de Markov para otimizar a frequência de descoberta. Vazintari e Cottis (2016) formulam um esquema autônomo baseado na teoria dos jogos para equilibrar a decisão de se mover fisicamente ou permanecer estático.

No roteamento, Eshghi *et al.* (2015) e Chen *et al.* (2024b) utilizam limiares de energia residual e dormência assíncrona para evitar o esgotamento de nós-chave. Outras abordagens incorporam métricas espaciais, como o framework DEEOF baseado em distância (Lu *et al.*, 2016), ou métricas sociais, como o protocolo SEBAR (Li *et al.*, 2017). Avanços recentes também exploram redes cognitivas com redes neurais pulsantes (Lent, 2024), algoritmos centrados na

informação (Verma *et al.*, 2024) e transferência de energia sem fio (Niyato *et al.*, 2015) para prolongar a vida útil da rede.

2.5 Simulação de redes móveis

2.5.1 Modelos de mobilidade

A avaliação de desempenho em redes móveis — englobando MANETs, VANETs e DTNs — depende de uma representação precisa do deslocamento de seus nós. Conforme apontado por Safaei *et al.* (2020), o modelo de mobilidade influencia a dinâmica da topologia da rede e a sua funcionalidade. Consequentemente, a utilização de modelos que não refletem a realidade pode comprometer a validade dos resultados, gerando distorções na taxa de entrega de pacotes e no atraso fim a fim (Abed *et al.*, 2014).

Para classificar esses modelos, a literatura propõe taxonomias abrangentes. Conforme ilustrado na Figura 2.9, Safaei *et al.* (2020) classificam inicialmente os modelos de mobilidade quanto à dependência de coletores (*sinks*).

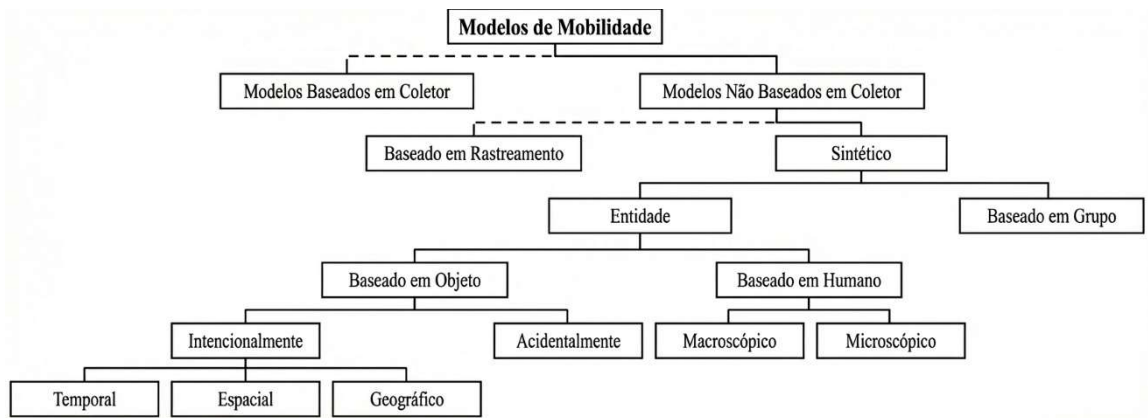


Figura 2.9 – Taxonomia geral dos modelos de mobilidade. Adaptado de Safaei *et al.* (2020).

Os modelos não-dependentes, que representam a maioria das aplicações, podem ser baseados em rastreamento (*trace-based*) ou sintéticos. Os modelos baseados em rastreamento oferecem alta fidelidade por refletirem dados empíricos, mas carecem de flexibilidade para cenários de larga escala. Por outro

lado, os modelos sintéticos utilizam formulações matemáticas para emular o movimento, sendo versáteis para simulações complexas (Safaei *et al.*, 2020; Gharib *et al.*, 2020).

No escopo dos modelos sintéticos, a dinâmica de movimentação condiciona a estrutura da rede. O deslocamento pode ser coletivo e interdependente, baseado em grupo, ou focado no comportamento autônomo de cada entidade. Ao modelar entidades, a literatura diferencia a complexidade inerente à mobilidade humana — analisada em suas rotinas sociais macroscópicas ou microscópicas — da movimentação de objetos genéricos. Para os objetos, o movimento varia desde a estocasticidade acidental até trajetórias governadas por regras lógicas intencionais. Para que os movimentos intencionais sejam condizentes com a realidade, eles são moldados por restrições temporais (dependência do histórico de velocidade e direção), espaciais (atração ou repulsão a pontos de referência) ou geográficas (confinamento imposto pela topologia do ambiente, como ruas e obstáculos).

Historicamente, dentro da categoria de modelos acidentais, o *Random Waypoint* (RWP) tem sido popular por sua simplicidade, mas falha ao capturar a mobilidade real. Modelos aleatórios podem gerar distribuições espaciais irreais (Mboup *et al.*, 2022) e superestimar métricas como taxas de *handover* (Khaki e Ghasemi, 2020). Para abrandar paradas abruptas, modelos com dependência temporal (ex: Gauss-Markov) suavizam as trajetórias (Abed *et al.*, 2014). Essa suavidade é crítica em redes aéreas, dadas as restrições físicas de *veículos aéreos não tripulados* (UAV) (He *et al.*, 2020; Chriki *et al.*, 2019).

No que diz respeito à dependência espacial, a modelagem de viagens em grupo é essencial, pois sua omissão gera estimativas imprecisas de infraestrutura (Kagho e Balac, 2023). Modelos como o *Reference Point Group Mobility* (RPGM) representam essa dinâmica, fazendo os nós seguirem um líder lógico e mantendo a coesão da rede (Safaei *et al.*, 2020).

Em ambientes urbanos, torna-se necessária a inclusão de restrições geográficas. Li *et al.* (2011) demonstram que a topologia do mapa frequentemente sobrepuja o impacto do próprio modelo de mobilidade, com interseções atuando como nós virtuais de alta densidade. Modelos baseados em

mapas restringem o movimento a caminhos predefinidos, otimizando a avaliação de protocolos de roteamento em comparação a modelos de espaço aberto (Kabilan *et al.*, 2018).

Aprofundando essa abordagem geográfica, Gharib *et al.* (2020) propõem modelar a cidade como um grafo, utilizando algoritmos de busca – como o de Dijkstra – para encontrar o caminho mais curto (*shortest path*) entre zonas de interesse. Essa técnica reflete o comportamento real, onde o movimento é confinado a vias e as rotas visam a eficiência do deslocamento. Por fim, a avaliação da escalabilidade sob esses modelos restritos é imperativa. Kabilan *et al.* (2018) evidenciam que a escalabilidade possui relação direta com o consumo de energia e com a entrega de pacotes.

Portanto, utilizar o modelo *Shortest Path Map-Based Movement* fornece o rigor necessário para validar os protocolos CASTRO e QL-CASTRO, garantindo que os indicadores sociais extraídos reflitam a previsibilidade típica de cenários urbanos.

2.5.2 Simulador ONE

O simulador ONE (*Opportunistic Networking Environment*) foi projetado para avaliar protocolos de roteamento e aplicações em DTNs. Keränen *et al.* (2009) destacam que o ONE oferece um *framework* de simulação extensível que suporta a geração de mobilidade e eventos, troca de mensagens, roteamento e uma modelagem de consumo de energia. Essa ferramenta visa preencher as lacunas dos simuladores de redes tradicionais (como ns-2 e OMNeT++), que possuem suporte nativo limitado para as características de cenários oportunistas e do paradigma *store-carry-forward*.

Em sua essência, o ONE é um motor de simulação de eventos discretos baseado em agentes. Keränen *et al.* (2010) explicam que as principais funções do simulador incluem a modelagem do movimento dos nós, a detecção de contatos usando várias interfaces de rádio, o roteamento e a manipulação de mensagens. Sua arquitetura modular permite configurar capacidades específicas para diferentes grupos de nós, definindo parâmetros como

armazenamento persistente, alcance de comunicação, taxa de transferência e modelos de consumo de energia.

A Figura 2.10 apresenta a arquitetura do simulador. Entre suas características, destaca-se a flexibilidade na modelagem de mobilidade. Keränen *et al.* (2009) descrevem que o simulador suporta tanto modelos sintéticos tradicionais (*Random Walk* e *Random Waypoint*) quanto modelos baseados em restrições de mapas reais (*Map-Based Movement*). Para ampliar o realismo, o simulador inclui modelos baseados no comportamento humano – como o *Working Day Movement* (WDM), que emula rotinas diárias de trabalho, lazer e descanso – e permite a importação de arquivos de rastreamento (*trace files*) de mobilidade do mundo real.

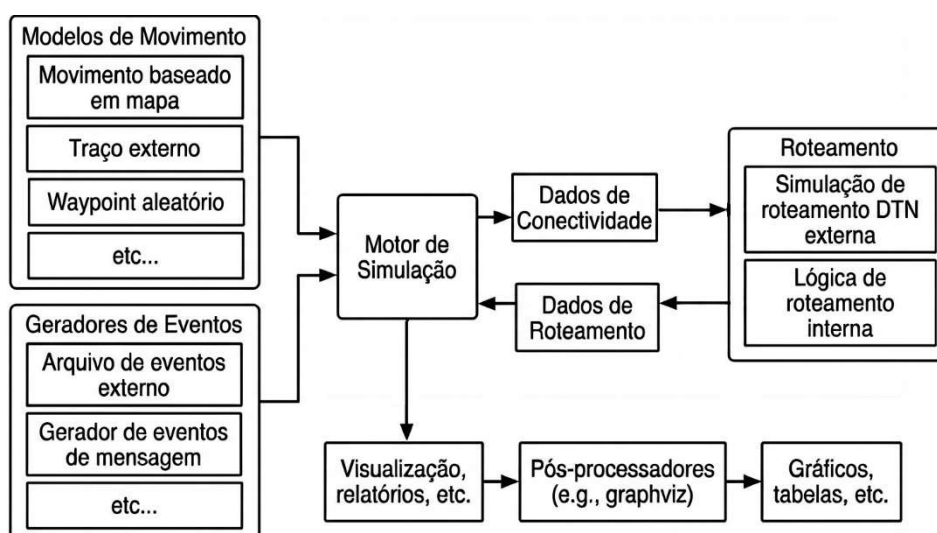


Figura 2.10 – Arquitetura do simulador ONE. Adaptado de Keränen *et al.* (2010)

Quanto ao roteamento, o simulador disponibiliza implementações de protocolos consolidados da literatura, facilitando a comparação de desempenho. Keränen *et al.* (2010) detalham que o ONE inclui nativamente protocolos de cópia única (*first contact* e *direct delivery*), roteamento por inundação (Epidêmico), esquemas de replicação controlada (*Spray and Wait*), e modelos baseados em estimativa de probabilidade (PRoPHET), dentre outros. Essa variedade, aliada à possibilidade de criação de novos módulos, define o ONE como uma das ferramentas mais utilizadas para o desenvolvimento de estratégias de encaminhamento.

Por fim, o ONE disponibiliza ferramentas de geração de relatórios e visualização para coleta e análise de resultados. Keränen *et al.* (2010) ressaltam que o simulador possui uma Interface Gráfica de Usuário (GUI) que exibe o estado da simulação em tempo real. Adicionalmente, módulos de relatórios permitem a extração de dados sobre métricas fundamentais – como probabilidade de entrega, latência e sobrecarga de rede – que subsidiam a posterior análise de desempenho dos protocolos.

2.5.3 Software QGIS

O QGIS, anteriormente conhecido como *Quantum GIS*, é um *sistema de informação geográfica* (SIG) de código aberto e multiplataforma voltado à visualização, edição, análise e gerenciamento de dados espaciais. Mantido pela *Open Source Geospatial Foundation* (OSGeo), o software suporta diversos formatos de vetores, *rasters* e bancos de dados. Devido à sua natureza extensível via complementos (*plugins*), o QGIS é amplamente utilizado para o geoprocessamento de dados em âmbitos acadêmicos e industriais.

No contexto de redes móveis e DTNs, a precisão topológica do cenário simulado é um fator relevante para a fidelidade dos resultados. Conforme discutido na subseção anterior, simuladores como o ONE utilizam modelos de mobilidade baseados em mapas (*Map-Based Movement*) para restringir o deslocamento dos nós a vias reais, como ruas, rodovias e calçadas. O QGIS possibilita a importação de dados cartográficos de plataformas colaborativas, como o *OpenStreetMap* (OSM), permitindo a manipulação e o ajuste de topologias conforme as exigências do cenário de teste.

A ferramenta permite filtrar camadas, isolar rotas específicas e simplificar a geometria de malhas viárias, contribuindo para a redução do custo computacional da simulação. Após o tratamento dos dados espaciais, o QGIS realiza a exportação da topologia resultante para o formato *Well Known Text* (WKT) – linguagem de marcação de texto padrão lida nativamente pelo simulador ONE para a construção de cenários de mobilidade urbana.

3. TRABALHOS RELACIONADOS

3.1 Introdução

Este capítulo apresenta uma revisão abrangente da literatura acerca dos protocolos de roteamento e encaminhamento de mensagens em redes oportunistas (OppNets) e redes tolerantes a atrasos e desconexões (DTNs). Devido à ausência de caminhos fim-a-fim preestabelecidos e à mobilidade dinâmica dos nós, diversas estratégias foram propostas para maximizar a taxa de entrega de mensagens e minimizar o consumo de energia e recursos de rede.

Para fins de organização, os trabalhos foram categorizados de acordo com os principais critérios utilizados para a tomada de decisão de roteamento. A Seção 3.2 apresenta protocolos baseados em inundação e replicação controlada, que servem como base de comparação na literatura. A Seção 3.3 discute abordagens baseadas em padrões de contato e histórico de encontros. A Seção 3.4 explora estratégias adaptativas e híbridas. Por fim, a Seção 3.5 faz algumas considerações sobre o estado da arte, comparando soluções existentes com a proposta desta tese, que será detalhada nos capítulos subsequentes.

3.2 Protocolos baseados em inundação e replicação controlada

O roteamento em redes com conectividade intermitente pode ser classificado de acordo com as estratégias de repasse de mensagens. Os esquemas de cópia única (*single-copy*), como a *transmissão direta*, minimizam o consumo de recursos de rede ao manter apenas uma cópia da mensagem com um único portador. No entanto, eles apresentam atrasos de entrega elevados e baixa confiabilidade (Spyropoulos *et al.*, 2008a, 2008b). Para contornar essa limitação, os esquemas de múltiplas cópias (*multi-copy*), fundamentados em inundação (*flooding*) ou replicação controlada, consolidaram-se como modelos de referência para o roteamento em DTNs.

O *roteamento epidêmico* (Vahdat e Becker, 2000) é o protocolo clássico mais representativo da categoria de inundação, frequentemente utilizado como

referência de desempenho. Neste modelo, cada nó mantém um *vetor resumo* (*summary vector*), contendo os identificadores das mensagens armazenadas em seu *buffer*. Durante o contato entre dois nós, ocorre o intercâmbio desses vetores para identificar as mensagens ausentes em cada par, seguido pela solicitação e transferência das réplicas correspondentes (Figura 3.1).

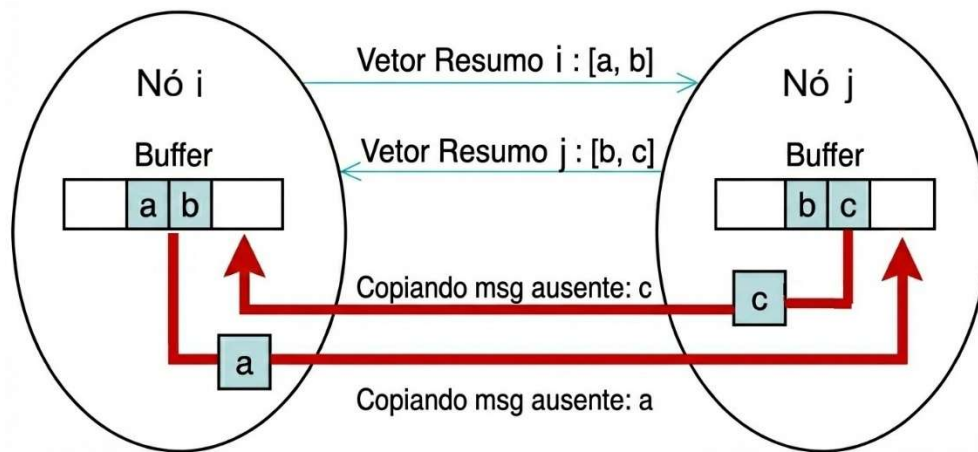


Figura 3.1 – Transmissão epidêmica de mensagens. Adaptado de Vahdat e Becker (2000)

Embora o protocolo clássico epidêmico maximize a taxa de entrega em cenários com recursos de armazenamento abundantes, sua estratégia de inundação irrestrita torna-o ineficiente quanto ao uso de largura de banda e energia, gerando um alto índice de mensagens redundantes. Para prevenir o esgotamento de recursos e reduzir o congestionamento da rede, a literatura propôs diversas otimizações ao longo dos anos. Tais esforços compreendem desde mecanismos de recuperação baseados em 'anti-pacotes' para a exclusão de cópias obsoletas (Zhang et al., 2007) até propostas recentes de gerenciamento adaptativo de *buffer* baseadas em múltiplas filas de prioridade (Guo et al., 2024). Adicionalmente, estudos investigaram a degradação do desempenho causada pelo comportamento não cooperativo (egoísmo) dos nós (Wang et al., 2019), enquanto avaliações fundamentadas em rastros reais de mobilidade urbana evidenciaram o elevado custo de rede intrínseco a essa abordagem (Amici et al., 2014).

Para reduzir a sobrecarga característica de esquemas de inundação sem a necessidade de mecanismos de recuperação ou gestão de *buffer*, o protocolo

Spray and Wait (SnW) (Spyropoulos *et al.*, 2005) introduziu uma estratégia de replicação controlada. O método estabelece uma quota para o número máximo de cópias (L) permitidas por mensagem, operando em duas fases distintas:

1. Fase *Spray* (difusão): O nó de origem inicia com L cópias da mensagem. A disseminação pode ocorrer via modelo fonte (*source*) ou binário. No modelo fonte, exclusivamente o nó de origem realiza as entregas, cedendo uma cópia para cada um dos primeiros $L - 1$ nós encontrados. No modelo binário — apontado pelos autores como o mais eficiente em cenários de mobilidade independente e identicamente distribuída (IID) — qualquer nó portador de $n > 1$ cópias transfere $\lfloor n/2 \rfloor$ ao encontrar um nó que ainda não possua a mensagem, restando $\lfloor n/2 \rfloor$ cópias para si. Essa estratégia visa acelerar a dispersão inicial das mensagens pela rede (Figura 3.2).

2. Fase *Wait* (espera): Ao restar apenas uma cópia no *buffer*, o nó interrompe a replicação e aguarda o encontro direto com o nó de destino para a entrega final. Nesta fase, o protocolo assume um comportamento passivo, dependendo da mobilidade do nó portador para a conclusão da trajetória.

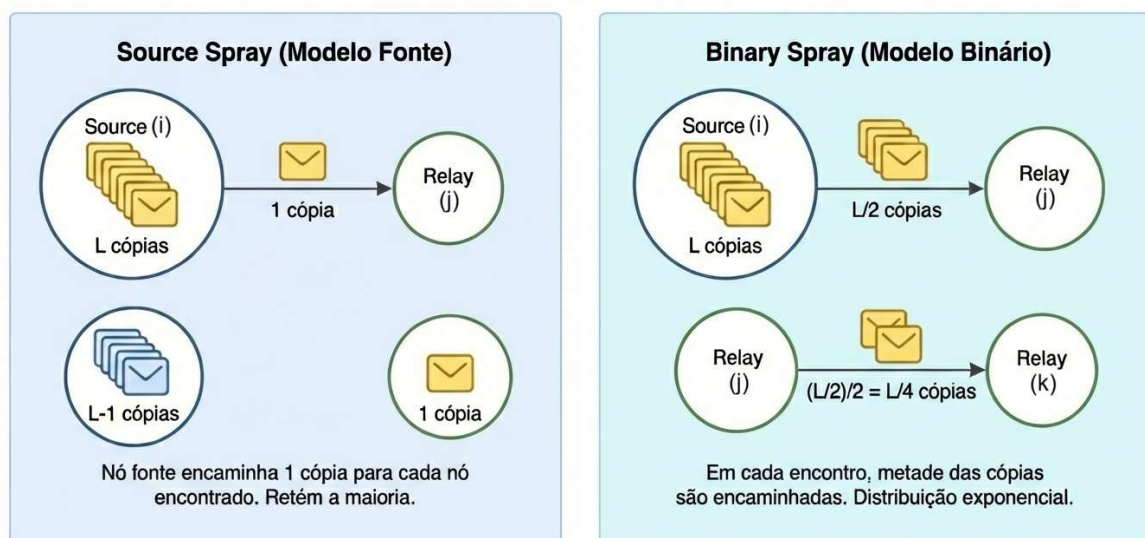


Figura 3.2 – Modelos *fonte* e *binário* da fase *Spray* – protocolo *Spray and Wait*. Baseado em Spyropoulos *et al.* (2005)

Embora o protocolo *Spray and Wait* reduza o volume de transmissões em relação ao Epidêmico, a passividade da fase *Wait* pode gerar latências elevadas,

especialmente em cenários onde a mobilidade dos nós é restrita a áreas locais. Para contornar essa limitação, Spyropoulos et al. (2008a) introduziram o *Spray and Focus*, no qual a fase de espera é substituída pela fase *Focus*. Nessa etapa, a última cópia da mensagem é encaminhada ativamente por meio de um roteamento baseado em utilidade, priorizando nós que possuam temporizadores de último encontro mais recentes com o destino.

A transição do *Spray and Wait* para o *Spray and Focus* evidencia a necessidade de extrair informações sobre a atividade social e os padrões de mobilidade entre os nós. O aprimoramento das funções de utilidade nas decisões de encaminhamento permite reduzir o consumo de recursos e evitar a saturação da rede. Essa abordagem, que explora o comportamento dos dispositivos, fundamenta os protocolos baseados em padrões de contato, que serão detalhados na próxima seção.

3.3 Protocolos baseados em padrões de contato

A mobilidade em redes oportunistas é frequentemente pautada por rotinas humanas, o que permite a identificação de padrões recorrentes. Conforme o levantamento de Tornell et al. (2015), o roteamento evoluiu de abordagens de "conhecimento zero" para protocolos baseados em utilidade, que exploram o histórico de contatos para otimizar as transmissões.

Um dos marcos dessa categoria é o protocolo PProPHET (Lindgren et al., 2003). Ele emprega uma métrica denominada *previsibilidade de entrega*, denotada por $P(i, j) \in [0, 1]$, que indica uma estimativa da probabilidade de o nó i encaminhar uma mensagem com sucesso ao nó j . O cálculo dessa métrica é regido por três equações fundamentais (Equações 3.1 – 3.3), ilustradas na Figura 3.3.

1. Atualização Direta: Quando dois nós se encontram, a previsibilidade de entrega entre eles é incrementada, utilizando uma constante de inicialização $P_{init} \in [0, 1]$:

$$P(i, j) = P(i, j)_{old} + (1 - P(i, j)_{old}) \cdot P_{init}. \quad (3.1)$$

2. Envelhecimento: Em períodos de ausência de contato, a previsibilidade de entrega entre um par de nós é reduzida gradualmente. Definindo $\gamma \in [0,1)$ como a constante de envelhecimento e k como o número de unidades de tempo transcorridas desde a última interação, o decaimento é modelado de forma exponencial:

$$P(i,j) = P(i,j)_{old} \cdot \gamma^k. \quad (3.2)$$

3. Transitividade: O protocolo baseia-se na premissa de que a frequência de encontros entre os pares de nós (i,j) e (j,k) qualifica o nó j como um intermediário apto para o encaminhamento de mensagens de i para k . Utilizando uma constante de impacto transitivo $\beta \in [0,1]$, a atualização da métrica ocorre de forma indireta:

$$P(i,k) = P(i,k)_{old} + (1 - P(i,k)_{old}) \cdot P(i,j) \cdot P(j,k) \cdot \beta. \quad (3.3)$$

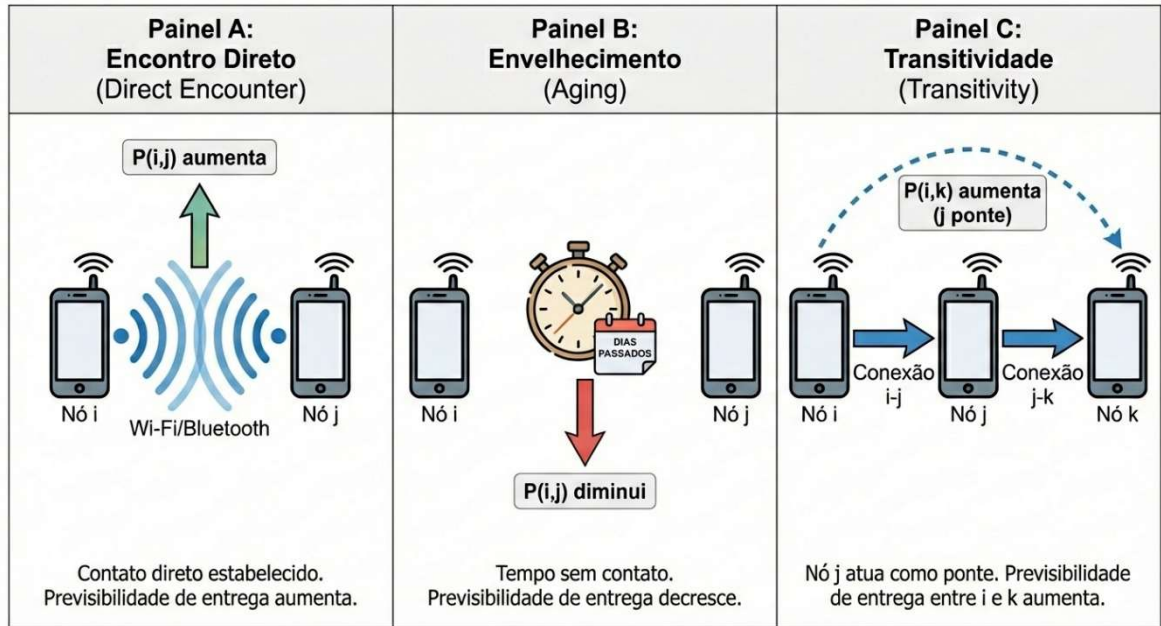


Figura 3.3 – Atualização da previsibilidade de entrega no protocolo PROPHET

Conforme destacado por Pathak *et al.* (2017), a base matemática do PROPHET serviu de fundamentação para diversos protocolos. Embora esses modelos busquem elevar a taxa de entrega e reduzir atrasos, a literatura indica que o desempenho é sensível à calibração das constantes $(P_{init}, \gamma, \beta)$, sob o risco de gerar sobrecarga na rede.

Apesar de sua eficácia, a previsibilidade do P_{Ro}PHET original pode apresentar flutuações imprecisas. Para estabilizar essa métrica, Peng *et al.* (2024) propuseram o TES-P_{Ro}PHET, que aplica a *suavização exponencial tripla* (TES) à série temporal de contatos. Visando reduzir o impacto do congestionamento em redes IoT, Mao *et al.* (2019) introduziram o Scheduling-PROPHET, que refina a seleção do próximo salto ao integrar a previsibilidade de entrega, a análise da contagem de saltos (*hop count*) e a carga de tráfego (*traffic load*) dos nós.

Expandindo a predição para o contexto de mobilidade estruturada, Yuan *et al.* (2012) propuseram o algoritmo PER (*Predict and Relay*). Este modelo utiliza um Processo Semi-Markoviano Homogêneo no Tempo (TH-SMP) para estimar a distribuição de probabilidade do momento do encontro entre pontos de referência (*landmarks*). Mais recentemente, Du *et al.* (2021) aprimoraram a seleção de nós em redes assistidas por drones (UAVs) ao introduzir um protocolo baseado em probabilidades de encontros e nos tempos de conexão persistente, visando identificar enlaces mais estáveis.

Seguindo a vertente de protocolos fundamentados no histórico de vizinhança, o protocolo MaxProp (Burgess *et al.*, 2006) diferencia-se do P_{Ro}PHET por basear suas decisões na probabilidade de caminhos (*path likelihoods*). Neste modelo, cada nó rastreia a probabilidade de encontro com seus pares, normalizando os valores de forma que sua soma seja unitária. Para contornar as limitações desse processo de normalização, Sadat e Mohiuddin (2014) propuseram um aprimoramento que substitui o custo de caminho original por um índice de vizinhança global (*Overall Neighborhood Index*). Essa modificação visa impedir a redução acentuada da probabilidade de entrega em nós cujos registros de encontro são mais antigos, conferindo maior estabilidade às decisões de roteamento.

Com a evolução da área, o histórico de encontros foi enriquecido com atributos sociais para qualificar as decisões de encaminhamento. O protocolo dLife (Moreira *et al.*, 2014) visa capturar o dinamismo da rede por meio das rotinas diárias dos usuários. O modelo baseia-se na métrica TECD (*Time-Evolving Contact Duration*), que emprega a duração média dos contatos,

segmentada em janelas horárias, para mapear a dinâmica social em intervalos específicos ao longo de dias consecutivos. Embora o dLife represente um avanço na captura do dinamismo social no modo ativo, ele não avalia a previsibilidade estatística dos períodos em que os nós permanecem desconectados, foco central desta tese.

Quanto ao gerenciamento de recursos físicos, Bista e Rawat (2016) propuseram o E-PRoPHET, condicionando os repasses à energia residual e ao espaço de *buffer* do nó receptor. Expandindo essa visão, Chen *et al.* (2024b) apresentaram o HP-ECD, que combina o PRoPHET com balanceamento de energia e dormência assíncrona. Buscando uma métrica unificada, Chen *et al.* (2024a) propuseram o CR-5, que calcula um *grau de retransmissor* a partir de cinco atributos dinâmicos: probabilidade de encontro, tempo de contato cumulativo, taxa de energia, taxa de cache e grau de entrada social. Contudo, observa-se que, embora o CR-5 utilize o tempo de contato cumulativo, ele não explora a regularidade dos intervalos de desconexão como fator de predição, lacuna que o Protocolo CASTRO visa preencher.

Por fim, no âmbito da segurança e confiabilidade, o comportamento não-cooperativo ou malicioso dos nós degrada a eficiência da rede. Para abordar este problema, Basu *et al.* (2018) introduziram o Wise-PRoPHET, que integra uma técnica de *Watchdog* ao histórico para gerar métricas de confiança e isolar nós que descartam pacotes deliberadamente, protegendo a integridade do roteamento.

3.4 Estratégias adaptativas e híbridas

Para equilibrar altas taxas de entrega com baixo consumo de recursos de rede, a literatura convergiu para estratégias híbridas. Estas abordagens frequentemente mesclam controles de réplica com métricas de utilidade, normalmente baseadas em históricos.

Han e Chung (2015) propuseram um protocolo que concilia a lógica Epidêmica com a do PRoPHET. O modelo utiliza a inundação irrestrita na fase inicial de disseminação, desde que os contadores de encaminhamento e de saltos da

mensagem permaneçam abaixo de limiares pré-definidos. Uma vez atingidos esses limites, o protocolo transita para a lógica do P_{Ro}PHET a fim de refinar o processo de entrega e reduzir o impacto da sobrecarga na rede.

Na mesma linha de hibridização, Tekouabou *et al.* (2021) introduziram o protocolo HDRP (*Hybrid DTN Routing Protocol*). A proposta alterna entre as estratégias de inundação e encaminhamento, baseando-se na proporção entre a duração dos contatos e o tempo total de conectividade dos nós. O modelo HDRP incorpora, ainda, uma política de gerenciamento que condiciona o repasse e a retenção das mensagens à disponibilidade de espaço em buffer e ao tempo de vida remanescente (TTL) dos pacotes, visando otimizar o consumo energético em ambientes de IoT.

