

UNIVERSIDADE FEDERAL FLUMINENSE

EDELBERTO FRANCO SILVA

Análise de Protocolos de Roteamento em Redes
Tolerantes a Atraso e Desconexões

NITERÓI

2011

UNIVERSIDADE FEDERAL FLUMINENSE

EDELBERTO FRANCO SILVA

Análise de Protocolos de Roteamento em Redes
Tolerantes a Atraso e Desconexões

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Paralelo e Distribuído.

Orientador:

Prof. Célio Vinicius Neves de Albuquerque

NITERÓI

2011

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

S586 Silva, Edelberto Franco

Análise de protocolos de roteamento em redes tolerantes a atraso e desconexões / Edelberto Franco Silva. – Niterói, RJ : [s.n.], 2011.
111 f.

Dissertação (Mestrado em Computação) - Universidade Federal Fluminense, 2011.

Orientador: Célio Vinicius Neves de Albuquerque.

1. Protocolo de comunicação de rede. 2. Roteamento. 3. Redes tolerantes a atrasos e desconexões. I. Título.

CDD 004.62

Análise de Protocolos de Roteamento em Redes Tolerantes a Atraso e
Desconexões

EDELBERTO FRANCO SILVA

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Paralelo e Distribuído.

Aprovada por:

Prof. Célio Vinicius Neves de Albuquerque / UFF
(Orientador)

Prof^a. Débora Christina Muchaluat Saade / UFF

Prof. Miguel Elias Mitre Campista / UFRJ

Niterói, 29 de Agosto de 2011.

“Pensée fait la grandeur de l’homme”

Blaise Pascal

A Deus, meus pais e meu irmão.

Agradecimentos

Em primeiro lugar agradeço a Deus pelo dom da vida, por seu olhar nos momentos mais íntimos de dificuldade e pelas bênçãos derramadas sobre minha vida. Agradeço a todos meus parentes e amigos que participaram dessa caminhada contribuindo para que eu me mantivesse firme e determinado. Agradeço a meu pai Lindolfo e minha mãe Leonor pela dedicação sempre dispensada a mim e, mesmo estando longe, sempre pude sentir o carinho, cuidado e amor de vocês. Agradeço a meu irmão Leonardo Franco pelas palavras de motivação, pelos finais de semana de estudos e pelos momentos de distração que me proporcionou. Não haveria melhor família!

Agradeço também aos amigos que aqui fiz e que espero para sempre manter a meu lado, em especial aos amigos de república Bruno Dembogurski e Carlos Sant’anna que me ajudaram desde o momento de adaptação, nas noites de estudos, além de sempre estarem dispostos a conversar sobre os mais diversos assuntos. Obrigado também por proporcionarem momentos ímpares de lazer. Não posso esquecer dos amigos Eduardo Pagani, um dos maiores responsáveis pelo meu ingresso no programa, e o sempre prestativo Gustavo Semaan por todo o auxílio na minha chegada a Niterói.

Agradeço a meu orientador Prof. Célio Vinicius pelo companheirismo e por toda a preocupação que sempre demonstrou tanto em relação à pesquisa quanto nas questões pessoais. Exponho aqui minha gratidão por ter trabalhado a seu lado e podido crescer lhe observando. Agradeço também ao amigo de pesquisa Etienne Oliveira, por sua motivação e empenho desde o início deste trabalho, sendo uma pessoa fundamental para sua conclusão.

Não poderia esquecer dos amigos do MídiaCom, em especial ao amigo Otávio, a nossa “mãe” Marister, e a Profa. Débora, que sempre proporcionaram muito mais que um ambiente de trabalho e discussão, um ambiente acolhedor e motivador.

Resumo

O presente trabalho tem como objetivo apresentar a análise dos mais relevantes protocolos de roteamento propostos para Redes Tolerantes a Atraso e Desconexões em cenários contrastantes, a fim de expor a necessidade de ciência de contexto nesse tipo de rede além de servir como base para uma proposta futura de um protocolo dinamicamente adaptável a contextos. Como principal contribuição deste trabalho, ressalta-se que os resultados obtidos demonstram o ganho em relação ao desempenho dos protocolos a partir da alteração dos valores dos parâmetros de cada um deles e ainda mensuram o quão distante a configuração padrão se encontra da melhor configuração em cada ambiente.

Palavras-chave: Redes Tolerantes a Atraso e Desconexões, DTN, Redes Desafiadoras, Protocolos de Roteamento

Abstract

This work analyses the impact contrasting scenarios have on the most important routing protocols proposed for Delay and Disruption Tolerant Networks, in order to expose the need for context awareness or even to serve as a basis for a proposal of a dynamically adaptive protocol. As its main contribution the results show how the turning of the parameters affect the performance of the protocols and also measures how far the default settings are from the best configuration in each environment.

Keywords: Delay and Disruption Tolerant Network, DTN, Challenged Networks, Routing Protocols

Abreviações

ACK Acknowledgement

ADUs Application Data Units

BP Bundle Protocol

BSR Bundle Status Reports

CARNIVORE Carnivore Adaptive Research Network in Varied Outdoor Remote Environments

CCSDS Consultative Committee for Space Data Systems

CHANTS CHALLENGED NeTworkS

CWC Current Window Counter

DD Direct Delivery

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DSL Digital Subscriber Line

DTN Delay and Disruption Tolerant Networks

DTNRG Delay Tolerant Network Research Group

E2E End-to-End

EBR Encounter-Based Routing

EDAQ Earliest Delivery with All Queues

EDLQ Earliest Delivery with Local Queuing

EID Endpoint Identifiers

EV Encounter Value

FDA Função de Distribuição Acumulada

FC First Contact

FTP File Transfer Protocol

GENI Global Environment for Network Innovations

GIS Geographic Information System

GPS Global Positioning System

ICMP Internet Control Message Protocol

IPN InterPlanetary Network

IRTF Internet Research Task Force

ISP Internet Service Provider

URI Uniform Resource Identifier

USB Universal Serial Bus

UTM Universal Transverse Mercator

LP Linear Programming

MAPs Mobile Access Points

MULEs Mobile Ubiquitous LAN Extensions

NECTAR Neighborhood Contact History Routing

Nidx Neighborhood Index

PDA Personal Digital Assistant

PRoPHET Probabilistic Routing Protocol

POI Points of Interest

PSN Pocket Switched Networks

RAPID Resource Allocation Protocol for Intentional DTN

RWP Random Way Point

SCPS Space Communication Protocol Standards

SMTP Simple Mail Transfer Protocol

SSP Scheme Specific Part

TCP Transmission Control Protocol

TTL Time To Live

UTM Universal Transverse Mercator

WDM Working Day Movement

WKT Well Known Text

WSN Wireless Sensor Network

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xiv
1 Introdução	1
1.1 Motivação	4
1.2 Objetivos	4
1.3 Contribuições	5
1.4 Organização	5
2 Redes Tolerantes a Atraso e Desconexões	7
2.1 Arquitetura	9
2.2 Camada de agregação	10
2.3 Tipos de contato e aplicações	11
2.3.1 Vila remota	11
Contatos previsíveis	12
2.3.2 Monitoramento de ambiente	14
Contatos oportunistas	15
2.3.3 Comunicação espacial	16
Contatos programados	17
2.3.4 Demais tipos de contatos	17
Contatos persistentes	18
Contatos sob demanda	18

2.4	Conclusão	19
3	Cenários Abordados	20
3.1	Introdução	20
3.1.1	UmassDieselNet	20
3.1.2	ZebraNet	21
3.1.3	<i>Working Day Movement</i> (WDM)	23
3.2	Configurações de cada cenário	24
3.3	Conclusão	26
4	Taxonomia e Protocolos de Roteamento em DTN	28
4.1	Trabalhos relacionados	28
4.2	Taxonomia dos protocolos de roteamento em DTN	29
4.3	Protocolos baseados em encaminhamento	31
4.4	Protocolos baseados em replicação	32
4.5	Protocolos avaliados	33
4.5.1	Spray and Wait	33
4.5.2	PRoPHET	35
4.5.3	MaxProp	36
4.5.4	RAPID	37
4.5.5	EBR	39
4.5.6	NECTAR	40
4.6	Configurações de cada protocolo	43
4.7	Conclusão	43
5	Resultados Obtidos	45
5.1	Simulador escolhido	45
5.2	Caracterização dos contatos	46

5.2.1	Duração dos contatos	46
5.2.2	Máximo teórico e o real obtido para mensagens transmitidas . .	49
5.3	Variação dos valores dos parâmetros e seu impacto	52
5.3.1	PRoPHET	53
5.3.2	Spray and Wait	56
5.3.3	MaxProp	59
5.3.4	RAPID	61
5.3.5	EBR	63
5.3.6	NECTAR	65
5.3.7	Ocorrência de protocolos em relação à entrega	66
5.3.8	Análise sobre a deficiência dos protocolos	68
5.3.9	Análise de ocorrência da configuração padrão	71
5.4	Conclusão	74
6	Considerações Finais	75
	Referências Bibliográficas	77
	Anexos	82
	Anexo A – Contribuições no código-fonte do RAPID	83
	Anexo B – Código-fonte do NECTAR	85

Lista de Figuras

2.1	Encaminhamento fim-a-fim e salto a salto	9
2.2	Camada de Agregação	10
2.3	Vilas remotas	12
2.4	Exemplo de contatos oportunistas	16
2.5	Exemplo de Contatos Programados	18
3.1	Componentes do projeto Umass DieselNet	21
3.2	Composição do colar de cada Zebra.	22
3.3	Uma das zebras do projeto ZebraNet.	22
3.4	Comparação entre contatos para o modelo WDM	25
3.5	Encontros totais vs. encontros únicos para WDM	25
4.1	Taxonomia dos protocolos de roteamento	30
5.1	Tempo de duração de contato para UmassDieselNet	47
5.2	Tempo de duração de contato para ZebraNet	48
5.3	Tempo de duração de contato para WDM	49
5.4	Carga de Mensagens na Rede $\leq 100\%$	73
5.5	Carga de Mensagens na Rede $> 100\%$	74

Lista de Tabelas

2.1	Aplicações DTN.	13
3.1	Características dos cenários e dos contextos	26
4.1	Intervalo de valores dos parâmetros	44
5.1	Valores médios do tempo duração dos contatos em segundos e comparação entre o teórico e o real de mensagens transmitidas na rede	50
5.2	Mensagens entregues para o melhor desempenho do PROPHET	53
5.3	Percentual de deficiência do PROPHET	54
5.4	Relação de valores para os parâmetros do PROPHET	55
5.5	Mensagens entregues para o melhor desempenho do Spray and Wait	57
5.6	Percentual de deficiência do Spray and Wait	58
5.7	Relação de valores para os parâmetros do Spray and Wait	58
5.8	Mensagens entregues para o melhor desempenho do MaxProp	60
5.9	Percentual de deficiência para MaxProp	61
5.10	Relação de valores para os parâmetros de MaxProp	61
5.11	Mensagens entregues para o melhor desempenho do RAPID	62
5.12	Percentual de deficiência do RAPID	63
5.13	Mensagens entregues para o melhor desempenho do EBR	64
5.14	Percentual de deficiência do EBR	65
5.15	Relação de valores para os parâmetros do EBR	65
5.16	Mensagens entregues para o melhor desempenho do NECTAR	66
5.17	Percentual de mensagens entregues para o melhor desempenho	67
5.18	Ocorrência de cada protocolo como melhor desempenho	68

5.19 Percentual de deficiência dos protocolos	69
5.20 Percentual da configuração padrão com melhor desempenho	71

Capítulo 1

Introdução

A Internet é consolidadamente uma solução tecnológica de sucesso, utilizada em todo mundo para interconectar os mais variados tipos de dispositivos de comunicação, em diferentes cenários e dando suporte a diversas aplicações. Tradicionalmente, seu ambiente é composto por uma infraestrutura que contempla diversas tecnologias, como por exemplo: satélite, celular, fibra ótica, linha telefônica, etc. Entretanto, algumas premissas necessárias ao bom funcionamento da arquitetura da Internet não são encontradas em determinados ambientes, tornando o perfil de protocolos da Internet inadequado e pouco robusto. Como exemplo a tais ambientes destacam-se: comunicações sem fio, comunicações entre dispositivos móveis, comunicações entre dispositivos móveis que apresentem restrições de energia, comunicações em ambientes rurais, comunicações em campo de batalha, comunicações em redes veiculares, comunicação em redes dinâmicas de pedestres, comunicações interplanetárias etc.

Destaca-se o crescimento nos últimos anos da utilização de dispositivos móveis, como telefones celulares, PDAs (*Personal Digital Assistants*), *tablets*, *notebooks* e *netbooks*, que vem contribuindo para o surgimento de novos ambientes de redes. Ambientes esses formados por esses tipos de dispositivos e que apresentam características como alta mobilidade, baixa autonomia de baterias e, comumente, ausência de infraestrutura. Esses ambientes são considerados “desafiadores” e possuem em comum a dificuldade de manter uma comunicação fim-a-fim entre a origem e o destino com baixa latência e pequena perda de pacotes. Nesses ambientes é recorrente a inexistência de um caminho fim-a-fim entre a origem e o destino por horas, dias, semanas ou ainda podendo jamais existir.

As características anteriormente citadas são compreendidas pelas de Redes Tolerantes a Atraso e Desconexões (*Delay and Disruption Tolerant Networks - DTNs*), como a

alta mobilidade, atrasos longos, interrupções de comunicação entre a origem e o destino e a ausência de infraestrutura. E, apesar de o termo DTN ser o mais utilizado na literatura, é possível encontrarmos também a terminologia de Redes com Desafios, ou Redes Desafiadoras (*CHALLENGED NeTworkS* - CHANTS), como referência a esse modelo de rede. Com o intuito de suprir tais características desafiadoras foi proposta uma arquitetura para esse tipo de rede, denominada arquitetura DTN (*Delay-Tolerant Networking Architecture*) [9] e em conjunto a esta arquitetura diversos protocolos de roteamento.

Os diversos protocolos de roteamento existentes em DTNs diferem, basicamente, em relação ao conhecimento que os nós têm sobre a rede e no modo como as mensagens são encaminhadas por esses nós. A classificação dos protocolos conforme seu conhecimento baseia-se nas informações que o nó possui sobre as características da rede como, por exemplo, sua topologia e o tempo médio de duração de um contato¹ e podem ser divididos entre aqueles aplicados a cenários determinísticos ou a cenários estocásticos. Sendo assim, caso os nós possuam tais informações serão classificados como determinísticos e caso não as possuam serão classificados como estocásticos. Protocolos estocásticos podem utilizar os dados coletados durante sua execução para tentar determinar as características da rede e assim definir suas políticas de encaminhamento, ou ainda replicar de forma não controlada suas mensagens a seus vizinhos como forma de maximizar a probabilidade de entrega de uma mensagem. Essa característica de replicação possibilita uma classificação secundária dos protocolos de roteamento, onde aqueles que geram cópias da mensagem original podem ser classificados como “baseados em replicação” (*replication-based*) e aqueles que não criam nenhuma cópia da mensagem na rede, são classificados como “baseados em encaminhamento” (*forwarding-based*). Neste trabalho estaremos analisando os protocolos estocásticos baseados em replicação, conforme serão apresentado no Capítulo 4.

O roteamento em DTNs se mostra uma das áreas mais desafiadoras neste modelo de rede, uma vez que enfrenta um maior número de limitações se comparado ao roteamento em redes tradicionais. O maior desafio enfrentado em DTN diz respeito à conectividade intermitente dos nós, além de sua dinamicidade gerar grandes alterações nos caminhos e conseqüente envelhecimento das rotas conhecidas. Os principais objetivos dos protocolos de roteamento em DTNs são maximizar o número de mensagens entregues (levando em consideração demais fatores como inserção de *overhead*

¹um contato existe, basicamente, a partir do momento em que um nó se encontra dentro do raio de alcance de outro nó e há a possibilidade de comunicação entre eles.

na rede) e minimizar o atraso, assim como preservar os recursos disponíveis.

O desenvolvimento de cada protocolo de roteamento tem se mostrado específico às características bem definidas de um tipo de cenário, sendo assim, cada protocolo de roteamento em DTN exibe resultados satisfatórios para aquele cenário específico. No entanto, quando aplicado a um cenário adverso à sua especificação inicial é notável seu detrimento e oscilações em relação às métricas avaliadas *a priori*.

Essa afirmativa pode ser contrastada por meio da exposição do desempenho dos protocolos *Epidemic* [57] e P_{Ro}PHET (*Probabilistic Routing Protocol*) [36] em três trabalhos com cenários distintos. De forma cronológica, temos em um primeiro cenário, o protocolo de roteamento *Epidemic* sendo exposto pela primeira vez em [57] e avaliado sob um cenário artificial composto por 50 nós aleatoriamente dispostos. A mobilidade imposta a este experimento se baseou na seleção de uma coordenada de forma aleatória com velocidade entre 0 e 20m/s. Ao chegar em seu ponto de destino cada nó escolhia uma nova coordenada e se movia até ela, e assim sucessivamente. Os resultados obtidos demonstraram um desempenho bastante satisfatório, pois em quatro dos cinco conjuntos de simulações a taxa de mensagens entregues ao destinatário foi de 100% e, no quinto, de 89,9%.

Porém, no segundo trabalho, foi proposto por [36] o protocolo de roteamento P_{Ro}PHET, o qual é comparado em dois cenários distintos ao protocolo *Epidemic*. Foi reproduzido o cenário utilizado no trabalho inicial do protocolo *Epidemic* [57] e, em um segundo cenário cuja movimentação foi modelada sob as características de uma vila remota. Já no primeiro cenário o protocolo P_{Ro}PHET apresentou resultados melhores que o protocolo *Epidemic* em algumas simulações, no segundo cenário, sob determinadas circunstâncias, a taxa de mensagens entregues foi equivalente ao dobro da taxa obtida pelo protocolo *Epidemic*.

No terceiro trabalho, em um cenário de rede veicular composta por 40 ônibus, foi proposto por [1] o protocolo de roteamento RAPID (*Resource Allocation Protocol for Intentional DTN*). Nesse trabalho, RAPID é confrontado com os protocolos P_{Ro}PHET, MaxProp [6] e *Spray and Wait* [55]. Para expor como o impacto da mudança de cenário pode ser determinante ao desempenho satisfatório do protocolo de roteamento, o P_{Ro}PHET se mostrou com o desempenho tão inferior a ponto de ser suprimido dos resultados apresentados.