Uma contribuição relevante neste cenário é o trabalho de Kuronuma *et al.* (2017), que figura como um dos primeiros a propor a renovação do tempo de vida (TTL) das mensagens. Ao monitorar o estado da replicação, o protocolo identifica mensagens prestes a expirar que possuem baixa disseminação na rede; nesses casos, o algoritmo reinicializa o TTL e incrementa a quota de cópias. Os resultados indicam que a gestão flexível do TTL é um fator determinante para a eficiência em redes oportunistas.

Uma linha de pesquisa proeminente que ilustra a evolução das abordagens híbridas é a conduzida por Cui *et al.* Inicialmente, o S-Prophet (2021) aprimorou o P_{Ro}PHET ao introduzir a *taxa de similaridade* baseada em vizinhos comuns e um mecanismo de reconhecimento (ACK) para gerenciamento de *buffer*. Posteriormente, o grupo propôs o P-AVF (2023), que introduziu a métrica de *flutuação média da conectividade* dos nós, utilizando o tempo de separação e a ocupação do cache para avaliar a confiabilidade dos enlaces. Essa base teórica resultou no STCESW (Huang *et al.*, 2024), uma proposta recente que expande o *Spray and Wait* ao integrar a capacidade de transmissão estável, o *buffer* residual e o ângulo entre os vetores de movimento dos nós para otimizar a difusão espacial.

Por fim, destacam-se também os protocolos HESnW e QoN-ASW, adotados como *benchmarks* nesta tese e explicados a seguir.

3.4.1 Protocolo HESnW

O HESnW (Gan *et al.*, 2017) explora a propriedade de "mundo pequeno" das redes sociais. Para evitar que o PProPHET descarte nós com baixa previsibilidade direta, mas que possuem contatos históricos com alta probabilidade de encontrar o destino, o HESnW introduz a métrica da *múltipla previsibilidade*. Em vez de restringir a decisão à previsibilidade direta ou à transitividade processada no instante do encontro, o protocolo analisa o histórico de interações do nó candidato para identificar potenciais rotas indiretas ao destino.

A Equação 3.4 define o cálculo da múltipla previsibilidade, em que s representa o nó atual, d o nó de destino e h identifica um nó presente no histórico de encontros de s .

$$mP_{s,h,d} = P_{s,h} \cdot P_{h,d}. \quad (3.4)$$

Diferente do protocolo *Spray and Wait* binário, que realiza uma divisão equitativa das cópias, o HESnW faz uma alocação proporcional aos valores de previsibilidade dos nós envolvidos. O protocolo define dois cenários para a distribuição da quota de mensagens:

Cenário A: Nó receptor selecionado via múltipla previsibilidade.

Este cenário ocorre quando o nó receptor (r) possui uma previsibilidade de entrega ao destino (d) inferior ao limiar de transferência (thr_h) do nó remetente (s). No entanto, o nó receptor apresenta um contato histórico (h) com elevada previsibilidade de entrega ao destino, resultando em um valor de múltipla previsibilidade que autoriza o repasse. As Equações 3.5 e 3.6 estabelecem a partição das cópias, onde L_s representa a quota retida pelo remetente e L_r , a quota transferida ao receptor:

$$L_s = \left\lfloor L \cdot \frac{P_{s,d} + \sum_{\forall h, mP_{s,h,d} > thrh_s} mP_{s,h,d}}{P_{s,d} + \sum_{\forall h, mP_{s,h,d} > thrh_s} mP_{s,h,d} + \sum_{\forall h, mP_{r,h,d} > thrh_s} mP_{r,h,d}} \right\rfloor, \quad (3.5)$$

$$L_r = L - L_s. \quad (3.6)$$

A quota total de mensagens (L) é distribuída proporcionalmente à soma das métricas de previsibilidade consideradas válidas – isto é, aquelas que superam o limiar de transferência (thr_h). O remetente (s) retém uma quantidade de cópias proporcional à sua previsibilidade direta e às múltiplas probabilidades extraídas de seu histórico, enquanto o receptor (r) recebe a quota remanescente.

Cenário B: Nó receptor selecionado via previsibilidade direta.

Este cenário ocorre quando o nó receptor (r) apresenta uma previsibilidade de entrega direta ao destino (d) superior ao limiar exigido pelo remetente ($P_{r,d} > thrh_s$). As Equações 3.7 e 3.8 definem a partição das cópias entre os nós envolvidos:

$$L_s = \left\lfloor L \cdot \frac{P_{s,d}}{P_{s,d} + P_{r,d}} \right\rfloor, \quad (3.7)$$

$$L_r = \left\lfloor L \cdot \frac{P_{r,d}}{P_{s,d} + P_{r,d}} \right\rfloor = L - L_s. \quad (3.8)$$

3.4.2 Protocolo QoN-ASW

O QoN-ASW (Cui *et al.*, 2019) utiliza a métrica de Qualidade do Nó (QoN), que combina funções de utilidade voltadas à capacidade de encaminhamento de mensagens e ao cálculo probabilístico de entrega. Na fase *Spray*, o protocolo realiza a alocação assimétrica de cópias baseada no valor de QoN e substitui a fase *Wait* passiva por uma estratégia de encaminhamento ativo.

O protocolo busca identificar nós com elevada capacidade de encaminhamento de dados. Para assegurar que a métrica reflita as variações dinâmicas da rede, o tempo é segmentado em janelas de tamanho fixo (TS), permitindo uma avaliação iterativa e ponderada.

A *capacidade estimada* de manipulação de mensagens ($EMHC$) é estabelecida pela Equação 3.9. O cálculo estima o volume de mensagens que o nó i é capaz de repassar na k -ésima janela, considerando a quantidade de mensagens encaminhadas com sucesso (m_{cur}) durante o tempo transcorrido desde o início da janela vigente (Δt), multiplicada pelo tamanho TS . Para janelas subsequentes ($k > 1$), a formulação isola o desempenho do período atual ao subtrair a somatória das mensagens processadas em janelas anteriores.

$$\begin{cases} EMHC_i^k = \frac{m_{cur}}{\Delta t} \cdot TS & \text{para } k = 1, \\ EMHC_i^k = \frac{m_{cur} - \sum_{j=1}^{k-1} m_j}{\Delta t} \cdot TS & \text{para } k > 1. \end{cases} \quad (3.9)$$

Para evitar oscilações abruptas na capacidade estimada, o QoN-ASW aplica uma técnica de suavização para obter a *capacidade de manipulação* (MHC), conforme a Equação 3.10. O cálculo integra o desempenho observado em janelas anteriores à estimativa do intervalo corrente, utilizando o fator de ponderação α para calibrar a influência do histórico. Dessa forma, um valor elevado de MHC sugere que o nó apresenta uma capacidade de encaminhamento consistente e eficiente ao longo do tempo.

$$\begin{cases} MHC_i^k = EMHC_i^k & \text{para } k = 1, \\ MHC_i^k = \alpha \cdot MHC_i^{k-1} + (1 - \alpha) \cdot EMHC_i^k & \text{para } k > 1. \end{cases} \quad (3.10)$$

Adicionalmente, o QoN-ASW incorpora a métrica da *força de conexão* (Equação 3.11) para ponderar o cálculo da previsibilidade de entrega do PROPHET (Equação 3.12). A força $S_{(i,j)}$ é definida pela razão entre o tempo

acumulado em que os nós i e j permaneceram conectados e o tempo total decorrido (T). Valores elevados nesta métrica sugerem a existência de enlaces mais persistentes entre o par de nós. A previsibilidade de entrega é recalculada conforme a Equação 3.12, enquanto os mecanismos de atualização por transitividade e envelhecimento permanecem inalterados, seguindo as definições das Equações 3.2 e 3.3:

$$S_{(i,j)} = \frac{\sum_{l=1}^n (Fim_{ij}^l - Inicio_{ij}^l)}{T}, \quad (3.11)$$

$$P_{(i,j)} = P_{(i,j)old} + (1 - P_{(i,j)old}) \cdot P_{init} \cdot \lambda^{S_{(i,j)}}. \quad (3.12)$$

A Qualidade do Nó (QoN) consolida os indicadores de MHC e de previsibilidade de entrega em um índice escalar que orienta o processo de tomada de decisão do protocolo. Para a quantificação deste valor, empregam-se funções de utilidade que normalizam o desempenho do nó em relação ao seu par de contato, conforme estabelecido pelas Equações 3.13 e 3.14:

$$U_{MHC} = \frac{MHC_i^k}{MHC_i^k + MHC_j^{k'}}, \quad (3.13)$$

$$U_P = \frac{P_{(i,d)}}{P_{(i,d)} + P_{(j,d)}}. \quad (3.14)$$

O índice QoN é quantificado por meio da Equação 3.15, na qual o fator de ponderação η regula a influência relativa entre a utilidade da capacidade operacional do nó (U_{MHC}) e sua previsibilidade de entrega (U_P). Este parâmetro permite ajustar o comportamento do protocolo conforme as prioridades de cada cenário de rede, resultando em um valor de QoN situado no intervalo $[0,1]$:

$$QoN_i = \eta \cdot U_{MHC} + (1 - \eta) \cdot U_P. \quad (3.15)$$

A Equação 3.16 estabelece a estratégia de divisão assimétrica de cópias na fase *Spray* do QoN-ASW, fundamentada no índice de Qualidade do Nó. Nesta formulação, L_{iold} representa a quota de mensagens do nó remetente (i) antes da interação, enquanto L_{inew} e L_{jnew} definem, respectivamente, o número de cópias retidas pelo emissor e a quantidade transferida ao nó receptor (j).

$$\begin{aligned} L_{inew}(m_d) &= \lfloor QoN_i \cdot L_{iold}(m_d) \rfloor, \\ L_{jnew}(m_d) &= L_{iold}(m_d) - L_{inew}(m_d). \end{aligned} \tag{3.16}$$

Ao restar apenas uma única cópia da mensagem m_d , inicia-se a fase *Wait*. O QoN-ASW diferencia-se do *Spray and Wait* clássico por adotar uma política de encaminhamento ativo nesta fase: em vez de aguardar passivamente o encontro direto com o destino, o nó remetente transfere a cópia para um intermediário sempre que este apresentar uma previsibilidade de entrega superior ($P_{j,d} > P_{i,d}$).

3.5 Protocolos baseados em aprendizado por reforço

A integração de técnicas de aprendizado por reforço (*reinforcement learning* — RL) no roteamento de redes oportunistas busca conferir adaptabilidade a topologias dinâmicas. O roteamento oportunista adota frequentemente o paradigma de *Q-Routing*, no qual o processo de aprendizado é direcionado à minimização de custos operacionais cumulativos, como a latência de entrega, a sobrecarga de rede e o consumo energético. Essa abordagem permite que os nós ajustem suas políticas de encaminhamento com base na exploração contínua do ambiente, substituindo heurísticas estáticas por modelos de decisão capazes de evoluir conforme os padrões de mobilidade e tráfego da rede se transformam.

Yao et al. (2023) realizaram um levantamento sobre o uso de algoritmos inteligentes em DTNs, destacando que o aprendizado por reforço é naturalmente compatível com o dinamismo dessas redes. A revisão detalha propostas que utilizam RL para gerenciar o encaminhamento e o controle de quota, integrando características sociais extraídas do histórico, como popularidade dos nós e força

de conexão, além de parâmetros operacionais como o tempo de vida (TTL) remanescente de mensagem e contagem de saltos. Contudo, os autores enfatizam que essas soluções muitas vezes sofrem de baixa capacidade de generalização e dependem de álgebra simplificada.

Wu, Yoshinaga e Ji (2017) desenvolveram um protocolo probabilístico para redes veiculares (VDTNs) que combina lógica difusa e aprendizado por reforço para refinar a seleção do próximo salto. O diferencial desta proposta é a integração de um *valor de competência* – derivado de variáveis como velocidade, centralidade e ocupação de buffer – diretamente na função de valor do *Q-Learning*. O protocolo utiliza o aprendizado para estimar a probabilidade de encontro multi-hop, aplicando um fator de desconto ao custo de entrega conforme o número de saltos aumenta.

Yuan et al. (2019) propuseram o protocolo DQLR (*Double Q-Learning Routing*), que foca na minimização do custo de entrega ao utilizar uma função de recompensa baseada no decaimento exponencial do número de saltos (e^{-h}), priorizando trajetórias que alcancem o destino com o menor caminho possível. Além disso, o DQLR implementa um mecanismo de valor intermediário que permite aos nós estimarem recompensas futuras para vizinhos ainda não encontrados, conferindo uma forma de transitividade ao aprendizado. Todavia, a estratégia de Yuan et al. baseia-se primordialmente na topologia de saltos e na frequência de encontros, não considerando a regularidade estatística dos intervalos de desconexão como variável de estado.

Singh et al. (2022) introduziram o protocolo RLCC (*Reinforcement Learning-based Congestion Control*), que utiliza *Q-Learning* e lógica difusa para selecionar as opções de encaminhamento em OppNets. O diferencial desta abordagem é o uso do aprendizado por reforço para determinar o número ideal de cópias de uma mensagem em circulação, baseando-se na densidade de nós em tempo real. O agente de aprendizado recebe entradas de um sistema de inferência difusa que avalia centralidade, o estado do *buffer* e da bateria, o tempo de vida restante (TTL) das mensagens e a contagem de saltos.

Nakayima et al. (2024) propuseram um algoritmo multiagente com controlador centralizado (CCMA) que integra redes definidas por *software* (SDN) e

aprendizado por reforço para otimizar o desempenho de redes veiculares. O diferencial desta abordagem reside na capacidade do agente de RL em atuar como um orquestrador de rede, selecionando dinamicamente o protocolo de roteamento (Epidêmico, PProPHET ou *Spray and Wait*) e a política de escalonamento de buffer mais adequados para o estado vigente do sistema. O modelo fundamenta-se em um espaço de estados baseado na qualidade dos enlaces de comunicação, no volume de mensagens trafegadas pela rede e nas diferenças entre capacidades de armazenamento entre os nós conectados, visando a minimização de custos operacionais como latência e sobrecarga.

Em síntese, os trabalhos mencionados sugerem que a aplicação de aprendizado por reforço favorece uma gestão de recursos e uma seleção de intermediários mais refinada quando comparada aos protocolos clássicos. Observa-se, contudo, que as propostas vigentes fundamentam seus estados de aprendizado e funções de custo prioritariamente em métricas de conectividade ativa, como a duração de contatos, a topologia de saltos ou a qualidade imediata dos enlaces. A integração da modelagem estatística dos intervalos de desconexão configura-se como um campo ainda pouco explorado por essas abordagens. A análise desses padrões sugere um potencial de aperfeiçoamento da capacidade preditiva em redes oportunistas.

3.6 Considerações

A revisão da literatura evidencia uma evolução incremental no roteamento oportunista. Partindo-se de modelos de inundação e de cópia única considerados clássicos, o estado da arte avançou para estratégias de replicação controlada ou orientada a previsibilidades. Mais recentemente, a convergência para modelos híbridos e adaptativos permitiu a inclusão de variáveis de contexto mais complexas, abrangendo a qualidade dos nós, a energia residual e atributos sociais.

O roteamento híbrido busca equilibrar a taxa de entrega e o controle da sobrecarga. Algoritmos de destaque recente – como HESnW, QoN-ASW, P-AVF, STCESW, entre outros – indicam que a frequência de encontros torna-se

um indicador mais robusto quando associada à força de conexão ou à análise de padrões temporais. Reforçando essa perspectiva, Koukis *et al.* (2024) destacam que a eficiência em ambientes desafiadores depende da modelagem adequada dos intervalos entre conexões (*inter-contact times*). Os autores sugerem que protocolos capazes de interpretar a regularidade desses intervalos históricos são propostas promissoras para contornar a escassez de recursos.

Contudo, a análise comparativa entre os protocolos clássicos e as propostas mais recentes revela uma lacuna científica no tratamento da intermitência. Embora trabalhos contemporâneos de destaque, como o P-AVF (Cui et al., 2023) e o STCESW (Huang et al., 2024), tenham iniciado a observação de métricas temporais de separação, eles o fazem de maneira reativa. Nesses modelos, os períodos de ausência de contato são utilizados majoritariamente para a atenuação ou o envelhecimento da previsibilidade de entrega e da capacidade de transmissão. Subsiste a carência de uma investigação proativa que trate os intervalos de desconexão como fontes primárias de informação social. A exploração da tendência e da regularidade estatística destes períodos permite inferir padrões de comportamento que precedem o encontro, transformando o tempo de separação em um indicador qualitativo de rota, em vez de tratá-lo apenas como um fator de degradação da confiança no enlace.

É neste cenário que o Protocolo CASTRO (detalhado nos Capítulos 4 e 5) se insere. Ao segmentar o histórico integral em intervalos de conexão e desconexão, o CASTRO avalia a previsibilidade e a inclinação estatística da reconexão. A introdução de *Scores*, baseados nas métricas de *força*, *tendência* e *regularidade*, permite uma difusão controlada que visa evitar a rigidez do *Spray and Wait* e a dependência de frequência do PROPHET, indicando potencial para alcançar elevadas taxas de entrega com consumo energético moderado.

Nesse sentido, visando abordar o equilíbrio entre a precisão da seleção social e a agilidade no encaminhamento, esta tese também introduz o Protocolo QL-CASTRO. Esta extensão fundamenta-se na premissa de que a inteligência das métricas sociais pode ser potencializada pela capacidade de aprendizado contínuo do aprendizado por reforço. Enquanto o CASTRO provê a base de seleção por regularidade e tendência, a camada de aprendizado é projetada para

estimar dinamicamente os atrasos de entrega em cada rota. O objetivo desta arquitetura híbrida é conciliar a robustez da taxa de entrega com uma tomada de decisão capaz de reagir ao dinamismo das OppNets com mais rapidez, integrando métricas sociais de desconexão a um modelo de decisão focado na minimização do custo de atraso.

A Tabela 3.1 sintetiza a comparação entre os principais protocolos representativos discutidos neste capítulo. Esta síntese permite destacar o posicionamento das propostas desta tese (CASTRO e QL-CASTRO) em relação ao estado da arte.

Tabela 3.1 – Tabela comparativa entre protocolos de roteamento para OppNets

Protocolo	Autor (ano)	Recurso de Decisão	Controle de Cópias	Gestão de Buffer	RL
Epidêmico	Vahdat (2000)	Nenhum	Ilimitado (Inundação)	FIFO	Não
SnW	Spyropoulos (2005)	Nenhum	Quota fixa	FIFO	Não
PRoPHET	Lindgren (2003)	Previsibilidade	Probabilístico	FIFO	Não
HESnW	Gan (2017)	Múltipla previsibilidade	Quota adaptativa	FIFO	Não
QoN-ASW	Cui (2019)	MHC e previsibilidade	Quota adaptativa	FIFO	Não
P-AVF	Cui (2023)	Confiabilidade e estreiteza de conexão	Probabilístico	Taxa de ocupação do cache	Não
STCESW	Huang (2024)	Estabilidade de enlace, eficiência de vazão, ângulo de movimento	Quota fixa	Priorização por <i>buffer</i> residual	Não
DQLR	Yuan (2019)	Duplo <i>Q-Learning</i> e contagem de saltos	Seleção via recompensa	FIFO	Sim
RLCC	Singh (2022)	<i>Q-Learning</i> e lógica difusa (<i>fuzzy</i>)	Quota flexível (RL)	Controle via lógica difusa (<i>fuzzy</i>)	Sim
CCMA-DRL	Nakayima (2024)	TTL e qualidade de enlace	Delegado a protocolo (Epidêmico, SnW ou PRoPHETv2)	Escalonamento Dinâmico	Sim
CASTRO	Rosa (2026)	Métricas sociais (força, tendência e regularidade)	Mérito social	FIFO + notificações de entrega	Não
QL-CASTRO	Rosa (2026)	Métricas sociais + atraso estimado	Mérito social ou via aprendizado	FIFO + aposentadoria + notificações	Sim

Com base na fundamentação teórica e na análise do estado da arte aqui apresentadas, o capítulo seguinte detalha as modelagens matemáticas de força, tendência e regularidade que compõem o núcleo analítico desta tese.

4. CONECTIVIDADE E SCORES SOCIAIS

Este capítulo apresenta as formulações matemáticas que fundamentam as contribuições desta tese. Na Seção 4.1, detalham-se as métricas de *tendência* (T) e *regularidade* (R) entre conexões, além da métrica de *força* (S), esta última adaptada da literatura de redes sociais. A Seção 4.2 descreve a composição dos indicadores denominados *Scores*. Este capítulo apresenta o teorema de máxima variabilidade, assegurando uma fundamentação teórica às propostas do Protocolo CASTRO, descrito no Capítulo 5.

4.1 Força, tendência e regularidade entre conexões

O histórico de encontros em uma rede oportunista (*OppNet*) pode ser modelado por meio de intervalos temporais. Quando os nós detectam a presença mútua dentro de seu raio de transmissão e estabelecem enlaces que viabilizam a troca de dados, define-se o modo ativo (*ON-mode*). Na ausência de conectividade, as interações situam-se no modo inativo (*OFF-mode*). A somatória das durações de todos os intervalos no modo ativo é denominada t_{ON} , enquanto a somatória dos intervalos no modo inativo é designada como t_{OFF} .

Nesta modelagem, os intervalos temporais em modo ativo são referidos como *intervalos de conexão*, ao passo que os intervalos fechados em modo inativo são denominados *intervalos de desconexão*.

A Figura 4.1 ilustra um cenário hipotético em uma janela temporal de $t = 300s$. O nó i estabelece seis intervalos de conexão com o nó j (10s cada) e dois intervalos com o nó k (30s cada). Em ambos os casos, os valores agregados resultam em $t_{ON} = 60s$ e $t_{OFF} = 240s$. Nota-se que o valor de $t = 300s$ é adotado como parâmetro de referência (*ref*) para as análises subsequentes.

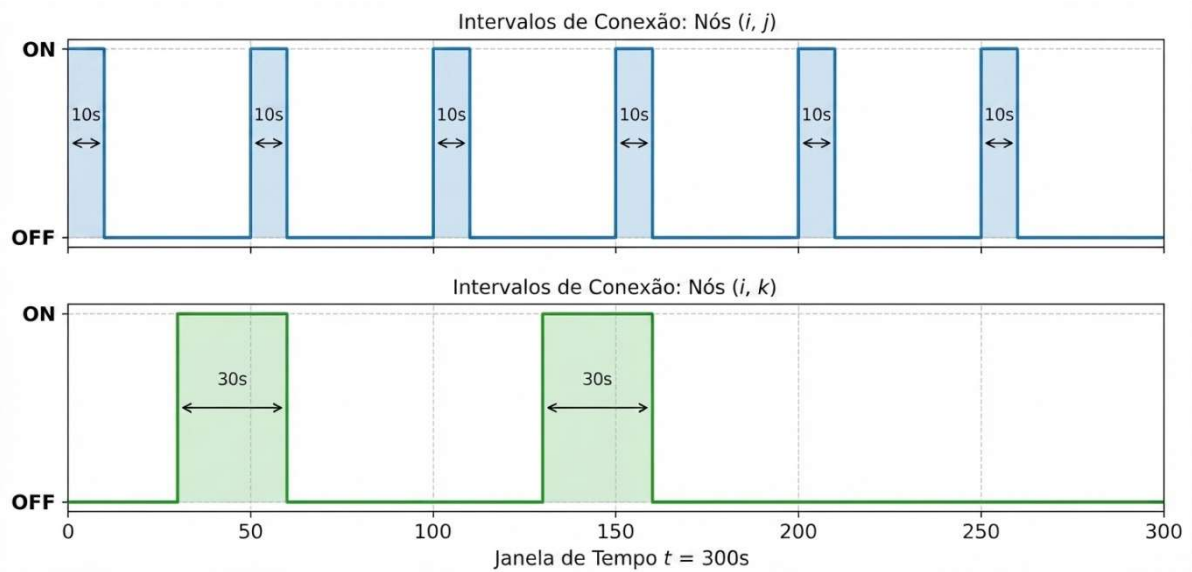


Figura 4.1 - Exemplo ilustrativo de intervalos de conexão

Os intervalos de desconexão correspondentes são exibidos na Figura 4.2. O par (i, j) apresenta cinco intervalos de desconexão uniformes (40 segundos cada), o par (i, k) registra dois intervalos de durações distintas (30s e 70s). Observa-se que, no instante $t = 300s$, o terceiro intervalo de inatividade entre i e k permanece em aberto, não sendo contabilizado como um intervalo de desconexão concluído.

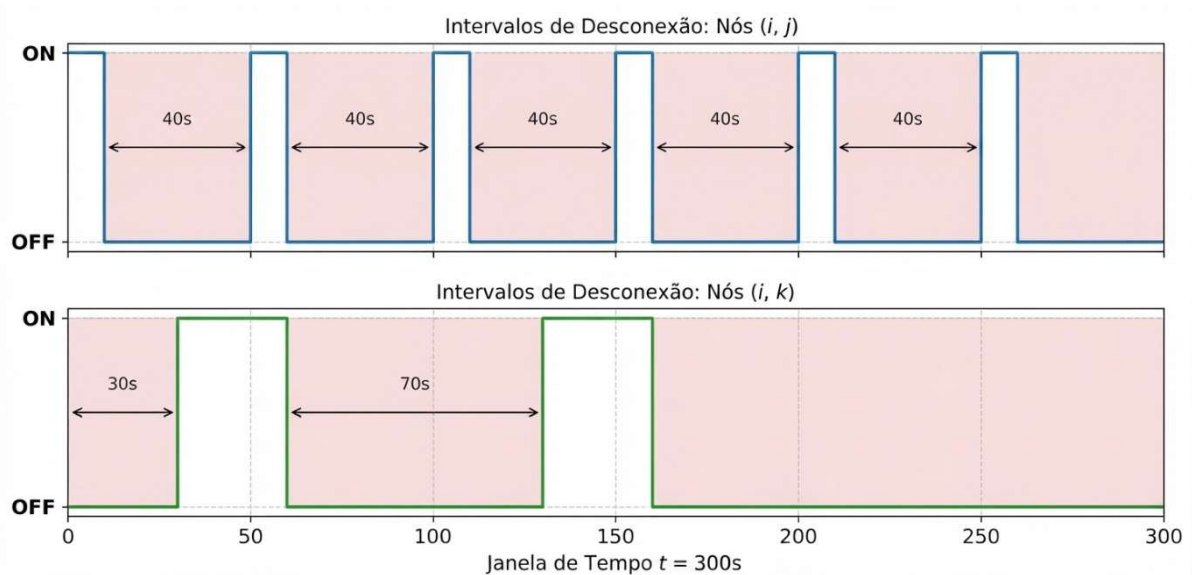


Figura 4.2 - Exemplo ilustrativo de intervalos de desconexão

4.1.1 Força de conexão

A *força de conexão* (S), métrica baseada na intensidade de contato, quantifica a fração de tempo em que dois nós permaneceram conectados em relação ao tempo total observado. A Equação 4.1 define esse cálculo em função de t_{ON} e t_{OFF} :

$$S_{(i,j)} = \frac{t_{on(i,j)}}{t_{on(i,j)} + t_{off(i,j)}}. \quad (4.1)$$

Magnitudes superiores de S indicam vínculos de contato mais intensos. No exemplo da Figura 4.1, o nó i possui força de conexão de 0,2 (ou 20%) em relação aos nós j e k . Contudo, a dinâmica de encontros do par (i,j) difere daquela do par (i,k) , o que justifica a introdução das métricas originais de tendência e regularidade para capturar nuances temporais que a métrica de força, isoladamente, não descreve.

4.1.2 Tendência à conexão

As durações dos intervalos de desconexão entre os nós i e j compõem o conjunto $I_{(i,j)}$. Cada elemento deste conjunto é um valor inteiro positivo, representado por $I_{n(i,j)}$, onde n indica a ordem cronológica do intervalo. Conforme a Figura 4.2, para o par (i,j) ,

$$I_{1(i,j)} = I_{2(i,j)} = \dots = I_{5(i,j)} = 40s,$$

enquanto para o par (i,k) ,

$$I_{1(i,k)} = 30s \text{ e } I_{2(i,k)} = 70s.$$

Cada intervalo $I_{n(i,j)}$ é classificado em uma categoria Ct , que estratifica a duração da desconexão. A categoria é calculada pela Equação 4.2, onde Ref é o intervalo de referência, configurado para 300 segundos. Este valor foi definido

por meio de experimentação preliminar, constatando-se adequado para a estratificação dos intervalos de desconexão nos cenários avaliados.

$$Ct_{(I_{n(i,j)})} = \text{int} \left(\frac{I_{n(i,j)}}{Ref} \right) + 1. \quad (4.2)$$

A categorização identifica a predisposição de dois nós em reestabelecer contato após períodos de separação. Os valores de Ct são transformados por meio do recíproco de sua raiz quadrada, resultando em valores no intervalo $[0,1]$. A média aritmética desses valores define a *tendência à conexão* (T), conforme a Equação 4.3, em que N representa o número de intervalos fechados em $I_{(i,j)}$:

$$T_{(i,j)} = \frac{1}{N} \cdot \sum_{n=1}^N \frac{1}{\sqrt{Ct_{(I_{n(i,j)})}}}. \quad (4.3)$$

O exemplo a seguir ilustra o cálculo da tendência entre dois nós, i e j , que apresentam um histórico social composto por intervalos de desconexão dados pelo conjunto $I_{(i,j)} = \{2102, 367, 607, 348, 609, 227, 372, 1127\}$, em segundos:

A Tabela 4.1 detalha a obtenção das categorias dos intervalos e o cálculo iterativo de $T_{(i,j)}$ em tempo real, após cada inserção de um novo intervalo no histórico.

Tabela 4.1 – Exemplo de cálculo de Tendência à Conexão

n	$I_{n(i,j)}$	$Ct_{(I_{n(i,j)})}$	$\frac{1}{\sqrt{Ct_{(I_{n(i,j)})}}}$	$T_{(i,j)}$
1	2102	$\text{int} (2102 / 300) + 1 = \mathbf{8}$	0,35	0,35
2	367	$\text{int} (367 / 300) + 1 = \mathbf{2}$	0,71	$(0,35 + 0,71) / 2 = \mathbf{0,53}$
3	607	$\text{int} (607 / 300) + 1 = \mathbf{3}$	0,58	$(0,35 + 0,71 + 0,58) / 3 = \mathbf{0,55}$
4	348	$\text{int} (348 / 300) + 1 = \mathbf{2}$	0,71	$(0,35 + \dots + 0,71) / 4 = \mathbf{0,59}$
5	609	$\text{int} (609 / 300) + 1 = \mathbf{3}$	0,58	$(0,35 + \dots + 0,58) / 5 = \mathbf{0,59}$
6	227	$\text{int} (227 / 300) + 1 = \mathbf{1}$	1,00	$(0,35 + \dots + 1,00) / 6 = \mathbf{0,66}$
7	372	$\text{int} (372 / 300) + 1 = \mathbf{2}$	0,71	$(0,35 + \dots + 0,71) / 7 = \mathbf{0,67}$
8	1127	$\text{int} (1127 / 300) + 1 = \mathbf{4}$	0,50	$(0,35 + \dots + 0,50) / 8 = \mathbf{0,65}$

Os valores apresentados na Tabela 4.1 demonstram que a tendência à conexão é incrementada quando os intervalos de desconexão são curtos, sinalizando maior predisposição a encontros iminentes. Em contrapartida, valores reduzidos de $T_{(i,j)}$ indicam que os nós i e j necessitam aguardar longos períodos até estabelecerem uma nova conexão, o que diminui o potencial desta parceria para o encaminhamento eficiente de mensagens em redes oportunistas.

4.1.3 Regularidade entre conexões

A *regularidade entre conexões* (R) mensura a similaridade entre as durações dos intervalos de desconexão, variando no intervalo $[0,1]$. Para avaliá-la, calcula-se a média aritmética $\mu_{I_{(i,j)}}$ e a variância populacional $\sigma_{(i,j)}^2$ do conjunto $I_{(i,j)}$:

$$\mu_{I_{(i,j)}} = \frac{1}{N} \cdot \sum_{n=1}^N I_{n(i,j)} . \quad (4.4)$$

A Equação 4.4 é usada no cálculo da variância $\sigma_{(i,j)}^2$, descrita na Equação 4.5:

$$\begin{aligned} \sigma_{(i,j)}^2 &= \frac{1}{N} \cdot \sum_{n=1}^N \left(I_{n(i,j)} - \mu_{I_{(i,j)}} \right)^2 , \\ \sigma_{(i,j)}^2 &= \frac{1}{N} \cdot \sum_{n=1}^N \left(I_{n(i,j)}^2 - 2\mu_{I_{(i,j)}} I_{n(i,j)} + \mu_{I_{(i,j)}}^2 \right) , \\ \sigma_{(i,j)}^2 &= \frac{1}{N} \cdot \sum_{n=1}^N (I_{n(i,j)}^2) - 2\mu_{I_{(i,j)}} \left(\frac{1}{N} \cdot \sum_{n=1}^N I_{n(i,j)} \right) + \frac{1}{N} \cdot N\mu_{I_{(i,j)}}^2 , \\ \sigma_{(i,j)}^2 &= \frac{1}{N} \cdot \sum_{n=1}^N (I_{n(i,j)}^2) - \mu_{I_{(i,j)}}^2 . \end{aligned} \quad (4.5)$$

Uma variância reduzida indica que os intervalos de desconexão possuem durações próximas à média, sugerindo alta previsibilidade. No caso particular em

que $\sigma_{(i,j)}^2 = 0$, todos os intervalos possuem durações idênticas. Para normalizar essa métrica, calcula-se a máxima variância possível ($\sigma_{max(i,j)}^2$) com base em um conjunto abstrato $I_{max(i,j)}$, o qual deve possuir a mesma média e o mesmo número de elementos de $I_{(i,j)}$, mas com a maior dispersão possível. Desse modo, a Equação 4.5 pode ser modificada para resultar na máxima variância, conforme a Equação 4.6.

$$\sigma_{max(i,j)}^2 = \frac{1}{N} \cdot \sum_{n=1}^N (I_{max,n(i,j)}^2) - \mu_{I(i,j)}^2. \quad (4.6)$$

Na Equação 4.6, o único componente que deve ser calculado de forma independente do histórico real é a soma dos quadrados dos elementos do conjunto $I_{max(i,j)}$. Para que esta soma seja a maior possível, os valores dos elementos de $I_{max(i,j)}$ são atribuídos do seguinte modo:

1. Duração unitária para $(N - 1)$ intervalos,
2. Duração remanescente de $\sum_{n=1}^N I_{n(i,j)} - (N - 1)$ para um único intervalo.

A demonstração formal para a construção de $I_{max(i,j)}$ fundamenta-se na maximização da soma dos quadrados sob a restrição de média constante.

Teorema 1: Dado um conjunto de N inteiros positivos, cuja soma de elementos é representada por Z , a máxima soma de quadrados é obtida quando um dos elementos do conjunto é igual a $Z - (N - 1)$ e os outros $(N - 1)$ elementos são iguais a 1.

Prova: Considera-se um conjunto de N inteiros positivos, $A = \{a_1, a_2, \dots, a_N\}$. A soma dos elementos de A é representada por Z_A e a soma dos quadrados é representada por Q_A . O objetivo é determinar valores para os elementos de A tais que Q_A tenha o maior valor possível.

Assume-se um conjunto de valores $A = \{x, 1, \dots, 1\}$, $x = Z_A - (N - 1)$. Nesse caso, a soma dos quadrados é dada por:

$$\mathcal{Q}_A = x^2 + (N - 1). \quad (4.7)$$

Em seguida, avalia-se outra combinação de valores para o conjunto A . Para tal finalidade, considera-se um inteiro positivo $y = y_1 + y_2 + \dots + y_{N-1}$, cujos adendos sejam inteiros não-negativos. Subtrai-se y de x , distribuindo seus adendos entre os demais elementos de A . O resultado é o conjunto Y e a soma dos quadrados de seus elementos é dada por $\mathcal{Q}_Y$:

$$Y = \{x - y, 1 + y_1, 1 + y_2, \dots, 1 + y_{N-1}\}, \quad (4.8)$$

$$\mathcal{Q}_Y = (x - y)^2 + (1 + y_1)^2 + (1 + y_2)^2 + \dots + (1 + y_{N-1})^2,$$

$$\mathcal{Q}_Y = (x - y)^2 + 1 + 2y_1 + y_1^2 + \dots + 1 + 2y_{N-1} + y_{N-1}^2,$$

$$\mathcal{Q}_Y = (x - y)^2 + (N - 1) + 2y + \sum_{n=1}^{N-1} y_n^2. \quad (4.9)$$

O somatório da Equação 4.9 pode ser reescrito em função de y^2 .

$$y^2 = (y_1^2 + y_2^2 + \dots + y_{N-1}^2) + 2 \cdot \sum_{1 \leq n < m \leq N-1} y_n y_m,$$

$$\sum_{n=1}^{N-1} y_n^2 = y^2 - 2 \cdot \sum_{1 \leq n < m \leq N-1} y_n y_m. \quad (4.10)$$

Inserindo a Equação 4.10 na Equação 4.9:

$$\mathcal{Q}_Y = (x - y)^2 + (N - 1) + 2y + y^2 - 2 \cdot \sum_{1 \leq n < m \leq N-1} y_n y_m,$$

$$\mathcal{Q}_Y = x^2 - 2xy + y^2 + (N - 1) + 2y + y^2 - 2 \cdot \sum_{1 \leq n < m \leq N-1} y_n y_m. \quad (4.11)$$

Após inserir a Equação 4.7 na Equação 4.11 e organizar os termos:

$$\mathcal{Q}_Y = \mathcal{Q}_A - 2 \left[y(x - y - 1) + \sum_{1 \leq n < m \leq N-1} y_n y_m \right]. \quad (4.12)$$

Como os elementos do conjunto Y são inteiros positivos, $x - y \geq 1$. Desse modo, $\mathcal{Q}_A \geq \mathcal{Q}_Y$ para todo $0 \leq y \leq x - 1$. A igualdade $\mathcal{Q}_A = \mathcal{Q}_Y$ é observada quando y e apenas um de seus adendos são, ambos, iguais a $x - 1$, porque esta condição garante que os subtraendos da Equação 4.12 sejam iguais a zero. De acordo com a Equação 4.12, a substituição de y e um de seus adendos por $x - 1$ resulta em combinações iguais de elementos para os conjuntos A e Y .

Portanto, a máxima soma de quadrados para o conjunto A é obtida quando $\mathcal{Q}_A = \mathcal{Q}_Y$, que corresponde ao caso em que um dos elementos deste conjunto seja igual a $Z_A - (N - 1)$ e os demais elementos sejam iguais a 1. $\square$

O Teorema 1 permite que a soma dos quadrados da Equação 4.6 e a máxima variância sejam reescritas em função dos elementos de $I_{(i,j)}$.

$$\sum_{n=1}^N (I_{\max,n(i,j)}^2) = (N - 1) + \left(\sum_{n=1}^N (I_{n(i,j)}) - (N - 1) \right)^2, \quad (4.13)$$

$$\sigma_{\max(i,j)}^2 = \frac{1}{N} \cdot \left[(N - 1) + \left(\sum_{n=1}^N (I_{n(i,j)}) - (N - 1) \right)^2 \right] - \mu_{I(i,j)}^2. \quad (4.14)$$

Finalmente, a *regularidade entre conexões* é definida pela Equação 4.15:

$$R_{(i,j)} = 1 - \frac{\sigma_{(i,j)}^2}{\sigma_{\max(i,j)}^2}. \quad (4.15)$$

Valores elevados de regularidade indicam que o par de nós está predisposto a se encontrar em intervalos previsíveis, enquanto valores baixos sugerem um

histórico de incerteza, tipicamente caracterizado por uma alternância aleatória entre períodos curtos e longos de desconexão.

A partir dos conceitos de tendência e regularidade, define-se a *consistência* φ do nó i como o somatório dos produtos $T_{(i,v)}R_{(i,v)}$, $v \in V_R$, onde V_R é o conjunto dos nós da rede com os quais i possua um valor de regularidade R (Equação 4.16).

$$\varphi_i = \sum_{v \in V_R} (T_{(i,v)} \cdot R_{(i,v)}). \quad (4.16)$$

A consistência está associada à capacidade de um nó em reestabelecer contatos com um grande número de nós da rede, de modo rápido e previsível, sendo interpretada como um indicador de centralidade. Desse modo, a consistência é um parâmetro utilizado na difusão e na manutenção de cópias de mensagens pela rede.

4.2 Scores

As métricas de força (S), tendência (T) e regularidade (R) fornecem uma visão holística do histórico de encontros entre os nós da rede. Ao considerar esses três parâmetros, o Protocolo CASTRO busca priorizar pares de nós que: (i) permaneçam conectados por períodos prolongados; (ii) reestabeleçam contato rapidamente; ou (iii) possuam um histórico de reconexão previsível.

Para essa finalidade, o Protocolo CASTRO utiliza os parâmetros supracitados para calcular indicadores sociais, denominados *Scores*, atribuídos a pares de nós para orientar o repasse de mensagens. Os indicadores dividem-se em duas categorias: diretos e transitivos, descritos nas subseções seguintes.

4.2.1 Scores diretos

O *Score direto* $DirSc_{(i,j)}$, calculado pelo nó i em relação ao nó j , é um valor normalizado entre 0 e 1, obtido pelas Equações 4.17. A formulação ajusta-se

conforme a existência de um ou mais intervalos de desconexão, incorporando a regularidade (R) apenas quando $N \geq 2$.

$$\begin{cases} DirSc_{(i,j)} = S_{(i,j)} + (1 - S_{(i,j)}) \cdot T_{(i,j)} & \text{para } N = 1, \\ DirSc_{(i,j)} = S_{(i,j)} + (1 - S_{(i,j)}) \cdot T_{(i,j)} \cdot R_{(i,j)} & \text{para } N \geq 2. \end{cases} \quad (4.17)$$

Conforme as Equações 4.17, se os nós i e j apresentarem uma elevada força de conexão $S_{(i,j)}$, o *Score* direto será próximo de 1. Caso permaneçam em contato direto por pouco tempo, o valor de $DirSc_{(i,j)}$ ainda poderá ser elevado caso o histórico exiba múltiplos intervalos de desconexão curtos e regulares.

4.2.2 Scores transitivos

A transitividade permite que um nó i avalie seu potencial de repasse a um destino j por intermédio de um terceiro nó k . A heurística transitiva $H_{(i,k,j)}$ entre i e j – tendo k como intermediário – é calculada pela média geométrica entre dois *Scores* diretos, conforme a Equação 4.18:

$$H_{(i,k,j)} = \sqrt{DirSc_{(i,k)} \cdot DirSc_{Upd(k,j)}}. \quad (4.18)$$

O *Score* direto entre o nó intermediário e o receptor deve ser previamente atualizado, conforme as Equações 4.19. Se os nós k e j estiverem conectados, $DirSc_{Upd(k,j)}$ utiliza os valores mais recentes de $S_{(k,j)}$, $T_{(k,j)}$ e $R_{(k,j)}$. Caso estejam desconectados, aplica-se um fator de envelhecimento baseado no tempo decorrido $\tau_{Dir(k,j)}$ desde a última transição $ON \rightarrow OFF$ entre k e j .

$$\begin{cases} DirSc_{Upd(k,j)} = S_{(k,j)} + (1 - S_{(k,j)}) \cdot T_{(k,j)} & \text{modo } ON (N = 1) \\ DirSc_{Upd(k,j)} = S_{(k,j)} + (1 - S_{(k,j)}) \cdot T_{(k,j)} \cdot R_{(k,j)} & \text{modo } ON (N \geq 2). \\ DirSc_{Upd(k,j)} = \xi^{\tau_{Dir(k,j)}} \cdot DirSc_{(k,j)} & \text{modo } OFF \end{cases} \quad (4.19)$$

O *Score transitivo* $TrSc_{(i,j)}$ resulta da comparação entre a heurística $H_{(i,k,j)}$ e o valor atualizado de $TrSc_{(i,j)}$, sendo este previamente submetido a um fator de envelhecimento. O maior valor entre ambos define o novo $TrSc_{(i,j)}$ (Equação 4.20), onde $\tau_{Tr(i,j)}$ representa o tempo decorrido desde a última atualização:

$$TrSc_{(i,j)} = \max(\xi^{\tau_{Tr(i,j)}} \cdot TrSc_{(i,j)}, H_{(i,k,j)}). \quad (4.20)$$

Na primeira computação, o valor inicial de $TrSc_{(i,j)}$ é zero, tornando o resultado igual à heurística $H_{(i,k,j)}$. Nas execuções subsequentes, aplica-se a Equação 4.20 após o envelhecimento do *Score* $TrSc_{(i,j)}$ e o cálculo da transitividade $H_{(i,k,j)}$.

A constante ξ , configurada como $\xi = 1 - 10^{-5}$ para o protocolo CASTRO com base em análises de sensibilidade preliminares, rege a taxa de envelhecimento dos *Scores* diretos e transitivos. Como as variáveis temporais τ_{Dir} e τ_{Tr} operam de forma independente, $DirSc_{(i,j)}$ e $TrSc_{(i,j)}$ decaem separadamente, permitindo flexibilidade na tomada de decisão do roteamento.

Nesta pesquisa, a integração dos *Scores* diretos e transitivos permite identificar nós com contatos locais frequentes e aqueles que atuam como pontes eficientes para destinos distantes. Essa abordagem, associada a uma estratégia de repasse estruturada detalhada no Capítulo 5, contribui para a obtenção de elevadas taxas de entrega na rede.

5. PROTOCOLO CASTRO

5.1 Introdução

No capítulo anterior, foram descritas a força, a tendência e a regularidade entre conexões. Enquanto a métrica de força representa um conceito adaptado da literatura, a formulação conjunta de tendência e regularidade constitui a base original para o cálculo de indicadores sociais entre pares de nós, denominados *Scores* (diretos e transitivos). Adicionalmente, a combinação desses parâmetros viabiliza a computação da consistência de um nó, que atua como um indicador de centralidade social.

Este capítulo apresenta o Protocolo de Roteamento CASTRO. O protocolo integra os *Scores* para computar chaves de roteamento (K), orientando nós emissores e intermediários na priorização de candidatos com potencial para entregar mensagens aos destinos, conforme definido na Equação 5.1:

$$K_{(i,j)} = DirSc_{(i,j)} + TrSc_{(i,j)}. \quad (5.1)$$

O Protocolo CASTRO utiliza essas chaves (Equação 5.1) em conjunto com a consistência dos nós (Equação 4.16) para estruturar o mecanismo de controle e decisão de transmissão de mensagens.

A próxima seção detalha os métodos de suporte, encarregados de: (i) atualizar parâmetros de conexão; (ii) atribuir *Scores*; (iii) gerenciar o tempo de vida útil (TTL) das mensagens; e (iv) realizar a gestão do *buffer*. Em seguida, descreve-se o algoritmo principal.

5.2 Métodos de suporte

O Algoritmo 5.1 apresenta o método *updParameters*, responsável por computar a tendência (linha 03) e a regularidade (linha 06) entre os nós i e j . O produto entre ambas as métricas é armazenado (linha 07) para uso posterior no cálculo da consistência φ_i .

Método *updParameters* (*i, j*)

01: **se** nó *i* encontrar nó *j* **então**
02: *i* atualiza $I_{(i,j)}$
03: *i* calcula $T_{(i,j)}$ (Eq. 4.3)
04: **se** $N \geq 2$ **então**
05: *i* obtém $\sigma_{(i,j)}^2$ e $\sigma_{max(i,j)}^2$ (Eq. 4.5 e 4.14)
06: *i* calcula $R_{(i,j)}$ (Eq. 4.15)
07: *i* armazena o produto $T_{(i,j)} \cdot R_{(i,j)}$
08: **fim se**
09: **fim se**

Algoritmo 5.1 – Método *updParameters*

O Algoritmo 5.2 descreve o método *updDirScore*, encarregado de atualizar a força de conexão (linha 02) e de calcular – ou envelhecer – o *Score* direto entre os nós *i* e *j* (linhas 03 ou 05).

Método *updDirScore* (*i, j*)

01: **se** nó *i* encontrar nó *j* **então**
02: *i* atualiza $S_{(i,j)}$ (Eq. 4.1)
03: *i* calcula $DirSc_{(i,j)}$ (Eq. 4.17)
04: **senão**
05: *i* envelhece $DirSc_{(i,j)}$ (Eq. 4.19, OFF)
06: **fim se**

Algoritmo 5.2 – Método *updDirScore*

O Algoritmo 5.3 detalha o método *updTrScores*. Quando o nó *i* contata o nó *j*, este último atua como intermediário para um conjunto de nós *V* com os quais *j* possui *Score* direto. O nó *i* atualiza seus *Scores* transitivos em relação aos elementos de *V* por intermédio de *j*. Para cada $v \in V$, o *Score* direto entre *j* e *v* é atualizado (linhas 04 – 08), a heurística transitiva é computada (linha 09) e o novo *Score* transitivo é estabelecido (linhas 10 – 14), atualizando-se o registro temporal de envelhecimento $\tau_{Tr(i,v)}$ (linha 15).

Método $updTrScores(i, j)$

```
01: se nó  $i$  encontrar nó  $j$  então
02:    $V =$  nós da rede com quem nó  $j$  tenha Score direto
03:   para todo  $v \in V$  faça
04:     se  $j$  e  $v$  estiverem conectados, então
05:        $j$  calcula  $DirSc_{Upd(j,v)}$  (Eq. 4.19, ON)
06:     senão
07:        $j$  envelhece  $DirSc_{(j,v)}$  (Eq. 4.19, OFF)
08:     fim se
09:      $i$  calcula  $H_{(i,j,v)}$  (Eq. 4.18)
10:     se  $TrSc_{(i,v)} = \{\emptyset\}$  ou  $(H_{(i,j,v)} \geq \xi^{\tau_{Tr(i,v)}} \cdot TrSc_{(i,v)})$  então
11:        $i$  atribui  $TrSc_{(i,v)} = H_{(i,j,v)}$  (Eq. 4.20)
12:     senão
13:        $i$  atribui  $TrSc_{(i,v)} = \xi^{\tau_{Tr(i,v)}} \cdot TrSc_{(i,v)}$  (Eq. 4.20)
14:     fim se
15:      $i$  atualiza  $\tau_{Tr(i,v)}$ 
16:   fim para
17: fim se
```

Algoritmo 5.3 – Método $updTrScores$

Com o propósito de reduzir a saturação dos canais de comunicação e evitar a retenção indefinida de mensagens, o Protocolo CASTRO emprega um mecanismo de rejuvenescimento proporcional de duplo gatilho, que restaura uma fração do TTL transcorrido com base em critérios de utilidade social.

Ao ocorrer um contato entre os nós i e j , o cálculo do novo TTL de uma mensagem m , destinada ao nó d , pondera o fator de vantagem social $f_{(d,i,j)}$ frente à margem temporal disponível entre o limite máximo (TTL_{max}) e o valor atual ($TTL(m)$). A Equação 5.2 assegura que a prorrogação respeite estritamente o teto máximo permitido:

$$TTL_{novo}(m) = \min\left(TTL_{max}, TTL(m) + f_{(d,i,j)} \cdot (TTL_{max} - TTL(m))\right). \quad (5.2)$$

Dessa forma, a extensão do tempo de vida mostra-se proporcional à qualidade da rota avaliada, permitindo que caminhos promissores recuperem margens temporais e reduzam o descarte prematuro de mensagens sem comprometer a estabilidade do tráfego.

O Algoritmo 5.4 define a apuração do fator $f_{(d,i,j)}$, derivado das chaves comparadas $K_{(i,d)}$ e $K_{(j,d)}$ (linhas 02 – 03), ou das consistências φ_i e φ_j (linhas 04 – 05). Na ausência de vantagem comparativa evidente, o fator adota um parâmetro fixo de recuo (*default*) (linhas 06 – 07).

Método factor (d, i, j)

```

01: se nó  $i$  encontrar nó  $j$  então
02:   se  $K_{(j,d)} > K_{(i,d)}$ 
03:      $f_{(d,i,j)} = K_{(j,d)} / (K_{(i,d)} + K_{(j,d)})$ 
04:   senão se  $\varphi_j > \varphi_i$ 
05:      $f_{(d,i,j)} = \varphi_j / (\varphi_i + \varphi_j)$ 
06:   senão
07:      $f_{(d,i,j)} = default$ 
08:   fim se
09: fim se

```

Algoritmo 5.4 – Método *factor*

Para preservar o equilíbrio de recursos, o parâmetro *default* varia conforme o contexto operacional. Em rotinas restritas à atualização topológica do TTL (troca de metadados sem repasse físico de pacotes), o recuo é nulo, evitando extensões artificiais de vida útil em rotas sem ganho estrutural comprovado. Por outro lado, na difusão física de cópias e em bonificações de percurso, o *default* adota o valor de 0,5.

O Algoritmo 5.5 exhibe o método *renewTTL*, responsável por atualizar o TTL de todas as mensagens presentes no buffer (linhas 02 – 06), fixando o parâmetro *default* em zero para o cálculo preliminar de $f_{(d,i,j)}$ (linha 04).

Método *renewTTL* (*i*, *j*)

01: **se** nó *i* encontrar nó *j* **então**
02: **para toda** $m \in \text{buffer}(i)$ **faça**
03: $d =$ nó de destino da mensagem m
04: $f_{(d,i,j)} = \text{factor}(d, i, j)$ // default = 0 (Algoritmo 5.4)
05: $TTL_{\text{novo}}(m) = \min(TTL_{\text{max}}, TTL(m) + f_{(d,i,j)} \cdot (TTL_{\text{max}} - TTL(m)))$ (Eq. 5.2)
06: **fim para**
07: **fim se**

Algoritmo 5.5 – Método *renewTTL* (Protocolo CASTRO)

O Algoritmo 5.6 corresponde ao método *bufferManagement*, focado na propagação de notificações de entrega de mensagens. Esse procedimento otimiza o uso do armazenamento local ao descartar mensagens previamente entregues aos nós de destino, reduzindo os riscos de saturação do *buffer*.

Método *bufferManagement* (*i*)

01: **Para todo** $v \in \{\text{Nós conectados a nó } i\}$ **faça**
02: **se** $((D_{(i,v)\text{past}} = \text{null}) \text{ ou } (D_{(i,v)\text{new}} \neq D_{(i,v)\text{past}}))$ **então**
03: **para toda** $m \in D_v$ **e** $m \notin D_i$ **faça**
04: *i* adiciona m a D_i
05: **se** $m \in i$ **então**
06: *i* remove m do seu *buffer*
07: **fim se**
08: **fim para**
09: **fim se**
10: **fim para**

Algoritmo 5.6 – Método *bufferManagement*

A estrutura $D_{(i)}$ representa a tabela de confirmações de entregas gerida por *i*. Diante de um novo contato com o nó *v*, o nó *i* precisa verificar se houve alterações em $D_{(v)}$ (linha 02). Denomina-se $D_{(i,v)\text{new}}$ a informação mais recente de $D_{(v)}$, que está sendo fornecida de *v* para *i*. De modo similar, $D_{(i,v)\text{past}}$ corresponde a uma informação prévia de $D_{(v)}$, fornecida ao nó *i* em um momento passado. Em caso de conexões inéditas, $D_{(i,v)\text{past}}$ será nula.

Esse procedimento cumpre dupla finalidade: atualizar o repositório local de notificações (linha 04) e remover do *buffer* as mensagens validadas como entregues (linha 06). Tais notificações são transmitidas sequencialmente por toda a rede. Na seção seguinte, apresenta-se o método principal de encaminhamento do Protocolo CASTRO.

5.3 Método *Main*

O Algoritmo 5.7 modela a estrutura principal do Protocolo CASTRO. Ele atribui L cópias de encaminhamento iniciais para cada mensagem, utilizando chaves e consistências para controlar os repasses pela rede. O procedimento integra as funções de suporte supramencionadas e aciona a rotina de envio (*sendMessages*).

Conforme o Algoritmo 5.7, ao detectar a presença de j , o nó i executa os Algoritmo 5.1 – Algoritmo 5.6 para calcular os dois tipos de *Scores* (linhas 03 e 04), gerenciar *buffer* (linha 05) e reajustar o TTL das mensagens (linha 06). Caso o nó i possua conexão direta com o destino final de alguma mensagem residente, o repasse é executado imediatamente, acompanhado da remoção do *buffer* e da atualização da tabela de confirmações (linhas 09 – 12).

Na ausência de destinos diretos, avalia-se a elegibilidade de repasse a nós intermediários (linhas 16 – 26), registrando os pares compatíveis em um vetor de tuplas *msg*. Essa verificação estende-se a todos os vizinhos ativos conectados ao nó i , checando a disponibilidade de cópias e a superioridade de chave ou consistência do receptor frente ao remetente.

Destaca-se que a chave orienta o repasse para nós com maior probabilidade de alcançar o destino final, enquanto a consistência prioriza nós dotados de elevada centralidade social, estimulando a difusão controlada pela topologia.

Consolidado o vetor *msg*, as ordens de transmissão são ordenadas por critério de antiguidade (linha 27) e submetidas ao método *sendMessages* (linha 28). Ao término da conexão com j , procede-se à atualização final do *Score* direto (linha 30), permitindo maior precisão em futuros cálculos transitivos na rede.

Método Main

```
01: se nó  $i$  encontrar nó  $j$  então
02:    $updParameters(i, j)$  (Algoritmo 5.1)
03:    $updDirScore(i, j)$  (Algoritmo 5.2)
04:    $updTrScores(i, j)$  (Algoritmo 5.3)
05:    $bufferManagement(i)$  (Algoritmo 5.6)
06:    $renewTTL(i, j)$  (Algoritmo 5.5)
07:   para toda  $m \in buffer(i)$  faça
08:      $d =$  nó de destino da mensagem  $m$ 
09:     se  $d \in \{\text{Nós conectados a nó } i\}$  então
10:        $i$  encaminha  $m$  ao nó  $d$ 
11:        $i$  adiciona  $m$  a  $D(i)$ 
12:        $i$  remove  $m$  do seu  $buffer$ 
13:     fim se
14:   fim para
15:   Defina  $msg =$  vetor de tuplas (mensagem, nó receptor)
16:   para todo  $v \in \{\text{Nós conectados a nó } i\}$  faça
17:     para toda  $m \in buffer(i)$  faça
18:        $d =$  nó de destino da mensagem  $m$ 
19:        $i$  e  $v$  calculam chaves e consistências (Eq. 4.16, 5.1)
20:       se  $L(m) > 1$  então
21:          $i$  adiciona a tupla  $(m, v)$  ao vetor  $msg$ 
22:       senão se  $L(m) == 1$  e  $(K_{(v,d)} > K_{(i,d)}$  ou  $\varphi_v > \varphi_i)$ 
23:          $i$  adiciona a tupla  $(m, v)$  ao vetor  $msg$ 
24:       fim se
25:     fim para
26:   fim para
27:    $i$  organiza  $msg$  priorizando as mensagens  $m$  mais novas
28:    $sendMessages(msg)$  (Algoritmo 5.8)
29: senão
30:    $updDirScore(i, j)$  (Algoritmo 5.2)
31: fim se
```

Algoritmo 5.7 – Método *Main* (Protocolo CASTRO)

O Algoritmo 5.8 detalha o método *sendMessages*, encarregado da divisão de cópias e da atualização síncrona do TTL nos casos em que ocorra algum encaminhamento.

Método *sendMessages* (*msg*)

Nó transmissor i :

```
01: Para cada tupla  $(m, v) \in msg, m \notin buffer(v)$  faça
02:    $L_{i_{old}}(m)$  = número de cópias de  $m$ 
03:    $d$  = nó destino de  $m$ 
04:    $f_{(d,i,v)} = factor(d, i, v)$  // default = 0,5 (Algoritmo 5.4)
05:   se  $L_{i_{old}}(m) > 1$  então
06:      $L_{v_{new}}(m) = \lfloor (1/2) \cdot f_{(d,i,v)} \cdot L_{i_{old}}(m) \rfloor$ 
07:      $L_{i_{new}}(m) = L_{i_{old}}(m) - L_{v_{new}}(m)$ 
08:   senão se  $L_{i_{old}}(m) = 1$  e  $BO_{(i)} < BO_{th}$  e  $BO_{(v)} < BO_{th}$  então
09:     se  $(K_{(v,d)} > K_{(i,d)})$  ou  $\varphi_v > \varphi_i$  então
10:        $L_{v_{new}}(m) = 1$ 
11:        $L_{i_{new}}(m) = 0$ 
12:     fim se
13:   fim se
14:   se  $L_{v_{new}}(m) > 0$  então
15:      $i$  encaminha  $L_{v_{new}}$  cópias de  $m$  a  $v$ 
16:      $i$  altera número de cópias de  $m$  para  $L_{i_{new}}$ 
17:   fim se
18:   Nós  $i$  e  $v$  atualizam  $TTL_{new(m)}$  (Eq. 5.2)
19: fim para
```

Algoritmo 5.8 – Método *sendMessages* (Protocolo CASTRO)

Conforme descrito pelo Algoritmo 5.8, o fator $f_{(d,i,v)}$ (linha 04) modula a divisão proporcional de cópias na fase de disseminação (linhas 05 – 07) e o reajuste do TTL da mensagem (linha 18).

Caso o transmissor disponha de apenas uma cópia remanescente, o sistema avalia os níveis de ocupação dos buffers (BO). Se ambos permanecerem abaixo do limiar estipulado ($BO_{th} = 90\%$; linha 08), as chaves ou as consistências são usadas como critérios de decisão para o encaminhamento ou retenção da última cópia (linhas 09 – 12).

Importa ressaltar que, ao repassar mensagens a intermediários, o nó emissor pode esgotar suas próprias cópias de repasse (linha 11) sem, contudo, remover

o item do armazenamento local. Essa estratégia preserva a mensagem para potenciais entregas diretas futuras ao destino. Desse modo, a remoção de mensagens do *buffer* restringe-se a quatro ocorrências: (i) entrega direta ao nó de destino; (ii) eliminação por confirmação de entrega recebida via rede (Algoritmo 5.6); (iii) expiração do TTL; ou (iv) ocupação total do *buffer*, acionando o descarte de registros mais antigos.

Essa retenção controlada viabiliza uma disseminação estruturada. Apenas nós com cotas ativas de repasse ($L(m) > 0$) encaminham mensagens a intermediários. Na prática, esta política funciona de modo similar à gestão de *tokens*, permitindo uma presença sistêmica superior à alocação inicial de réplicas. Desse modo, o Protocolo CASTRO realiza uma difusão mais ampla que o *Spray And Wait* e, ao mesmo tempo, mais restrita e eficaz que o PProPHET ou o protocolo Epidêmico.

5.4 Análise de Complexidade Computacional

Do ponto de vista analítico, a complexidade computacional do Algoritmo 5.7 vincula-se à densidade da vizinhança e ao volume do *buffer* local, sendo parametrizada pelo número de vizinhos ativos $|V|$ e pelo quantitativo de mensagens residentes $|B|$.

A checagem de entregas diretas demanda uma varredura linear de ordem $O(|B|)$. Na sequência, a seleção de mensagens para repasse realiza avaliações aninhadas, comparando todas as $|B|$ mensagens contra todos os $|V|$ vizinhos, impondo uma complexidade de tempo limitada a $O(|V| \cdot |B|)$. Como os parâmetros internos de chave (K) e consistência (φ) são consultados via estruturas de tabelas *hash* com complexidade $O(1)$, não se verificam sobrecargas adicionais de processamento aninhado.