1.1 Motivação

Embora as oscilações no desempenho dos protocolos de roteamento para DTN em cenários contrastantes sejam esperadas, a literatura não dispõe de trabalhos que tenham mensurado a dimensão dessas oscilações, nem o quanto protocolos parametrizáveis podem apresentar um desempenho superior à configuração sugerida como padrão.

Com essa premissa acreditamos ser de grande valia à comunidade este levantamento. A partir deste trabalho, visamos auxiliar na configuração dos protocolos de roteamento aqui expostos, definindo quais valores deverão ser utilizados a fim de se obter um melhor desempenho nos cenários abordados. Isto garante que uma comparação entre os protocolos descritos neste trabalho e outros que venham a ser futuramente propostos, seja coerente. Essa afirmação se baseia na recomendação de valores para os parâmetros de cada um dos protocolos analisados para alguns dos contextos analisados, que são expostos no Capítulo 5.

1.2 Objetivos

O presente trabalho tem como objetivo apresentar a análise dos mais relevantes protocolos de roteamento propostos para Redes Tolerantes a Atraso e Desconexões em cenários contrastantes, a fim de expor a necessidade de ciência de contexto nesse tipo de rede além de servir como base para uma proposta futura de um protocolo dinamicamente adaptável a contextos.

Para alcançar este objetivo foram selecionados três cenários distintos, onde o primeiro se baseou nos arquivos de registro de movimentação de ônibus realizado pelo projeto da universidade de Massachusetts denominado UmassDieselNet [3]. Este cenário possibilita, portanto, a partir da utilização de dados reais, reproduzirmos um ambiente de grande mobilidade. Já o segundo cenário visa representar um ambiente esparso e real. Para tal foram utilizados os dados de movimentação reais coletados pelo projeto ZebraNet [58], no Quênia. E ainda um terceiro cenário, contrastante aos anteriores, foi criado artificialmente. Baseado no modelo WDM (*Working Day Movement*) [14] e de acordo com o mapa da área central da cidade de Helsinki, na Finlândia, avaliou-se o comportamento dos protocolos sobre uma DTN composta quantitativamente por pedestres, ônibus e carros. A ideia deste último cenário é reproduzir um ambiente urbano com características próximas às reais.

Atendendo ao objetivo principal deste trabalho, serão mensuradas as oscilações dos protocolos de roteamento DTN em relação às suas respectivas propostas originais de configuração para seus parâmetros e uma gama de variações de valores. Para tanto foram selecionados protocolos de roteamento que permitissem ajustes em seus parâmetros, e estivessem entre os mais citados pela literatura, além de operarem de forma distinta. Sendo assim abordaremos os protocolos *Spray and Wait* [55], MaxProp [6], PRoPHET [36], RAPID [1], EBR [40] e NECTAR [46]. Ainda, ao final, o desempenho de todos os protocolos são também comparados ao protocolo Epidemic [57]. A intenção dessa comparação é utilizar o protocolo Epidemic como um limiar de desempenho aos demais protocolos, exatamente por não haver nenhuma restrição, *a priori*, do próprio protocolo, somente àquelas oferecidas pelas próprias características da rede, como a limitação de espaço de armazenamento.

1.3 Contribuições

Como contribuição deste trabalho pode-se destacar a extensa análise de um número significativo de protocolos de roteamento parametrizáveis para DTN, sendo selecionados os mais relevantes na literatura atual. Ainda é concluído, a partir dos resultados gerados, que há a necessidade de adaptação a contextos como forma de otimização de métricas (atraso, taxa de entrega de mensagens etc.). É também mensurado o ganho de cada protocolo em relação ao ajuste dos valores dos parâmetros em três cenários distintos além de expor aqueles que obtiveram melhor desempenho. São ainda, sintetizados os valores com maior ocorrência para cada parâmetro de cada protocolo.

Os Anexos A e B são referentes, respectivamente, a contribuição realizada no código-fonte do simulador The ONE para o protocolo RAPID e a implementação do protocolo NECTAR nesse mesmo simulador.

Resultados preliminares deste trabalho foram publicados como “Promovendo Adaptação a Contextos em DTNs” nos anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos [45].

1.4 Organização

O trabalho está organizado em um total de seis capítulos. Após este primeiro capítulo introdutório, são apresentados no Capítulo 2 os conceitos fundamentais às DTNs

como: sua arquitetura, a camada de agregação, os tipos de contato e seus cenários de aplicação. No Capítulo 3, são expostas as características dos cenários escolhidos para experimentação neste trabalho. O Capítulo 4 apresenta primeiramente alguns trabalhos relacionados, e logo em seguida uma taxonomia dos protocolos de roteamento em DTN assim como um detalhamento das funcionalidades de cada protocolo selecionado para os experimentos. Os resultados e sua análise encontram-se expostos no Capítulo 5. Este trabalho conclui com a apresentação das considerações finais no Capítulo 6.

Capítulo 2

Redes Tolerantes a Atraso e Desconexões

Com a crescente abrangência dos meios de comunicação, e em especial a Internet, novos cenários surgem para atender às mais diversas necessidades de comunicação. Dentre os diversos novos cenários existentes, a DTN visa tratar algumas limitações e restrições não abordadas no modelo tradicional da Internet. O principal objetivo deste modelo de redes é minimizar os problemas causados pelo atraso e desconexões na comunicação de forma a não restringi-la às limitações dos protocolos hoje utilizados pela Internet que têm como base a exigência de uma comunicação fim-a-fim entre a origem e o destino, baixo atraso, baixa taxa de erro, caminho reverso disponível etc.

A seguir são descritas as principais características nas quais as DTNs devem suportar.

- **Particionamento de rede:** eventualmente não haverá conexão fim-a-fim entre a origem e o destino devido ao particionamento da rede. Esses particionamentos podem ocorrer por conta da distância geográfica entre os pontos, ausência do sinal de rádio, baixa potência de sinal etc.
- **Falhas, interrupções e heterogeneidade:** devido a natureza das DTNs, geralmente empregadas em ambientes hostis e adversos, os nós que compõem este tipo de rede estão sujeitos a inúmeras falhas de operação. Estas falhas podem causar interrupções na comunicação da rede. Outra premissa é que nem sempre as DTNs se conectarão a outras DTNs, formando um ambiente homogêneo em relação aos protocolos e aplicações, para tanto as DTNs deverão ser capazes de se comunicar com outros modelos de rede.
- **Alta taxa de erros:** *a priori* a combinação entre baixa conectividade, alta mobilidade e sinal fraco são fatores que agravam a taxa de erros, o que por sua vez gera baixa confiabilidade na transmissão neste meio.

- **Atraso longo e variável:** a conectividade intermitente presente nas DTNs dificulta a estimação em relação ao atraso. Os dados a serem transmitidos geralmente necessitam ser mantidos em memória, um espaço dedicado ao armazenamento das mensagens, para posterior envio, o que pode demorar minutos, horas ou dias, dependendo da mobilidade e conectividade da rede.
- **Taxas assimétricas de transmissão de dados:** devido às características intermitentes de comunicação e de ambiente, os *links* entre os nós de uma DTN poderão apresentar taxas de transmissão assimétricas, ou seja, as taxas de transmissão de dados para o envio e o recebimento podem não ser as mesmas.
- **Restrições de energia, largura de banda e outros:** assim como nas redes sem fio *ad hoc*, as DTNs muitas vezes apresentam limites em relação à energia disponível, se tornando um recurso valioso para o funcionamento da rede. Técnicas de economia de energia podem ser empregadas neste ambiente para a maximização dos recursos, por exemplo, desligando o rádio ou ainda diminuindo sua potência. Outro fator desafiador para as DTNs são os gargalos em relação à largura de banda disponível, muitas vezes, além do baixo tempo de conexão entre os nós, os dados são transferidos em uma taxa muito baixa, na ordem de 16kbps. É necessário levar em consideração também a limitação, quase sempre presente, em relação ao espaço de armazenamento das mensagens (o *buffer*). E, esse *buffer* tende a se tornar ainda mais restrito, uma vez que o intervalo entre os contatos é potencialmente longo.

Na literatura os cenários abordados pelo modelo DTN se encaixam nos conceitos de Redes Desafiadoras (*Challenged Networks*), assim como Kevin Fall em [15] as definiu. O conceito de DTN surge a partir da aplicação da Internet sobre uma rede interplanetária (*Inter-Planetary Network* - IPN). Nesse cenário as características da DTN podem ser muito bem visualizadas, uma vez que apresenta atraso na propagação de sinal e uma significativa taxa de erros, além de haver conectividade de forma ocasional e taxa de transmissão assimétrica.

De acordo com Durst *et. al* [13], não é possível a utilização de protocolos modelados para a arquitetura atual da Internet em uma IPN hipotética. Para tanto são analisadas as performances do protocolo de transporte TCP (*Transmission Control Protocol*), de aplicações de transferência de arquivos (FTP - *File Transfer Protocol*) e de correio eletrônico (SMTP - *Simple Mail Transfer Protocol*). Os autores ponderam, inclusive, a adequação do protocolo TCP para operar em cenários com atrasos fim-a-fim

em torno de 9 horas, no entanto concluem que o uso da arquitetura da Internet em cenários inadequados implica em um desempenho insatisfatório, o que inviabilizaria a operação em uma Internet Interplanetária.

Neste capítulo serão abordados, de forma geral, os principais conceitos da arquitetura DTN, definições e tipos de contato previstos.

2.1 Arquitetura

As DTNs são consideradas redes especiais no sentido de apresentarem características peculiarmente particulares que a distinguem dos demais modelos. Estas principais características são compreendidas por longos e variáveis atrasos, frequentes desconexões, presença de mobilidade, taxa assimétrica de transmissão, altas taxas de erros, suporte à não existência de um caminho fim-a-fim, dentre outras.

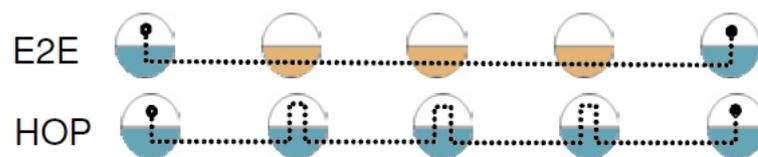


Figura 2.1: Encaminhamento fim-a-fim e salto a salto. Baseado em [11].

Como recorrentemente DTNs não apresentam um caminho fim-a-fim disponível, é utilizada uma nova arquitetura de armazenamento persistente e comutação de mensagens denominada *store-carry-and-forward* na forma de um protocolo adaptado a atender às necessidades destes ambientes desafiadores. Este modelo permite que um nó carregue uma mensagem cujo destino seja um nó fora de seu alcance. Sendo assim, é necessário levar em consideração também o espaço de armazenamento disponível a estes nós.

A Figura 2.1 ilustra como uma mensagem é encaminhada em ambientes de redes com conexão fim-a-fim entre a origem e o destino (representada por E2E - *end-to-end*), e também a comutação de mensagem sobre uma rede salto a salto (HOP) através de nós intermediários. Como já citado, as redes DTNs utilizam-se de contatos intermitentes para realizar o encaminhamento das mensagens em contraste à semântica fim-a-fim presente no protocolo TCP, ocorrendo, portanto, salto a salto.

2.2 Camada de agregação

Para suprir as necessidades presentes nas DTNs, em 2002, o *Internet Research Task Force* (IRTF) [22], comunidade que se dedica à pesquisa sobre o avanço tecnológico da Internet, criou um grupo dedicado à pesquisa das DTNs, chamado de *Delay Tolerant Network Research Group* (DTNRG) [12]. Por meio deste grupo instituiu-se que haveria a necessidade de uma sobrecamada, permanecendo entre a camada de aplicação e a camada de transporte, chamada de camada de agregação (*Bundle Layer*). Este modelo de camadas está presente em todos os nós pertencentes à DTN e pode ser visualizado na Figura 2.2. Esta nova camada tem como função armazenar os dados de forma persistente e comutar mensagens. A intenção da localização da camada de agregação acima da camada de transporte é manter uma interoperabilidade com qualquer tipo de rede [15]. Fica claro a partir da Figura 2.2 que em uma rede DTN serão utilizados os serviços das camadas inferiores como atualmente na Internet, como pode ser visto no exemplo com a utilização do TCP/IP. Para maiores informações sobre o funcionamento e utilização dos padrões da camada de agregação, indicamos a leitura da RFC 5050 (*Bundle Protocol Specification*) [49].

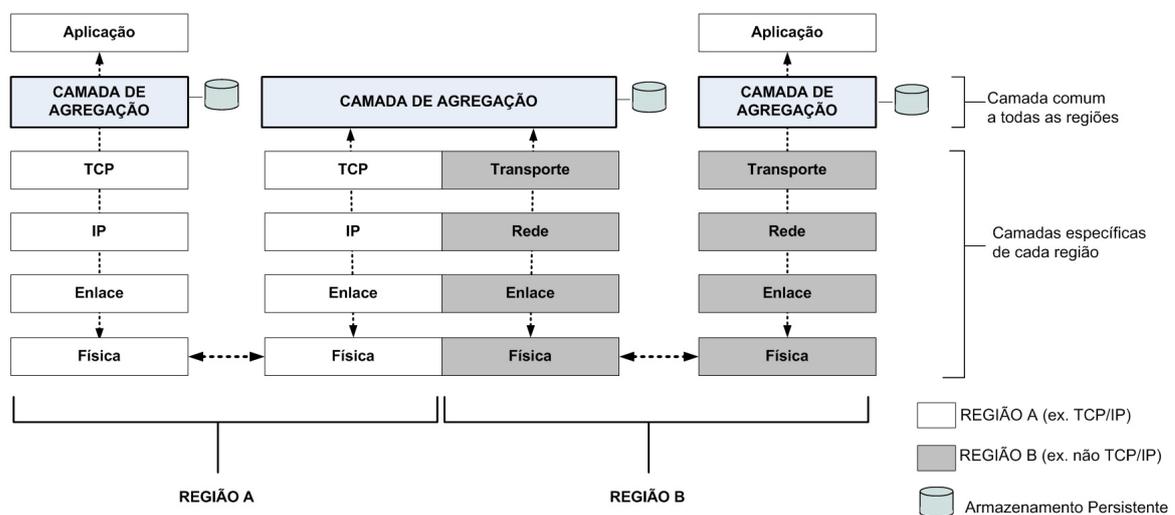


Figura 2.2: Camada de Agregação [43]

Considerado a principal contribuição da arquitetura DTN, o conceito de armazenamento persistente auxilia na diminuição da perda de dados durante interrupções na conexão entre os nós além de incluir a transferência salto a salto com entrega confiável de dados. Várias outras características de gerenciamento são abordadas pelo Protocolo de Agregação (BP - *Bundle Protocol*) [49].

2.3 Tipos de contato e aplicações

O contato é conceituado por Cerf *et. al* [9] como a oportunidade de dois nós se comunicarem. Contatos se classificam em função da previsibilidade e se alguma ação é necessária para que um contato possa ocorrer. Como adição, em [15] é definido que contatos podem ser parametrizados pelo tempo de início e término (relativos ao nó DTN fonte), sua capacidade, latência e direção. Todas as várias aplicações existentes em DTN se encaixam sob, ao menos, um tipo de contato. A fim de elucidar os tipos de contatos existentes juntamente com a sua aplicação em um ambiente real, serão abordadas as seguintes aplicações: vila remota, o monitoramento de ambientes e a comunicação espacial. A exposição dessas aplicações tem como finalidade exemplificar alguns dos possíveis cenários onde a arquitetura DTN pode ser empregada.

2.3.1 Vila remota

O cenário de Vila Remota aqui apresentado tem como objetivo principal oferecer suporte ao acesso à Internet com baixo custo em áreas urbanas e, principalmente, em comunidades de áreas remotas que, seja por falta de cobertura de outras tecnologias, questões econômicas, ou ainda por questões geográficas não podem usufruir desse acesso.

Neste cenário DTN, são utilizadas mulas de dados (pode-se representá-las, por exemplo, por ônibus equipados com um dispositivo de comunicação sem fio e com grande capacidade de armazenamento) que se deslocam por um trajeto previamente estabelecido. Ao encontrar outros dispositivos de comunicação sem fio sob sua área de cobertura é possível enviar ou receber mensagens entre si. Este mesmo processo se repete até o momento que a mula de dados encontra seu “gateway” de acesso à Internet. Nesse cenário, existe previsibilidade em relação ao instante de tempo em que o contato ocorrerá entre o dispositivo sem fio e o ônibus, sendo assim, é possível estimar uma relação entre a capacidade de transmissão, o tempo de contato e a capacidade de armazenamento.

A seguir é descrito com maior ênfase o tipo de contato que ocorre no cenário de Vila Remota.

Contatos previsíveis

Com base em históricos de contato é possível que se faça previsões sobre a realização de um futuro encontro entre os nós assim como seu provável tempo de duração de contato. Esta estimativa não garante que o contato realmente ocorrerá da mesma maneira historicamente registrada, porém, serve como auxílio ao conhecimento sobre um possível momento de contato. Pode se utilizar um certo grau de confiabilidade de estimativas com base nas informações de encontros colhidas de tal forma que haja um nível maior de acerto nestas previsões.

Um exemplo que ilustra de forma clara este tipo de contato são as redes rurais, exploradas por vários trabalhos na área de DTN, como [51] [47] [16] [18]. Abordaremos o cenário exposto na Figura 2.3 que demonstra o funcionamento da DTN do projeto DakNet [47]. Para o encaminhamento a longas distâncias e com grande atraso entre vilas localizadas remotamente e um ponto central de acesso à Internet localizado, por exemplo, em uma cidade, foi-se desenvolvido o projeto DakNet. Neste projeto são instalados quiosques nas vilas remotas que têm a função de armazenar as requisições de serviços com a Internet. Estes quiosques realizam conexões ponto-a-ponto de curta distância com os Pontos de Acesso Móveis (MAPs - *Mobile Access Points*). MAPs podem ser bicicletas, motos, ônibus e carregarão os dispositivos de comunicação sem fio para conexão com os quiosques em sua rota. Portanto, o MAP armazenará a requisição, levará até o ponto central e retornará com a resposta ao pedido para a vila remota em questão.



Figura 2.3: Exemplo de transmissão em vilas remotas. Baseado em [47].

Como complemento, a Tabela 2.1 expõe alguns trabalhos relacionados a cenários que utilizam mulas de dados para a comunicação e merecem destaque. São expostos em (a) os projetos DakNet [47], KioskNet [18] e MotoPost [39] que têm a finalidade de introdução à comunicação digital em áreas rurais. Em (b) dois projetos são expostos em áreas diferentes às citadas em (a), onde WiderNet [38] visa auxiliar na educação em áreas remotas levando e distribuindo conteúdo de sites educacionais, já o projeto TrainNet [61] tem o objetivo de transportar por longas distâncias dados que não necessitam ser enviados em tempo-real. A tabela consiste do nome do projeto, os tipos de nós que compõem a DTN, a variação do tempo de atraso entre os contatos e o trajeto realizado pela mula de dados.

Tabela 2.1: Aplicações DTN.

Comunicação digital para áreas rurais			
Projeto	DakNet	KioskNet	MotoPost
Nós	Motos, carroças, quiosques etc.	Palm conduzido por pessoas, quiosques, <i>desktop</i> conectados via <i>dial-up</i> , Ponto de Acesso	
Atraso	Minutos ou horas	Horas ou dias	Horas
Trajeto	Semi-aleatório, dependendo do transporte utilizado	Conforme o trajeto dos ônibus ou aleatoriamente quando carregados por pessoas	

(a)

Auxílio na área educacional (África, Índia, Bangladesh e Haiti)		Transporte massivo de dados a longas distâncias
Projeto	WiderNet	TrainNet
Nós	Computadores com grande espaço de armazenamento para armazenar sites educacionais	Trens e estações
Atraso	Dias ou meses	Minutos ou horas
Trajeto	Semi-aleatório, visitando os locais sob-demanda	Fixo, com pontos ao longo da linha férrea

(b)

2.3.2 Monitoramento de ambiente

O monitoramento de ambientes pode ser compreendido desde a análise dos hábitos de animais em parques de preservação a situações de desastres naturais e ambientes hostis gerados por conflitos, como guerras. Muitas vezes a comunicação de dados inexistirá nessas regiões pela ausência de infraestrutura e/ou cobertura. É assim quando pensamos no monitoramento de regiões silvestres não civilizadas, onde, geralmente, não há cobertura por nenhum meio de comunicação com outras regiões (não existindo uma infraestrutura de fibra-ótica, cobertura de satélites etc).

São vários os projetos relacionados a esse tipo de aplicação, merecendo destaque:

- **CARNIVORE** (*Carnivore Adaptive Research Network in Varied Outdoor Remote Environments*) [48]: surge sob a necessidade de observar os hábitos e interações entre predadores, entre predadores e o ecossistema e entre esses e a população. Nesse projeto foi criada uma rede de sensores sem fio (WSN - *Wireless Sensor Network*) por meio de colares presos ao pescoço de coiotes na região da cidade de Santa Cruz, Califórnia. Dados são coletados com o intuito de monitorar a fisiologia e o comportamento dos animais. A topologia da rede é composta de nós móveis, representados pelos predadores, e nós fixos, equivalentes às estações rádio-base. Uma vez recebidos os dados dos predadores pelas estações rádio-base, essas são responsáveis por encaminhá-los para a Internet. Nesse cenário fica clara a interrupção de conexão entre os nós envolvidos, assim como sua provável topologia esparsa. A fim de suportar os problemas de desconexão recorrentes, todos os contatos, seja entre predadores ou entre predadores e antenas, são aproveitados para propagar os dados coletados. Ou seja, conexões de um único e múltiplos saltos de modo *ad hoc* com estabelecimento de conexões apenas entre os pares.
- **ZebraNet**: sucintamente, este projeto monitora o comportamento de Zebras em uma reserva ambiental, armazenando suas informações de localização em um colar que serão posteriormente enviadas a uma estação-base, diretamente ou não. Este projeto foi escolhido como um de nossos cenários para experimento e será detalhado na sub-seção 3.1.2.

Cenários sob a premissa de monitoramento de ambientes utilizam, portanto, a forma de contatos oportunistas para gerir sua comunicação e a seguir este tipo de contato é descrito.

Contatos oportunistas

Conforme apresentado em [7], os nós pertencentes à DTN podem necessitar aproveitar um momento de contato não previsto para se comunicar. Neste caso, esse tipo de contato será chamado de contato oportunista, onde, diferentemente dos contatos previsíveis, transmissões poderão ser realizadas durante um contato ocasional. Este tipo de contato se mostra uma das áreas mais interessantes de desenvolvimento em DTN, onde vários protocolos de comunicação são propostos pela literatura¹. O cenário propício a este tipo de contato pode ser composto por: pessoas, veículos, aviões, animais ou satélites em movimento. A partir do momento em que há visada, energia e alcance de rádio a comunicação pode ser estabelecida.

Um exemplo exposto em [7] e que explicita de forma muito compreensível tal modelo de contato diz respeito à comunicação interpessoal. Quando uma pessoa deseja falar com outra, espera o momento até que a encontre para realizar esta tarefa. Assim que esta pessoa encontra seu destinatário ela então transmite o que deseja. Ou seja, oportunisticamente uma pessoa, que faz o papel da origem pôde transmitir sua informação à outra, que desempenha o papel de destinatário.

Outro exemplo, exposto em [43] cita a comunicação entre dispositivos portáteis através do paradigma de comunicação entre usuários, denominado *Pocket Switched Networks* (PSNs) [21] [33]. Nesse trabalho, é considerado um novo modelo de redes que atua dentro do contexto de DTN, pois realiza a comunicação na ausência de conectividade fim-a-fim, obtendo vantagem de qualquer oportunidade de transmissão ao longo do trajeto do dispositivo móvel para que o encaminhamento das mensagens seja realizado até o destino.

A Figura 2.4 demonstra um exemplo para contatos oportunistas. Dividido em três quadros, pode-se acompanhar, da esquerda para a direita, a movimentação dos nós que compõem tal ambiente em relação ao tempo. Os nós desta rede estão representados por dispositivos móveis.

No exemplo da Figura 2.4, a fonte, identificada pelo dispositivo “Fonte”, deseja enviar uma mensagem ao dispositivo identificado por “Destino”. Como não há um caminho completo estabelecido entre os nós “Fonte” e “Destino”, a mensagem deverá ser encaminhada salto-a-salto entre os nós da rede para que essa chegue a seu destino.

¹Protocolos de roteamento que objetivam maximizar diversas métricas em cenários cujo os contatos são oportunistas, serão abordados por este trabalho.

Sendo assim, o nó fonte, ao encontrar o nó 1, encaminha sua mensagem a este, que por sua vez, ao encontrar o nó 3 repassa a mensagem original. O nó 3, ao se locomover, encontra o nó de “Destino” e então repassa a mensagem, entregando-a a seu destino final. Assim, aproveitando de forma oportunista os encontros ocorridos neste ambiente, a mensagem pôde ser enviada do nó “Fonte” ao nó de “Destino”.

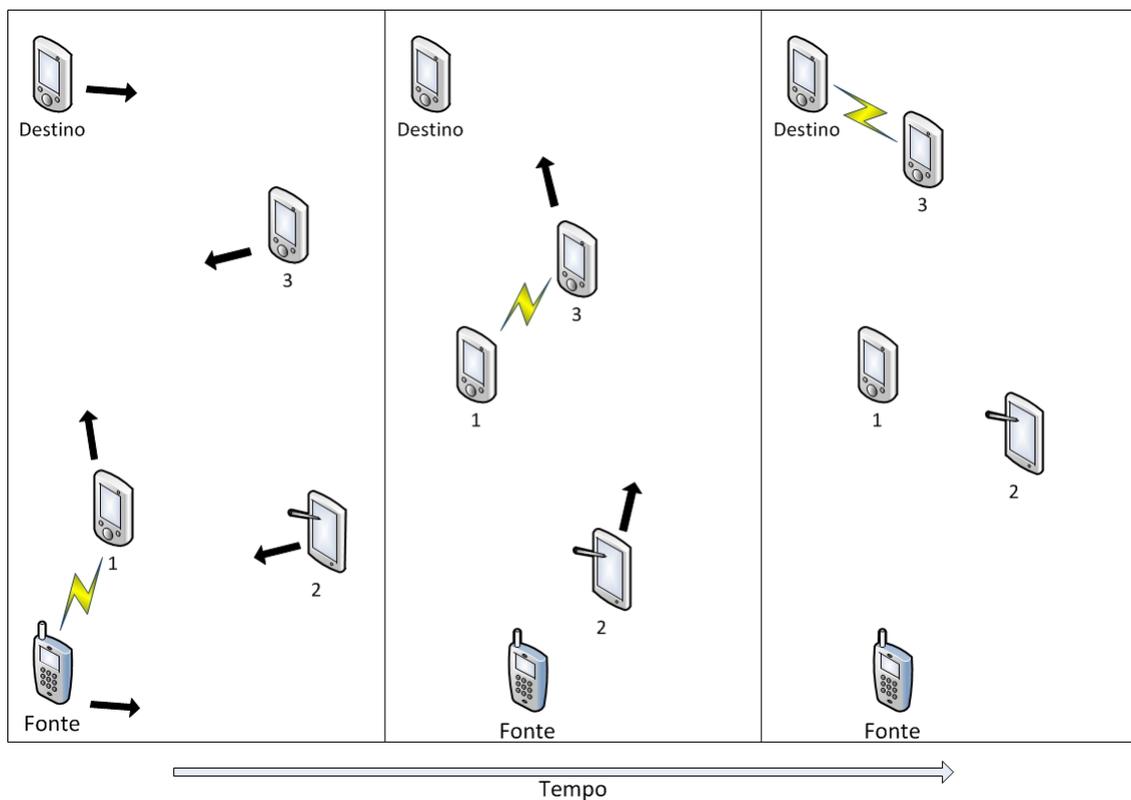


Figura 2.4: Exemplo de contatos oportunistas.

Portanto, esse tipo de contato acontecerá conforme sua disponibilidade, e poderá ter o auxílio de uma previsão anterior à sua ocorrência baseado em um histórico de encontros.

2.3.3 Comunicação espacial

A comunicação espacial, resultado da comunicação por redes interplanetárias, apresenta características particulares em relação a vários fatores, como: latência na propagação do sinal, baixa taxa de transmissão, conectividade programada e possivelmente intermitente devido a interposição de outros corpos estelares (planetas, asteróides, satélites etc.) entre o transmissor e o receptor, além de outras características [7].

Esse cenário descreve com clareza o conceito por trás das DTNs, onde é necessário o preenchimento do requisito relativo à persistência das mensagens que deverão ser

encaminhadas, uma vez que podem ocorrer atrasos na ordem de horas ou dias. Para garantir a comunicação, o *Consultative Committee for Space Data Systems* (CCSDS) [53] desenvolveu, a partir da arquitetura TCP/IP, um conjunto de protocolos denominado *Space Communication Protocol Standards* (SCPS) [50] que inclui um protocolo de aplicação para transmissão de arquivos (*SCPS File Protocol*), um protocolo para controle da transmissão (*SCPS Transport Protocol*), um protocolo de roteamento (*SCPS Network Protocol*), entre outros protocolos.

Este cenário possui as características do tipo de contato programado que é descrito a seguir.

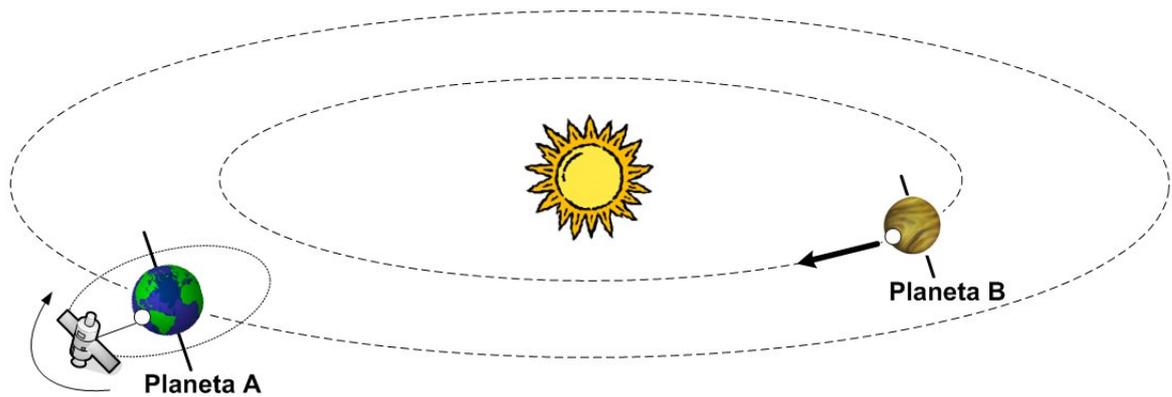
Contatos programados

Contatos programados são aqueles onde o contato é determinístico, previamente definido para certo momento. Este tipo de contato exige que os nós participantes estejam em sincronia de tempo de rede. Nos tipos de rede que utilizam este modelo de contato é mais simples saber quanto os nós poderão transmitir durante seu tempo de contato do que nos modelos já descritos (a saber: contatos oportunistas e previsíveis), o que facilita muito o dimensionamento da carga e atraso da rede [7].

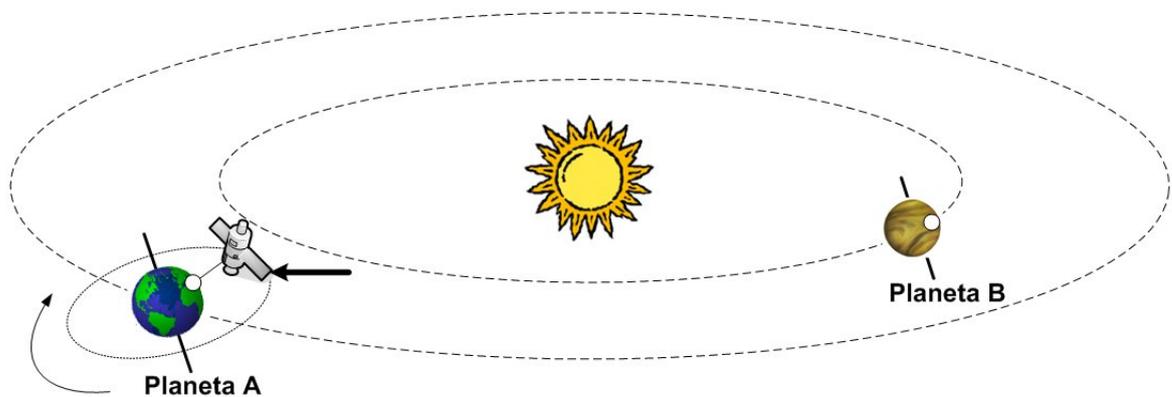
O exemplo de uma rede de sensores onde seus nós “dormem” e “acordam” em horários programados para poupar energia é simples e ajuda na caracterização deste modelo. Outro exemplo que deve ser citado, e pode ser mais claramente visualizado com a exposição da Figura 2.5, caracteriza um ambiente de comunicação interplanetária, onde nós localizados nos planetas A e B desejam se comunicar, porém não existe uma conexão persistente entre os dois planetas. O que existe é apenas a conexão persistente do nó residente no planeta A com seu satélite. O nó do planeta B, que possui uma informação previamente armazenada em seu *bundle*, deseja enviá-la ao planeta A, porém, conforme a Figura 2.5(a) o nó do planeta B não possui uma conexão persistente com o satélite do planeta A. No momento programado em que há visada entre o nó do planeta B e o satélite do planeta A, a mensagem é então enviada com sucesso, conforme ilustra a Figura 2.5(b).

2.3.4 Demais tipos de contatos

A título de complementação a esta seção, são expostos dois tipos de contatos abordados em DTN. São eles: contatos persistentes e contatos sob demanda. E, serão de-



(a) O Planeta B não tem visada direta com satélite do Planeta A no momento do envio da mensagem.



(b) A mensagem chega ao satélite do Planeta A exatamente no momento da visada direta.

Figura 2.5: Exemplo de Contatos Programados [43]

scritos logo a seguir:

Contatos persistentes

O conceito de contato persistente é facilmente compreendido uma vez que segue o modelo mais difundido atualmente. Nesse modelo a conexão está sempre disponível, independentemente de sua utilização ou não. O exemplo clássico deste tipo de contato é a conexão do tipo DSL (*Digital Subscriber Line*).

Contatos sob demanda

Este tipo de contato é definido por ter funcionalidade idêntica ao apresentado anteriormente, o contato persistente, porém sob a necessidade de uma ação prévia para

efetivação desta persistência. O contato sob demanda será considerado persistente até o momento em que haja sua finalização.

Um exemplo que deixa claro este cenário é a conexão discada (*dial-up*) realizada por um usuário em sua casa. Até certo momento sua linha de telefone se encontrava livre para utilização convencional, porém a partir de um momento este usuário realizou uma conexão com seu provedor de acesso à Internet (ISP - *Internet Service Provider*). Após o usuário ter demonstrado necessidade (demanda) de uma conexão com a Internet ele assim a fez e pôde usufruir do mesmo modelo encontrado no contato persistente até que desejasse se desconectar.

Outro exemplo que ilustra bem a questão da demanda pode ser ilustrado por uma rede de sensores onde em certo se deseja enviar uma mensagem a todos os sensores e então é enviada uma mensagem para “acordá-los”. Ou seja, somente quando houve a necessidade (demanda) os sensores foram despertados.

2.4 Conclusão

Esse capítulo teve o intuito não só de abordar diretamente a motivação deste trabalho, mas também expor e embasar o leitor sobre a tecnologia envolvida nas DTNs. A partir de então, é possível iniciarmos uma exposição dos cenários escolhidos como base de mobilidade para as experimentações.

Capítulo 3

Cenários Abordados

São apresentadas neste capítulo as principais características dos cenários selecionados, assim como detalhes em relação ao seu padrão de mobilidade. A seleção destes cenários objetiva encontrar a relação de interferência existente entre suas características particulares e o desempenho dos protocolos de roteamento para DTNs. Para tanto, foram selecionados três cenários com características contrastantes, onde dois desses foram criados a partir de registros de movimentação reais e o terceiro produzido artificialmente, porém com características que modelam a movimentação de nós sobre um mapa real.