O principal encargo computacional concentra-se na fase de escalonamento. Considerando uma lista agregada de encaminhamento de cardinalidade máxima $|F_{total}|$ (onde $|F_{total}| \leq |V| \cdot |B|$), a ordenação de tarefas impõe um custo

de $O(|F_{total}| \log |F_{total}|)$. Logo, a complexidade temporal global do algoritmo é dada por $O(|V| \cdot |B| + |F_{total}| \log |F_{total}|)$.

Tendo consolidado o protocolo CASTRO, o próximo passo desta pesquisa é conferir aos nós da rede a capacidade de aprendizado contínuo, visando à minimização dos tempos de atraso. Nesse sentido, o Capítulo 6 aborda os fundamentos de aprendizado por reforço aplicados a redes oportunistas, preparando a base teórica para a variação aprimorada QL-CASTRO, descrita no Capítulo 7.

6. APRENDIZADO POR REFORÇO EM REDES OPORTUNISTAS

6.1 Aprendizado de máquina no roteamento de dados

As redes oportunistas operam em ambientes de conectividade intermitente e topologia dinâmica, cenários nos quais os algoritmos heurísticos tradicionais frequentemente demonstram limitações de flexibilidade e adaptabilidade (Liaq *et al.*, 2025). Para contornar esses desafios, técnicas de inteligência artificial – com destaque para o aprendizado de máquina (*machine learning*) – vêm sendo adotadas para otimizar a tomada de decisão no roteamento (Gandhi e Narmawala, 2024).

Em contraste com as estratégias baseadas em regras estáticas, os modelos de aprendizado de máquina processam dados históricos e contextuais da rede — como durações de contato, intervalos de desconexão e ocupação de *buffer* — para estimar padrões de mobilidade e probabilidades de entrega de mensagens. Como ilustração, Alaoui *et al.* (2022) desenvolveram um sistema que utiliza classificadores (árvores de decisão) e técnicas de interpretação (valores SHAP) para prever o sucesso na entrega de *bundles*, reduzindo o desperdício de recursos e a sobrecarga na rede (*overhead*).

A gestão eficiente de recursos escassos também se beneficia dessas abordagens, uma vez que modelos analíticos de alta complexidade podem ser complementados ou substituídos por previsões mais rápidas, essenciais em redes de dispositivos IoT com restrições de energia e de capacidade de armazenamento (Liaq *et al.*, 2025).

Embora os métodos supervisionados sejam eficazes em tarefas de previsão e classificação de enlaces, o roteamento oportunista é um problema de decisão sequencial sob forte incerteza. Para lidar de forma nativa com a evolução temporal do ambiente e estabelecer políticas de transmissão adaptativas, o aprendizado por reforço (*reinforcement learning*) destaca-se como uma abordagem robusta, permitindo que os nós atuem como agentes autônomos na otimização de repasses (Gandhi e Narmawala, 2024). Este paradigma é detalhado nas próximas seções.

6.2 Processos de decisão de Markov

O roteamento em redes oportunistas impõe aos nós a necessidade de tomar sequências contínuas de decisões em ambientes estocásticos, nos quais os efeitos das ações não são estritamente determinísticos. Para formalizar matematicamente essa dinâmica, diversos estudos utilizam a formulação de Processos de Decisão de Markov (*Markov Decision Processes* - MDP) (Oualhaj et al., 2018; Liu et al., 2019; Saad et al., 2023). A teoria de MDP provê o embasamento formal necessário para o aprendizado por reforço, viabilizando a resolução de problemas de decisão sequencial com foco na maximização de retornos cumulativos a longo prazo (Sutton e Barto, 2018).

O problema de aprendizado é formalizado por meio de uma interface clássica de interação entre agente e ambiente, conforme ilustrado na Figura 6.1.

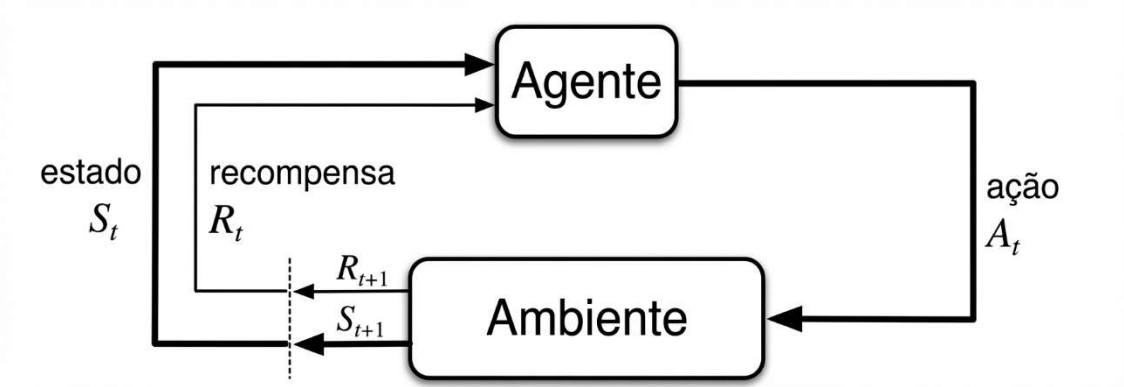


Figura 6.1 – Interação agente-ambiente em um processo de decisão de Markov. Traduzido de Sutton e Barto (2018).

Em cada instante de tempo discreto $t = 0, 1, 2, \dots$, o *agente* observa o *estado* corrente do ambiente $S_t \in \mathcal{S}$ e seleciona uma *ação* $A_t \in \mathcal{A}(s)$. Como consequência dessa ação, no instante subsequente, o agente recebe um sinal de *recompensa* numérica $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$, e o sistema transita para um novo estado S_{t+1} . Em um MDP finito, essa dinâmica ambiental é caracterizada por uma função de probabilidade de transição p , definida pela Equação 6.1:

$$p(s', r | s, a) \doteq \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}. \quad (6.1)$$

A função p estabelece a probabilidade conjunta de ocorrência de uma transição para o estado s' acompanhada da emissão da recompensa r , dado que o agente executou a ação a a partir do estado s no instante t . Esta formulação captura formalmente a estocasticidade inerente ao meio, abrangendo tanto a volatilidade topológica quanto a variabilidade dos retornos obtidos.

Desse modo, a partir da Equação 6.1, a *função de probabilidade de transição de estado* é definida como a soma das probabilidades conjuntas de transição para o estado s' considerando todas as recompensas r possíveis, de acordo com a Equação 6.2:

$$\mathcal{P}(s' | s, a) \doteq \sum_{r \in \mathcal{R}} p(s', r | s, a). \quad (6.2)$$

Dada a natureza estocástica das redes oportunistas, a execução repetida de uma mesma ação a em um estado s pode conduzir a múltiplos desfechos topológicos e operacionais. Para tratar essa incerteza, a *recompensa esperada* para um par estado-ação é calculada ponderando-se cada ganho potencial r por sua respectiva probabilidade de ocorrência, nos moldes da Equação 6.3:

$$\mathcal{R}(s, a) \doteq \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r | s, a). \quad (6.3)$$

No escopo do aprendizado por reforço, o objetivo global do agente consiste em maximizar o valor esperado do retorno cumulativo ao longo de suas interações (Sutton e Barto, 2018). Para garantir que os cálculos funcionem em redes de longa duração ou sem tempo limite definido, emprega-se o *fator de desconto* γ ($0 \leq \gamma \leq 1$). Formalmente, o retorno descontado G_t é definido na Equação 6.4:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \quad (6.4)$$

O parâmetro γ modula o peso das recompensas futuras: um ganho obtido k passos à frente é desvalorizado por um fator de γ^k . No contexto do roteamento oportunista, essa escolha impacta diretamente a diretriz comportamental do nó móvel: com γ próximo de 0, o agente prioriza ganhos míopes orientados estritamente ao salto imediato; em contrapartida, com γ próximo de 1, o agente adquire uma perspectiva de planejamento abrangente, ponderando as consequências estendidas de suas decisões de roteamento ao longo da rede.

6.2.1 Roteamento oportunista sob a quintupla do MDP

Embora a formulação teórica de Sutton e Barto (2018) trate as transições de estado e as recompensas de maneira estocástica conjunta (Equação 6.1), a literatura aplicada à engenharia de redes adota frequentemente uma representação modular por meio da quintupla formal $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$.

Na adaptação dessa estrutura para redes oportunistas, os estados operacionais da rede são mapeados diretamente nos componentes do MDP, onde o agente decisor corresponde tipicamente ao nó portador da mensagem – seja um veículo, um dispositivo IoT ou um roteador. Os elementos da quintupla estruturam-se da seguinte forma:

- Estados ($\mathcal{S}$): Representam o escopo de informações locais acessíveis ao nó no exato instante da tomada de decisão. Em cenários com restrições energéticas severas – como redes de sensores com coleta de energia – Oualhaj, Kobbane e Ben-Othman (2018) definem o estado $s_t = (b_t, q_t, h_t)$, integrando o nível de carga da bateria, a presença de pacotes na fila e a taxa de captação energética no instante t . Em paralelo, Saad *et al.* (2023) parametrizam o estado com base na qualidade dos enlaces em redes veiculares, ao passo que Sammour (2025) adota discretizações focadas na densidade de nós e nos regimes de mobilidade.

- Ações ($\mathcal{A}$): Correspondem às decisões de roteamento disponíveis. As formulações mais básicas reduzem-se a escolhas binárias: encaminhar ou reter a mensagem (Oualhaj, Kobbane e Ben-Othman, 2018). Modelagem avançadas, contudo, ampliam esse espaço de decisão. Liu, Wang e Wu (2019) definem a

ação como a determinação da quota exata de cópias a serem disseminadas em cada intervalo temporal (*time slot*), enquanto Sammou (2025) modela a ação como a seleção dinâmica, em tempo de execução, do protocolo clássico adequado ao estado corrente (Epidêmico, PProPHET, *Spray and Wait* ou MaxProp).

- Recompensas ($\mathcal{R}$): Orientam o comportamento do agente para a maximização do desempenho e o respeito às limitações operacionais da rede. Para ponderar múltiplos objetivos, Liu, Wang e Wu (2019) formulam funções de custo que penalizam atrasos de propagação e a redundância excessiva de cópias. Da mesma forma, Sammou (2025) emprega uma função de recompensa composta que bonifica métricas de maximização (taxa de entrega e energia residual) e penaliza métricas de minimização (latência, número de saltos e sobrecarga).

O objetivo dessa modelagem consiste em derivar uma política de mapeamento ótimo entre estados e ações, capaz de maximizar o retorno esperado (G_t). Todavia, a resolução exata de um MDP por meio de métodos clássicos de programação dinâmica exige o conhecimento prévio e irrestrito da função de transição $\mathcal{P}(s' | s, a)$.

Em redes oportunistas reais, a matriz de transição mostra-se inacessível e estocasticamente instável devido à mobilidade irregular dos nós, à intermitência dos contatos e à imprevisibilidade topológica (Saad et al., 2023). Para contornar essas limitações, recorre-se a algoritmos de aprendizado por reforço livres de modelo, com destaque para o *Q-Learning*, detalhado a seguir.

6.3 O algoritmo *Q-Learning*

Como mencionado anteriormente, a carência de informações a priori acerca das probabilidades de transição $\mathcal{P}$ em redes oportunistas impede a aplicação direta de técnicas determinísticas de programação dinâmica. A alternativa metodológica consiste no uso de abordagens *livres de modelo*, com destaque para o *Q-Learning* (Watkins e Dayan, 1992).

O *Q-Learning* é um algoritmo de aprendizado por reforço do tipo *off-policy*, o que permite ao agente convergir para a política ótima independentemente da política de exploração efetivamente executada. O núcleo deste algoritmo é a função de valor de ação, denominada função Q (*Quality*), que quantifica a utilidade esperada de executar uma ação a a partir de um dado estado s .

No contexto de redes oportunistas, cada dispositivo atua de forma descentralizada como um agente autônomo. Para tanto, o nó mantém uma estrutura tabular (*Q-table*), encarregada de registrar as estimativas $Q(s, a)$ para cada par estado-ação. A cada ciclo de interação, o agente adota uma decisão de roteamento, coleta a recompensa imediata r_{t+1} , observa o estado subsequente s' e atualiza a entrada correspondente em sua tabela com base na *Equação de Bellman* adaptada (Equação 6.5):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]. \quad (6.5)$$

Onde:

- $Q(s_t, a_t)$: Estimativa corrente do valor associado à execução da ação a_t no estado s_t ;
- $\alpha \in (0,1]$: Taxa de aprendizado, que define o peso atribuído às novas observações em relação ao histórico acumulado;
- r_{t+1} : Recompensa imediata coletada imediatamente após a efetivação da ação;
- $\gamma \in [0,1]$: Fator de desconto, responsável por ponderar a relevância de retornos futuros;
- $\max_{a'} Q(s_{t+1}, a')$: Maior valor de ação estimado para o estado seguinte s' .

Na dinâmica da Equação 6.5, taxas de aprendizado elevadas conferem ao nó capacidade de rápida adaptação a mudanças topológicas, enquanto valores baixos preservam a estabilidade do aprendizado frente a oscilações transitórias. De forma análoga, valores de γ próximos da unidade estimulam o planejamento

de rotas de longo alcance, enquanto parâmetros próximos de zero concentram a otimização em ganhos imediatos.

Para conciliar a descoberta de novas rotas eficientes com o aproveitamento do conhecimento já adquirido, o processo decisório é regido por políticas de exploração e exploração, com destaque para a estratégia ϵ -greedy. Sob esse mecanismo, o agente seleciona uma ação estocástica aleatória com probabilidade ϵ – fomentando a exploração de rotas alternativas – e opta pela ação de maior valor Q armazenado com probabilidade $1 - \epsilon$, promovendo a exploração do conhecimento prévio (Do *et al.*, 2025). Em ambientes dinâmicos, o parâmetro ϵ decresce progressivamente ao longo do tempo mediante um fator de decaimento, assegurando ampla exploração nos estágios iniciais de operação e transicionando para regimes de exploração estável à medida que o agente acumula mais conhecimento sobre o ambiente (Sammou, 2025; Ma *et al.*, 2025).

6.4 Aplicações de Q-Learning no roteamento oportunista

A literatura especializada apresenta diversas adaptações do *Q-Learning* e de suas extensões para resolver problemas específicos em DTNs, variando a concepção de estados, ações e funções de recompensa:

Otimização de protocolos clássicos: Conforme abordado na Seção 6.2, Sammou (2025) propõe um agente capaz de selecionar dinamicamente um protocolo (Epidêmico, P_{Ro}PHET etc.), calculando recompensas mediante combinações ponderadas de métricas normalizadas. Na mesma linha, Do *et al.* (2025) incorporam o *Q-Learning* ao roteamento Epidêmico, parametrizando o sinal de recompensa em função do sucesso de entrega para reduzir transmissões redundantes.

Métricas físicas e mobilidade: Em redes subaquáticas restritas, Hu e Fei (2010) formulam uma recompensa que equilibra a distância ao destino, a densidade local de vizinhos e a energia residual dos nós. Em redes veiculares, Wu, Yoshinaga e Ji (2017) estimam probabilidades de encontros multi-saltos

descontando a recompensa em função do número de saltos, integrando lógica difusa para ponderar simultaneamente mobilidade, centralidade e *buffer*.

Comportamento social e econômico: Para modelar cenários habitados por nós egoístas, Dong *et al.* (2020) propuseram o QMCR, integrando o valor de crédito do nó à recompensa do *Q-Learning*, enquanto Hajiaghajani e Biswas (2017) focam na disseminação de conteúdo comercial, onde a recompensa reflete o ganho econômico abatido dos custos operacionais de repasse.

Mitigação de viés e *aprendizado por reforço profundo* (DRL): Para contornar os efeitos de superestimação introduzidos pelo operador de máximo do *Q-Learning* tradicional, Yuan *et al.* (2019) propuseram o DQLR, utilizando *Q-Learning duplo* e recompensas dinâmicas baseadas na contagem de saltos. Em topologias complexas como VANETs, Ma *et al.* (2025) utilizam redes neurais profundas baseadas em Q (*Deep Q-Networks* – DQN) em unidades de beira de estrada para selecionar segmentos viários com base na saturação de tráfego. O emprego combinado de DQN e *Q-Learning* também é apontado como a principal tendência para a automação, em tempo real, de tomadas de decisão em redes de drones para emergências (López-Villegas *et al.*, 2026).

6.5 Considerações

A formulação do roteamento via Processos de Decisão de Markov (MDP) resolvida por meio de algoritmos de *Q-Learning* provê uma base teórica consistente para a tomada de decisão autônoma em redes oportunistas. Ao capacitar os nós a aprenderem políticas ótimas de repasse com base em retornos cumulativos e interações empíricas contínuas, essas abordagens eliminam a dependência de rotas estáticas pré-estabelecidas.

Contudo, a ausência de visibilidade topológica global e a alta estocasticidade dos enlaces geram, com frequência, atrasos na convergência das tabelas de valor. Nas fases iniciais de operação (*cold-start*), a carência de informações sobre rotas inexploradas induz o agente a decisões temporariamente subótimas, elevando a latência de entrega e o consumo de largura de banda.

Para contornar tais restrições operando exclusivamente com dados locais, faz-se necessário vincular o aprendizado a parâmetros que reflitam a estrutura subjacente da rede. Nesse sentido, o Capítulo 7 apresenta o protocolo QL-CASTRO, concebido para reduzir essa deficiência ao fundir os indicadores sociais de conectividade do CASTRO (força, tendência e regularidade) com a mecânica adaptativa do *Q-Learning*. Essa estratégia híbrida utiliza o histórico consolidado de interações para inicializar e orientar as estimativas de valor, acelerando significativamente a convergência e assegurando um mecanismo de roteamento descentralizado, eficiente e robusto.

7. PROTOCOLO QL-CASTRO

7.1 Introdução

No Capítulo 5, apresentou-se o Protocolo CASTRO, estruturado a partir da interpretação sistemática do histórico de interações para obtenção de métricas sociais, como força, tendência e regularidade entre conexões. No entanto, as estratégias de repasse do Protocolo CASTRO não priorizam explicitamente a minimização da latência de entrega, e a alocação fracionária de cópias pode resultar em um aproveitamento subótimo da capacidade de armazenamento dos *buffers*.

Com o propósito de mitigar tais limitações, este capítulo introduz o Protocolo QL-CASTRO (*Q-Learning Applied to CASTRO Protocol*). O modelo preserva a avaliação social do CASTRO original, mas reformula a tomada de decisão ao incorporar uma camada de aprendizado por reforço (*Q-Learning*), voltada à estimativa dinâmica do atraso de entrega (Q_{delay}). Adicionalmente, o QL-CASTRO adota um mecanismo de expiração e aposentadoria de mensagens (*message retirement*), mantendo a estratégia de remoção de mensagens já entregues na rede.

A Seção 7.2 detalha os métodos de suporte dedicados ao aprendizado de máquina e à gestão do tempo de vida (TTL). Em seguida, a Seção 7.3 descreve o funcionamento central do roteamento QL-CASTRO.

7.2 Métodos de suporte

Para integrar o *Q-Learning* ao ambiente oportunista e estruturar a retenção controlada de pacotes, o QL-CASTRO emprega um conjunto de variáveis temporais, sociais e analíticas. A formalização da nomenclatura base adota os seguintes parâmetros:

- t : Instante de tempo atual na simulação;
- i, j : Nó emissor/agente (i) e o seu vizinho receptor/candidato (j);

- d : Nó de destino de uma mensagem;
- v : Nós conhecidos do vizinho j ;
- Q : Estimativa de atraso entre dois nós armazenada na Tabela-Q;
- t_{last} : Instante de tempo do último encontro registrado entre dois nós. Valores negativos indicam que os nós nunca se encontraram;
- Δt : Defasagem temporal, calculada como a diferença entre o tempo atual e o tempo do último encontro ($t - t_{last}$);
- Q_{est} : Estimativa dinâmica de atraso entre dois nós, ajustada pelo fator de envelhecimento temporal Δt ;
- Q_{max} : Penalidade máxima de atraso atribuída a destinos desconhecidos ou com ausência de rota;
- α : Taxa de aprendizado (*learning rate*) do algoritmo *Q-Learning*;
- m : Mensagem em trânsito pela rede;
- t_{origin} : Tempo original de criação de uma mensagem;
- Age : Idade da mensagem na rede, calculada como $t - t_{origin}$.
- Age_{max} : Idade limite da mensagem a partir da qual cessam as renovações de seu Tempo de Vida (TTL);
- φ : Consistência de um nó, herdada do Protocolo CASTRO (Cap. 4).

Com base nessa nomenclatura, o Algoritmo 7.1 detalha o método *getEstimatedDelay*, o qual atualiza de forma dinâmica a estimativa de atraso de um nó até um destino. Como a conectividade em DTNs é esporádica, os parâmetros de aprendizado não podem permanecer estáticos. Conforme o Algoritmo 7.1, caso o nó i jamais tenha entrado em contato com o destino d , o método atribui de imediato a penalidade Q_{max} (linhas 01 – 02). Caso contrário, a estimativa atualizada Q_{est} resulta da soma entre a defasagem temporal

$\Delta t(i, d)$ e o valor nominal $Q(i, d)$ (linhas 03 – 04). Esse mecanismo de envelhecimento reduz a competitividade de rotas desatualizadas, mitigando o risco de repasses por caminhos topologicamente obsoletos.

Método $getEstimatedDelay(d)$

```

01: se  $t_{last}(i, d) < 0$  então
02:    $Q_{est}(i, d) = Q_{max}$ 
03: senão
04:    $\Delta t(i, d) = t - t_{last}(i, d)$ 
05:    $Q_{est}(i, d) = Q(i, d) + \Delta t(i, d)$ 
06: fim se

```

Algoritmo 7.1 – Método $getEstimatedDelay$

O aprendizado por reforço é acionado para a propagação de atrasos transitivos no momento do contato entre os nós i e j , mediante uma adaptação da Equação de Bellman voltada à minimização de custos temporais. Conforme sintetizado por Mammeri (2019) em sua revisão sobre roteamento adaptativo, essa formulação transiciona do escopo clássico de maximização de recompensas (*Q-Learning*) para o paradigma de roteamento de menor custo (*Q-Routing*). As adaptações algébricas sobre a Equação de Bellman compreendem:

1. A substituição do operador de maximização (max) pela busca do limite inferior correspondente ao menor atraso estimado;
2. A fixação do fator de desconto em $\gamma = 1$, visto que a latência constitui métrica temporal aditiva e cumulativa ao longo dos saltos;
3. A substituição da recompensa imediata pelo custo real de transição, mensurado pelo intervalo transcorrido desde o último encontro (Δt).

A partir dessas premissas, a Equação de Bellman para a diferença temporal de custo assume a forma

$$Q(i, v) = Q(i, v) + \alpha[Q_{est}(j, v) - Q(i, v)], \quad (7.1)$$

onde $Q_{est}(j, v)$ engloba tanto a estimativa futura do vizinho quanto o custo temporal decorrido. Fatorando os termos, obtém-se a Equação 7.2:

$$Q(i, v) = (1 - \alpha) \cdot Q(i, v) + \alpha \cdot Q_{est}(j, v). \quad (7.2)$$

Essa formulação viabiliza a propagação transitiva de informações na rede, permitindo que o nó absorva gradualmente o conhecimento de sua vizinhança para descobrir e atualizar os caminhos indiretos de menor atraso. Esse procedimento é operacionalizado pelo método *updateTransitiveDelays* (Algoritmo 7.2).

Método updateTransitiveDelays (j)

```

01: Para todo  $v \in \{Nós\ conhecidos\ por\ j\}$  faça
02:   se  $v \neq i$  então
03:      $Q_{est}(j, v) = j.getEstimatedDelay(v)$  (Algoritmo 7.1)
04:     se  $Q_{est}(j, v) < Q(i, v)$  então
05:        $Q(i, v) = (1 - \alpha) \cdot Q(i, v) + \alpha \cdot Q_{est}(j, v)$  (Eq. 7.2)
06:     fim se
07:   fim se
08: fim para

```

Algoritmo 7.2 – Método *updateTransitiveDelays*

Para cada nó v conhecido por j (linha 01), o nó i avalia a estimativa atualizada $Q_{est}(j, v)$ (linha 03) e, caso identifique uma rota mais rápida do que a sua própria, atualiza $Q(i, v)$ ponderando a nova informação pela taxa de aprendizado α (linhas 04 – 05).

O Algoritmo 7.3 apresenta o método *renewTTL*, encarregado de estender o tempo de vida útil das mensagens quando a vizinhança dispõe de rotas vantajosas. A prorrogação é restrita a mensagens cuja idade esteja abaixo do limiar de aposentadoria (linha 04) e que satisfaçam ao menos um critério: (i) estimativa de atraso do vizinho $Q_{est}(j, d)$ inferior à do portador $Q_{est}(i, d)$; ou (ii) consistência do vizinho φ_j superior à do portador φ_i . Em ambos os casos, o TTL

é incrementado pelo bônus b_m (linhas 07 – 08), assegurando que apenas as mensagens com perspectiva de encaminhamento conservem recursos da rede.

Método $renewTTL(j)$

```

01: Para toda  $m \in buffer(i)$  faça
02:    $d = \text{nó de destino da mensagem } m$ 
03:    $Age(m) = t - t_{origin}(m)$ 
04:   se  $Age(m) < Age_{max}$  então
05:      $Q_{est}(i, d) = getEstimatedDelay(d)$  (Algoritmo 7.1)
06:      $Q_{est}(j, d) = j.getEstimatedDelay(d)$  (Algoritmo 7.1)
07:     se  $Q_{est}(j, d) < Q_{est}(i, d)$  ou  $\varphi_j > \varphi_i$  então
08:        $TTL_{novo(m)} = \min(TTL_{(m)} + b_m, TTL_{max})$ 
09:     fim se
10:   fim se
11: fim para

```

Algoritmo 7.3 – Método $renewTTL$ (Protocolo QL-CASTRO)

Adicionalmente, para evitar a retenção indefinida de mensagens e a saturação de *buffers*, a bonificação do tempo extra é protegida por um teto matemático. Conforme expresso na lógica de renovação, a função $\min()$ assegura que o tempo de vida total restante da mensagem jamais exceda o seu tempo de vida máximo original (TTL_{max}).

Nesta implementação, os parâmetros Age_{max} e b_m foram definidos em 3h e 30 min, respectivamente, admitindo ajustes conforme o cenário de aplicação.

7.3 Métodos principais

O Algoritmo 7.4 apresenta a estrutura principal do Protocolo QL-CASTRO, combinando chaves, consistências e estimativas temporais obtidas por aprendizado por reforço no direcionamento do tráfego. O protocolo realiza o gerenciamento de *buffer* do protocolo base, adicionando os métodos da seção anterior.

Método Main

```
01: se nó  $i$  encontrar nó  $j$  então
02:    $updParameters(i, j)$  (Algoritmo 5.1)
03:    $updDirScore(i, j)$  (Algoritmo 5.2)
04:    $updTrScores(i, j)$  (Algoritmo 5.3)
05:    $bufferManagement(i)$  (Algoritmo 5.6)
06:    $Q(i, j) = (1 - \alpha) \cdot Q(i, j)$ 
07:    $t_{last}(i, j) = t$ 
08:    $updateTransitiveDelays(j)$  (Algoritmo 7.2)
09:    $renewTTL(j)$  (Algoritmo 7.3)
10:   para toda  $m \in buffer(i)$  faça
11:      $d =$  nó de destino da mensagem  $m$ 
12:     se  $d \in \{\text{Nós conectados a nó } i\}$  então
13:        $i$  encaminha  $m$  ao nó  $d$ 
14:        $i$  adiciona  $m$  a  $D(i)$  e atualiza  $Q(i, d) = 0$ 
15:        $i$  remove  $m$  do seu  $buffer$ 
16:     fim se
17:   fim para
18:   Defina  $msg =$  vetor de tuplas (mensagem, nó receptor)
19:   para todo  $v \in \{\text{Nós conectados a nó } i\}$  faça
20:     para toda  $m \in buffer(i)$  faça
21:        $d =$  nó de destino da mensagem  $m$ 
22:        $i$  e  $v$  calculam  $K, \varphi$  e  $Q_{est}$ 
23:       se  $L_i(m) > 1$  então
24:          $i$  adiciona a tupla  $(m, v)$  ao vetor  $msg$ 
25:       senão se  $L(m) = 1$  então
26:         se  $(Q_{est}(v, d) < Q_{est}(i, d) \text{ ou } K_{(v,d)} > K_{(i,d)} \text{ ou } \varphi_v > \varphi_i)$ 
27:            $i$  adiciona a tupla  $(m, v)$  ao vetor  $msg$ 
28:         fim se
29:       fim se
30:     fim para
31:   fim para
32:    $i$  organiza  $msg$  priorizando mensagens mais novas
33:    $sendMessages(msg)$  (Algoritmo 7.5)
34: senão
35:    $updDirScore(i, j)$  (Algoritmo 5.2)
36: fim se
```

Algoritmo 7.4 – Método *Main* (Protocolo QL-CASTRO)

Entre as linhas 02 e 09, o método *Main* processa as métricas sociais do protocolo base, atualiza os atrasos e aplica as rotinas de renovação e aposentadoria de TTL. A verificação de entrega direta (linhas 10 – 17) valida se o nó adjacente corresponde ao destinatário final de mensagens residentes no *buffer*, executando a entrega imediata, a limpeza do registro local e a atualização da lista de mensagens entregues.

Para reduzir desvios nas estimativas de latência, o modelo estabelece um estado terminal formal: ao efetuar a entrega direta de uma mensagem ao destino d , o nó i reconhece a conclusão da rota e ancora a estimativa de atraso para zero ($Q(i, d) = 0$, linha 14). Esse mecanismo propaga rapidamente a configuração de entrega pela rede, acelerando a convergência para caminhos de menor atraso.

Não sendo o vizinho o destino final, as mensagens elegíveis ao repasse a intermediários são agrupadas em um vetor de tuplas (linhas 18 – 31). Mensagens com múltiplas cópias ($L_i(m) > 1$) são selecionadas de imediato, enquanto aquelas restritas a uma única cópia ($L_i(m) = 1$) passam por filtragem seletiva, sendo priorizados apenas se o receptor apresentar menor atraso estimado (Q_{est}) ou maior relevância social (K ou φ). A ordenação final obedece à política *Newest-First* (linha 32), favorecendo a propagação de dados recentes antes da execução da rotina de distribuição.

A efetivação do repasse é delegada ao método *sendMessages* (Algoritmo 7.5), que atua como regulador de fluxo ao estipular o número de cópias para transferência ao nó adjacente.

Para cada mensagem candidata, o protocolo verifica o estoque local de cópias (L_{old}). Em fases iniciais de difusão ($L_{old} > 1$), aplica-se a partilha binária, repassando metade das réplicas ao receptor (linhas 06 – 07) para acelerar a disseminação inicial. Nas etapas sob regime de cota unitária (linhas 08 – 14), a última cópia remanescente é cedida exclusivamente se o receptor v demonstrar vantagem temporal (Q_{est} menor) ou superioridade social (K ou φ maiores), além de ter uma taxa de ocupação de *buffer* inferior a um limite ($BO_{th} = 90\%$). Essa estratégia híbrida protege a rede contra redundâncias excessivas,

condicionando o repasse da última réplica a expectativas consistentes de otimização da rota.