3.1 Introdução

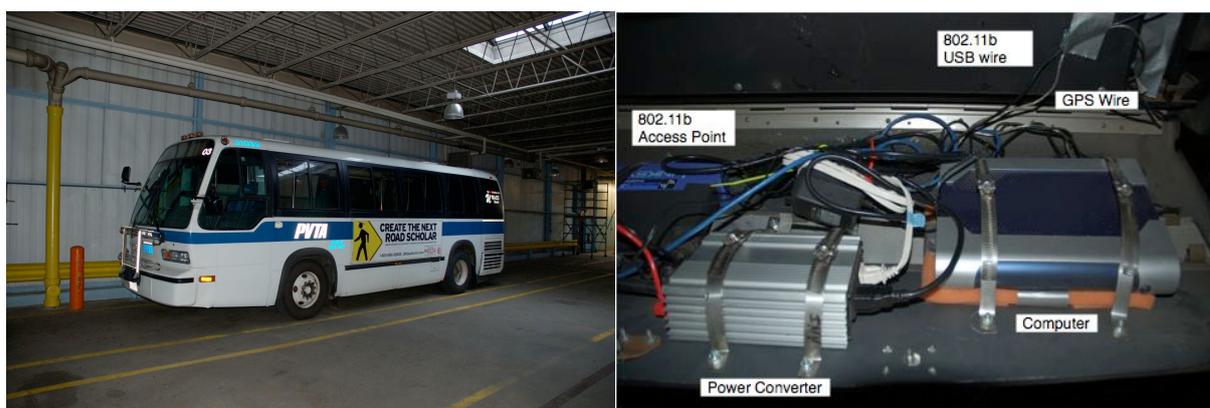
A movimentação dos nós da rede é essencial para a avaliação do desempenho de um protocolo em DTN. O primeiro cenário descrito será o UmassDieselNet [3], onde ocorrem conexões entre ônibus e entre ônibus e estações-base (pontos fixos) em um ambiente real. Já o segundo cenário objetiva avaliar um ambiente de monitoramento em uma reserva ambiental, com movimentações bem particulares, e é denominado ZebraNet [58]. O terceiro, e último ambiente, se denomina WDM (*Working Day Movement*) e é baseado no modelo de movimentação com mesmo nome proposto em [14] e possui a representação do centro da cidade de Helsinki, na Finlândia, com seus respectivos atores (ônibus, carros e pedestres).

3.1.1 UmassDieselNet

O projeto intitulado DieselNet é parte de um *testbed* chamado de GENI (*Global Environment for Network Innovations*), e encontra-se aberto a experimentos de pesquisadores

externos. Constituído de 35 ônibus conectados entre si e a estações-base fixas, utiliza o padrão IEEE 802.11b e conduz seus experimentos na cidade de Amherst, nos EUA. É descrito que cada ônibus possui o que é chamado de *Diesel Brick* que é baseado na arquitetura P6 com 577Mhz de CPU, 256MB RAM, 40GB de disco rígido e sistema operacional Linux. Essa *brick* é conectada a três rádios onde o primeiro é um ponto de acesso que provê acesso via DHCP aos passageiros e a quem mais estiver a seu alcance, o segundo é uma interface USB (*Universal Serial Bus*) que constantemente varre o espectro em busca de ofertas DHCP (*Dynamic Host Configuration Protocol*) a fim de armazenar estes dados como representação da ocorrência de encontros com outros nós na rede; já o terceiro rádio, esse de longo alcance, tem o intuito de se comunicar com a estação-base. Nos arquivos de atividades armazenados por este projeto ainda são coletados dados de localização geográfica, por meio de GPS (*Global Positioning System*).

As Figuras 3.1a e 3.1b mostram, respectivamente um dos ônibus que compõem o *testbed* do Umass DieselNet e o *Diesel Brick*.



(a) Ônibus

(b) Diesel Brick

Figura 3.1: Componentes do projeto Umass DieselNet.

Vários são os trabalhos que já utilizaram os dados coletados neste *testbed* para validação de seus resultados, com destaque na avaliação dos protocolos de roteamento para DTN, MORA [8] e RAPID [1], além de diversos outros trabalhos como [28, 2, 5].

3.1.2 ZebraNet

O projeto ZebraNet [26], idealizado e realizado pela Universidade de Princeton, nos Estados Unidos, tem com principal objetivo o monitoramento do comportamento de zebras na reserva ambiental de *Sweetwaters Game Reserve*, em Nanyuki, Quênia.

Nesse projeto são utilizados nós sensores sem fio em colares colocados nas zebras e que, periodicamente, coletam dados sobre a localização desses animais por meio de um GPS em formato UTM (*Universal Transverse Mercator*) e os armazena em memória não volátil.

Na Figura 3.3 é possível visualizar a composição de cada colar das zebras. Nele há uma placa GPS-MS1E que possui um GPS, uma CPU e uma área de armazenamento representada por uma memória *flash*. Há ainda um rádio de curto alcance, responsável pela comunicação entre as zebras e a estação-base móvel. Na imagem não são exibidos o painel solar, as baterias e o circuito de ligação desses componentes. Maiores informações sobre a parte técnica deste projeto podem ser encontradas em [26].

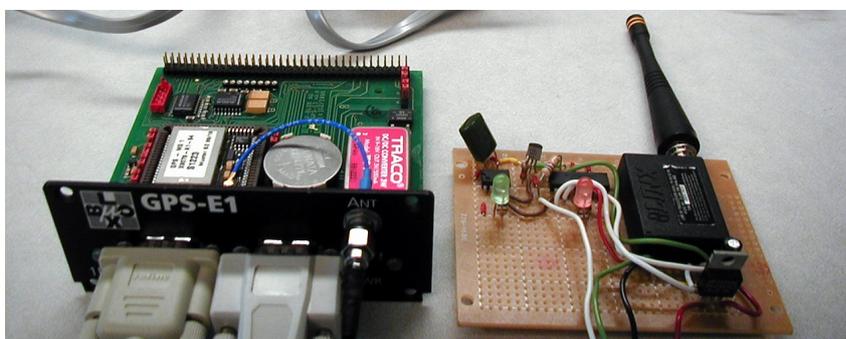


Figura 3.2: Composição do colar de cada Zebra.

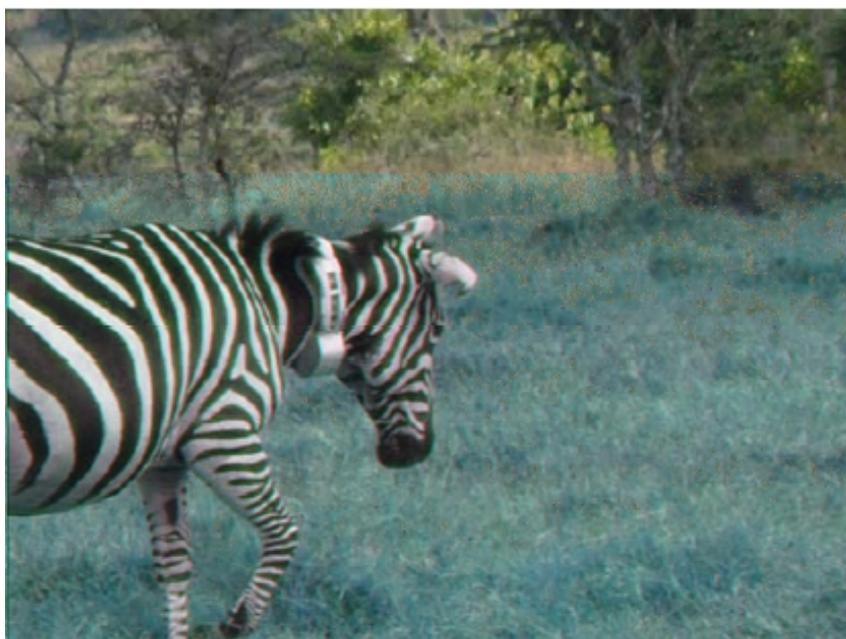


Figura 3.3: Uma das zebras do projeto ZebraNet.

Devido à inexistência de cobertura por qualquer tipo de serviço de comunicação na região, as informações são armazenadas persistentemente nos colares, para que,

em uma oportunidade propícia, sejam enviadas para uma estação-base, que também é móvel. Seguindo a abordagem do projeto, a disponibilidade dessa estação-base móvel é esporádica, pois ela corresponde a um carro conduzido pelos pesquisadores que recebem os dados enquanto o veículo está em movimento pela região.

Um fator relevante diz respeito também à cobertura dessa estação-base móvel com as zebras, já que é natural que nem sempre uma estação-base consiga cobrir todas as zebras, sendo assim, conexões sem fio entre as zebras devem ser utilizadas. Esse tipo de conexão é conhecida como conexão sem fio *ad hoc* e possibilita que os dados sejam encaminhados através dos nós (representados pelas zebras, neste caso) até a estação-base, salto-a-salto. Dessa forma é ampliada a possibilidade de que um número maior de informações chegue até a estação-base.

3.1.3 Working Day Movement (WDM)

Este modelo surge a partir da motivação de uma contraproposta aos modelos de movimentação abstratos, como *Random Waypoint* (RWP) [24], que não reproduzem de forma confiável as propriedades de movimentação em cenários reais [60]. A introdução do WDM [14] tem como foco a aplicação de um modelo mais próximo à movimentação real de pessoas, ônibus e carros, que possa ser utilizado em simulações em DTNs. Nesse modelo são realizadas distribuições coerentes tanto em relação ao tempo entre contatos quanto à duração destes.

Este modelo utiliza sub-modelos propostos em outros trabalhos na área de mobilidade, sendo, por base, utilizadas três atividades básicas do dia-a-dia de uma pessoa: quando se está em casa, no trabalho ou realizando alguma atividade a noite com amigos [14]. Em cada uma dessas atividades sub-tarefas são realizadas, como pode ser observado por meio da descrição de dois submodelos de atividades destacados a seguir:

- **Submodelo de atividades em casa:** cada nó tem um ponto no mapa referente a sua casa, a ser visitado ao final do dia. Chegando a esse local, o nó anda a uma curta distância e ali permanece até a hora de despertar. Neste caso atividades de movimentação dentro da casa não são consideradas relevantes e, portanto, desconsideradas.
- **Submodelo de atividades de transporte:** ao se locomover durante as atividades diárias as pessoas podem utilizar um dos três meios de transporte suportados,

que são: a pé, de carro ou de ônibus. Cada modelo terá seu trajeto definido por um mapa em formato WKT (*Well Known Text*) – este tipo de arquivo pode ser gerado/editado a partir de dados de mapas reais usando programas GIS (*Geographic Information System*) como o OpenJUMP¹ – e por seu POIs (*Points of Interest* - Pontos de Interesse), que são equivalentes a atrações turísticas, *shoppings* ou restaurantes.

Os mesmos autores desta proposta participaram do desenvolvimento do simulador utilizado neste trabalho, o The ONE [31], e como validação compararam seu modelo proposto (WDM) ao modelo RWP e a *traces* de movimentação reais retirados do trabalho realizado em Dartmouth por [52].

É possível visualizar nas Figuras 3.4 e 3.5 a confiabilidade claramente superior à utilização deste modelo em relação à modelos de movimentação abstrata e randômica. Na Figura 3.4 é possível avaliar a discrepância em relação aos contatos para o modelo RWP em relação ao modelo real Dartmouth e a proximidade com a realidade alcançada pelo WDM. Já na Figura 3.5 é exibido o gráfico de dispersão para os contatos únicos em relação ao contatos totais da simulação, onde é possível mensurar o grau de heterogeneidade de encontros gerados no modelo real reproduzido pelo WDM em relação à simplicidade do RWP.

Apesar de não ser o foco deste trabalho, a abordagem dos principais conceitos envolvidos sobre o modelo de movimentação WDM visa justificar sua escolha como cenário de experimentos.

3.2 Configurações de cada cenário

As características constantes a cada cenário são descritas nesta seção juntamente com a Tabela 3.1 que visa facilitar a visualização global da composição dos ambientes de experimentos.

O cenário UmassDieselNet utiliza os dados de conexão relatados pelos *logs* entre outubro e novembro de 2007 sobre a DTN de mesmo nome, e podem ser encontrados em [4]. Dentre os arquivos de movimentação disponíveis na base, foram selecionados aqueles referentes ao registro de contato somente entre ônibus, e descartados os contatos entre ônibus e as estações rádio-base. Foi realizada essa filtragem a fim de que

¹<http://openjump.org/>

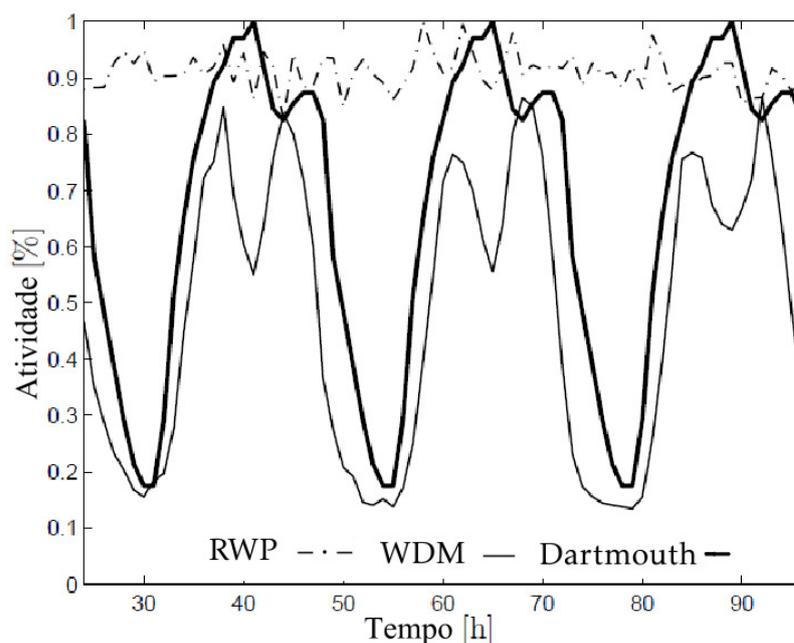
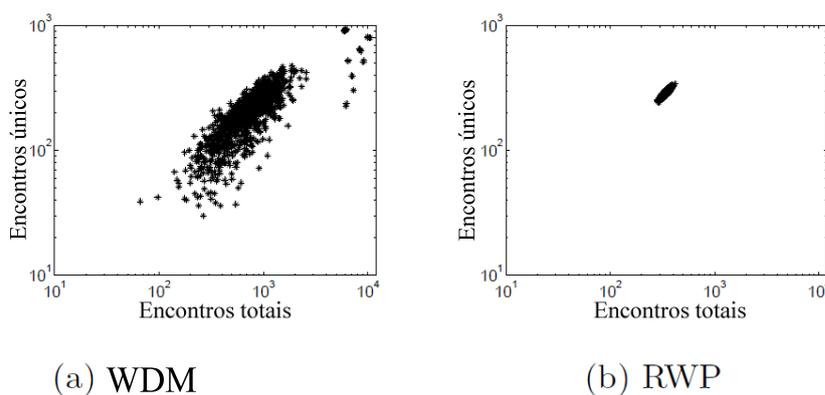


Figura 3.4: Contatos por hora (normalizado) para RWP, WDM e Dartmouth. Baseado em [14].



(a) WDM

(b) RWP

Figura 3.5: Encontros totais versus encontros únicos. Nós representados por pontos. Baseado em [14].

fossem consideradas somente conexões sem fio *ad hoc* entre os nós móveis da rede. Foi selecionado um período de aproximadamente uma semana e realizada sua conversão para o formato de entrada compreendido pelo simulador The ONE. Seu raio de alcance não é relatado na Tabela 3.1 uma vez que os nós obedeciam às informações referentes a existência ou não de contato, como relatado pelo arquivo fornecido na base CRAW-DAD [4].

O segundo cenário, referenciado neste trabalho como ZebraNet, constitui-se de movimentos reais de zebras em uma reserva ambiental (como descrito anteriormente) e para tanto foram utilizados arquivos de movimentação da base de dados CRAW-DAD

disponíveis em [59], coletados em Janeiro de 2004. Os arquivos de movimentação disponibilizados também foram convertidos para que pudessem ser interpretados pelo simulador The ONE. Foram gerados cenários com 10, 20, 30 e 40 nós em uma duração de 45 minutos de simulação sobre a área que respeitasse a equivalência à reserva, de 100km^2 . Como os dados desses arquivos de registro referenciavam a localização geográfica das zebras, foi possível simular a variação do raio de alcance do rádio.

No cenário WDM foi utilizado como base para o deslocamento dos nós o mapa da área central da cidade de Helsinki, na Finlândia, assim como realizado pelo trabalho original deste modelo [14]. Já o padrão de mobilidade dos nós tenciona representar o comportamento de pedestres, carros e ônibus. A proporção de cada um dos personagens deste cenário foi escolhido de forma arbitrária, e apresenta uma distribuição quantitativa na proporção de, 50% de pedestres, 30% de carros e 20% de ônibus. As simulações representam as atividades relativas a seis horas reais de um dia. Vale ressaltar neste momento que a escolha deste período de simulação ocorreu devido à viabilidade de análise e obtenção de resultados em respeito ao tempo hábil para o desenvolvimento deste trabalho uma vez que as taxas de processamento do simulador se tornavam cada vez mais lentas conforme a quantidade de nós, raio e mensagens aumentava. Isso se deve, principalmente, pela complexidade do modelo de movimentação utilizado neste cenário, o WDM.

Para todos os cenários foram gerados cinco arquivos de tráfego distintos com o objetivo de inserir maior confiança aos resultados obtidos. Foi estipulado para cada mensagem o tamanho de 10KBytes e fixada a taxa de transmissão em 1Mbps.