Método sendMessages (msg)

Nó transmissor i :

```

01: Para cada tupla  $(m, v) \in msg, m \notin buffer(v)$  faça
02:    $L_i(m) =$  número de cópias de  $m$  em  $buffer(i)$ 
03:    $L_{old} = L_i(m)$ 
04:    $L_{new} = 0$ 
05:    $d =$  nó destino de  $m$ 
06:   se  $L_{old} > 1$  então
07:      $L_{new} = \lfloor L_{old} / 2 \rfloor$ 
08:   senão se  $L_{i_{old}}(m) = 1$  e  $BO_{(v)} < BO_{th}$  então
09:      $Q_{est}(i, d) = getEstimatedDelay(d)$  (Algoritmo 7.1)
10:      $Q_{est}(v, d) = v.getEstimatedDelay(d)$  (Algoritmo 7.1)
11:     se  $Q_{est}(v, d) < Q_{est}(i, d)$  ou  $K_{(v,d)} > K_{(i,d)}$  ou  $\varphi_v > \varphi_i$ 
12:        $L_{new} = 1$ 
13:     fim se
14:   fim se
15:    $L_i(m) = L_{old} - L_{new}$  // Novo número de cópias de  $m$  em  $buffer(i)$ 
16:    $L_v(m) = L_{new}$  // Novo número de cópias de  $m$  em  $buffer(v)$ 
17: fim para

```

Algoritmo 7.5 – Método sendMessages (Protocolo QL-CASTRO)

7.4 Análise de Complexidade Computacional do QL-CASTRO

A integração da camada de aprendizado demanda validação de desempenho para assegurar a viabilidade computacional em nós móveis. Denotando por $|P|$ o quantitativo de destinos catalogados na tabela de valores, a rotina de sincronização e atualização requer uma varredura linear de ordem $O(|P|)$ para o recálculo das diferenças temporais de Bellman, executada em paralelo com as checagens de entrega direta e renovação de TTL de ordem $O(|B|)$.

A montagem do vetor de tuplas preserva a complexidade estrutural aninhada do protocolo base, confrontando o conjunto de $|B|$ mensagens com os $|V|$

vizinhos ativos, com custo de $O(|V| \cdot |B|)$. A ordenação subsequente demanda $O(|F_{total}| \log |F_{total}|)$ operações. Contudo, a supressão de operações de ponto flutuante do fator social (f) nas etapas de alocação reduz a carga computacional por iteração no Algoritmo 7.5.

Consequentemente, a complexidade temporal global do QL-CASTRO consolida-se em $O(|P| + |V| \cdot |B| + |F_{total}| \log |F_{total}|)$. Ao dispensar estruturas de laços aninhados na atualização de valores da Tabela-Q e empregar operações escalares de tempo constante ($O(1)$) na rotina de Bellman, a arquitetura apresenta características adequadas para a redução da latência, preservando a escalabilidade requerida por ambientes IoT com restrições energéticas e de processamento.

Em suma, o Protocolo QL-CASTRO constitui uma evolução direcionada à redução de atrasos em relação ao Protocolo CASTRO base, preservando os patamares de taxa de entrega. O modelo simplifica a fase de difusão e concentra a tomada de decisão na fase de espera, prevenindo saturações prematuras. A integração do *Q-Learning* adaptado à otimização de latência, associada às políticas de aposentadoria e de remoção de mensagens previamente entregues, confere ao protocolo robustez operacional na aceleração do fluxo informacional.

Tendo estabelecido a fundamentação teórica e a modelagem matemática dos protocolos apresentados nos capítulos 5 e 7, o próximo passo consiste na validação empírica. O Capítulo 8 detalhará a metodologia de avaliação, descrevendo os cenários de mobilidade, as configurações de parâmetros implementadas no simulador ONE e os critérios de qualidade adotados. Através dessas configurações de teste, os resultados das simulações serão apresentados e discutidos, permitindo uma comparação do Protocolo CASTRO e de sua versão aprimorada, QL-CASTRO, frente a alguns protocolos clássicos da literatura e a outros mais recentes, do estado da arte.

8. AVALIAÇÃO EXPERIMENTAL E RESULTADOS

8.1 Cenários e configurações de teste

Para validar empiricamente os protocolos propostos nesta pesquisa, CASTRO e QL-CASTRO, em comparação a modelos clássicos e do estado da arte (PRoPHET, *Spray and Wait*, QoN-ASW e HE SnW), utilizou-se o *Opportunistic Network Environment* (ONE Simulator).

Para garantir a robustez da avaliação, todos os protocolos avaliados foram implementados utilizando o mesmo mecanismo de gerenciamento de *buffer*. Esse mecanismo funciona como um sistema de *acknowledgement* (ACK) epidêmico, no qual os nós notificam uns aos outros sobre as mensagens que já alcançaram seus destinos finais, instruindo-os a remover sumariamente as cópias dessas mensagens de seus *buffers*.

Os testes não se limitaram a um único ambiente. Foram modelados dois cenários com diferentes perfis de mobilidade, densidade de nós e hardware de comunicação. A escolha das áreas geográficas foi essencial ao planejamento metodológico. O mapa do centro urbano de Helsinki foi adotado por representar o cenário clássico da literatura, com ruas curtas, intersecções próximas e alta densidade populacional. Em contrapartida, foi extraída e modelada a topologia viária de Manaus, resultando em um mapa criado para testar a resiliência dos protocolos em condições de conectividade esparsa e altas velocidades de deslocamento.

8.1.1 Cenário 1: Rede densa de pedestres (Helsinki)

O primeiro cenário simula uma rede *wearable* e de Internet das Coisas (IoT) urbana. Utilizou-se o mapa do centro da cidade de Helsinki, nativo do ONE, com dimensões de 4.500 x 3.400 metros (Figura 8.1). Com a alocação padrão de 500 nós movendo-se a velocidades de pedestres (0,5 a 1,5 m/s), obteve-se alta densidade, com cerca de 32 nós/km². O cenário favorece encontros frequentes, mas com forte propensão a congestionamentos e elevado uso de *buffer*.



Figura 8.1 – Mapa do centro da cidade de Helsinque (4.500m x 3.400m), no simulador ONE

A interface de rede modelada foi o *Bluetooth Low Energy* (BLE), com curto alcance (10 metros) e taxa de transferência de 250 kbps (equivalente ao *PHY rate* do padrão BLE 5.0). O modelo de energia foi parametrizado com base em hardwares IoT típicos (como a família Nordic nRF52), operando a 3,3 V com consumos restritos à ordem de miliwatts.

8.1.2 Cenário 2: Rede esparsa veicular (Manaus)

O segundo cenário reflete uma rede veicular *ad-hoc* (*Vehicular Ad-hoc Network* – VANET) em ambiente metropolitano. Utilizando a ferramenta QGIS (Apêndice), extraiu-se a topologia viária da cidade de Manaus, compreendendo uma área de 35.000 x 35.000 metros (Figura 8.2). Com os mesmos 500 nós, a densidade é reduzida para 0,4 nós/km², criando um ambiente esparso. Os nós movem-se a velocidades veiculares (30 a 60 km/h), resultando em encontros curtos e menos frequentes.



Figura 8.2 – Mapa da cidade de Manaus (35.000m x 35.000m), no simulador ONE

Para compensar a janela de contato mínima, a interface foi modelada como *Wi-Fi 6* (IEEE 802.11ax), com alcance de 100 metros e taxa de transferência de 25 MBps. O modelo energético baseou-se no módulo Panasonic PAN9019, refletindo o consumo de computadores de bordo com potências de transmissão (TX) na casa de 640 mW e recepção (RX) em 231 mW.

8.1.3 Configurações

A Tabela 8.1 sintetiza os parâmetros gerais fixos estabelecidos para os testes em ambos os cenários de mobilidade (Helsinki e Manaus), assegurando que fatores como velocidade, alcance de rádio e modelagem energética permaneçam constantes e correspondam aos ambientes propostos. Os três últimos parâmetros – taxa de aprendizado, bônus do TTL e idade de aposentadoria – são específicos do QL-CASTRO.

Tabela 8.1 – Configurações dos cenários de teste

Parâmetro	Cenário 1 (Helsinki)	Cenário 2 (Manaus)
Dimensões do mapa	4.500 x 3.400 m	35.000 x 35.000 m
Modelo de mobilidade	<i>Shortest Path Map-Based</i>	<i>Shortest Path Map-Based</i>
Duração da simulação	12 horas (43.200 s)	15 horas (54.000 s)
Velocidade dos nós	0,5 a 1,5 m/s	8,33 a 16,66 m/s
Interface e alcance	Bluetooth LE (10 m)	Wi-Fi 6 (100 m)
Taxa de transmissão	250 kbps	25 MBps
Tamanho da mensagem	300 a 500 kB	1 a 5 MB
Potência TX; RX	15,84 mW; 10,89 mW	640,5 mW; 231,4 mW
Taxa de Aprendizado (α)	0,3	0,3
Bônus TTL (b_m)	30 minutos	30 minutos
Idade de aposentadoria	3 horas	3 horas

Os parâmetros do aprendizado por reforço foram calibrados empiricamente, com a taxa de aprendizado fixada em $\alpha = 0,3$, a fim de promover um equilíbrio ideal entre a absorção rápida de novas topologias e a resiliência matemática do histórico de encontros. Além destes parâmetros, a metodologia de testes adotada exige a definição de valores *default* (padrões) para as variáveis que serão o foco das avaliações: número de nós, tamanho do *buffer*, tempo de vida das mensagens (TTL) e intervalo de geração de mensagens. A Tabela 8.2 descreve esses valores.

Tabela 8.2 – Padrões adotados para variáveis de análise

Variável de análise	Padrão adotado
Número de nós	500 dispositivos
Tempo de vida (TTL)	60 minutos
Intervalo de geração	15 segundos
Número inicial de cópias	6 cópias
Tamanho de <i>buffer</i> (Helsinki)	5 MB
Tamanho de <i>buffer</i> (Manaus)	50 MB

Deve-se ressaltar que as avaliações de desempenho seguem um modelo de isolamento de variáveis. Durante as baterias de testes, enquanto uma variável de análise é iterada gradativamente, todas as demais são mantidas fixas em seus respectivos valores *default*, garantindo que as flutuações de desempenho registradas sejam causadas unicamente pelo parâmetro em observação.

8.2 Critérios de Avaliação

Para mensurar o desempenho do roteamento, utilizou-se um conjunto de quatro métricas: Taxa de entrega de mensagens, atraso médio, sobrecarga em rede (*overhead*) e consumo energético. Todos os nós participaram do processo de geração e repasse de mensagens no cenário.

- Taxa de entrega (em %):

$$Taxa\ de\ Entrega = \frac{N_{ent}}{N_{cr}} \cdot 100. \quad (8.1)$$

Indica a eficácia do algoritmo, dada pela razão entre o número de mensagens entregues com sucesso (N_{ent}) e o total de mensagens criadas (N_{cr}).

- Atraso Médio ou Latência (em minutos):

$$Atraso\ Médio = \frac{\sum_{x=1}^{N_{ent}} At_x}{N_{ent}}. \quad (8.2)$$

Representa o tempo médio necessário para o transporte de uma mensagem da origem ao destino, calculado pela soma dos tempos de entrega (At_x) dividida pelo número de mensagens entregues com sucesso.

- Sobrecarga em rede ou *Overhead*:

$$Overhead = \frac{N_{rep} - N_{ent}}{N_{ent}}. \quad (8.3)$$

O numerador desta expressão indica a diferença entre o número de mensagens encaminhadas (N_{rep}) pela rede e de mensagens entregues ao destino (N_{ent}). O *overhead* indica a necessidade de retransmitir uma mensagem diversas vezes pela rede, até a sua entrega.

- Consumo energético (em Joules):

Somatório da energia consumida (em Joules) pelas interfaces de rádio nas operações de recepção (RX) e transmissão (TX) de mensagens.

8.3 Análise de resultados: Rede densa de pedestres

O cenário configurado com o mapa da zona central da cidade de Helsinki representa um ambiente de alta densidade de nós pedestres, empregando a interface *Bluetooth Low Energy* (BLE). Esta configuração gera encontros frequentes e propensão ao rápido esgotamento da capacidade de *buffer*, sendo adequada para o teste da eficiência dos protocolos. Para as métricas de sobrecarga em rede e consumo energético, são apresentados dois gráficos em cada subseção: o primeiro ilustra todos os protocolos, evidenciando o comportamento fora de escala do P_{Ro}PHET, e o segundo foca na análise comparativa entre os demais protocolos.

Os valores representados nos gráficos correspondem à média de 15 simulações realizadas a partir de sementes aleatórias de geração de mensagens. O quantitativo de 15 execuções foi adotado por margem de segurança metodológica, após constatar-se que a estabilização estatística dos resultados já ocorria a partir de 10 a 12 repetições. Optou-se por omitir as barras de erro nos gráficos para manter a clareza visual das curvas, visto que a dispersão amostral observada foi residual e homogênea em todas as execuções.

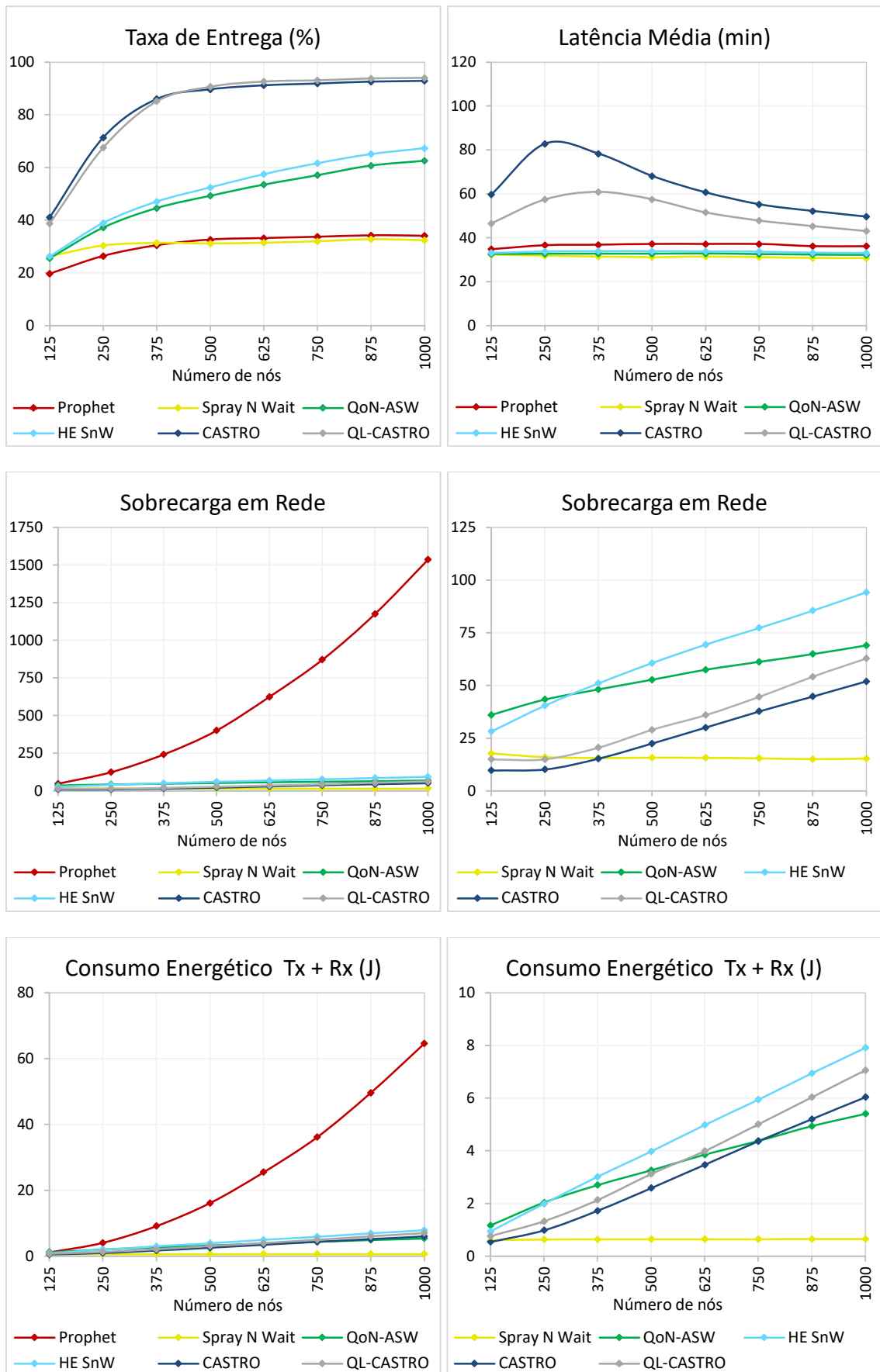
8.3.1 Variação do número de nós

A variação da densidade da rede, testada de 125 a 1000 nós, demonstra o impacto direto da quantidade de dispositivos nas oportunidades de roteamento, conforme ilustrado nos Gráficos 8.1.

Em relação à taxa de entrega, os protocolos CASTRO e QL-CASTRO são notavelmente mais eficazes. A partir de 375 nós, ambos superam a marca de 80% de taxa de entrega e convergem para acima de 90% nas configurações mais densas. Em contraste, o HESnW e o QoN-ASW exibem taxas inferiores, variando de 60% a 70%, enquanto o PRoPHET e o *Spray and Wait* convergem para um valor em torno de 35%. A eficiência dos modelos propostos indica que a utilização das métricas de força, tendência e regularidade mapeia adequadamente a topologia social da rede, garantindo que as mensagens alcancem seus destinos mesmo com o aumento da complexidade estrutural.

Na análise da latência média, observa-se o principal ganho arquitetural do QL-CASTRO em relação ao seu predecessor. O protocolo CASTRO apresenta uma elevação na latência ao redor de 250 nós, atingindo a marca de 80 minutos. Esse comportamento é esperado, visto que o protocolo consegue entregar um volume muito maior de mensagens para nós difíceis de alcançar, o que eleva a média de tempo. O QL-CASTRO, no entanto, introduz a estimativa temporal Q_{est} , reduzindo o pico de latência para menos de 60 minutos e mantendo uma curva estritamente inferior à do CASTRO em todas as densidades. Os demais protocolos apresentam latência inferior, entre 30 a 40 minutos, porque falham em entregar diversas mensagens e contabilizam apenas os tempos das entregas mais simples.

Nos gráficos de sobrecarga e consumo energético, o perfil de inundação do PRoPHET gera um crescimento exponencial e insustentável, ultrapassando o valor de 1.500 para sobrecarga e 60 Joules de consumo. Ao isolar os demais protocolos no segundo gráfico, nota-se que o QL-CASTRO e o CASTRO operam com níveis de sobrecarga (entre 50 e 60) e de energia (cerca de 6 Joules) comparáveis às métricas dos protocolos do estado da arte (QoN-ASW e HESnW), confirmando que os repasses seletivos evitam a saturação da rede.



Gráficos 8.1 – Desempenho para número de nós variável – rede densa de pedestres

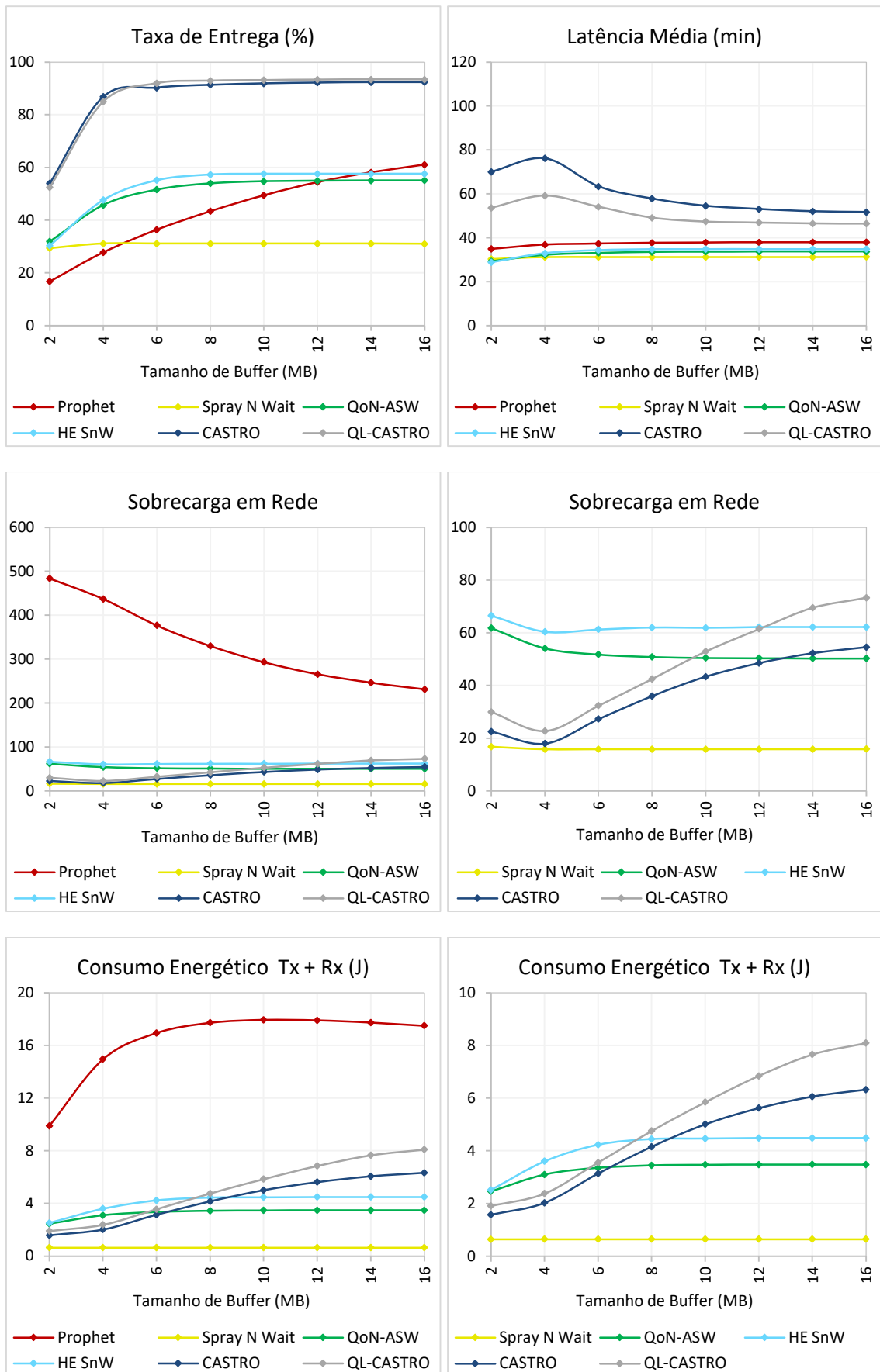
8.3.2 Variação do tamanho de *buffer*

A variação do tamanho de *buffer*, compreendida entre 2 MB e 16 MB, exerce influência direta sobre a eficácia de entrega das mensagens. Os impactos dessa variável no desempenho dos protocolos são detalhados nos Gráficos 8.2.

A taxa de entrega evidencia a resiliência dos modelos propostos nos casos de maiores restrições de memória. Mesmo com apenas 4 MB de *buffer*, os protocolos CASTRO e QL-CASTRO alcançam aproximadamente 85% de entrega. Os protocolos concorrentes exigem *buffers* maiores para apresentar taxas de entrega que se aproximam de 60%. O gerenciamento implementado pelo descarte preemptivo de mensagens entregues mostra-se eficaz para evitar a saturação dos *buffers*.

Em termos de latência, as restrições de *buffer* forçam os protocolos propostos a reduzir o número de encaminhamentos, retendo mensagens por mais tempo até encontrarem o contato ideal. Novamente, o QL-CASTRO demonstra controle superior ao de seu predecessor, mantendo o atraso médio abaixo de 60 minutos e reduzindo gradualmente este valor à medida que a capacidade de armazenamento aumenta.

Os gráficos de sobrecarga e de consumo energético ilustram um comportamento peculiar: nos algoritmos CASTRO e QL-CASTRO, à medida que o *buffer* aumenta de 2 MB para 16 MB, a sobrecarga e o consumo energético apresentam um leve crescimento. Este fenômeno ocorre porque a disponibilidade de memória permite que os nós retenham mensagens válidas por mais tempo, viabilizando repasses que, em outros casos, não seriam realizados. Apesar desse incremento, os valores absolutos de consumo energético permanecem baixos e não constituem um risco ao tempo de vida útil dos nós.



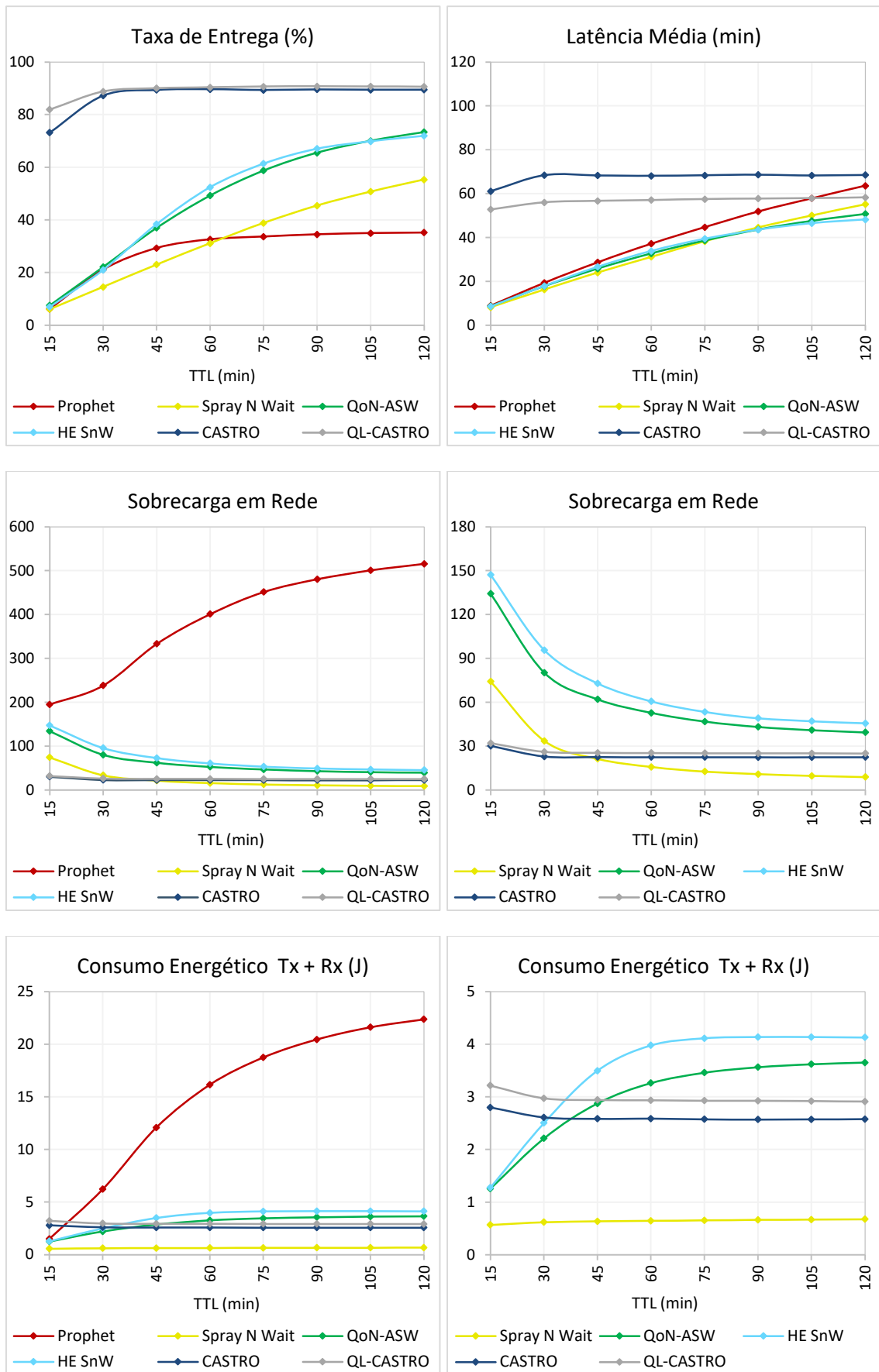
Gráficos 8.2 – Desempenho para tamanho de *buffer* variável – rede densa de pedestres

8.3.3 Variação do tempo de vida das mensagens (TTL)

O parâmetro de tempo de vida das mensagens (TTL) foi avaliado no intervalo de 15 a 120 minutos, com o objetivo de analisar o comportamento da rede diante da expiração precoce e da retenção prolongada de dados e mensagens. Os resultados são ilustrados nos Gráficos 8.3.

Na taxa de entrega, o gráfico revela que os algoritmos da literatura (PRoPHET, *Spray and Wait*, QoN-ASW e HE SnW) têm seus desempenhos afetados pelo tempo de vida das mensagens. Valores de TTL baixos significam que as mensagens são frequentemente descartadas antes que sejam entregues ao destino. Esse problema é contornado pelos protocolos CASTRO e QL-CASTRO, que possuem a política de renovação de TTL das mensagens, permitindo-lhes que alcancem aproximadamente 80% de entrega no menor tempo de vida avaliado. Quando o TTL é igual a 30 minutos, os modelos propostos convergem para valores próximos a 90%, enquanto os demais protocolos necessitam de maiores TTLs para as mensagens chegarem a seus destinos. Os protocolos concorrentes mais recentes e eficazes, QoN-ASW e HE SnW, necessitam de 105 minutos de TTL para entregar 70% das mensagens.

No gráfico dos atrasos médios, observa-se que os protocolos da literatura apresentam latências reduzidas para valores baixos de TTL. Esse comportamento é consequência da baixa taxa de entrega, visto que as únicas mensagens entregues são as de transmissão rápida aos nós de destino, enquanto as demais são expiradas. Ao analisar a curva dos protocolos propostos, que efetivamente entregam o volume massivo da rede, o CASTRO atinge o maior valor de latência, próximo a 70 minutos. O QL-CASTRO, por sua vez, demonstra a eficácia do limiar de aposentadoria e da aplicação do *Q-Learning*. Ao vetar a renovação de TTL para mensagens muito antigas, o algoritmo impede o aumento contínuo da latência, mantendo a sua curva abaixo dos 60 minutos, com vantagem sobre o protocolo base.