Tabela 3.1: Características dos cenários e dos contextos

Cenário	Características dos cenários			Características dos contextos		
	Quantidade de Nós	Duração	Dimensão (km^2)	Raio de Alcance (m)	Quantidade de Mensagens	Capacidade de Armazenamento
UmassDieselNet	15 e 31	7 dias e 20 horas	241,40	–	50, 250,	500 KB, 2.5 MB,
ZebraNet	10, 20, 30 e 40	45 minutos	100	50, 150 e 250	500 e	5 MB e
WDM	75 e 150	6 horas	15,30	50, 150 e 250	1.000	10 MB

3.3 Conclusão

Após a exposição das características e configurações de cada um dos cenários selecionados é possível expor uma classificação taxonômica e a descrição de cada um dos

protocolos de roteamento a serem avaliados neste trabalho.

Capítulo 4

Taxonomia e Protocolos de Roteamento em DTN

Este capítulo expõe primeiramente alguns trabalhos relacionados. Em seguida são descritos os conceitos de protocolos de roteamento em DTN assim como sua classificação de forma taxonômica, a fim de facilitar o entendimento e classificação dos protocolos. Juntamente à sua classificação, é descrita sua abordagem e objetivo, e ainda, ao final são listados os parâmetros de cada um dos protocolos e suas variações utilizadas nas simulações.

4.1 Trabalhos relacionados

Várias propostas de protocolos de roteamento apresentam como forma de validação a comparação do desempenho do protocolo proposto com alguns dos principais protocolos, porém respeitando-se as respectivas configurações padrão. Como é o caso dos protocolos APRP-Group [42] e EBR [40]. Em [29] encontramos o único trabalho similar à proposta de análise deste trabalho. Os autores avaliam, porém limitados a um único cenário, os protocolos PRoPHET, MaxProp e uma versão adaptada de cada um desses protocolos. É observada a relação entre a variação da duração de um contato, do intervalo entre esses contatos, e os parâmetros dos referidos protocolos. No entanto, o objetivo principal dos autores é analisar o impacto da escala de tempo e não mensurar o quanto protocolos parametrizáveis podem oscilar em relação à configuração padrão.

Um dos objetivos deste trabalho é demonstrar a necessidade de um protocolo de roteamento adaptativo para DTN, e para tanto, citamos algumas propostas da literatura que intencionam incrementar o desempenho através de algum mecanismo adap-

tativo. Em [46] os autores refinaram o protocolo NECTAR, que mescla o modelo de disseminação epidêmica com o modelo probabilístico. Durante a fase epidêmica, o NECTAR é capaz de se adaptar às características da rede e aumentar ou reduzir a permanência na referida fase. Huang *et al.* [20] adicionaram ao protocolo PROPHET quatro novas métricas de ponderação para a tomada de decisão com relação ao encaminhamento de mensagens, que são: capacidade de armazenamento, energia restante, taxa de transmissão e popularidade. Nesse trabalho é demonstrado que a ponderação dessas métricas é capaz de incrementar a taxa de mensagens entregues e reduzir a latência. Em [32] os autores demonstram que, em função da densidade da rede, da velocidade dos nós e do tamanho da mensagem, a seleção dinâmica dos protocolos de roteamento, AODV, Epidemic ou Spray and Wait, é possível incrementar o desempenho de diversas métricas.

4.2 Taxonomia dos protocolos de roteamento em DTN

O roteamento em DTNs se mostra uma das áreas mais desafiadoras neste modelo de rede, uma vez que enfrenta um maior número de limitações se comparado ao roteamento em redes tradicionais. O maior desafio enfrentado em DTN diz respeito à conectividade intermitente dos nós, além de sua dinamicidade gerar grandes alterações nos caminhos e conseqüente envelhecimento das rotas conhecidas. Os principais objetivos dos protocolos de roteamento em DTN são maximizar o número de mensagens entregues ou minimizar o atraso, preservando os recursos disponíveis na rede.

Nos trabalhos de Balasubramanian *et. al* [1] e Spyropoulos *et. al* [56], é proposta uma taxonomia para protocolos de roteamento em DTNs sob a perspectiva da criação ou não de cópias de mensagens. Onde, caso sejam criadas cópias da mensagem original, temos a classe dos protocolos conhecidos como “baseados em replicação” (*replication-based*). Porém, se não é criada nenhuma cópia da mensagem na rede, são denominados “baseados em encaminhamento” (*forwarding-based*). Uma classe adicional a essa taxonomia é apresentada por este trabalho, onde são classificados como protocolos híbridos aqueles que apresentam as duas classes citadas anteriormente como fases distintas de operação. Assim, em uma primeira fase, onde, geralmente, os recursos dos nós da rede (como o espaço de armazenamento para as mensagens – *buffer*) se encontram mais disponíveis, seu comportamento equivale ao de um protocolo “baseado em replicação”, e em uma segunda fase, onde esses recursos, de alguma forma, já não se encontram tão disponíveis, seu comportamento será equivalente ao de um protocolo

“baseado em encaminhamento”. A lógica dessa escolha se dá como forma de maximizar o aproveitamento dos recursos dos nós da rede, aumentando assim, as chances de entregar mensagens a seus destinatários.

É relevante citar que também é possível classificar os protocolos conforme seu conhecimento sobre o estado atual da rede, ou seja, caso os nós possuam informações sobre características como, por exemplo, a topologia da rede, tempo médio de encontro, e entre encontros são considerados do grupo de protocolos determinísticos. Caso não possuam informações *a priori* sobre a rede e suas características, são chamados de protocolos estocásticos. Essa classe de protocolos utilizará dados coletados durante sua execução, como a ocorrência de encontros, o tempo médio de contato etc. para determinar as características da rede.

Este trabalho introduz ambos os conceitos, porém enfoca nos protocolos estocásticos baseados em replicação e nos híbridos. É possível visualizar a taxonomia dos protocolos expostos por esta seção na Figura 4.1.

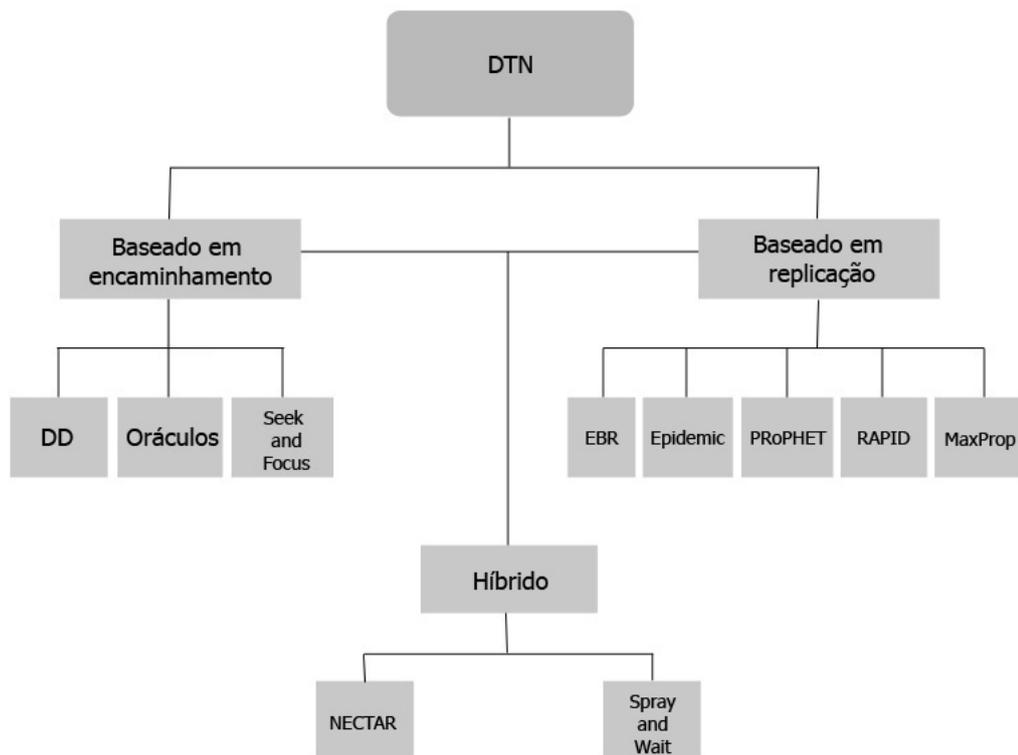


Figura 4.1: Taxonomia dos protocolos de roteamento

4.3 Protocolos baseados em encaminhamento

Como já exposto, esta classe de protocolos é determinada pela não geração de várias cópias de uma mesma mensagem.

Considerado o primeiro trabalho da classe dos protocolos baseados em encaminhando, o *Direct Delivery* (DD) é introduzido em [17] como um protocolo de funcionalidade simples, onde o nó fonte encaminhará sua mensagem somente ao nó de destino quando, e se, o encontrar. Obviamente essa abordagem gera um incremento potencial em relação ao atraso.

Em [23], os autores apresentam uma avaliação onde todas as variáveis que compõem o ambiente de uma DTN são conhecidas, o que facilita o cálculo de melhores rotas. É demonstrado como seu desempenho decai a partir do momento em que esse conhecimento se torna limitado. Os autores propuseram entidades denominadas de oráculos, onde essas entidades dispõem de informações relevantes aos nós e funcionam como bases de conhecimentos centralizadas. São divididos em: oráculos com resumo de contatos (*Contacts Summary Oracle*), com informações resumidas sobre o tempo médio de intervalo entre contatos; oráculos de contatos (*Contacts Oracle*), que além do tempo médio de intervalo entre os contatos conta também com o tempo de duração; oráculos de ocupação (*Queuing Oracle*), que têm ciência da taxa de ocupação de cada nó na rede; e, por fim o oráculo de demanda de tráfego (*Traffic Demand Oracle*), onde informações referentes às demandas de tráfego necessárias atualmente e futuramente são disponibilizadas.

Os autores comparam sua proposta com uma gama variada de algoritmos de roteamento, partindo daquele com utilização mínima de conhecimento da rede (FC - *First Contact*¹) até àqueles que fazem maior utilização dos conhecimentos dos oráculos (EDLQ - *Earliest Delivery with Local Queuing*² e EDAQ - *Earliest Delivery with All Queues*³). Ao final realizam uma comparação entre esses algoritmos com uma formulação em Programação Linear (LP - *Linear Programming*) que considera todos os oráculos e é definida como a solução ótima para os algoritmos.

Os resultados demonstram que quanto maior o conhecimento dos algoritmos ac-

¹Determina que as mensagens serão encaminhadas para um vizinho escolhido aleatoriamente, ou seja, nenhum oráculo é utilizado.

²Usa o oráculo de contatos e informações sobre tamanho das filas de transmissão dos seus enlaces locais.

³Utiliza os oráculos de contatos e de ocupação (mantendo informações globais sobre as filas de transmissão).

erca das informações da rede, mais eficiente será o algoritmo de roteamento. Porém, deve-se ressaltar que na prática não é trivial implementar o conceito de oráculo, uma vez que conexões esperadas podem não ocorrer, seja por fatores como o clima, interferência do rádio, ou algum outro fator que gere falha na comunicação [1].

Diversos outros protocolos baseados em encaminhamento foram propostos no início das pesquisas em DTNs, como o *Minimum Estimated Expected Delay* (MEED) [25], uma variante do *Minimum Expected Delay* (MED) [23], *Single-copy routing* [54] e ainda, mais recentemente, especificamente para redes sub-aquáticas de sensores, o *Prediction Assisted Single-copy Routing* (PASR) [19].

4.4 Protocolos baseados em replicação

Esta classe de protocolos pode ser definida como aquela onde diversas cópias de uma mesma mensagem são geradas geralmente com a intenção de minimizar o atraso e maximizar o número de mensagens entregues ao destino. Dentre os diversos protocolos propostos pela literatura podemos destacar o *Epidemic* [57], PRoPHET [36], RAPID [1], MaxProp [6], EBR [40] e ainda os híbridos⁴ *Spray and Wait* [55] e NECTAR [46].

O protocolo *Epidemic* foi o primeiro protocolo proposto com a funcionalidade de encaminhamento através de vizinhos e que possui algum controle sobre as mensagens armazenadas e encaminhadas aos vizinhos da DTN [57]. Esse controle realizado pelo protocolo *Epidemic* pode ser exemplificado pela troca de vetores referentes às mensagens armazenadas quando dois nós se encontram. Ao encontrar um vizinho, o nó primeiramente realizará a sessão anti-entropia (*anti-entropy session*), que é a troca de seus vetores de índices referentes às mensagens armazenadas *summary vector*, a partir desse momento o protocolo tem a possibilidade de decidir se receberá ou não mensagens que não possua. Essas decisões podem ser tomadas com base, por exemplo, no espaço em *buffer*. Outro método utilizado pelo *Epidemic* como esforço para a diminuição dos recursos dispendidos é a manutenção de um *cache* de encontros recentes a fim de que não hajam conexões redundantes e desnecessárias. Um dos maiores problemas do *Epidemic* é sua simplicidade na escolha da replicação das mensagens, sendo baseada apenas na existência ou não da mensagem pelo nó contactado, gerando um alto consumo dos recursos disponíveis em prol da diminuição do atraso e maxi-

⁴Protocolos híbridos são aqueles que apresentam as duas características distintas por meio de fases. Por exemplo, uma primeira fase epidêmica (baseada em replicação) e uma segunda fase conservadora (baseada em encaminhamento).

mização da taxa de entrega.

A grande gama de protocolos propostos pela literatura têm utilizado técnicas de replicação, uma vez que no artigo base do protocolo *Spray and Wait* [55] é exposto que as técnicas de replicação tendem a melhorar a performance em relação às técnicas baseadas em encaminhamento. Esse ganho ocorrerá, principalmente, quando em conjunto as técnicas de replicação se utilizam técnicas de controle na disseminação das mensagens na rede e gerência de *buffer*. Sejam essas técnicas baseadas na utilização de um histórico, na probabilidade de contato ou entrega da mensagem a seu destinatário [46, 36, 6, 1] ou no uso de reconhecimento passivo, por meio do uso de ACKs (*Acknowledgments*), para eliminação de mensagens já entregues ao destino. Exemplos de protocolos que utilizam a técnica de reconhecimento passivo são o MaxProp [6] e o NECTAR [46].

4.5 Protocolos avaliados

Foram seis os protocolos avaliados por este trabalho, onde os três primeiros são os protocolos *Spray and Wait*, MaxProp e PRoPHET, selecionados por estarem entre os protocolos parametrizáveis mais citados na literatura e apresentarem modos de operação com características distintas entre si. Os protocolos RAPID, EBR e o NECTAR são relevantes na literatura por conta de sua atualidade e o bom desempenho que vêm demonstrando quando comparados a outras propostas. Esses se encontram com menos variações de parâmetros, onde o EBR apresenta as mesmas variações escolhidas no artigo da proposta original, o RAPID é variado sobre suas três métricas distintas de operação, e o NECTAR é avaliado apenas em sua configuração padrão, com o intuito de comparação no desempenho global entre os protocolos.

As subseções a seguir descrevem brevemente cada um dos protocolos selecionados.

4.5.1 Spray and Wait

Com base na premissa de redução do *overhead* e do alto custo de energia dispendidos na transmissão por protocolos de roteamento baseados em inundação, é proposto o protocolo *Spray and Wait* [55]. Baseado em uma abordagem oportunista para estimar a quantidade de nós na rede e determinar a quantidade ótima de réplicas das mensagens, combina a velocidade na disseminação de mensagens do *Epidemic* [57] com a simplicidade de um encaminhamento de mensagens diretamente para o destinatário.

O mecanismo de encaminhamento de mensagens deste protocolo requer duas fases distintas, onde a primeira é denominada *spray* e a segunda *wait*, expostas a seguir:

1. durante a primeira fase, *spray*, para cada mensagem gerada pelo nó de origem, uma quantidade máxima de L cópias dessa mensagem é propagada pela rede, de forma que as mensagens são transportadas pelo nó de origem e por outros $L - 1$ nós da rede;
2. ao iniciar a segunda fase, *wait*, caso o destinatário não tenha sido alcançado na fase anterior, cada um dos L nós que contém uma cópia da mensagem somente poderá efetuar a entrega diretamente para o destinatário, como um protocolo de roteamento *Direct Delivery* (DD).

É possível habilitar o modo de operação binário (*Binary Spray and Wait*), que de maneira sucinta é a divisão das cópias armazenadas em um nó de origem ao seu vizinho que não possuía nenhuma cópia desta mensagem. Para ficar mais claro, este processo pode ser exemplificado conforme apresentado em [55]: primeiramente o nó A de origem será inicializado com um número L de cópias da mensagem X . Caso A , que tem $n > 1$ cópias da mensagem X , encontre com um nó B que não possua uma única cópia da referida mensagem, caberá ao nó A transferir $\lfloor n/2 \rfloor$ cópias para B , permanecendo com $\lceil n/2 \rceil$ cópias. Esse processo se repetirá até que $n = 1$, quando a mensagem X somente poderá ser transmitida para o destinatário.

Sobre a seleção dos valores de L , como não há um valor único para o número de cópias L explicitamente estipulado pela proposta deste protocolo de roteamento foi determinado que os valores de cópias L seria encontrado a partir do *Lemma 4.3* do referente trabalho original [55], que aqui encontra-se exposto pela Equação 4.1. Com base Acreditamos assim termos escolhido os valores de L de forma mais justa; em contraste às comparações da literatura que comumente arbitram valores para L sem qualquer embasamento.

Para encontrar os valores mínimos de L , para a minimização do atraso, a partir apenas do conhecimento do número de nós na rede, é utilizada a Equação 4.1:

$$(H_M^3 - 1, 2)L^3 + \left(H_M^2 - \frac{\pi^2}{6}\right)L^2 + \left(a + \frac{2M - 1}{M(M - 1)}\right)L = \frac{M}{M - 1}, \quad (4.1)$$

onde, $H_n^r = \sum_{i=1}^n \frac{1}{i^r}$ é o n -ésimo número harmônico de ordem r e M representa o número de nós.

Avaliamos a relação de impacto das variações dos parâmetros L , referente ao número inicial de cópias de uma mensagem, e da ativação ou não do modo de encaminhamento binário.

4.5.2 PRoPHET

O PRoPHET (*PRobabilistic Routing Protocol using History of Encounters and Transitivity*) é um protocolo de roteamento DTN baseado no histórico de encontros para estimar a probabilidade de entrega [36, 34]. Cada nó a deve calcular a métrica denominada *Delivery Predictability*, $P_{(a,b)} \in [0,1]$ para cada destino conhecido b , conforme a Equação 4.2. Quando ocorre um contato entre dois nós a e b , o nó a efetua o encaminhamento da mensagem m se b apresentar uma maior probabilidade de entrega para o destino d , ou seja, se $P_{(a,d)} < P_{(b,d)}$.

O valor de $P_{(a,b)}$ aumenta a cada novo encontro entre a e b . No entanto, se a e b deixam de se encontrar frequentemente, $P_{(a,b)}$ tem o valor reduzido à proporção que o tempo decorre. Essa redução é controlada por uma constante de envelhecimento, identificada por γ , elevada à potência de k , que corresponde à quantidade de unidades de tempo decorridas desde a última vez que a e b se encontraram, conforme Equação 4.3. Já a propriedade de transitividade, apresentada pela Equação 4.4, é aplicada caso um nó a encontre frequentemente um nó b , e o nó b encontre frequentemente um nó c . Desta forma, o nó c apresenta-se como uma boa escolha para encaminhar mensagens para o nó a e vice-versa.

A probabilidade de entrega de mensagens é calculada a partir de:

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init}, \quad (4.2)$$

onde P_{init} é uma constante de inicialização;

O envelhecimento das rotas é encontrando, solucionando a Equação 4.3, conforme:

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k, \quad (4.3)$$

onde γ é uma constante de envelhecimento, e k a quantidade de unidades de tempo decorridas desde a última atualização da métrica *Delivery Predictability*;

Já as rotas descobertas por meio de nós intermediários, são determinadas a partir da Equação 4.4, como segue:

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \times P_{(a,b)} \times P_{(b,c)} \times \beta, \quad (4.4)$$

onde β é a constante que determina o impacto da transitividade sobre a métrica *Delivery Predictability*.

Este trabalho avaliará quão impactante serão as alterações dos valores dos parâmetros γ , referente a constante de envelhecimento e β , responsável pelo impacto na transitividade.

4.5.3 MaxProp

O protocolo de roteamento MaxProp [6] faz uso do conhecimento adquirido nos encontros anteriores entre nós da rede para otimizar o encaminhamento de mensagens. Esse protocolo trabalha da seguinte forma, cada nó mantém um vetor denominado *delivery likelihood* que armazena um custo para alcançar cada destinatário conhecido. Quando dois nós se encontram, ocorre a troca do vetor *delivery likelihood* para que seja possível calcular o menor caminho para cada destinatário. O algoritmo que tem a função executar o cálculo do custo de cada destino baseia-se nas seguintes regras:

- cada nó tem um vetor $F^i = (f_0^i, \dots, f_{j-1}^i)$, que é normalizado, de forma que a soma dos elementos seja igual a 1;
- quando um nó x encontra um nó y , o elemento do vetor correspondente ao nó y é incrementado de uma unidade, e todos os elementos do vetor são divididos por 2, mantendo o vetor sempre normalizado;

Dessa forma, o nó encontrado pela última vez apresenta sempre o maior valor, o encontrado pela penúltima vez a metade do valor, e assim sucessivamente, ordenando os nós sempre em função dos encontros. De posse dos valores para os outros nós, um nó x pode calcular o custo, $c(i, i+1, \dots, d)$ de cada caminho possível para um destino d , até o limite de n saltos. Logo, o custo de um caminho usando os nós $(j, j+1, \dots, d)$ será a soma das probabilidades de cada conexão ao longo do percurso não ocorrer, representada por um menos a probabilidade de cada conexão ocorrer, conforme a Equação 4.5:

$$c(j, j+1, \dots, d) = \sum_{x=j}^{d-1} [1 - (f_{x+1}^x)] \quad (4.5)$$

Essencialmente, o protocolo MaxProp mantém uma lista de pacotes ordenada com base no destino de cada mensagem, que por sua vez obedece a uma ordenação estimada a partir de seu vetor *delivery likelihood*.

Assim como o PRoPHET, o MaxProp tem a característica de considerar o último encontro como o mais importante, porém, o processo de envelhecimento destes dois protocolos se mostra diferente quando levado em consideração o peso destes encontros no cálculo das probabilidades. O PRoPHET considera como base para seu cálculo uma constante de envelhecimento e o tempo decorrido desde o último encontro com o nó em questão, já para o MaxProp, em sua configuração padrão, não é levado em consideração uma medida de pesos para os encontros anteriores durante o cálculo de suas probabilidades. Sendo assim, seguindo o padrão de incremento deste protocolo, ou seja, com o valor de $\alpha = 1$, tem-se uma maior relevância ao último encontro em relação ao peso dos encontros anteriores. Para tanto é proposto em [29] uma variação desta variável de envelhecimento α a fim de tornar mais justo o cálculo de sua função de probabilidade conforme o perfil de cada cenário. A partir de tal generalização o protocolo em questão é passível de comparações em relação ao impacto sob a alteração do valor de α para qualquer valor maior que 0. Neste trabalho não serão expostas as provas para variação do parâmetro α , para tanto, é indicada a leitura de [29].

Outro parâmetro utilizado neste trabalho com a finalidade de verificação do impacto sobre sua variação é o *MaxSize*, que tem como finalidade o armazenamento das probabilidades conhecidas pelo nó em questão sobre os demais nós da rede. Esse armazenamento é realizado por meio de um vetor que, como já citado, é conhecido como *delivery likelihood*.

4.5.4 RAPID

Com a proposta de suprir uma lacuna existente nos diversos protocolos DTNs propostos pela literatura é concebido o RAPID (*Resource Allocation Protocol for Intentional DTN*) [1]. Essa lacuna diz respeito à satisfação de uma métrica de forma intencional. Partindo da premissa de que protocolos de roteamento DTNs em geral se preocupam apenas com a descoberta de caminhos até o destino a partir das limitadas informações disponíveis na rede e, sendo assim, demais métricas como o respeito a um atraso máximo e a diminuição do atraso médio de entrega serão acrescentadas somente de forma não intencional, o RAPID apresenta modos de operação que visam intencionalmente minimizar uma das três métricas seguintes: atraso médio, pior caso de atraso e o número

de mensagens perdidas por expiração (*deadline*).

É apresentada a função utilidade U_i calculada para cada mensagem i em relação à métrica selecionada, que visará atender ao objetivo esperado, respeitando os recursos disponíveis aos nós.

As três métricas são apresentadas a seguir juntamente com sua respectiva descrição e função utilidade.

Métrica 1: minimizar o atraso médio (*minimizing average delay*): para minimizar o atraso médio de entrega das mensagens na rede é definida uma função utilidade para cada mensagem, conforme a Equação 4.6, onde o atraso esperado para cada mensagem influencia no cálculo do atraso esperado. Nesta métrica o RAPID trabalha de forma gulosa (*greedy*) replicando um alto número de mensagens em busca da diminuição do atraso total na rede.

$$U_i = -D(i) \quad (4.6)$$

Métrica 2: minimizar mensagens perdidas por expiração (*minimizing missed deadlines*): a fim de minimizar o número de mensagens perdidas por conta da expiração de seu tempo de vida (TTL - *Time To Live*), a função utilidade desta métrica é baseada na probabilidade da mensagem ser entregue respeitando seu TTL. Onde $L(i)$ representa o TTL da mensagem.

$$U_i = \begin{cases} P(a(i) < L(i) - T(i)), & L(i) > T(i) \\ 0, & \text{caso contrário} \end{cases} \quad (4.7)$$

Uma mensagem que tenha seu TTL expirado recebe o valor de 0.

Métrica 3: minimizar o atraso máximo (*minimizing maximum delay*): a métrica mais conservadora entre as propostas tem sua função utilidade determinada pela Equação 4.8, onde S representa o conjunto de todas as mensagens no armazenados no *buffer* de X . Sendo assim, a função utilidade U_i tem o atraso esperado negativo se i é a mensagem com o atraso máximo dentre todos armazenados por Y . Dessa forma, a replicação é utilizada somente para aquela mensagem cujo atraso é máximo, *i.e.*, avalia-se se haverá replicação para aquela mensagem com atraso máximo e após esse passo é recalculado para todas as demais.

$$U_i = \begin{cases} -D(i), & D(i) \geq D(j) \forall j \in S \\ 0, & \text{caso contrário} \end{cases} \quad (4.8)$$

É a partir de tais funções utilidade apresentadas que o RAPID intencionalmente visa maximizar uma de suas métricas levando em consideração o espaço de armazenamento disponível e a banda limitada das DTNs.

4.5.5 EBR

Proposto como um protocolo que visa maximizar a entrega de mensagens com uma baixa sobrecarga em relação ao processamento e recursos disponíveis na rede, o EBR (*Encounter-Based Routing*) [40], como o próprio nome sugere, utiliza seu aprendizado em encontros anteriores como métrica para o encaminhamento de réplicas de suas mensagens.

Esse trabalho percebe a necessidade de replicação das mensagens na rede, e visa, a partir de uma implementação mais simplista, demonstrar que um protocolo que apresente um número inicial limitado de réplicas de mensagens (assim como é o caso do *Spray and Wait*), mesmo que baseado em um algoritmo com menor complexidade, pode apresentar resultados similares ou até mesmo superiores a protocolos puramente baseados em replicação baseando-se na tomada de decisão em relação a quantidade de réplicas que serão encaminhadas a um nó vizinho.

Sua funcionalidade se baseia em observações locais da ocorrência de encontros, representadas pelos valores de EV (*Encounter Value*) e CWC (*Current Window Counter*). Onde, EV significa a taxa de encontros no passado (que é exponencialmente proporcional à movimentação do nó), e já o CWC é utilizado para obter o número de encontros ocorrido durante o intervalo entre encontros. A Equação 4.9 representa tal funcionalidade.

$$EV \leftarrow \alpha \times CWC + (1 - \alpha) \times EV, \quad (4.9)$$

onde α representa o peso dado ao CWC do último encontro.

Como EV, resultante da Equação 4.9, representa a previsão de ocorrência de encontros futuros por um intervalo de tempo, o protocolo considera que o nó com maior

EV representará uma maior probabilidade de sucesso na entrega de uma mensagem. Sendo assim, quando dois nós se encontram, comparam seus EVs e decidem o número de réplicas que irão transferir. A seleção desse número de réplicas para dois nós, A e B, para cada mensagem M_i é representado pela Equação 4.10.

$$\left\lfloor m_i \times \frac{EV_B}{EV_A + EV_B} \right\rfloor, \quad (4.10)$$

onde, m_i é o número total de réplicas de M_i armazenada pelo nó.

Para a análise deste protocolo foram variados os valores dos parâmetros de entrada α e L (que representa o número inicial de cópias). Os valores escolhidos são os mesmos utilizados no artigo de proposta original do protocolo EBR [40].

4.5.6 NECTAR

O protocolo de roteamento NECTAR (*Neighborhood Contact History Routing*) [44, 46] utiliza uma métrica de roteamento baseada em encontros, denominada *Neighborhood Index* (Nidx). Este protocolo parte da premissa que cenários DTN estocásticos têm uma certa probabilidade de reencontros entre seus vizinhos e, como forma de evitar um consumo desnecessário dos recursos da rede, cada nó mantém uma relação de vizinhos encontrados recentemente, evitando sua re-avaliação.

Quando dois nós se encontram, inicialmente, são transmitidas as mensagens cujo destino seja o nó que estabeleceu o contato. Em seguida, ocorre a troca de informações sobre a vizinhança (troca do *Neighborhood Index*) e, eventualmente, o encaminhamento das demais mensagens armazenadas conforme política de encaminhamento exposta mais a frente. A propagação das informações sobre a vizinhança de cada nó possibilita um conhecimento mais apurado da topologia da rede, o que, intrinsecamente, favorece o trabalho de roteamento. A partir desse conhecimento é possível identificar a melhor forma de encaminhamento das mensagens e conseqüentemente a redução do consumo pelos nós da rede.

As principais funcionalidades desse protocolo podem ser divididas em três módulos: cálculo do *Neighborhood Index*, algoritmo para escalonamento de mensagens e política de descarte de mensagens, como descritas a seguir.

Cálculo do Nidx

O *Neighborhood Index* (Nidx) é baseado no histórico de contatos recentes, de forma

que nós cujos contatos sejam duradouros e frequentes apresentem um valor superior ao valor de nós cujos contatos sejam de curta duração e eventuais. Ao ocorrer o primeiro contato entre os nós i e j , $Nidx_{(i,j)}$ é inicializado com 1. É prevista também o cálculo por transitividade. Quando, por exemplo, $Nidx_{(j,d)} > Nidx_{(i,d)}$ ou i não conhece uma rota para d , $Nidx_{(i,d)}$ terá j como $NextHop_{(i,d)}$ e será atualizado de forma ponderada a partir de uma métrica de distância $hops_{(j,d)}$ (quantidade de saltos entre j e d) e a quantidade de tempo em que j e d não se encontram, elevada à uma constante de envelhecimento (σ). Conforme a Equação 4.11:

$$Nidx' = \frac{Nidx_{(j,d)}}{(hops_{(j,d)} + 1) \times (TS - ts_update_{(j,d)})^{\sigma_{(j,d)}}}, \quad (4.11)$$

onde, TS é o tempo atual e ts_update o tempo do último contato entre j e d .

Caso i conheça uma rota para o destino d , porém seu $Nidx$ seja menor que de seu vizinho j , então o cálculo de um novo $Nidx$ é realizado porém de forma ponderada a fim de que essas novas informações não tenham um impacto drástico em relação ao conhecimento prévio daquela rota. Conforme a Equação 4.12:

$$Nidx_{(i,d)} = \begin{cases} Nidx', & \text{caso } i \text{ não tenha rota para } d \\ \frac{(Nidx_{(i,d)} \times \omega) + Nidx'_{(i,d)}}{\omega + 1}, & \text{caso } i \text{ tenha rota para } d, \end{cases} \quad (4.12)$$

onde, ω é o valor do peso dado ao índice de vizinhança anterior no cálculo de atualização.

O processo de envelhecimento do $Nidx$ só ocorre se não houver atualização das informações de roteamento em um intervalo de tempo superior, denominado $NR_contact$. Nesse caso, o índice de envelhecimento aplicado é inversamente proporcional ao valor do $Nidx$, ou seja, quanto maior o valor do $Nidx$, menor será a variável de envelhecimento (σ). Essa abordagem permite que índices de vizinhança associados a valores maiores, que representam rotas mais estáveis, sejam envelhecidos de forma mais amena.

Escalonamento das mensagens

Como em DTNs a capacidade de transmissão da rede pode ser consideravelmente limitada, torna-se imprescindível que os nós possam selecionar e priorizar as mensagens que devem ser transmitidas, aproveitando ao máximo o tempo de contato disponível, e evitando, dessa forma, a perda da oportunidade de transmissão.

Ao encontrar um vizinho, mensagens cujo destino seja o nó contactado são encaminhadas prioritariamente. Posteriormente, mensagens cujo próximo salto *NextHop* seja o nó encontrado também são encaminhadas. Outra técnica utilizada como forma de economia de recursos dos nós é a utilização de um campo TTL (*Time-To-Live*) que contabiliza os saltos restantes à mensagem, e a partir do momento em que a mensagem se encontra a apenas um salto restante, esta poderá ser encaminhada somente ao nó de destino (como o protocolo *Direct Delivery*).

O NECTAR utiliza a técnica de anti-pacote por meio de ACKs (*Acknowledgements*) [27] (assim como o MaxProp [6]) com a intenção de avisar aos nós da rede que a mensagem já foi entregue ao seu destino. Essa técnica se baseia em, após o nó *i* entregar a mensagem a seu destino *d*, guarda-se seu cabeçalho em um vetor denominado *PassiveAck* por um tempo pré-determinado (*ttl_deleted*), e quando este nó *i* encontra outras nós na rede, troca-se esse vetor para que sejam feitas as devidas exclusões.

Outro fator interessante deste protocolo sobre a gerência das mensagens se dá em razão da disponibilidade ou não de recursos na rede. O protocolo propõe que dinamicamente seja avaliada a capacidade de armazenamento e que seja decidido se a replicação das mensagens será realizada de forma epidêmica ou probabilística.

Política de descarte de mensagens

Como característica de uma DTN, geralmente, os nós apresentam capacidade limitada de armazenamento, gerando a necessidade de uma política eficiente de descarte de mensagens. Desta forma o NECTAR associa dois parâmetros às mensagens recebidas, que são: M_{repl} , que identifica a quantidade de réplicas da mensagem efetuadas pelo nó em questão; e M_{slot} , que indica o tempo em que a mensagem se encontra no *buffer* daquele nó.

A política de descarte é acionada a partir do momento em que o *buffer* se encontra cheio e objetiva manter as mensagens mais recentes e aquelas menos encaminhadas. Essa ordenação é gerada a partir da Equação 4.13:

$$M_{age} = M_{repl} \times M_{slot}. \quad (4.13)$$

4.6 Configurações de cada protocolo

A Tabela 4.1 relaciona os parâmetros selecionados, seus respectivos valores padrão e o intervalo dos valores utilizados nas simulações para cada protocolo. Deve-se ressaltar que a implementação do protocolo MaxProp no simulador The ONE foi baseada em [29], possibilitando que o incremento utilizado para atualização do vetor *delivery likelihood* não ficasse restrito a uma unidade. Essa implementação permitiu a avaliação do impacto causada pela variação desse incremento, identificado pelo parâmetro α , porém respeitando-se a normalização da proposta original. Já o parâmetro *MaxSize* limita a quantidade de informações armazenadas no vetor *delivery likelihood* e, como não há na literatura um valor padrão para a referido parâmetro, assumimos neste trabalho o valor sugerido na implementação do simulador The ONE. O parâmetro P_{init} do protocolo PROPHET foi excluído em função da influência limitada. Já o protocolo *Spray and Wait* é o único que prevê a alteração da configuração padrão, porém de forma estática. O valor de L associado à configuração padrão depende da quantidade de nós da rede e foi calculado utilizando o *Lema 4.3* apresentado em [55]. Em EBR os parâmetros selecionados são os mesmos variáveis no artigo original de proposta desse protocolo, onde α referencia o peso aos encontros e L o número de cópias iniciais. O NECTAR não apresenta variações em seus parâmetros, utilizando assim a configuração padrão deste protocolo, onde ω representa o peso dado ao índice de vizinhança anterior no cálculo de atualização σ representa a variável de envelhecimento e o γ a ocupação máxima do *buffer* para determinação da fase epidêmica. O protocolo RAPID apresenta como variação suas métricas propostas.