Gráficos 8.3 – Desempenho para TTL variável – rede densa de pedestres

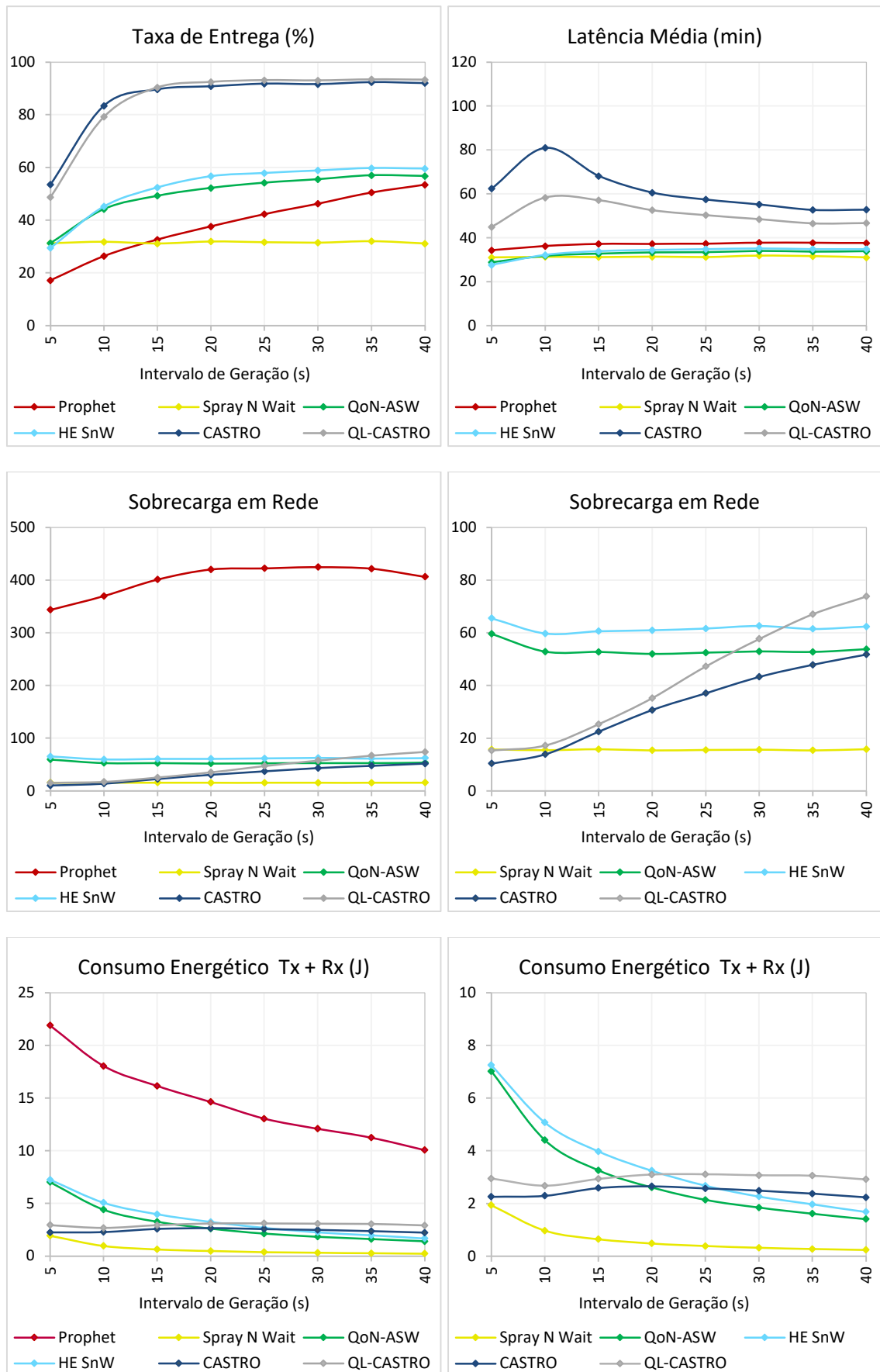
A análise conjunta de sobrecarga e de consumo energético revela comportamentos distintos entre as abordagens. Para os protocolos QoN-ASW e HESnW, nota-se um fenômeno peculiar – enquanto a sobrecarga relativa da rede decai, o consumo energético se eleva. Esse comportamento está associado à melhoria em suas taxas de entrega. Com a expansão do TTL, esses protocolos ganham tempo hábil para concluir o roteamento, elevando substancialmente o volume de mensagens entregues ao destino final (N_{ent}). O aumento desse denominador provoca a redução da sobrecarga, de aproximadamente 130 para 40. No entanto, a efetivação física dessas entregas demanda uma quantidade maior de operações absolutas de transmissão e recepção (TX/RX), o que justifica a elevação gradual do consumo de energia.

Diferindo dos protocolos anteriores, o QL-CASTRO e o CASTRO exibem um perfil mais conservador, onde a sobrecarga e o consumo energético entram em convergência precocemente. Essa observação indica que os modelos propostos reduzem as replicações e a geração de tráfego redundante na rede.

8.3.4 Variação do intervalo de geração de mensagens

A variação do referido intervalo, de 5 a 40 segundos, representa um teste de congestionamento na rede. Avalia-se a capacidade dos protocolos de lidar com um grande tráfego de informações, conforme os Gráficos 8.4.

No gráfico de taxa de entrega, o cenário de alta demanda de recursos da rede (5 segundos) expõe as limitações da maioria dos protocolos da literatura. O excesso de novas mensagens causa a remoção das mais antigas, seja por descarte preemptivo ou por expiração forçada. Nesse cenário desafiador, as taxas de protocolos como PProPHET, HESnW e QoN-ASW sofrem degradação, operando em faixas que variam de 20% a 30% de entrega. Por outro lado, protocolos como o CASTRO e o QL-CASTRO demonstram maior resiliência, obtendo taxas de entrega na faixa de 50% a 55% no cenário de pior caso. A partir de intervalos mais largos (15 segundos), ambos os protocolos se tornam capazes de entregar 90% das mensagens.



Gráficos 8.4 – Desempenho para intervalo de geração variável – rede densa de pedestres

Esse comportamento indica que as decisões baseadas em métricas sociais (chaves e consistências) e aprendizado por reforço reduzem o tempo de encaminhamento de mensagens a nós de alta conectividade, evitando descartes anteriores ao tempo de expiração decorrentes de congestionamentos na rede.

A análise do gráfico da latência mostra que os menores intervalos de geração saturam a rede mais rápido, obrigando os protocolos mais eficazes a aguardar pelas melhores oportunidades de repasse, aumentando os atrasos. O protocolo CASTRO original apresenta latências que podem chegar a 80 minutos, mas o modelo QL-CASTRO tem latências menores por adotar a política baseada na estimativa de atraso (Q_{est}), a qual acelera as decisões de encaminhamento. Como resultado, o QL-CASTRO é mais resistente ao impacto do tráfego intenso, registrando latências inferiores a 60 minutos e evidenciando a eficácia do aprendizado por reforço para redução de atrasos.

Por fim, os resultados de sobrecarga em rede e de consumo energético revelam que a alta densidade de mensagens preenche rapidamente o armazenamento dos *buffers*, destacando a diferença de comportamento entre os protocolos propostos – CASTRO e QL-CASTRO – e os modelos do estado da arte – QoN-ASW e HESnW. Isso se explica pelo fato de os modelos propostos não realizarem transmissões de mensagens a nós que tenham taxas de ocupação de *buffer* superiores a um limiar máximo, reduzindo o consumo energético dos nós e o risco de perdas por *overflow*.

À medida que o intervalo de geração aumenta e a geração de mensagens se torna mais esparsa (20 a 40 segundos), os nós conseguem preservar mais espaço livre de *buffer*. Nesses casos, os protocolos CASTRO e QL-CASTRO exibem um crescimento moderado na sobrecarga (entre 50 a 75) e no consumo energético (3 Joules). Esse incremento atesta que, ao dispor de recursos computacionais extras, os protocolos propostos os utilizam de forma controlada para tornar os repasses mais eficientes, operando dentro de uma margem de consumo energético compatível com tecnologias como o *Bluetooth Low Energy*.

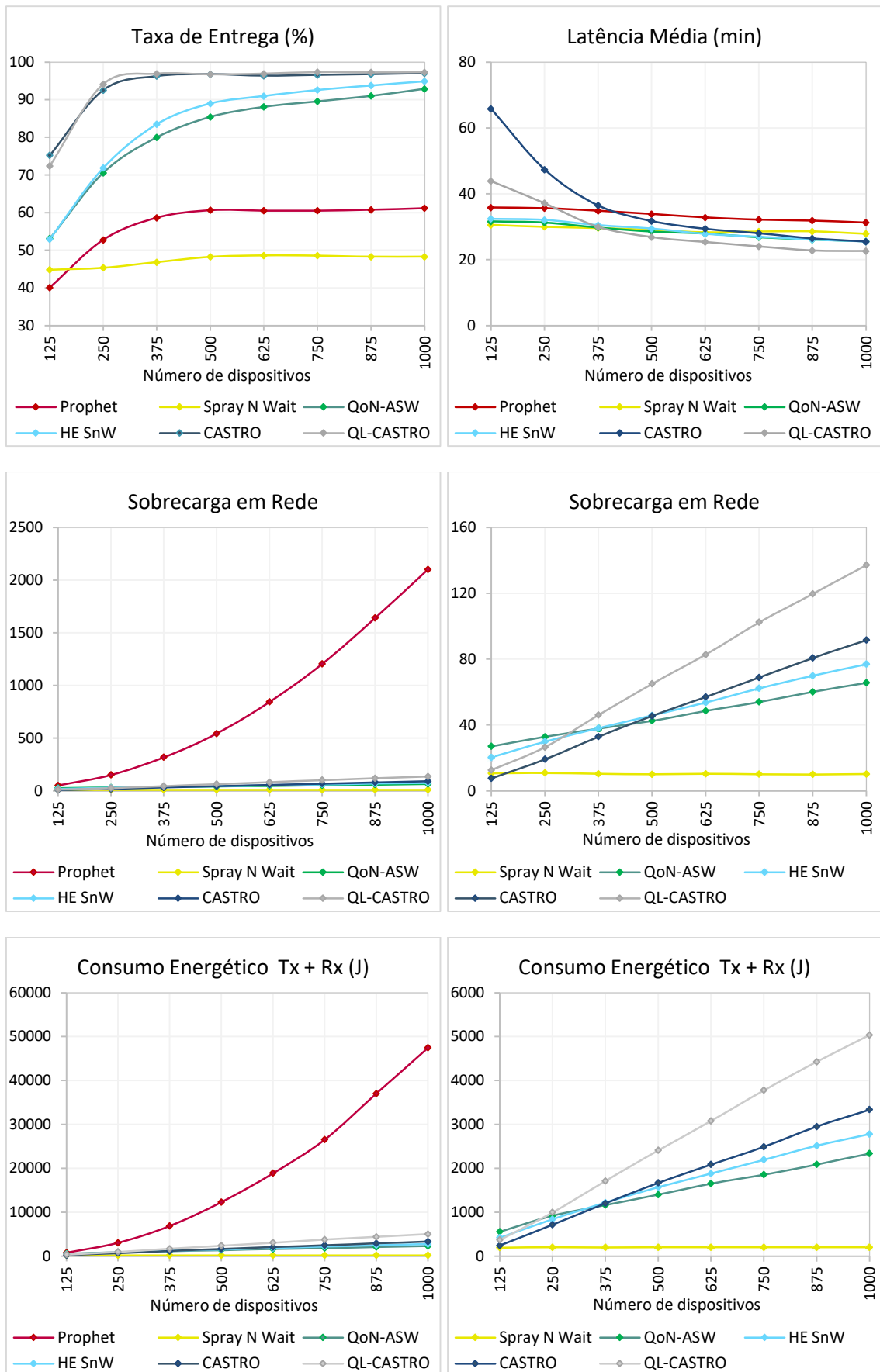
8.4 Análise de resultados: Rede esparsa veicular

O cenário do mapa da cidade de Manaus representa uma rede veicular que atua em uma área de dimensões metropolitanas, na qual os nós são dotados de interface de comunicação Wi-Fi 6 e de maior velocidade de deslocamento. Como o número de nós foi mantido da seção anterior, o presente cenário opera com uma menor densidade de nós, resultando em uma rede veicular esparsa de alta mobilidade. Esse contexto é desafiador para os protocolos de roteamento, que devem aproveitar as breves janelas de contato para tomar decisões. Os valores representados nos gráficos correspondem à média de 15 simulações feitas a partir de sementes aleatórias de geração de mensagens.

8.4.1 Variação do número de nós

A análise da taxa de entrega sob a variação de 125 a 1000 dispositivos revela a resiliência dos modelos propostos ao distanciamento geográfico, conforme os Gráficos 8.5. Mesmo na configuração mais esparsa (125 nós), os protocolos CASTRO e QL-CASTRO iniciam com taxas de entrega próximas a 75%, enquanto protocolos recentes da literatura (HESnW e QoN-ASW) registram valores em torno de 50% a 55%, e o PRoPHET opera abaixo de 50%. Conforme a rede interliga 250 nós ou mais, os modelos propostos convergem rapidamente para 95% de entrega, demonstrando alta eficiência na utilização dos poucos encontros disponíveis.

A vantagem da aplicação do *Q-Learning* torna-se mais visível no gráfico da latência. Para a configuração de 125 nós, o protocolo CASTRO apresenta retenção prolongada de mensagens, resultando em uma latência média de 65 minutos. A introdução do QL-CASTRO reduz essa métrica inicial para cerca de 45 minutos. À medida que a densidade aumenta, o QL-CASTRO atinge as menores latências do experimento, com aproximadamente 22 minutos para 1000 nós. Esta redução mostra que o aprendizado focado na estimativa de atraso Q_{est} aprimora a seleção de nós para repasse.



Gráficos 8.5 – Desempenho para número de nós variável – rede esparsa veicular

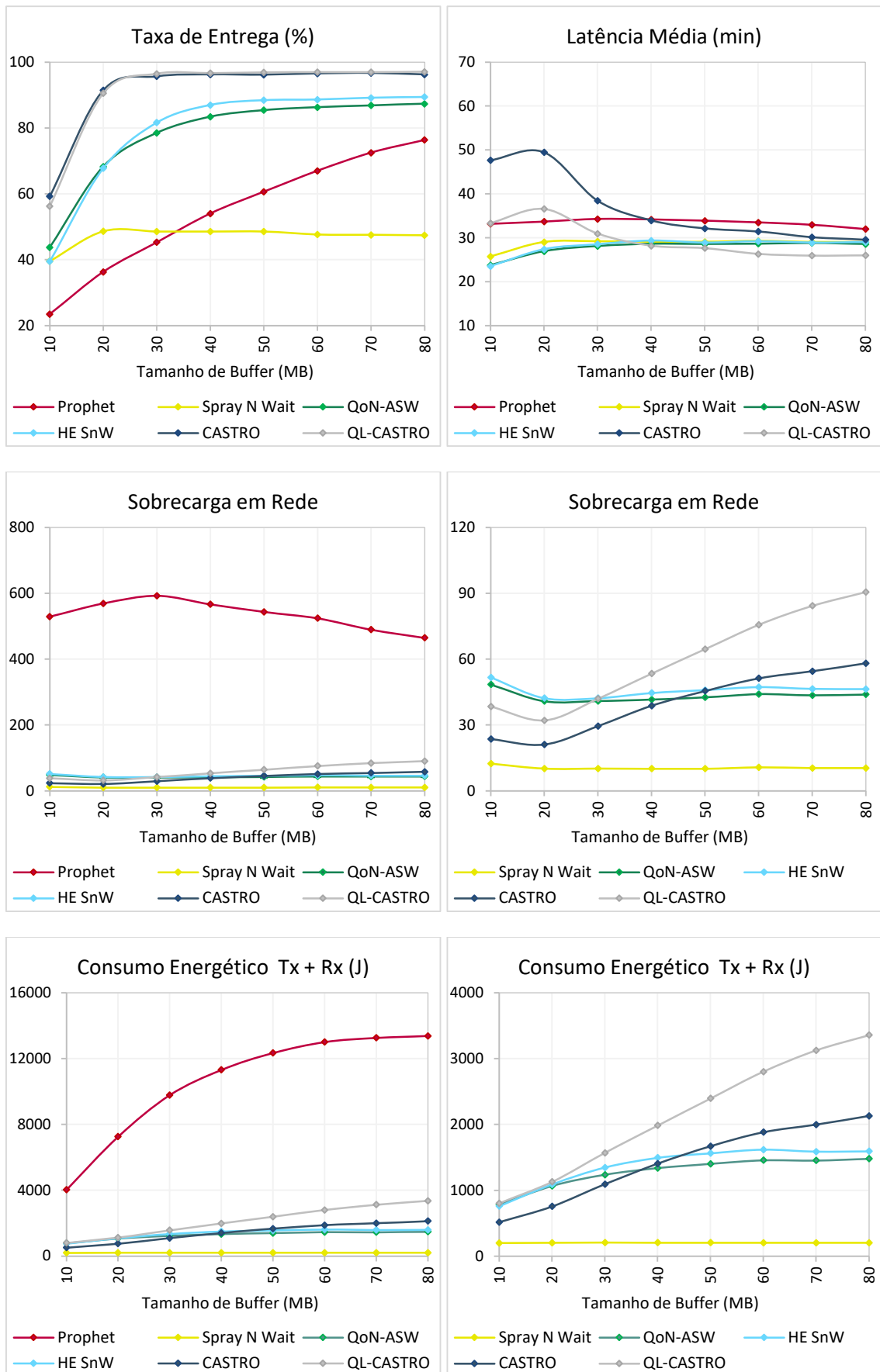
Quanto à sobrecarga e ao consumo energético, a natureza de inundação do PProPHET resulta em um crescimento exponencial, ultrapassando o valor de 2000 para sobrecarga e 47 kJ de consumo total dos nós da rede. Isso torna sua aplicação em redes veiculares reais custosa. Em contrapartida, os protocolos CASTRO e QL-CASTRO mantêm o valor de sobrecarga abaixo de 140. Observa-se que o QL-CASTRO apresenta sobrecarga e consumo energético levemente superiores aos do protocolo CASTRO básico, sendo este acréscimo justificado pela exploração de novas rotas impulsionada pelo agente de aprendizado. Trata-se de um compromisso aceitável, considerando-se a substancial redução de latência obtida.

8.4.2 Variação do tamanho do *buffer*

A restrição de *buffer* (variando de 10 MB a 80 MB) para mensagens de até 5 MB impõe alta probabilidade de descarte por *overflow*. Os Gráficos 8.6 ilustram o desempenho dos protocolos para o cenário de redes veiculares esparsas, com tamanhos variáveis de *buffer*.

No limiar mais crítico (10 MB), os protocolos PProPHET, QoN-ASW e HESnW, perdem eficiência, com taxas de entrega que variam entre 20% e 45%. Os protocolos CASTRO e QL-CASTRO, em contraste, asseguram cerca de 60% de entrega. A partir de 20 MB, esse percentual sobe para 90%, em virtude do uso de métricas sociais, da renovação do tempo de vida de mensagens promissoras e da verificação da taxa de ocupação dos *buffers* de nós receptores.

No que concerne à latência sob restrição de memória, o protocolo CASTRO apresenta um pico próximo de 50 minutos em *buffers* de 20 MB – comportamento característico de sua política de repasse baseada em chaves e consistências sociais. Em contrapartida, o protocolo QL-CASTRO reduz a latência para menos de 40 minutos em toda a escala, registrando o menor valor observado a partir de 50MB de *buffer*. Esse desempenho é reflexo da integração entre o mecanismo de limiar de aposentadoria e o aprendizado por reforço.



Gráficos 8.6 – Desempenho para tamanho de *buffer* variável – rede esparsa veicular

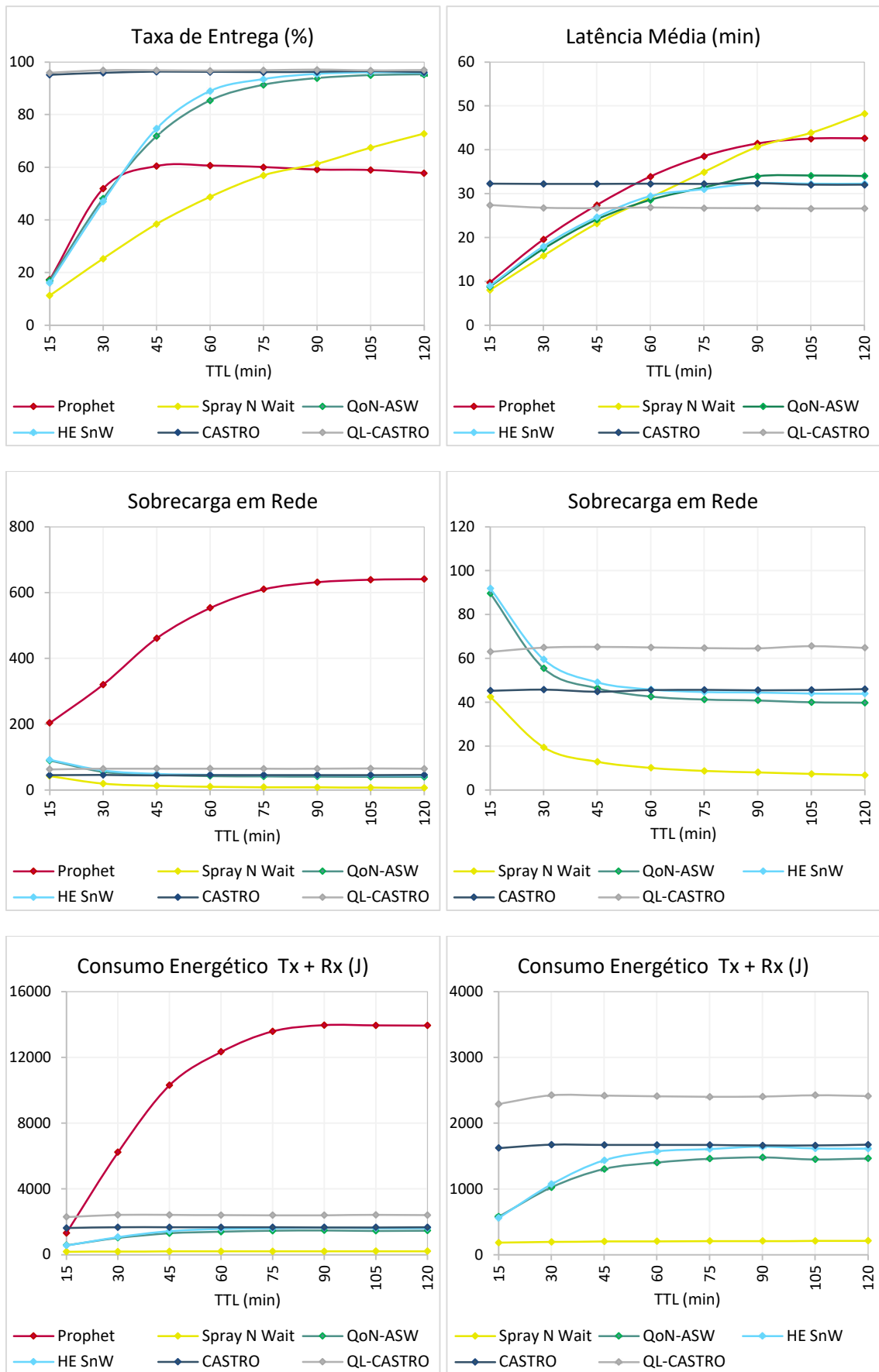
A análise de sobrecarga e de consumo energético evidencia o contraste entre as abordagens. O P_{Ro}PHET ultrapassa 14 kJ de consumo e pode chegar a um valor de sobrecarga igual a 600, demonstrando ser um protocolo que necessita de muitos recursos de rede. Os protocolos QoN-ASW e HESnW operam em um estrato intermediário, convergindo para um valor de sobrecarga igual a 45 e de consumo igual a 1,5 KJ, no total. Em contrapartida, o modelo CASTRO opera com controle de difusão, mantendo o consumo energético em até 2 kJ, enquanto o valor de sobrecarga chega a 60. O QL-CASTRO registra sobrecarga e consumo total superiores aos do CASTRO original, decorrente do processo exploratório do *Q-Learning*. Porém, este acréscimo é aceitável, considerando-se a manutenção das elevadas taxas de entrega e as reduções de latência.

8.4.3 Variação do tempo de vida das mensagens (TTL)

O parâmetro de tempo de vida (TTL) estabelece a janela temporal limite para a conclusão do roteamento. Em uma topologia esparsa como a do cenário veicular, as mensagens demandam mais tempo para percorrer distâncias geográficas entre as zonas de conectividade. Os Gráficos 8.7 mostram o desempenho dos protocolos em cenários com TTL variável, de 15 a 120 minutos.

No caso mais difícil, em que o TTL é igual a 15 minutos, os protocolos da literatura apresentam taxas de entrega próximas a 20%. Em contrapartida, os protocolos CASTRO e QL-CASTRO utilizam as métricas sociais para prorrogar o prazo de validade das mensagens, obtendo uma taxa de entrega superior a 90%. O QL-CASTRO também inclui o aprendizado por reforço como critério para prorrogação de TTL. Isso mostra que a concessão de tempo extra é fundamental para a entrega de mensagens em redes descentralizadas e de topologia variável.

A análise do gráfico da latência aponta o crescimento esperado à medida que o TTL se expande, uma vez que mensagens submetidas a trajetos mais longos conseguem alcançar o destino, elevando a média global da latência. Todavia, a curva do QL-CASTRO é substancialmente mais achatada, não ultrapassando 30 minutos, mesmo para os maiores valores de TTL (120 minutos). O limiar de aposentadoria interrompe a renovação indefinida de tempo de vida em rotas ineficientes, impedindo a o aumento contínuo da latência.



Gráficos 8.7 – Desempenho para TTL variável – rede esparsa veicular

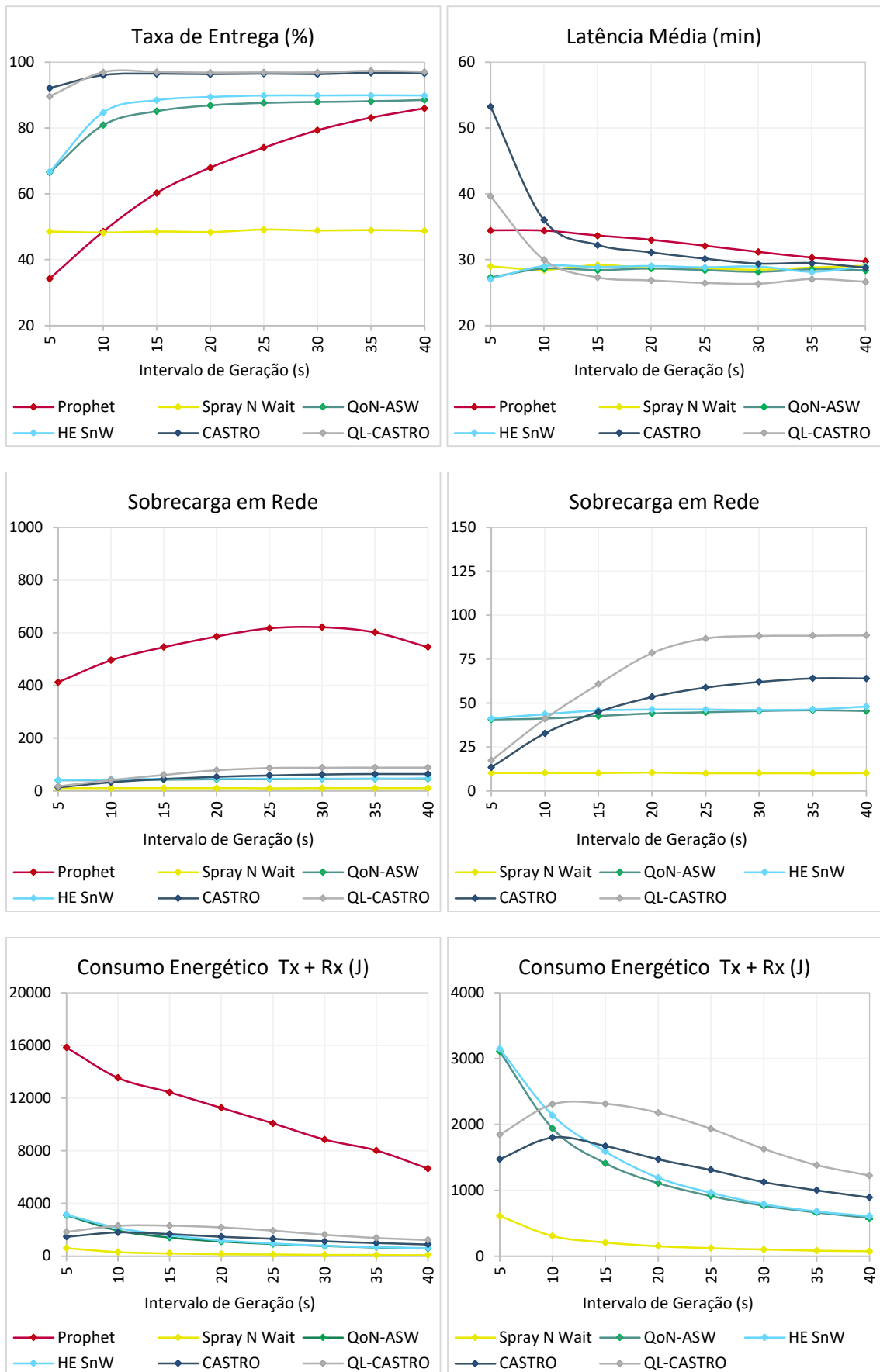
O aumento do TTL permite o surgimento de novas oportunidades de encontro. Protocolos probabilísticos de inundação, como o P_{Ro}PHET, convertem essa janela estendida em transmissões redundantes, resultando em um consumo energético de 14 kJ e uma sobrecarga superior a 600. Os demais protocolos da literatura também aumentam suas taxas de entrega, consequentemente elevando a latência e reduzindo a sobrecarga. No entanto, eles conseguem manter um consumo energético baixo, inferior a 2 kJ.

Os protocolos CASTRO e QL-CASTRO, por sua vez, mantêm os indicadores de sobrecarga e consumo energético constantes ao longo de toda a variação de TTL. Essa estabilidade mostra que as políticas de encaminhamento adotadas pelos protocolos propostos restringem as replicações aos caminhos que oferecem vantagem social ou temporal, assegurando que o tempo de vida prolongado não seja convertido em saturação do canal de comunicação.

8.4.4 Variação do intervalo de geração de mensagens

A alteração no intervalo de criação de mensagens (variando de 5 a 40 segundos) avalia a resiliência da rede contra congestionamentos de dados. O desempenho dos protocolos é ilustrado nos Gráficos 8.8. No limiar de máximo estresse (intervalos de 5 segundos), os protocolos da literatura sofrem redução na taxa de entrega devido a descartes prematuros. Nestes cenários, o CASTRO e o QL-CASTRO sustentam taxas de entrega superiores a 90%, à frente dos 65% do QoN-ASW e do HESnW. O repasse criterioso de mensagens, considerando-se apenas métricas sociais ou temporais aprimoradas, confere um desempenho mais eficaz aos modelos propostos.

O impacto das decisões é evidenciado no gráfico da latência. O protocolo CASTRO apresenta uma escalada acentuada no tempo de entrega, atingindo cerca de 55 minutos para o intervalo de 5 segundos. Esse comportamento reflete a inserção contínua de mensagens na rede que, associada à política de retenção pautada por métricas sociais, eleva os tempos de espera. Para reduzir este efeito adverso, o QL-CASTRO promove a sinergia entre a avaliação de rotas rápidas (*Q-Learning*), o repasse de mensagens recentes (*Newest-First*) e o limiar de aposentadoria para reduzir a latência máxima para menos de 40 minutos.



Gráficos 8.8 – Desempenho para intervalo de geração variável – rede esparsa veicular

No aspecto de sobrecarga e de consumo energético, a alta frequência de geração de mensagens penaliza os protocolos tradicionais. O aumento na oferta de dados faz com que protocolos como o PROPHET maximizem o gasto energético em transmissões dotadas de poucas informações para roteamento.

Os protocolos CASTRO e QL-CASTRO elevam a quantidade de repasses quando os nós da rede possuem maior capacidade de armazenamento, o que normalmente ocorre quando os intervalos de geração são mais largos. Isso tende a aumentar a sobrecarga e o consumo energético destes protocolos. No entanto, a utilização de métricas sociais (chaves e consistências), em combinação com as políticas de gerenciamento de *buffer*, assegura que a rede veicular suporte o aumento do tráfego sem entrar em colapso por exaustão de bateria ou memória.

8.5 Síntese do desempenho

A avaliação experimental valida as formulações propostas nos Capítulos 5 e 7, evidenciando o bom desempenho de roteamento dos protocolos CASTRO e QL-CASTRO. Os cálculos de métricas sociais e temporais foram associados a estratégias de repasse, de ajuste de TTL das mensagens e de gerenciamento de *buffer*, resultando em elevadas taxas de entrega e um consumo energético baixo ou moderado. A inclusão do *Q-Learning* também proporcionou a redução do atraso médio das entregas.

O protocolo CASTRO básico, por adotar uma postura de retenção conservadora baseada em métricas como chaves e consistências sociais, proporciona um menor consumo de bateria e de recursos de rede. No entanto, ele apresenta uma latência média mais elevada, em comparação com a sua versão modificada baseada em *Q-Learning*. Este efeito é uma consequência do acúmulo de mensagens no aguardo para as transmissões mais promissoras.