4.7 Conclusão

Nesse capítulo foram introduzidas as funcionalidades de cada um dos protocolos de roteamento DTN selecionados assim como sua classificação. A partir desse embasamento se tornarão mais claras as avaliações realizadas sobre os experimentos descritos. Esse capítulo justificou ainda a escolha dos protocolos de roteamento por suas características de parametrização e relevância atual na literatura.

Tabela 4.1: Intervalo de valores dos parâmetros

Protocolo	Parâmetro	Conf. Padrão	Intervalo de Valores
MaxProp	α	1	0,01, 0,1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 50, 100, 250, 500 e 1.000
	MaxSize	50	10, 25, 50, 75, 100, 250, 500 e 1.000
PRoPHET	β	0,25	de 0,05 até 0,95, com incrementos de 0,05
	γ	0,98	de 0,05 até 0,95, com incrementos de 0,05, 0,98 e 0,999
Spray and Wait	<i>binary</i>	<i>true</i>	<i>true</i> e <i>false</i>
	<i>L</i>	*	de 1 até 15, 20, 25, 30, 40, 50, 75, 100, 150, 200, 250, 300, 500 e 1.000
EBR	α	0,85	0,5 e 0,85
	<i>L</i>	11	5, 11 e 20
NECTAR	ω	5	5
	σ	0,05	0,05
	γ	0,90	0,90
RAPID	métricas	<i>minimizing average delay</i>	<i>minimizing average delay, minimizing maximum delay e minimizing missed deadlines</i>

* os resultados de *L* para cada cenário foram suprimidos desta tabela por questão de espaço.

Capítulo 5

Resultados Obtidos

Serão apresentados neste capítulo os resultados obtidos, analisando-os sob diversas métricas. Como intuito de validação deste trabalho será prioritariamente analisada a relação entre as diversas métricas de desempenho e os valores atribuídos aos parâmetros de cada protocolo de roteamento. Por apresentarem maior relevância a análise destes resultados, foram selecionadas as seguintes métricas: taxa de mensagens entregues, latência e o desempenho conforme o espaço de armazenamento disponível e a densidade da rede.

Cronologicamente este capítulo apresenta a caracterização de contato nos cenários simulados e analisa os valores possíveis de mensagens encaminhadas e a quantidade média alcançada por cada protocolo. Posteriormente, é realizada a análise individual de cada protocolo conforme suas respectivas variações de valores dos parâmetros para os diversos ambientes gerados. É exposto, também, o percentual de deficiência em relação a configuração de melhor desempenho, para a configuração padrão e a configuração com pior desempenho de cada protocolo. Verificamos, ainda, a ocorrência da configuração padrão em relação ao espaço de armazenamento disponível. Ao final é apresentada a conclusão sobre os resultados apresentados.

5.1 Simulador escolhido

O simulador escolhido para a validação dos experimentos propostos por este trabalho foi o simulador The ONE [30, 31]. Uma de suas maiores motivações é sua ampla utilização nas atuais pesquisas sobre DTNs, como podemos ver em: [10], [37], [42], [29] dentre diversos outros. Esse simulador é baseado em eventos discretos. Para maiores informações é aconselhável o trabalho [31], onde são detalhadas desde as funcionalidades mais básicas, passando pelos protocolos e modelos de movimentação suportados

pelo simulador até uma análise de seu núcleo.

Este trabalho utilizou a versão 1.3 do simulador The ONE. Durante a etapa de avaliação dos protocolos, foram incorporadas funcionalidades ao código-fonte do protocolo RAPID no simulador The ONE. Essas funcionalidades encontram-se no Anexo A, e sua finalidade é permitir ao usuário selecionar uma das três métricas suportadas por esse protocolo sem a necessidade de recompilação do simulador. Já como contribuição maior à comunidade, foi implementado o protocolo NECTAR no simulador The ONE, como pode ser observado no Anexo B, que compreende seu código-fonte.

5.2 Caracterização dos contatos

Esta seção analisa as características de cada cenário em relação ao tempo de duração de cada contato. São comparados os valores máximos possíveis de mensagens transmitidas a cada contato (máximo teórico) e os valores médios reais alcançados por cada protocolo de roteamento. As figuras, que apresentam os gráficos para exposição dos resultados do tempo de contato, foram expressos a partir de uma Função de Distribuição Acumulada (FDA) em relação ao tempo em segundos representado em escala logarítmica. A intenção desta seção é facilitar a identificação das características de movimentação dos cenários escolhidos sob a perspectiva do tempo de duração de contatos.

5.2.1 Duração dos contatos

Quando da análise do tempo de duração de um contato, é necessário levar em consideração as mensagens disponíveis no *buffer* do nó. Essa afirmativa existe porque mesmo que um contato perdure por um tempo suficientemente grande, o número de mensagens encaminhadas será limitado pela quantidade existente no *buffer* do nó.

A Figura 5.1 exibe o gráfico referente à distribuição da duração do tempo de contato no cenário UmassDieselNet (representado pela abscissa), onde a característica variável corresponde à quantidade de nós (ônibus) na rede, equivalente a 15 e 31 e representado por $N=15$ e $N=31$. Como já citado na Seção 3.2, os dados de movimentação desse cenário foram obtidos a partir de *traces* reais cujos dados se referem ao momento de início e fim de um contato, sendo assim, não há variações de raio. Analisando as informações dispostas pela Figura 5.1 podemos verificar a longa duração dos contatos neste cenário. Pode-se observar que 70% dos contatos se encontram com probabilidade de

duração acima de 1000 segundos. Essa relevante duração de tempo de contato, beneficiará a todos os protocolos de roteamento, uma vez que há condições suficientes para a realização de todas as fases de sincronismo (*e.g.* troca de vetores, escalonamento de mensagens, descarte de mensagens etc.), assim como envio e recebimento das mensagens do nó vizinho contatado.

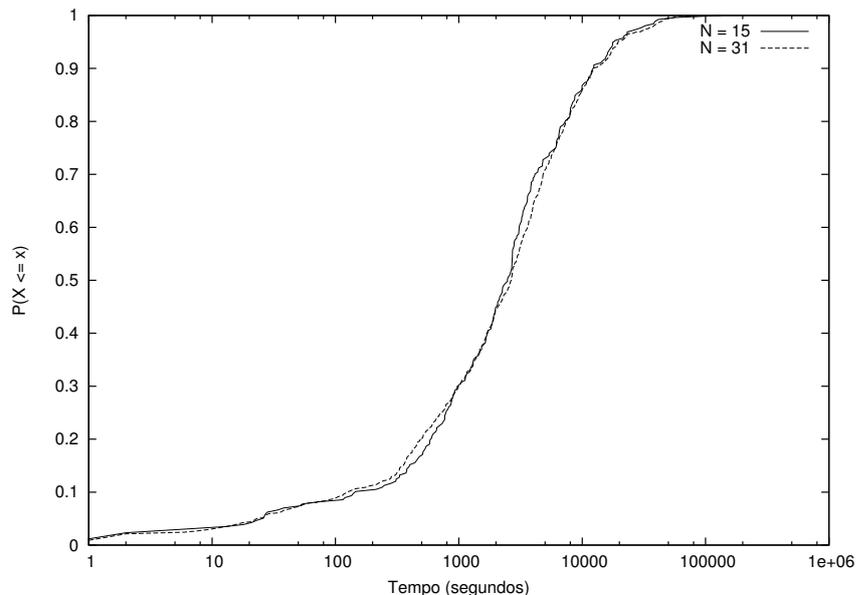


Figura 5.1: FDA para o tempo de duração de contato entre os nós no cenário Umass-DieselNet.

O cenário ZebraNet possui variações em seu número de nós ($N=10, 20, 30$ e 40) e, diferentemente do UmassDieselNet, possui também variações de raio ($R=50, 150$ e 250). Esse tipo de característica pode refletir no desempenho de cada um dos protocolos de forma diferente, uma vez que cada protocolo apresenta sua política em relação, principalmente, ao encaminhamento e descarte de mensagens.

As Figuras 5.2a, 5.2b, 5.2c e 5.2d, refletem a baixa duração dos contatos no cenário ZebraNet. É possível observar, analisando as Figuras 5.2a e 5.2d, que quanto maior o número de nós e o raio de alcance do rádio, maior será a probabilidade de duração do contato. Essa observação fica mais clara quando observada na Figura 5.2a, equivalente a rede menos densa, onde 90% dos 10 nós da rede com raio igual a 50, terão probabilidade de contatos com duração próxima a 10 segundos. Já a Figura 5.2d, quando a rede se encontra mais densa para esse cenário, demonstra que aproximadamente 80% dos contatos para os 40 nós com raio igual a 250, têm a probabilidade de que o tempo de contato seja acima de 100 segundos.

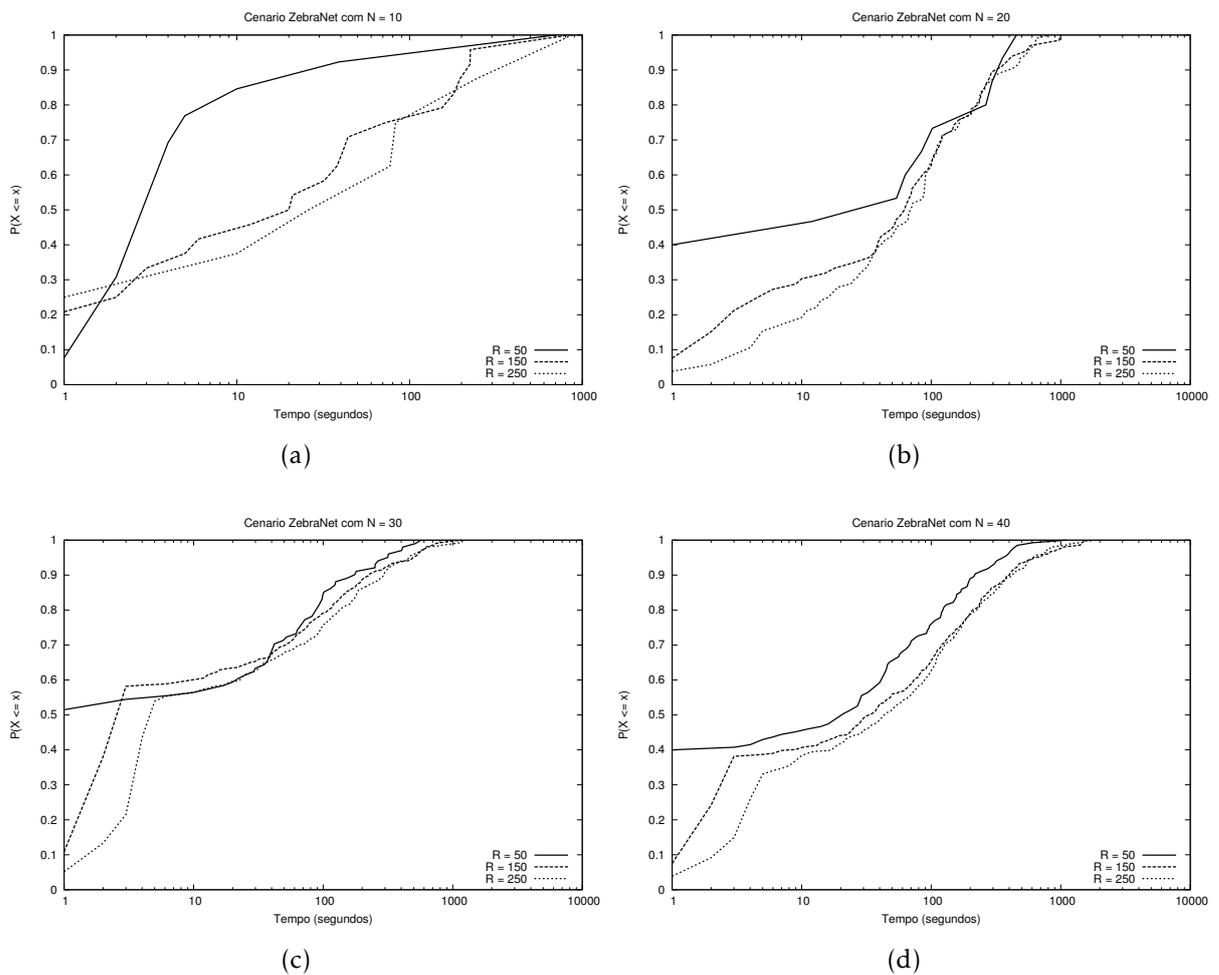


Figura 5.2: FDA para o tempo de duração de contato entre os nós no cenário ZebraNet com raios 50, 150 e 250.

Assim como no cenário ZebraNet, o WDM apresenta uma variação em relação ao raio e ao número de nós, onde pode-se observar, a partir das Figuras 5.3a e 5.3b, que a maior diferença entre as características se dá pelo tempo de duração de contato, que cresce juntamente com o raio.

Nota-se que as Figuras 5.3a e 5.3b não apresentam a mesma forma de distribuição em relação ao tempo de contatos que nas Figuras 5.2a, 5.2b, 5.2c e 5.2d, referentes ao ZebraNet. Nas Figuras do WDM percebe-se encontros com menor tempo de duração, porém, ao analisarmos os demais dados envoltos nas simulações, encontra-se um número superior da quantidade total de encontros para esse cenário se comparado ao ZebraNet e UmassDieselNet. Sendo assim, o cenário WDM apresenta contatos mais curtos, porém mais frequentes.

Outra observação ocorre por conta do tempo total de simulação para o WDM, que

foi de apenas 6 horas. Para uma maior diversificação entre as características de cenário, como exposta na Seção 3.1.3, seria necessário simular períodos maiores, como uma semana, porém, isso se tornou inviável em tempo de processamento para o número de simulações geradas.

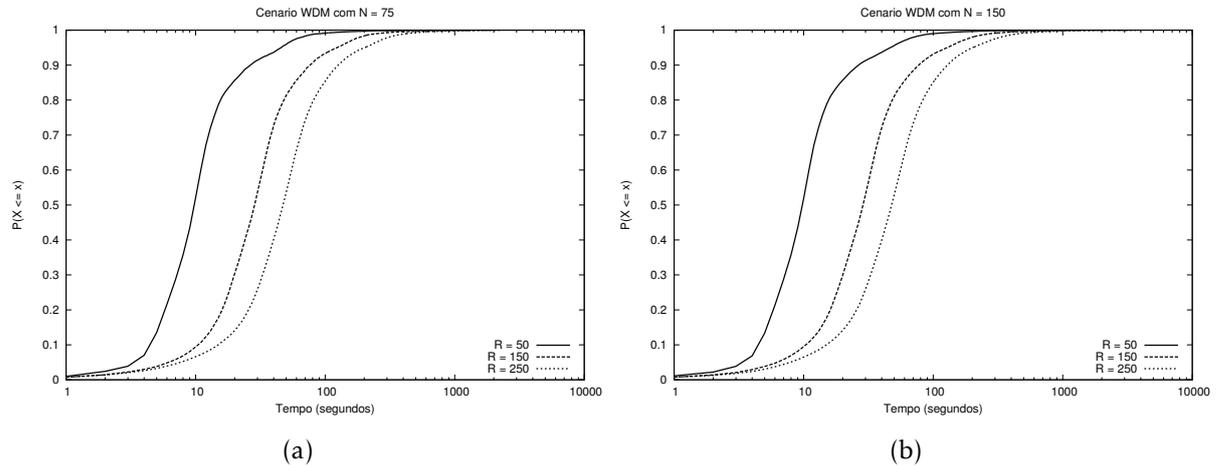


Figura 5.3: FDA para o tempo de duração de contato entre os nós no cenário WDM com raios 50, 150 e 250.

Até este momento foi realizado apenas um breve levantamento das características de contatos nos cenários abordados. Esse levantamento inicial permite uma análise mais profunda, por exemplo, do impacto do tempo de contato nos protocolos de roteamento. Porém, como não é o foco principal deste trabalho a modelagem da mobilidade nos cenários ou o impacto da mobilidade nos resultados, este trabalho analisará somente a relação entre a capacidade de transmissão de mensagens nos cenários e a quantidade real de mensagens transmitidas por cada um dos protocolos.

5.2.2 Máximo teórico e o real obtido para mensagens transmitidas

Foi traçado um comparativo entre o valor máximo de mensagens que teoricamente poderiam ser transmitidas durante o tempo de contato entre os nós e o número de mensagens realmente transmitidas por cada um dos protocolos a cada contato. Os resultados são exibidos de forma resumida pela Tabela 5.1.

O valor máximo teórico não leva em consideração o tempo gasto entre, por exemplo, troca de vetores, seleção de mensagens, enfileiramento ou qualquer outra atividade da rede ou do protocolo. Para encontrar o máximo teórico foi calculado o tempo médio de duração de contato entre dois nós e verificado, conforme sua banda disponível

e, sabendo-se *a priori* o tamanho de cada mensagem, quantas mensagens poderiam ser transmitidas. Confrontando esse valor com o real obtido por cada protocolo é possível traçarmos um paralelo em consideração à complexidade e funcionamento de cada protocolo para uma visão mais clara da possível transmissão efetiva nos ambientes simulados.

Como já citado, este trabalho utilizou em todos os seus cenários o tamanho de mensagem fixo de 10kB e com taxa de transmissão de 1Mbps. E, para verificar o maior número possível de mensagens transmitidas em relação aos cenários, foi eliminada a questão da limitação do espaço de armazenamento, analisando somente os cenários cujo os nós apresentavam espaço de armazenamento igual a 10MB. Foram, ainda, suprimidos alguns dos resultados para os cenários ZebraNet e WDM em função de não demonstrarem relevância nos resultados.

Tabela 5.1: Valores médios do tempo duração dos contatos em segundos e comparação entre o teórico e o real de mensagens transmitidas na rede

Cenário	No. de nós	Raio	Duração média (s)	Máximo teórico	Valor real obtido					
					<i>Spray and Wait</i>	PRoPHET	MaxProp	RAPID	EBR	NECTAR
DieselNet	N=15	-	2692	33653	5,27	3,21	0,21	6,33	11	2,97
	N=31	-	2738	34225	3,60	2,50	2,29	3,11	6,53	1,12
ZebraNet	N=10	R=50	30	376	3,64	1,65	1,11	2,50	3,34	2,14
		R=150	44	553	3,14	1,4	3	3,33	3	2,10
		R=250	82	1028	3,08	1,11	2,12	3,67	1,98	2,02
WDM	N=75	R=50	7	95	89	80	80	66	10	66
		R=150	20	289	72	75	112	76	7	67
		R=250	34	428	64	63	302	160	6	63

Para encontrar os valores referentes à média de transmissão de cada um dos protocolos, baseou-se no número de mensagens encaminhadas durante a execução de sua configuração padrão.