A estrutura do QL-CASTRO reduz a latência média do roteamento ao incluir a otimização do tempo (Q_{est}) como critério de encaminhamento de mensagens. Dessa forma, o QL-CASTRO evita a dependência de métricas puramente sociais. Ademais, a ordenação de mensagens baseada em tempo de criação e a utilização de um limiar de aposentadoria são recursos extras que auxiliam na redução da latência média.

Em contrapartida, a redução da latência exige um custo operacional previsível. O QL-CASTRO registra maior sobrecarga e um consumo energético superior ao do CASTRO original. Este acréscimo é uma consequência direta da natureza exploratória do *Q-Learning* — que necessita testar novas rotas para convergir o aprendizado — e da sua maior agilidade na liberação de cópias.

Apesar deste incremento operacional em relação ao seu predecessor, ao contrastar seus resultados com os protocolos do estado da arte, conclui-se que o QL-CASTRO é uma alternativa eficaz para a obtenção de elevadas taxas de entrega em redes oportunistas descentralizadas, mantendo o consumo energético sob patamares controlados. Os resultados evidenciam sua adaptabilidade a diferentes perfis de mobilidade (densos ou esparsos) e adequada escalabilidade, tornando-o um protocolo viável e eficiente para o escopo de redes tolerantes a atrasos e desconexões.

9. CONCLUSÕES

Esta tese abordou o problema do roteamento de dados em Redes Tolerantes a Atrasos e Desconexões (DTNs) e Redes Oportunistas, ambientes caracterizados pela ausência de conectividade fim-a-fim, mobilidade dinâmica e restrições de recursos de hardware. O objetivo central consistiu em formular, simular e avaliar protocolos de roteamento descentralizados orientados a elevar as taxas de entrega de mensagens, mantendo o consumo energético e a sobrecarga da rede em patamares sustentáveis para dispositivos restritos.

Como primeira grande contribuição, desenvolveu-se o Protocolo CASTRO. Enquanto abordagens consolidadas e do estado da arte da literatura fundamentam suas decisões majoritariamente na frequência agregada de encontros ou em estimativas puramente probabilísticas, o CASTRO avançou ao segmentar e analisar estruturalmente a duração dos contatos e os intervalos de desconexão. A modelagem matemática das métricas de Força (S), Tendência (T) e Regularidade (R) permitiu extrair *Scores* Diretos e Transitivos do histórico social local, resultando em uma seleção de rotas mais refinada. Os resultados experimentais indicaram que o CASTRO restringe a replicação redundante de pacotes, assegurando taxas de entrega elevadas com baixo custo energético comparado a modelos tradicionais.

Apesar dos ganhos em eficiência energética e taxa de entrega, observou-se que a diretriz conservadora do CASTRO — focada estritamente na confiabilidade social — gerava impactos desfavoráveis sobre a latência média de entrega. Para mitigar essa limitação, a segunda contribuição consistiu na concepção do Protocolo QL-CASTRO. A integração de uma camada de Aprendizado por Reforço (RL) — adaptada para estimar dinamicamente o atraso de propagação e combinada com políticas rigorosas de gestão de *buffer* (limiar de aposentadoria e escalonamento *Newest-First*) — conferiu maior dinamismo à tomada de decisão. As avaliações experimentais conduzidas nos cenários de Helsinki e Manaus indicaram que o QL-CASTRO reduz os tempos de atraso em relação ao protocolo base, mantendo taxas de entrega elevadas e superando propostas recentes do estado da arte.

Em suma, as arquiteturas propostas oferecem mecanismos escaláveis e autônomos que potencializam o roteamento em ambientes desprovidos de infraestrutura fixa. O QL-CASTRO posiciona-se como uma alternativa eficiente para cenários de computação em névoa e Internet das Coisas (IoT), operando com inteligência local e descentralizada.

Como desdobramentos naturais e perspectivas de trabalhos futuros, apontam-se as seguintes direções:

1. Estudos de ablação: Executar uma análise decomposta do QL-CASTRO para isolar e quantificar o ganho de desempenho gerado separadamente pelas métricas sociais, pelo *Q-Learning*, pela limpeza de *buffer* e pelas políticas de renovação de *TTL*.
2. Comparação com outras abordagens de RL: Realizar avaliações comparativas aprofundadas frente a outros protocolos de roteamento oportunista baseados em aprendizado por reforço profundo (*Deep Reinforcement Learning*).
3. Abordagens espectrais baseadas em Grafos (GSP): Investigar a aplicação de ferramentas de Processamento de Sinais em Grafos (*Graph Signal Processing* – GSP) para modelar a dinâmica topológica e a propagação temporal de pacotes sob uma perspectiva espectral, estendendo a análise de métricas locais para domínios de frequência em redes oportunistas (Da Rosa *et al.*, 2022).
4. Validação em ambiente físico: Promover a transição dos arcabouços teóricos para protótipos embarcados, explorando mecanismos dinâmicos de alternância (*handoff*) entre interfaces de rádio de baixo consumo (BLE) para descoberta de vizinhos e de alta vazão (Wi-Fi 6) para a transmissão de cargas úteis elevadas, visando o atendimento a restrições reais de bateria e vazão (*throughput*).

PRODUÇÃO CIENTÍFICA DA TESE

Os conceitos, arquiteturas e resultados avaliados nesta tese deram origem ao seguinte produto científico publicado em periódico indexado:

Artigo em Periódico Científico:

- DA ROSA, W. C.; CARVALHO, C. B.; SILVA, M. W. R.; GUEDES, R. M.; MENDES, A. C.; JUNIOR, W. S. S. Connectivity Assessment: Strength, Trend, and Regularity in Opportunistic Networks. **Electronics**, v. 15, n. 11, p. 2351, 2026. DOI: <https://doi.org/10.3390/electronics15112351>.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABBAS, A.; KRICHEN, M.; ALROOBAEA, R.; MALEBARY, S.; TARIQ, U.; PIRAN, M. J. An opportunistic data dissemination for autonomous vehicles communication. **Soft Computing**, v. 25, p. 11899–11912, 2021.
- ABED, A. K.; OZ, G.; AYBAY, I. Influence of mobility models on the performance of data dissemination and routing in wireless mobile ad hoc networks. **Computers and Electrical Engineering**, v. 40, p. 319–329, 2014.
- ALAOUI, E. A. A.; TEKOUABOU, S. C. K.; MALEH, Y.; NAYYAR, A. Towards to intelligent routing for DTN protocols using machine learning techniques. **Simulation Modelling Practice and Theory**, v. 117, p. 102475, 2022.
- AMICI, R.; BONOLA, M.; BRACCIALE, L.; RABUFFI, A.; LORETI, P.; BIANCHI, G. Performance assessment of an epidemic protocol in VANET using real traces. **Procedia Computer Science**, v. 40, p. 92–99, 2014.
- ARAFATH, M. S.; KHAN, K. U.; SUNITHA, K. V. Incorporating privacy and security in military application based on opportunistic sensor network. **International Journal of Internet Technology and Secured Transactions**, v. 7, 2017.
- AVOUSOUKPO, C. B.; OGUNSEYI, T. B.; TCHENAGNON, M. Securing and Facilitating Communication Within Opportunistic Networks: A Holistic Survey. **IEEE Access**, v. 9, p. 55009–55038, 2021.
- AYELE, E. D.; MERATNIA, N.; HAVINGA, P. J. M. Towards A New Opportunistic IoT Network Architecture for Wildlife Monitoring System. **2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)**, p. 1–5, 2018.
- AZABAL, M. J.; HERRERA, J. L.; LASO, S.; GALÁN-JIMÉNEZ, J. OPPNets and Rural Areas: An Opportunistic Solution for Remote Communications. **Wireless Communications and Mobile Computing**, v. 2021, ID 8883501, 2021.
- BACANLI, S. S.; TURGUT, D. Energy-efficient unmanned aerial vehicle scanning approach with node clustering in opportunistic networks. **Computer Communications**, v. 161, p. 76–85, 2020.
- BASU, S.; BISWAS, A.; ROY, S.; DASBIT, S. Wise-PRoPHET: A Watchdog supervised PRoPHET for reliable dissemination of post disaster situational information over smartphone based DTN. **Journal of Network and Computer Applications**, v. 109, p. 11–23, 2018.
- BAUDIC, G.; AUGER, A.; RAMIRO, V.; LOCHIN, E. Using emulation to validate applications on opportunistic networks. **Advances in Delay-Tolerant Networks (DTNs) Architecture and Enhanced Performance**, 2nd Ed, Elsevier, p. 273–280, 2020.
- BENAMAR, N.; SINGH, K. D.; BENAMAR, M.; EL OUADGHIRI, D.; BONNIN, J.-M. Routing protocols in Vehicular Delay Tolerant Networks: A comprehensive survey. **Computer Communications**, v. 48, p. 141–158, 2014.
- BHARAMAGOUDAR, S. R.; SABOJI, S. V. Routing in Opportunistic Networks: Taxonomy, Survey. **2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)**, p. 301–306, 2017.

- BIABANI, M.; YAZDANI, N.; FOTOUHI, H. EE-MSWSN: Energy-Efficient Mobile Sink Scheduling in Wireless Sensor Networks. **IEEE Internet of Things Journal**, v. 9, n. 19, p. 18360–18377, 2022.
- BISTA, B. B.; RAWAT, D. B. Enhancement of PROPHET Routing in Delay Tolerant Networks from an Energy Prospective. **2016 IEEE Region 10 Conference (TENCON)**, p. 1579–1582, 2016.
- BOLDRINI, C.; LEE, K.; ÖNEN, M.; OTT, J.; PAGANI, E. Opportunistic Networks. **Computer Communications**, v. 48, 2014.
- BORREGO, C.; BORRELL, J.; ROBLES, S. Hey, influencer! Message delivery to social central nodes in social opportunistic networks. **Computer Communications**, v. 137, p. 81–91, 2019.
- BRANDES, U. A Faster Algorithm for Betweenness Centrality. **Journal of Mathematical Sociology**, v. 25, n. 2, p. 163–177, 2001.
- BRANDES, U.; PICH, C. Centrality Estimation in Large Networks. **Department of Computer and Information Science**, University of Konstanz, 2006.
- CASTILLO, A.; JUIZ, C.; BERMEJO, B. Delay and Disruption Tolerant Networking for Terrestrial and TCP/IP Applications: A Systematic Literature Review. **Network**, v. 4, n. 3, p. 237–259, 2024.
- ÇELEBI, H.; YAPICI, Y.; GÜVENÇ, İ.; SCHULZRINNE, H. Load-Based On/Off Scheduling for Energy-Efficient Delay-Tolerant 5G Networks. **IEEE Transactions on Green Communications and Networking**, v. 3, n. 4, p. 955–970, 2019.
- CHEN, H.; LOU, W. Contact expectation based routing for delay tolerant networks. **Ad Hoc Networks**, v. 36, p. 244–257, 2016.
- CHEN, J.; YANG, X.; XU, G.; HUANG, B. CR-5: Resource optimization opportunistic network routing algorithm based on node dynamic attributes. **Ad Hoc Networks**, v. 154, p. 103356, 2024.
- CHEN, J.; BIE, P.; NIE, J.; WEI, Z. HP-ECD: Heuristic Prophet protocol based on energy balance, cache optimization, and asynchronous dormancy. **Journal of King Saud University - Computer and Information Sciences**, v. 36, n. 1, p. 101861, 2024.
- CHRIKI, A.; TOUATI, H.; SNOUSSI, H.; KAMOUN, F. FANET: Communication, mobility models and security issues. **Computer Networks**, v. 163, p. 106877, 2019.
- CUI, J.; CAO, S.; CHANG, Y.; WU, L.; LIU, D.; YANG, Y. An Adaptive Spray and Wait Routing Algorithm Based on Quality of Node in Delay Tolerant Network. **IEEE Access**, v. 7, p. 35274–35286, 2019.
- CUI, J.; WU, S.; CHANG, Y.; HUANG, D. Probabilistic Routing Algorithm Based on Node Similarity Rate in Opportunity Network. **Journal of Chinese Computer Systems**, v. 42, n. 3, p. 609–614, 2021.
- CUI, J.; WU, Y.; CHANG, Y.; SUN, J.; YU, D. Probabilistic Routing Algorithm Based on Connection Separation Time in Delay Tolerant Network. **Journal of Chinese Computer Systems**, v. 44, n. 6, p. 1257–1265, 2023.
- CUKA, M.; ELMAZI, D.; BYLYKBASHI, K.; SPAHO, E.; IKEDA, M.; BAROLLI, L. A Fuzzy-based System for Selection of IoT Devices in Opportunistic Networks Considering IoT Device Storage, Waiting Time and Node Centrality Parameters. **2018 IEEE 32nd International**

- Conference on Advanced Information Networking and Applications (AINA)**, p. 710–717, 2018.
- DA ROSA, W. C.; DANTAS, P. V.; JUNIOR, W. S. S.; CARVALHO, C. B. Graph Signal Processing and Applications: A Survey. **2022 IEEE International Conference on Consumer Electronics (ICCE)**, p. 1–4, 2022.
- DEEBAK, B. D.; HWANG, S. O. Intelligent drone-assisted robust lightweight multi-factor authentication for military zone surveillance in the 6G era. **Computer Networks**, v. 225, 2023.
- DO, Q. H.; ABREU, T.; KUSI, B.; DIOP, N. C.; SOUIHI, S. An efficient Epidemic Routing Protocol with Reinforcement Learning algorithm in Opportunistic Networks. **2025 IEEE International Conference on Communications (ICC): IoT and Sensor Networks Symposium**, p. 3497–3502, 2025.
- DONG, Y.; ZHANG, F.; JOE, I.; LIN, H.; JIAO, W.; ZHANG, Y. Learning for Multiple-Relay Selection in a Vehicular Delay Tolerant Network. **IEEE Access**, v. 8, p. 175602–175611, 2020.
- DU, Z.; WU, C.; YOSHINAGA, T.; CHEN, X.; WANG, X.; YAU, K.-L. A.; JI, Y. A Routing Protocol for UAV-Assisted Vehicular Delay Tolerant Networks. **IEEE Open Journal of the Computer Society**, v. 2, p. 85–98, 2021.
- ESHGHI, S.; KHOUZANI, M. H. R.; SARKAR, S.; SHROFF, N. B.; VENKATESH, S. S. Optimal Energy-Aware Epidemic Routing in DTNs. **IEEE Transactions on Automatic Control**, v. 60, n. 6, p. 1554–1569, 2015.
- EZIFE, F.; LI, W.; YANG, S. A Survey of Buffer Management Strategies in Delay Tolerant Networks. **2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)**, p. 599–603, 2017.
- GAN, S.; ZHOU, J.; WEI, K. HESnW: History Encounters-Based Spray-and-Wait Routing Protocol for Delay Tolerant Networks. **Journal of Information Processing Systems**, v. 13, n. 3, p. 618–629, 2017.
- GANDHI, J.; NARMAWALA, Z. A comprehensive survey on Machine Learning techniques in opportunistic networks: Advances, challenges and future directions. **Pervasive and Mobile Computing**, v. 100, p. 101917, 2024.
- GAUTAM, T.; DEV, A. Opportunistic Network Routing Protocols: Challenges, Implementation and Evaluation. **2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)**, Noida, India, p. 100-106, 2019.
- GHARIB, M.; FOROOZANI, A.; REZAEI, S.; HEMMATYAR, A. M. A.; MOVAGHAR, A. An area-scalable human-based mobility model. **Computer Networks**, v. 177, p. 107300, 2020.
- GUO, H.; SU, B.; YU, J. Adaptive Epidemic Routing Based on Buffer Management for DTNs. **2024 International Seminar on Artificial Intelligence, Computer Technology and Control Engineering (ACTCE)**, p. 77–81, 2024.
- HAJIAGHAJANI, F.; BISWAS, S. Learning based Gain-aware Content Dissemination in Delay Tolerant Networks. **2017 9th International Conference on Communication Systems and Networks (COMSNETS)**, p. 198–205, 2017.
- HAN, S. D.; CHUNG, Y. W. An Improved PRoPHET Routing Protocol in Delay Tolerant Network. **The Scientific World Journal**, v. 2015, Article ID 623090, 2015.

- HE, D.; SUN, W.; SHI, L. The Novel Mobility Models Based on Spiral Line for Aerial Backbone Networks. **IEEE Access**, v. 8, p. 11297–11314, 2020.
- HSU, Y.-F.; HU, C.-L. Enhanced Buffer Management for Data Delivery to Multiple Destinations in DTNs. **IEEE Transactions on Vehicular Technology**, v. 65, n. 10, p. 8735–8739, 2016.
- HU, T.; FEI, Y. An Adaptive and Energy-efficient Routing Protocol based on Machine Learning for Underwater Delay Tolerant Networks. **2010 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems**, p. 381–384, 2010.
- HUANG, F.; CUI, J.; CHANG, Y.; WANG, T.; WANG, M.; YANG, Y. An Improved Spray and Wait Routing Algorithm Based on Stable Transmission Capacity Evaluation of Nodes in Opportunistic Network. **International Journal of Distributed Sensor Networks**, v. 2024, Article ID 7471329, 2024.
- ISLAM, M. A.; IQBAL, M. A.; ALEEM, M.; HALIM, Z. Analysing Connectivity Patterns and Centrality Metrics for Opportunistic Networks. **2017 International Conference on Communication, Computing and Digital Systems (C-CODE)**, p. 64–69, 2017.
- KABILAN, K.; BHALAJI, N.; SELVARAJ, C.; KUMAAR, B. M.; KARTHIKEYAN, P. T. R. Performance analysis of IoT protocol under different mobility models. **Computers and Electrical Engineering**, v. 72, p. 154–168, 2018.
- KAGHO, G. O.; BALAC, M. Identifying and Planning for Group Travellers in On-Demand Mobility Models. **IEEE Open Journal of Intelligent Transportation Systems**, v. 4, p. 785–799, 2023.
- KAISAR, S.; KAMRUZZAMAN, J.; KARMAKAR, G.; RASHID, M. M. Decentralized content sharing in mobile ad-hoc networks: A survey. **Digital Communications and Networks**, v. 9, p. 1363–1398, 2023.
- KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. The ONE Simulator for DTN Protocol Evaluation. **2nd International Conference on Simulation Tools and Techniques (SIMUTools)**, p. 1–10, 2009.
- KERÄNEN, A.; KÄRKKÄINEN, T.; OTT, J. Simulating Mobility and DTNs with the ONE. **Journal of Communications**, v. 5, n. 2, p. 92–105, 2010.
- KHAKI, M.; GHASEMI, A. The impact of mobility model on handover rate in heterogeneous multi-tier wireless networks. **Computer Networks**, v. 182, p. 107454, 2020.
- KIM, S.-K.; KIM, J.-U. Measure for Centrality Based Forwarding Schemes in Opportunistic Networks. **2016 6th International Conference on Information Communication and Management**, p. 105–108, 2016.
- KLEINBERG, J. The small-world phenomenon: an algorithmic perspective. **Proceedings of the thirty-second annual ACM symposium on Theory of computing (STOC)**, p. 163–170, 2000.
- KOSMIDES, P.; LAMBRINOS, L. Intelligent Routing in Mobile Opportunistic Networks. **2018 Global Information Infrastructure and Networking Symposium (GIIS)**, p. 1–5, 2018.
- KOUKIS, G.; SAFOURI, K.; TSAOUSSIDIS, V. All about Delay-Tolerant Networking (DTN) Contributions to Future Internet. **Future Internet**, v. 16, n. 4, p. 129, 2024.
- KURONUMA, Y.; SUZUKI, H.; KOYAMA, A. An Adaptive DTN Routing Protocol Considering Replication State. **2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)**, p. 421–426, 2017.

- LE, T.; KALANTARIAN, H.; GERLA, M. A Buffer Management Strategy Based on Power-Law Distributed Contacts in Delay Tolerant Networks. **25th International Conference on Computer Communication and Networks (ICCCN)**, 2016.
- LE, T. A unified utility-based framework for joint scheduling and buffer management in Delay Tolerant Networks. **Computer Communications**, v. 249, p. 108432, 2026.
- LEE, E. H.; SEO, D. Y.; CHUNG, Y. W. An Efficient Routing Protocol Using the History of Delivery Predictability in Opportunistic Networks. **Applied Sciences**, v. 8, n. 11, p. 2215, 2018.
- LENT, R. Experimental Evaluation of a Cognitive Routing Strategy for Efficient Energy Management in DTN. **IEEE Journal of Radio Frequency Identification**, v. 8, p. 506–515, 2024.
- LI, X.; WANG, J.; JIN, X.; ZHAO, T.; YAN, W. A Framework to Analyze the Impact of Maps on VANET Simulation. **Chinese Journal of Electronics**, v. 20, n. 3, p. 401–406, 2011.
- LI, F.; JIANG, H.; LI, H.; CHENG, Y.; WANG, Y. SEBAR: Social-Energy-Based Routing for Mobile Social Delay-Tolerant Networks. **IEEE Transactions on Vehicular Technology**, v. 66, n. 8, p. 7195–7206, 2017.
- LIAQ, M.; SHARIF, S.; ZEADALLY, S.; EJAZ, W. Utilization of machine learning in future wireless networks for resource optimization: A survey. **Ad Hoc Networks**, v. 178, p. 103983, 2025.
- LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. **ACM SIGMOBILE Mobile Computing and Communications Review**, v. 7, n. 3, p. 19–20, 2003.
- LIU, L.; WANG, R.; WU, J. A time-inhomogeneous Markov chain and its distributed solution for message dissemination in OUSNs. **Journal of Parallel and Distributed Computing**, v. 130, p. 179-192, 2019.
- LOHACHAB, A.; JANGRA, A. Opportunistic Internet of Things (IoT): Demystifying the Effective Possibilities of Opportunistic Networks Towards IoT. **2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)**, p. 1100–1105, 2019.
- LÓPEZ-VILLEGAS, I.; MARTÍNEZ-RIOS, E. A.; IZQUIERDO-REYES, J.; BUSTAMANTE-BELLO, R.; FALCONE, F. A systematic literature review of emergency communications assisted by unnamed aerial vehicles. **Ad Hoc Networks**, v. 182, p. 104063, 2026.
- LU, Y.; WANG, W.; CHEN, L.; ZHANG, Z.; HUANG, A. Distance-Based Energy-Efficient Opportunistic Broadcast Forwarding in Mobile Delay-Tolerant Networks. **IEEE Transactions on Vehicular Technology**, v. 65, n. 7, p. 5512–5524, 2016.
- MA, H.; LI, S.; WU, H.; XING, L.; ZHANG, X. QRAVDR: A deep Q-learning-based RSU-Assisted Video Data Routing algorithm for VANETs. **Ad Hoc Networks**, v. 171, p. 103790, 2025.
- MAJEED, M. R.; NAEEM, F.; SALAM, A.; KHAN, W. U.; IRFANULLAH, S. Enhanced buffer management scheme to avoid congestion in delay tolerant social network. **Proc. IEEE**, 2018.
- MAMMERI, Z. Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches. **IEEE Access**, v. 7, p. 55916–55950, 2019.
- MAO, Y.; ZHOU, C.; LING, Y.; LLORET, J. An Optimized Probabilistic Delay Tolerant Network (DTN) Routing Protocol Based on Scheduling Mechanism for Internet of Things (IoT). **Sensors**, v. 19, n. 2, p. 243, 2019.

- MBROUP, D.; DIALLO, C.; CHERIFI, H. Temporal Networks Based on Human Mobility Models: A Comparative Analysis With Real-World Networks. **IEEE Access**, v. 10, p. 5912–5935, 2022.
- MOREIRA, W.; MENDES, P.; SARGENTO, S. Opportunistic Routing Based on Daily Routines. **arXiv preprint arXiv:1407.8368**, 2014.
- NAKAYIMA, O.; SOLIMAN, M. I.; UEDA, K.; MOHAMED, S. A. E. Combining Software-Defined and Delay-Tolerant Networking Concepts With Deep Reinforcement Learning Technology to Enhance Vehicular Networks. **IEEE Open Journal of the Vehicular Technology**, v. 5, p. 721–736, 2024.
- NIYATO, D.; WANG, P.; TAN, H.-P.; SAAD, W.; KIM, D. I. Cooperation in Delay-Tolerant Networks With Wireless Energy Transfer: Performance Analysis and Optimization. **IEEE Transactions on Vehicular Technology**, v. 64, n. 8, p. 3740–3754, 2015.
- OUALHAJ, O. A.; KOBANE, A.; BEN-OTMAN, J. A Decentralized control of Autonomous Delay Tolerant Networks: Multi Agents Markov Decision Processes Framework. **IEEE International Conference on Communications (ICC)**, p. 1-6, 2018.
- PATHAK, S.; GONDALIYA, N.; RAJA, N. A Survey on PROPHET Based Routing Protocol in Delay Tolerant Network. **2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)**, p. 110–115, 2017.
- PELUSI, L.; PASSARELLA, A.; CONTI, M. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. **IEEE Communications Magazine**, v. 44, n. 11, p. 134–141, 2006.
- PENG, D.; CHANG, Y.; DUAN, X.; CUI, J. A Predictiveness-enhanced PROPHET Based on Triple Exponential Smoothing In Mobile Opportunistic Networks. **2024 International Conference on Ubiquitous Communication (Ucom)**, p. 314–319, 2024.
- PIROZMAND, P.; WU, G.; JEDARI, B.; XIA, F. Human mobility in opportunistic networks: Characteristics, models and prediction methods. **Journal of Network and Computer Applications**, v. 42, p. 45–58, 2014.
- PRASAD, A.; GURUNG, S.; SHARMA, K. Challenges and opportunities to Enhance Buffer Management for Disaster Area in Delay Tolerant Network: A Review with Performance Analysis. **2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)**, 2024.
- RAHMOUNI, I.; EL KAMILI, M.; EL FENNI, M. R.; OMARI, L.; KOBANE, A. Optimal Buffer Management Policies in DTNs: A POMDP Approach. **IEEE ICC 2014 - Ad-hoc and Sensor Networking Symposium**, p. 94–99, 2014.
- ROOPA, M. S.; PATTAR, S.; BUYYA, R.; VENUGOPAL, K. R.; IYENGAR, S. S.; PATNAIK, L. M. Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions. **Computer Communications**, v. 139, p. 32–57, 2019.
- SAAD, A.; YAN, P.; DE GRANDE, R. E. MDP-based connectivity and availability models for Internet of Vehicles. **Internet of Things**, v. 24, p. 100963, 2023.
- SADAT, M. N.; MOHIUDDIN, M. T. An Improved MaxProp Based on Neighborhood Contact History for Delay Tolerant Networks. **16th Int'l Conf. Computer and Information Technology**, p. 287–291, 2014.
- SAFAEI, B.; MOHAMMADSALEHI, A.; KHOOSANI, K. T.; ZARBAF, S.; MONAZZAH, A. M. H.; SAMIE, F.; BAUER, L.; HENKEL, J.; EIJALI, A. Impacts of Mobility Models on RPL-

- Based Mobile IoT Infrastructures: An Evaluative Comparison and Survey. **IEEE Access**, v. 8, p. 167779–167829, 2020.
- SAKTHIVEL, T., BALARAM, A. The Impact of Mobility Models on Geographic Routing in Multi-Hop Wireless Networks and Extensions – A Survey. **International Journal of Computer Networks and Applications**, v. 8, n. 5, 2021.
- SALEEM, Y.; MITTON, N.; LOSCRI, V. DIVINE: Data offloading in vehicular networks with QoS provisioning. **Ad Hoc Networks**, v. 123, 2021.
- SAMMOU, E. M. Intelligent routing agent based on Q-learning and Markov decision processes for routing optimization in DTN networks. **International Journal of Intelligent Networks**, v. 6, p. 97–112, 2025.
- SATI, S.; PROBST, C.; GRAFFI, K. Analysis of Buffer Management Policies for Opportunistic Networks. **Proc. IEEE**, 2016.
- SHEN, H.; BAI, G. Routing in wireless multimedia sensor networks: A survey and challenges ahead. **Journal of Network and Computer Applications**, v. 71, p. 30–49, 2016.
- SILVA, P.; BURLEIGH, S.; HIRATA, C. M.; OBRACZKA, K. Congestion control in disruption-tolerant networks: A comparative study for interplanetary and terrestrial networking applications. **Ad Hoc Networks**, v. 44, p. 1–18, 2016.
- SINGH, J.; DHURANDHER, S. K.; WOUNGANG, I.; CHATZIMISIOS, P.; RODRIGUES, J. J. P. C. Reinforcement Learning Based Congestion Control Mechanism for Opportunistic Networks. **2022 IEEE Globecom Workshops (GC Wkshps)**, p. 67–73, 2022.
- SOBIN, C. C.; RAYCHOUDHURY, V.; MARFIA, G.; SINGLA, A. A survey of routing and data dissemination in Delay Tolerant Networks. **Journal of Network and Computer Applications**, v. 67, p. 128–146, 2016.
- SOCIEVOLE, A.; DE RANGO, F. Energy-aware Centrality for Information Forwarding in Mobile Social Opportunistic Networks. **2015 IEEE International Conference on Communications (ICC)**, p. 622–627, 2015.
- SOHAIL, M.; LATIF, Z.; JAVED, S.; BISWAS, S.; AJMAL, S.; IQBAL, U.; RAZA, M.; KHAN, A. U. Routing protocols in Vehicular Adhoc Networks (VANETs): A comprehensive survey. **Internet of Things**, v. 23, p. 100837, 2023.
- SONKAR, N.; KUMAR, S. Harmony search algorithm based buffer management in Delay Tolerant Networks. **2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS)**, p. 331–335, 2022.
- SPAHO, E., DHOSKA, K., VOICI, H., MOEZZI, R., & ANNUK, A. A novel energy-efficient routing protocol for delay-tolerant networks. **Proceedings of the Estonian Academy of Sciences**, v. 73, n. 3, pp. 254–263, 2024.
- SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05)**, p. 252–259, 2005.
- SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case. **IEEE/ACM Transactions on Networking**, v. 16, n. 1, p. 77–90, 2008a.
- SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case. **IEEE/ACM Transactions on Networking**, v. 16, n. 1, p. 63–76, 2008b.

- SREYA, K.; RAJYALAXMI, M.; NIREESHA, K.; KUMAR, K. P.; BHAVANI, A. Review on Energy Consumption in Delay Tolerant Network (DTN). **Proceedings of the International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)**, p. 656–663, 2024.
- SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. 2. ed. Cambridge: MIT Press, 2018.
- TEKOUABOU, S. C. K.; ALAOUI, E. A. A.; GALLAIS, A. Efficient forwarding strategy in HDRP protocol based Internet of Things. **Computer Communications**, v. 170, p. 164–176, 2021.
- TORNELL, S. M.; CALAFATE, C. T.; CANO, J.-C.; MANZONI, P. DTN Protocols for Vehicular Networks: An Application Oriented Overview. **IEEE Communications Surveys & Tutorials**, v. 17, n. 2, p. 868–887, 2015.
- TULU, M. M.; MKIRAMWENI, M. E.; HOU, R.; FEISSO, S.; YOUNAS, T. Influential nodes selection to enhance data dissemination in mobile social networks: A survey. **Journal of Network and Computer Applications**, v. 169, p. 102768, 2020.
- VAHDAT, A.; BECKER, D. Epidemic routing for partially connected ad hoc networks. **Technical Report CS-200006**, Department of Computer Science, Duke University, 2000.
- VAZINTARI, A.; COTTIS, P. G. Mobility Management in Energy Constrained Self-Organizing Delay Tolerant Networks: An Autonomic Scheme Based on Game Theory. **IEEE Transactions on Mobile Computing**, v. 15, n. 6, p. 1401–1411, 2016.
- VEGNI, A. M.; IGLESIAS, C. B.; LOSCRÍ, V. MOVES: A MemOry-based VEhicular Social forwarding technique. **Computer Networks**, v. 197, 2021.
- VERMA, R.; SONI, D.; GOYAL, A.; GUPTA, M.; ALANAZI, F.; NANTHAAMORNPHONG, A. A context-aware energy saving information centric (CAES) model for wireless communication network. **Journal of Engineering Research**, v. 13, p. 2340–2347, 2024.
- WANG, X.; PAN, W. W.; LI, X. H.; SUN, X. F.; ZHANG, W.; XU, T. DONM: Domain-based Opportunistic Networks Model in Emergency Scenarios. **Proceedings of the 3rd International Conference on Wireless Communication and Sensor Networks (WCSN 2016)**, v. 44, p. 310-314, 2016.
- WANG, E.; YANG, Y.; WU, J. Energy Efficient Beaconing Control Strategy Based on Time-Continuous Markov Model in DTNs. **IEEE Transactions on Vehicular Technology**, v. 66, n. 8, p. 7411–7421, 2017.
- WANG, R.; WANG, Z.; MA, W.; DENG, S.; HUANG, H. Epidemic Routing Performance in DTN With Selfish Nodes. **IEEE Access**, v. 7, p. 65560–65568, 2019.
- WARD, T.; MARTINEZ, K.; CHOWN, T. Adding support for delay tolerance to IPv6 networks. **Procedia Computer Science**, v. 56, p. 217–224, 2015.
- WARTHMAN, F. Delay- and Disruption-Tolerant Networks (DTNs): A Tutorial. Version 3.2 [S.l.]: **Warthman Associates**, 2015.
- WATKINS, C. J. C. H.; DAYAN, P. Q-learning. **Machine Learning**, v. 8, n. 3-4, p. 279–292, 1992.
- WEI, K.; GUO, S.; ZENG, D.; XU, K.; LI, K. Exploiting Small World Properties for Message Forwarding in Delay Tolerant Networks. **IEEE Transactions on Computers**, v. 64, n. 10, p. 2809–2818, 2015.

- WU, C.; YOSHINAGA, T.; JI, Y. A Learning-based Probabilistic Routing Protocol for Vehicular Delay Tolerant Networks. **Proceedings of the 2017 4th International Conference on Information and Communication Technologies for Disaster Management**, p. 1-6, 2017.
- WU, H.; HAN, X.; ZHU, H.; CHEN, C.; YANG, B. An Efficient Opportunistic Routing Protocol with Low Latency for Farm Wireless Sensor Networks. **Electronics**, v. 11, n. 13, 2022.
- XIAN, J.; WU, H.; MEI, X.; CHEN, X.; YANG, Y. Low-Delay and Energy-Efficient Opportunistic Routing for Maritime Search and Rescue Wireless Sensor Networks. **Remote Sensing**, v. 14, n. 20, p. 5178, 2022.
- XU, X.; ZHOU, Z.; YANG, L.; MA, X. A Routing Strategy with Small World Feature and Node Authority in Energy Constrained Delay Tolerant Networks. **2015 IEEE**, 2015.
- YAO, L.; BAI, X.; WANG, Z. An Overview of Intelligent Algorithm-Based Routing in Delay Tolerant Networks. **2023 8th International Conference on Computer and Communication Systems (ICCCS)**, p. 345-350, 2023.
- YUAN, Q.; CARDEI, I.; WU, J. An Efficient Prediction-Based Routing in Disruption-Tolerant Networks. **IEEE Transactions on Parallel and Distributed Systems**, v. 23, n. 1, p. 19–31, 2012.
- YUAN, P.; LIU, P.; TANG, S. Exploiting Partial Centrality of Nodes for Data Forwarding in Mobile Opportunistic Networks. **2014 IEEE 17th International Conference on Computational Science and Engineering**, p. 1435–1442, 2014.
- YUAN, P.; FAN, L.; LIU, P.; TANG, S. Recent progress in routing protocols of mobile opportunistic networks: A clear taxonomy, analysis and evaluation. **Journal of Network and Computer Applications**, v. 62, p. 163–170, 2016.
- YUAN, F.; WU, J.; ZHOU, H.; LIU, L. A Double Q-Learning Routing in Delay Tolerant Networks. **2019 IEEE International Conference on Communications (ICC)**, p. 1-6, 2019.
- ZHANG, X.; NEGLIA, G.; KUROSE, J.; TOWSLEY, D. Performance modeling of epidemic routing. **Computer Networks**, v. 51, n. 10, p. 2867–2891, 2007.
- ZHANG, J.; HUANG, H.; MIN, G.; MIAO, W.; WU, D. Social-Aware Routing in Mobile Opportunistic Networks. **IEEE Wireless Communications**, v. 28, n. 2, p. 152–158, 2021.
- ZHU, H.; LU, K. Resilient Opportunistic Forwarding: Issues and Challenges. **2007 IEEE Wireless Communications and Networking Conference**, p. 4033–4037, 2007.

APÊNDICE

Para avaliar o desempenho dos protocolos de roteamento em um ambiente urbano realista, faz-se necessário modelar a topologia viária da cidade. Na presente pesquisa, a cidade de Manaus, Amazonas, serviu como base para o modelo de mobilidade restrito a mapas (*Map-Based Movement*) em um dos cenários de teste. A extração, o tratamento e a formatação dos dados espaciais foram realizados utilizando o software QGIS (versão 3.22).

A primeira etapa consistiu na aquisição dos dados cartográficos. Utilizou-se o complemento *QuickOSM* integrado ao QGIS para consultar e descarregar a malha viária da região de interesse diretamente da base de dados colaborativa *OpenStreetMap* (OSM). A consulta foi filtrada pela chave *highway* com o valor espacial delimitado para "Manaus, Amazonas" (Figura A.1), resultando em uma camada vetorial contendo todas as vias mapeadas da cidade (Figura A.2).

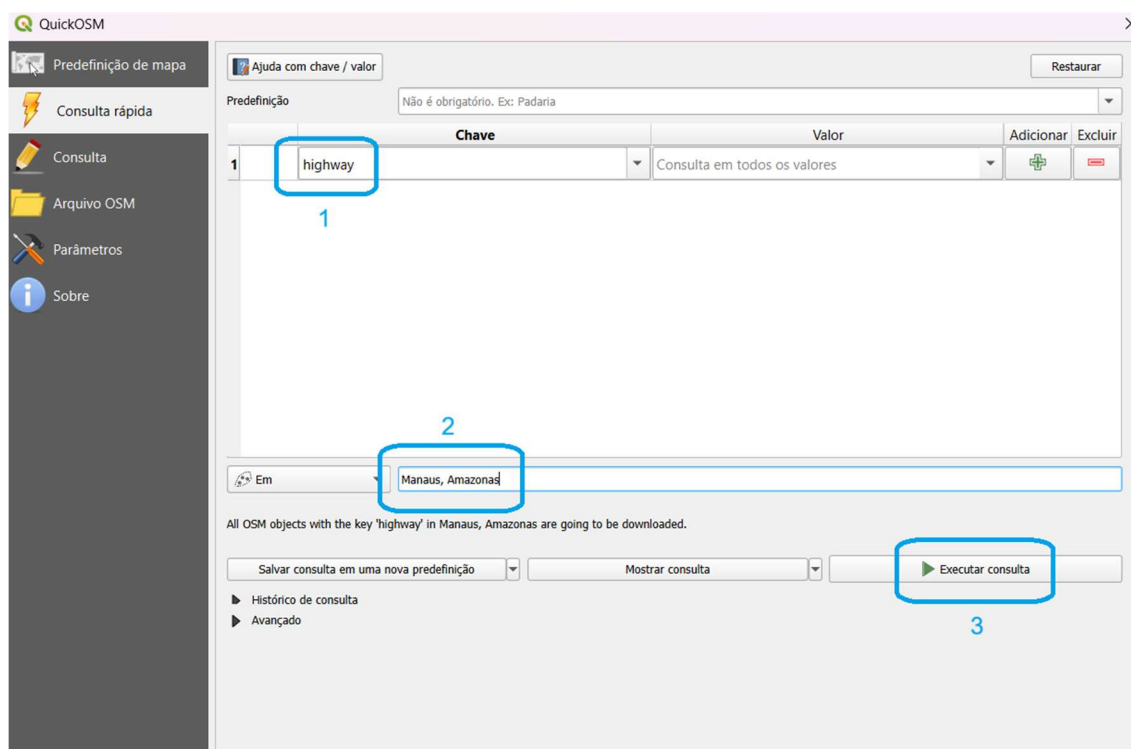


Figura A.1 – Consulta de malha viária pela base OSM.

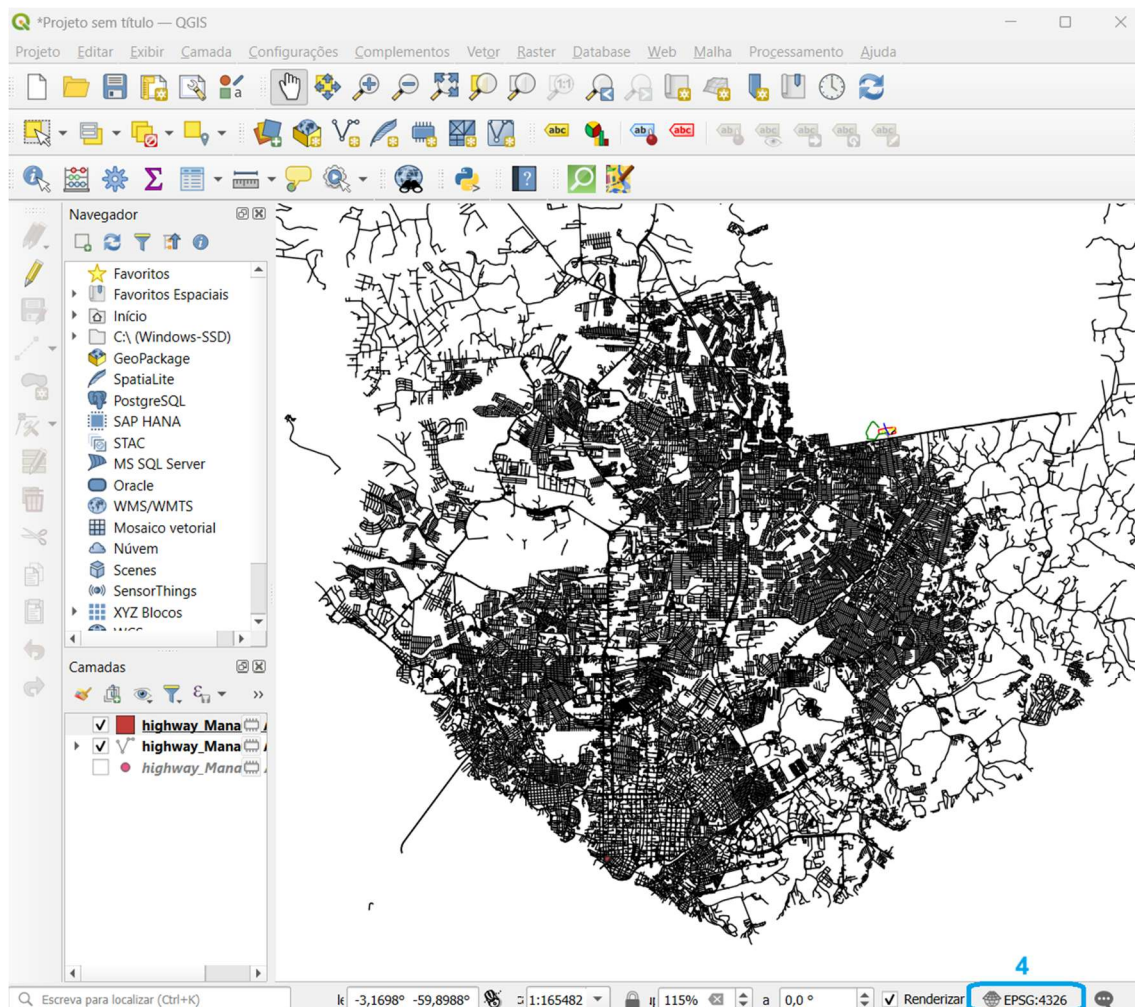


Figura A.2 – Mapeamento das vias da cidade

Como os dados nativos do OSM são fornecidos no Sistema de Referência de Coordenadas (SRC) geográficas WGS 84 (EPSG:4326), foi necessário realizar a reprojeção da camada. Para que o simulador ONE calcule corretamente as distâncias, velocidades e alcances de rádio em unidades métricas, a camada vetorial foi reprojeta para o sistema plano UTM Zona 21S (EPSG:32721), adequado para a longitude da região de Manaus (Figura A.3). O resultado foi salvo em um banco de dados *GeoPackage* para garantir a integridade dos atributos durante o processamento.

Após a reprojeção ser concluída, o mapa da cidade foi recortado para delimitar a área de roteamento. Uma consequência do recorte foi o surgimento de inconsistências na topologia da rede viária, que normalmente causam anomalias nos testes (por exemplo, nós móveis presos em vias isoladas).

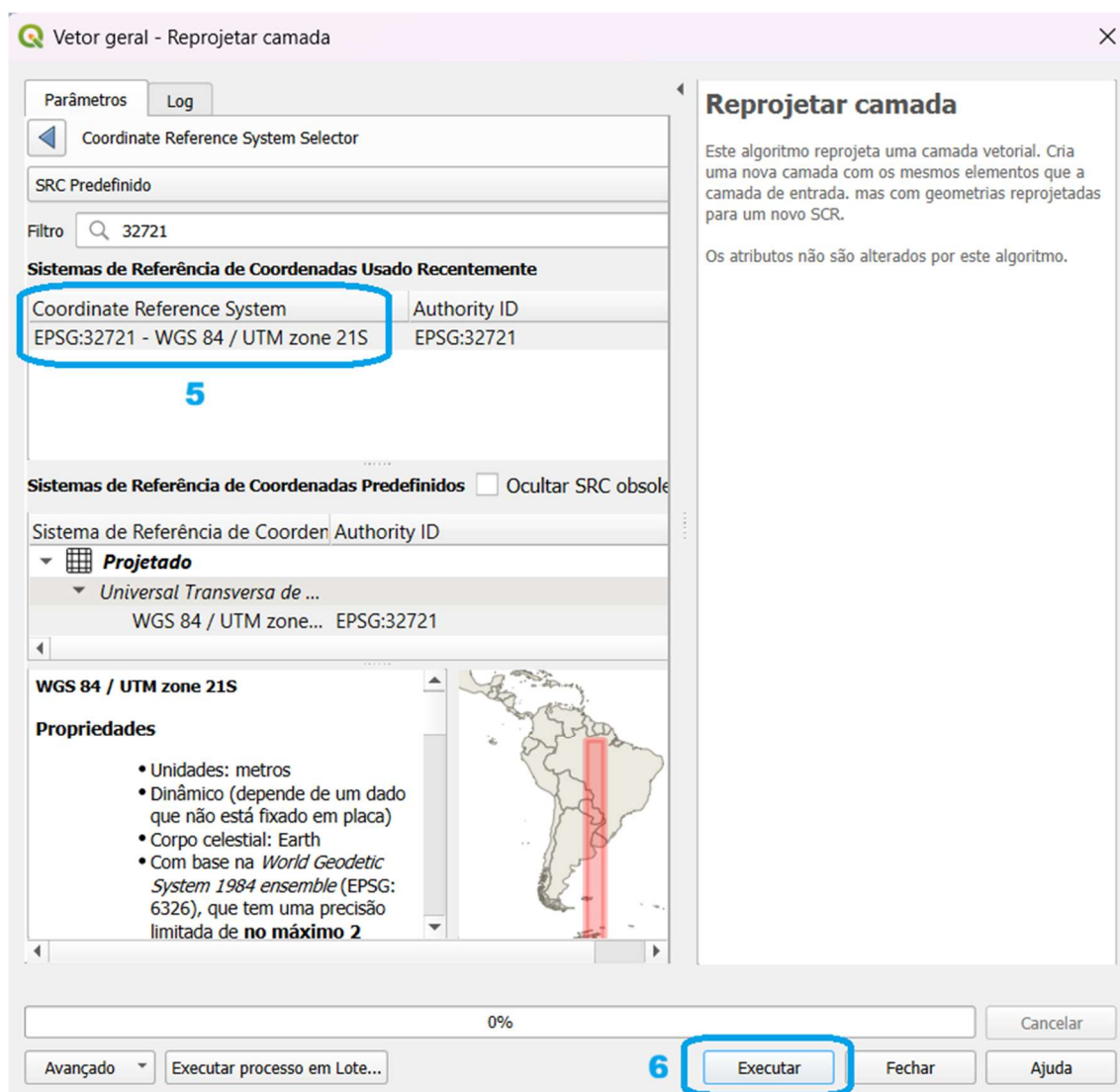


Figura A.3 – Reprojeção de Camada Vetorial

Para resolver ou mitigar esse problema, o QGIS disponibiliza a ferramenta *Disconnected Islands* (Figura A.4), capaz de analisar a conectividade dos vértices da malha viária e categorizar os segmentos. A partir dessa análise, foram identificados e excluídos os elementos isolados (Figura A.5 e Figura A.6), conhecidos como ilhas topológicas, resultando em um grafo viário contínuo e totalmente conectado (Figura A.7).

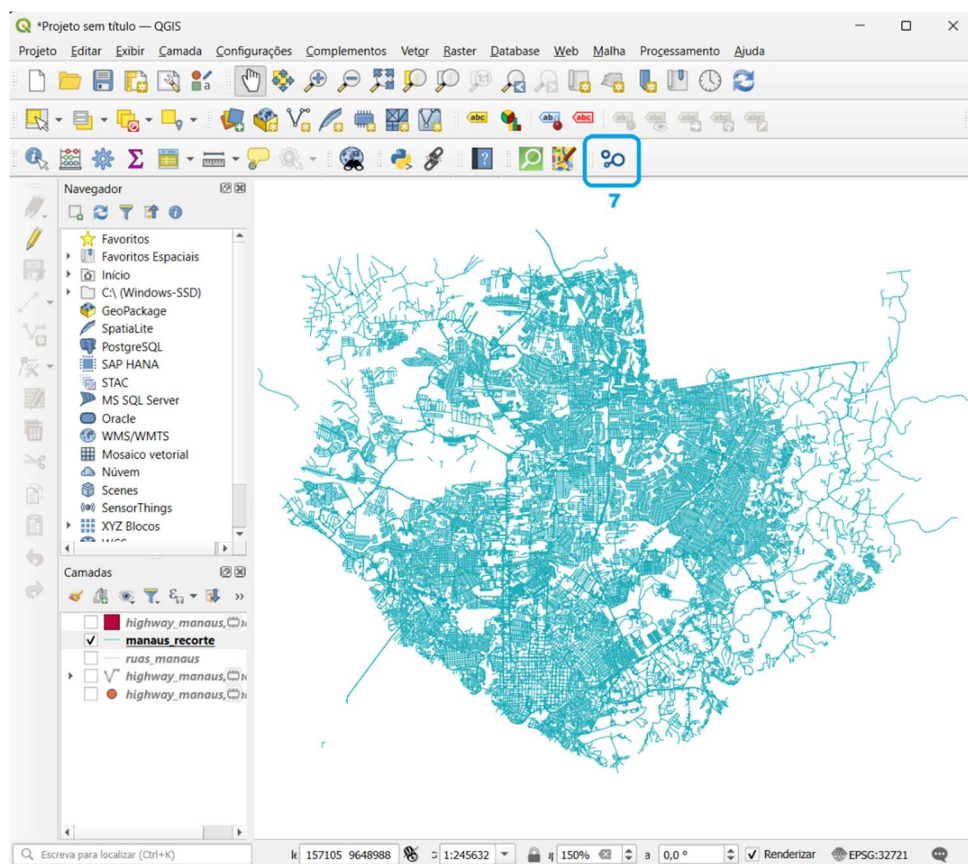


Figura A.4 – Mapa recortado e ferramenta *Disconnected Islands*

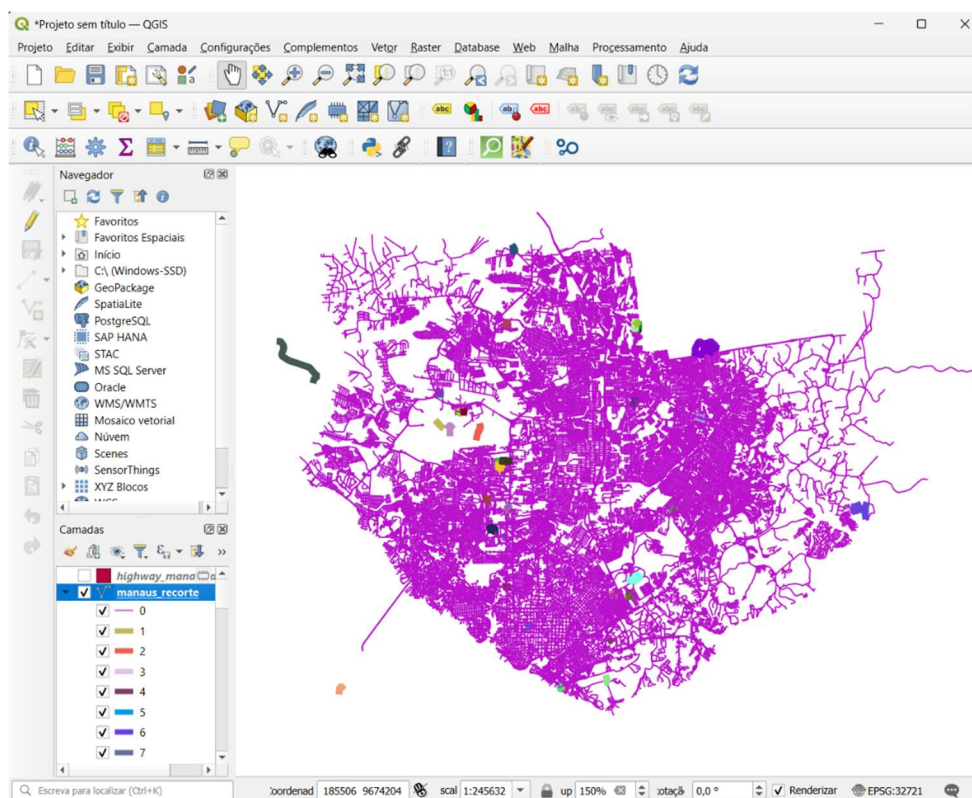


Figura A.5 – Detecção de ilhas topológicas

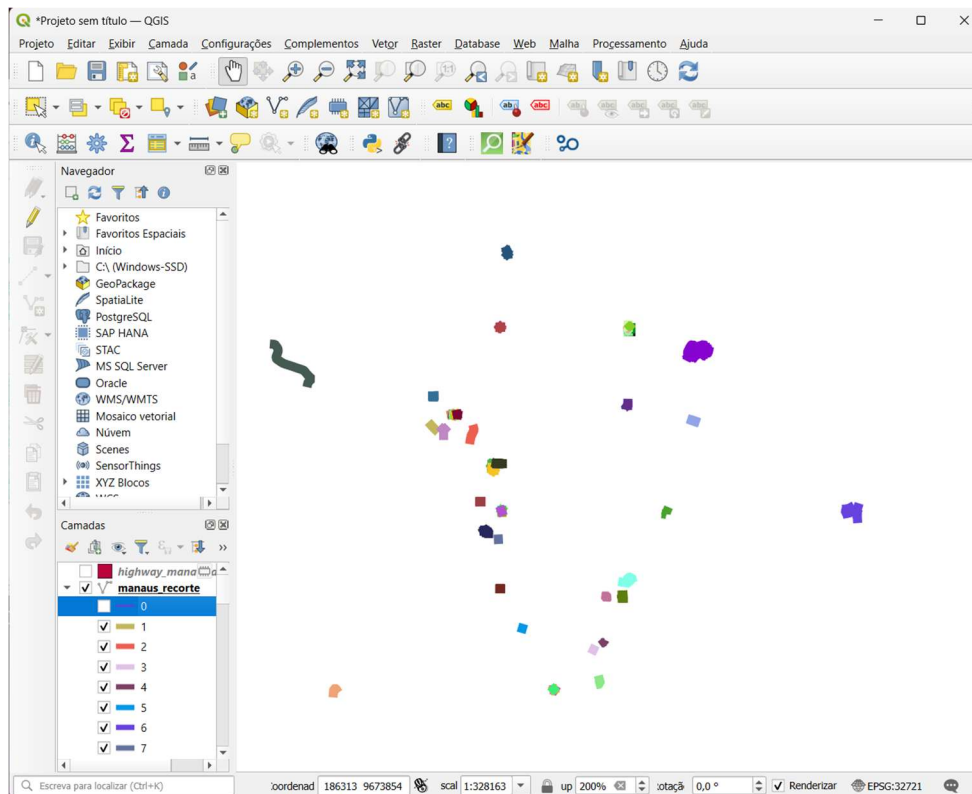


Figura A.6 – Remoção de ilhas topológicas

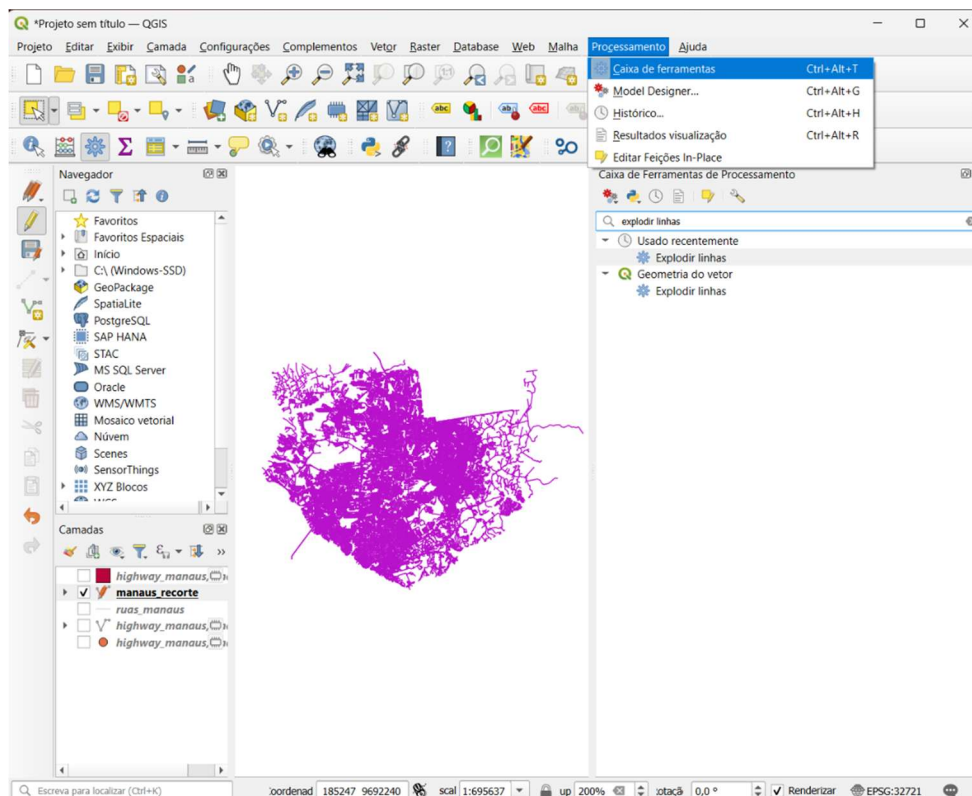


Figura A.7 – Grafo conectado e explosão de linhas

Para adequar a complexidade das geometrias ao motor de roteamento do simulador ONE, a malha viária limpa foi submetida à ferramenta *Explodir Linhas* (Figura A.7). Esse procedimento decompõe polilinhas complexas, compostas por múltiplos vértices intermediários, em segmentos de reta simples, compostos apenas por um ponto inicial e um ponto final. Essa simplificação geométrica reduz a sobrecarga computacional do simulador ao calcular as rotas mais curtas (*Shortest Path Map-Based Movement*).

Por fim, a camada processada foi exportada para o formato de texto exigido pelo simulador. Utilizou-se a função de exportação do QGIS para salvar os elementos em um arquivo de Valores Separados por Vírgula (CSV), configurando a opção de camada GEOMETRY como AS_WKT. Essa configuração extrai as coordenadas de cada segmento de reta no formato *Well-Known Text* (WKT), especificamente como feições do tipo LINESTRING. O arquivo resultante passou por uma etapa final de formatação em um editor de texto simples, onde aspas duplas geradas automaticamente pelo exportador CSV foram substituídas e removidas, adequando a sintaxe ao padrão de leitura do simulador ONE (Figura A.8).

Arquivo	Editar	Exibir
LINESTRING	(164039.801595587 9664063.91850863,163923.843107424 9663995.56533618)	
LINESTRING	(163923.843107424 9663995.56533618,163895.362300147 9663981.56109592)	
LINESTRING	(163895.362300147 9663981.56109592,163881.823136147 9663976.47583932)	
LINESTRING	(163881.823136147 9663976.47583932,163867.403785474 9663971.78659557)	
LINESTRING	(163867.403785474 9663971.78659557,163852.605387426 9663967.3508716)	
LINESTRING	(163852.605387426 9663967.3508716,163836.302733358 9663963.64147895)	
LINESTRING	(163836.302733358 9663963.64147895,163805.648985483 9663958.68542573)	
LINESTRING	(163805.648985483 9663958.68542573,163787.346917565 9663957.68229386)	
LINESTRING	(163787.346917565 9663957.68229386,163769.176115995 9663957.48755132)	
LINESTRING	(163769.176115995 9663957.48755132,163752.415840328 9663958.24867351)	
LINESTRING	(163752.415840328 9663958.24867351,163736.655984056 9663959.38893397)	
LINESTRING	(163736.655984056 9663959.38893397,163708.236687023 9663963.16135219)	
LINESTRING	(163708.236687023 9663963.16135219,163682.257347088 9663969.82956356)	
LINESTRING	(163682.257347088 9663969.82956356,163657.818061671 9663978.88188522)	
LINESTRING	(163657.818061671 9663978.88188522,163636.760912873 9663988.16505004)	
LINESTRING	(163636.760912873 9663988.16505004,163593.385251622 9664008.13359163)	
LINESTRING	(163593.385251622 9664008.13359163,163573.177309337 9657316.8152165)	
LINESTRING	(163573.177309337 9657316.8152165,165028.277298021 9657232.20294346)	
LINESTRING	(165028.277298021 9657232.20294346,165024.072979202 9657157.29992206)	
LINESTRING	(165024.072979202 9657157.29992206,165020.884404636 9657096.93303463)	
LINESTRING	(165020.884404636 9657096.93303463,165016.708284508 9657031.70415772)	
LINESTRING	(165016.708284508 9657031.70415772,165014.379294346 9656997.73864564)	
LINESTRING	(165014.379294346 9656997.73864564,165012.726959832 9656980.14572005)	
LINESTRING	(165012.726959832 9656980.14572005,165011.999954606 9656969.73903801)	
LINESTRING	(165011.999954606 9656969.73903801,165010.656101225 9656941.49886426)	
LINESTRING	(165010.656101225 9656941.49886426,165007.376651804 9656885.73629769)	
LINESTRING	(165007.376651804 9656885.73629769,165007.352788081 9656882.39347094)	
LINESTRING	(165007.352788081 9656882.39347094,165007.098162402 9656878.03166334)	
LINESTRING	(165007.098162402 9656878.03166334,165006.207478135 9656862.58823819)	
LINESTRING	(165006.207478135 9656862.58823819,165005.762215618 9656854.8388539)	
LINESTRING	(165005.762215618 9656854.8388539,165004.656101225 9656838.03166334)	
Ln 1, Col 1	12.447.954 caracteres	Texto sem formatação 100%

Figura A.8 – Arquivo .wkt para o mapa de Manaus