Analisando a Tabela 5.1, verifica-se que, apesar de todos os protocolos apresentarem alguma política de controle para a disseminação de mensagens, os protocolos RAPID e MaxProp obtiveram um número de mensagens encaminhadas a cada contato mais

próximo ao máximo teórico no cenário WDM. Em relação ao RAPID, atribuímos esse resultado por sua métrica padrão (*minimizing average delay*) ter um comportamento considerado guloso quanto à replicação das mensagens. Já para o MaxProp, acreditamos que este tenha tentado aproveitar ao máximo o tempo de contato entre os nós no cenário WDM, replicando assim um alto número de mensagens ao encontrar nós considerados potenciais. O protocolo PRoPHET, também apresenta valores próximos aos alcançados pelos protocolos *Spray and Wait*, NECTAR e RAPID para o cenário DieselNet, ZebraNet e WDM. O protocolo MaxProp se mostra de forma bastante controlada no encaminhamento das mensagens no cenário DieselNet. Acreditamos que as características deste cenário tenham influenciado neste resultado, uma vez que é possível que não hajam mensagens a serem encaminhadas. Apesar dos baixos números de encaminhamento de mensagens do protocolo MaxProp para os cenários ZebraNet e DieselNet, esse protocolo não tem seu desempenho em relação a taxa de entrega de mensagens prejudicado, como poder ser visto na Seção 5.3.9, nas Figuras 5.4 e 5.5. Destacamos também a baixa taxa de encaminhamento das mensagens pelo protocolo EBR, principalmente, por seu número baixo de cópias de mensagens inicialmente geradas e sua política de encaminhamento.

Apesar de observarmos valores relativamente baixos de encaminhamento das mensagens para todos os protocolos nos cenários DieselNet e ZebraNet, foi observado que o tempo de contato seria suficiente para a replicação de todas as mensagens armazenadas em cada nó. Esta comparação pôde ser feita a partir dos dados de número máximo teórico e valor real obtidos por cada um dos protocolos, apresentado na Tabela 5.1. Desta forma, seu desempenho irá depender de suas políticas de encaminhamento, que englobam diversas outras características, como o intervalo entre contatos, o reencontro de nós, e não somente das características de duração de um contato.

Disseminar de forma controlada e coerente com a real melhoria da probabilidade de entrega de uma mensagem a seu destino, é fundamental aos protocolos de roteamento em DTNs, como poderá ser observado nas próximas seção, e principalmente na Seção 5.3.9.

Essa subseção expôs uma visão sobre a quantidade de mensagens encaminhadas pelos protocolos abordados por este trabalho, o que demonstra que todos eles apresentam uma replicação de forma mais controlada. Ou seja, ratificando a premissa de que mesmo que haja possibilidade de se encaminhar diversas mensagens, muitas vezes apenas um número reduzido de mensagens será efetivamente encaminhada, na tenta-

tiva de reduzir uma possível inserção de sobrecarga de forma desnecessária.

5.3 **Variação dos valores dos parâmetros e seu impacto**

Por conta da variedade de características dos cenários, os resultados extraídos das simulações foram discretizados e tabulados em função de duas dimensões: densidade e carga de mensagens na rede. Caso o somatório da área de alcance de todos os nós de um determinado cenário seja igual ou menor que o dobro da área de simulação, a densidade será igual a 1 (D 1). Caso seja maior ou igual ao dobro, porém menor que o triplo da área de simulação, a densidade será igual a 2 (D 2). Se a quantidade total de mensagens geradas inicialmente na rede (chamado por este trabalho de carga de mensagens na rede) possa ser armazenado pelo nó, ocupando apenas 5% de sua área de armazenamento, então este será representado por 5B. Sendo assim, quando considerada a carga de mensagens na rede igual a 50B, todas as mensagens da rede podem ser armazenadas pelo nó ocupando 50% de seu *buffer*. Até 100B o nó tem capacidade de armazenar todas as mensagens inicialmente geradas na rede, porém seu *buffer* estaria 100% ocupado. Vejamos outros exemplos: para o valor de carga de mensagens na rede igual a 200B, os nós deveriam ter o dobro da capacidade de armazenamento para suportar todas as mensagens geradas, ou seja, um *buffer* com tamanho de duas vezes seu tamanho atual. E assim por diante para os valores de 500B (cinco vezes o tamanho do *buffer*) e 2000B (vinte vezes o tamanho do *buffer*).

5.3.1 PRoPHET

Tabela 5.2: Percentual de mensagens entregues do PRoPHET - Configuração com melhor desempenho

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	γ	0,60	0,999	0,999	0,999	0,999	0,999
	β	0,45	0,95	0,05	0,15	0,95	0,95
	λ	10,70%	99,00%	67,00%	88,00%	100,00%	80,80%
50B (250 e 500 Mensagens)	γ	0,95	0,60	0,999	0,999	0,999	0,999
	β	0,10	0,95	0,10	0,15	0,95	0,70
	λ	7,22%	96,75%	53,60%	87,00%	100,00%	74,30%
100B (50, (250, 500 e 1.000 Mensagens)	γ	0,40	0,90	0,999	0,999	0,999	0,999
	β	0,55	0,90	0,10	0,15	0,95	0,90
	λ	7,76%	97,25%	55,28%	87,40%	100,00%	76,00%
200B (500 e 1.000 Mensagens)	γ	0,98	0,80	0,999	0,999	0,98	0,999
	β	0,05	0,05	0,10	0,05	0,05	0,05
	λ	6,43%	90,70%	49,35%	84,20%	97,90%	70,80%
500B (250 Mensagens)	γ	0,999	0,80	0,999	0,999	0,999	0,999
	β	0,10	0,05	0,10	0,05	0,05	0,10
	λ	7,65%	71,50%	52,90%	75,80%	88,20%	60,20%
2000B (1.000 Mensagens)	γ	0,98	0,85	0,999	0,999	0,999	0,999
	β	0,20	0,05	0,20	0,05	0,05	0,10
	λ	6,10%	53,50%	34,70%	60,40%	70,40%	49,60%

λ - corresponde ao percentual de mensagens entregues pela configuração.

Observando a primeira célula da Tabela 5.2, identificamos que, para o cenário ZebraNet com densidade igual a 0 (D 0) e com capacidade de armazenar todas as mensagens da rede ocupando entre 5% e 20% (carga de mensagens na rede de 5B a 20B) do total de seu *buffer*, a configuração que obteve o melhor desempenho foi capaz de entregar 10,70% das mensagens geradas com $\gamma = 0,60$ e $\beta = 0,45$. Já a primeira célula da Tabela 5.3, indica que a configuração padrão deveria melhorar 4,90% para se igualar a configuração com melhor desempenho, e que a configuração com pior desempenho deveria melhorar 5,94%. Destacamos que algumas dimensões das Tabelas 5.2 e 5.3 foram suprimidas em função de não agregarem informações relevantes.

Tabela 5.3: Percentual de deficiência da configuração padrão e com pior desempenho para mensagens entregues do PROPHET

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B, e 20B (50 Mensagens)	padrão	(4,90%)	(2,06%)	(1,82%)	(1,38%)	(0,00%)	(0,00%)
	pior	(5,94%)	(3,12%)	(13,18%)	(22,22%)	(0,40%)	(22,42%)
50B (250 e 500 Mensagens)	padrão	(2,12%)	(0,52%)	(1,04%)	(1,40%)	(0,81%)	(0,13%)
	pior	(4,94%)	(1,31%)	(15,64%)	(17,25%)	(1,01%)	(26,15%)
100B (50, 250, 500 e 1.000 Mensagens)	padrão	(2,11%)	(0,78%)	(1,21%)	(1,27%)	(0,70%)	(0,13%)
	pior	(4,30%)	(5,63%)	(13,05%)	(18,83%)	(0,86%)	(24,29%)
200B (500 e 1.000 Mensagens)	padrão	(1,26%)	(2,78%)	(1,33%)	(3,82%)	(9,51%)	(2,16%)
	pior	(5,41%)	(5,71%)	(15,44%)	(15,82%)	(28,82%)	(18,59%)
500B (250 Mensagens)	padrão	(2,00%)	(24,78%)	(2,52%)	(3,55%)	(22,50%)	(2,73%)
	pior	(5,52%)	(28,14%)	(23,02%)	(26,33%)	(72,94%)	(24,90%)
2000B (1.000 Mensagens)	padrão	(0,83%)	(29,54%)	(1,17%)	(10,22%)	(41,37%)	(5,98%)
	pior	(7,02%)	(44,59%)	(18,84%)	(36,04%)	(100,00%)	(31,22%)

A análise não só dos resultados expostos pelas Tabelas 5.2 e 5.3, mas todos os demais resultados, possibilitou obtermos algumas conclusões interessantes. Averiguamos que a partir do *draft* mais atual publicado do PROPHET [35] é proposta a alteração do valor de γ de 0,98 para 0,999; essa mudança se faz de forma acertada, pois o valor de γ igual a 0,999 esteve presente nas configurações que apresentaram o melhor desempenho em 56,74% dos cenários avaliados, sendo que esse valor chegou a 91,67% no cenário DieselNet. Contudo, deve-se observar que quanto maior o valor de γ , mais lento será o processo de envelhecimento de rotas antigas que podem estar obsoletas. Já o incremento do valor de β , passando de 0,25 para 0,9 não nos parece adequado, pois as configurações com β igual a 0,05 foram responsáveis pelo melhor desempenho em 31,24% dos cenários. Essa constatação confirma que, como β determina o impacto da transitividade, o peso associado a rotas obtidas de forma indireta em cenários com alta mobilidade deveria ser reduzido, ou seja, com valores de β menores que o padrão de 0,25, evitando, assim, associar pesos significativos a rotas que tendem a se tornar obsoletas rapidamente. O impacto de transitividade pode ainda ser agravado por uma constante de envelhecimento que implica em um processo de convergência lento, a partir do valor de γ igual a 0,9. Por fim, a partir do instante que os nós não dispõem de capacidade para armazenar todas as mensagens presentes na rede (carga de mensagens

na rede de 200B a 2000B), o valor de β manteve-se entre 0,20 e 0,05 e, na maioria das simulações, o valor de γ manteve-se entre 0,98 e 0,999. Cabe ressaltar que as alterações propostas pelo *draft* mais atual [35], não se limitam a alterar os valores de γ e β .

A Tabela 5.4 relaciona os valores dos parâmetros γ e β com maior ocorrência nas simulação em relação à densidade e a carga de mensagens na rede. Os resultados são exibidos de forma sucinta para a carga de mensagens na rede. Onde, caso o nó tenha capacidade de armazenar em seu *buffer* todas as mensagens geradas inicialmente na rede, a carga de mensagens na rede se encontrará como menor ou igual a 100%, ou seja, um ambiente menos restritivo. Ao contrário de quando a carga de mensagens na rede se encontra maior do que 100%, onde as políticas de escalonamento e descarte de cada protocolo terão forte influência na administração do espaço de armazenamento e consequentemente no desempenho do protocolo.

Para a carga de mensagens na rede menor ou igual a 100%, temos o equivalente na Tabela 5.2 às linhas de 5B até 100B. Já para a carga de mensagens na rede maior que 100%, temos o equivalente na Tabela 5.2, às linhas de 200B a 2000B. A intenção da exibição desse resumo de valores é guiar quem deseje realizar comparações deste protocolo com algum outro de forma mais justa.

Tabela 5.4: Relação entre os parâmetros γ e β e as características do cenário (densidade local e carga de mensagens na rede)

Carga de Mensagens na Rede	Densidade 0		Densidade 1		Densidade 2	
	γ	β	γ	β	γ	β
$\leq 100\%$	0,40	0,90	0,60	0,95	0,999	0,95
$> 100\%$	0,999	0,15	0,999	0,05	0,999	0,05

Essa exposição de resultados é também considerada para os protocolos a serem apresentados: *Spray and Wait*, MaxProp e EBR. Já o NECTAR não apresenta essa exposição por não apresentar configurações de distintas de seus parâmetros para as simulações, e o RAPID por ter sido avaliado em suas métricas e não sob a variância de valores de seus parâmetros.

5.3.2 Spray and Wait

Assim como na análise do protocolo PROPHET, o protocolo *Spray and Wait* (e todos os demais) apresentam duas tabelas comparativas entre o melhor resultado encontrado e a configuração equivalente. As Tabelas 5.5 e 5.6 exibem esses resultados. Intuitivamente, na Tabela 5.5 os valores referentes a ativação ou não do modo binário (*binary*), se encontram na linha especificada por este parâmetro, assim como o número de cópias, representado por L .

Como pode ser observado a partir da análise das Tabelas 5.5 e 5.6, o protocolo *Spray and Wait* apresenta em todos os resultados de melhor configuração o parâmetro *binary* ativo (*true*), confirmando sua premissa de que aumenta-se a probabilidade de entrega de uma mensagem se enviado um maior número de cópias ao potencial vizinho encontrado. É possível notar também que os valores referentes ao número de cópias enquanto o *buffer* consegue armazenar todas as mensagens da rede (carga de mensagens na rede de 5B a 100B) é o mais alto quanto possível, porém a partir do momento em que o *buffer* se encontra mais restrito (carga de mensagens na rede de 200B a 2000B), os valores tendem a cair. Isso ocorre porque quanto maior o número de cópias circulando na rede, maior será a chance dessa mensagem alcançar seu destino, porém é de suma importância levar em consideração os recursos disponíveis na rede.

Tabela 5.5: Percentual de mensagens entregues do Spray and Wait - Configuração com melhor desempenho

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	<i>binary</i>	Btrue	Btrue	Btrue	Btrue	Btrue	Btrue
	<i>L</i>	500	500	500	75	500	75
	λ	12,20%	99,00%	72,40%	90,00%	100,00%	81,20%
50B (250 e 500 Mensagens)	<i>binary</i>	true	true	true	true	true	true
	<i>L</i>	1000	75	1000	75	300	75
	λ	7,78%	97,00%	55,85%	89,10%	99,90%	74,50%
100B (50, (250, 500 e 1.000 Mensagens)	<i>binary</i>	true	true	true	true	true	true
	<i>L</i>	1000	500	500	75	500	500
	λ	8,57%	97,50%	57,12%	89,35%	99,75%	76,25%
200B (500 e 1.000 Mensagens)	<i>binary</i>	true	true	true	true	true	true
	<i>L</i>	500	300	50	50	300	500
	λ	6,90%	97,50%	47,15%	88,30%	97,90%	74,50%
500B (250 Mensagens)	<i>binary</i>	true	true	true	true	true	true
	<i>L</i>	40	500	500	20	500	200
	λ	8,25%	97,00%	50,60%	84,80%	99,20%	72,00%
2000B (1.000 Mensagens)	<i>binary</i>	true	true	true	true	true	true
	<i>L</i>	6	50	75	9	150	13
	λ	5,75%	97,00%	35,70%	62,00%	84,00%	55,80%

λ - corresponde ao percentual de mensagens entregues pela configuração.

A partir dos dados expostos na Tabela 5.6 verifica-se a distância considerável da configuração padrão e também da pior configuração em relação a de melhor desempenho. Isso ocorre devido ao valor de cópias selecionado. A configuração de pior desempenho em todos os casos tinha como valor *L* (número de cópias) igual a 1, portanto, trabalhando da mesma forma que o protocolo DD, encaminhando somente ao destino sua mensagem, o que conseqüentemente também incrementará a latência, como poderá ser visto ainda neste trabalho.

Tabela 5.6: Percentual de deficiência da configuração padrão e com pior desempenho para mensagens entregues do Spray and Wait

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B, e 20B (50 Mensagens)	padrão	(10,91%)	(7,61%)	(31,16%)	(4,17%)	(70,07%)	(10,33%)
	pior	(69,44%)	(43,48%)	(153,15%)	(122,17%)	(278,79%)	(160,26%)
50B (250 e 500 Mensagens)	padrão	(6,58%)	(4,31%)	(19,47%)	(4,45%)	(77,44%)	(14,37%)
	pior	(49,04%)	(50,00%)	(120,32%)	(91,20%)	(246,88%)	(177,04%)
100B (50, 250, 500 e 1.000 Mensagens)	padrão	(7,12%)	(4,84%)	(20,89%)	(4,20%)	(75,93%)	(13,05%)
	pior	(53,86%)	(48,29%)	(125,33%)	(91,12%)	(251,23%)	(173,90%)
200B (500 e 1.000 Mensagens)	padrão	(4,55%)	(4,84%)	(11,33%)	(3,03%)	(75,13%)	(11,89%)
	pior	(43,15%)	(48,29%)	(104,11%)	(93,64%)	(228,52%)	(167,54%)
500B (250 Mensagens)	padrão	(5,77%)	(4,86%)	(8,35%)	(1,19%)	(77,14%)	(11,46%)
	pior	(51,38%)	(49,23%)	(90,23%)	(73,77%)	(259,42%)	(166,67%)
2000B (1.000 Mensagens)	padrão	(0,00%)	(2,65%)	(3,48%)	(0,65%)	(72,13%)	(5,68%)
	pior	(23,66%)	(49,23%)	(62,27%)	(39,64%)	(183,19%)	(96,48%)

A Tabela 5.7 exibe os valores equivalentes para as configurações dos parâmetros *binary* e *L*. Para todas as configurações do parâmetro *binary* seu valor se encontra igual a *true*, ou seja, ativado. Já o valor referente ao número de cópias, por apresentar grande oscilação, ficou resumido por um cálculo simples que leva em consideração a quantidade máxima de mensagens que podem ser armazenadas no *buffer* daquele nó (representado neste momento por *b*). Como exemplo, vamos supor que a densidade seja igual a 0 (D 0) e o percentual de ocupação do *buffer* seja $> 100\%$ (o equivalente a 200B até 2000B). Seguindo a referência de cálculo da Tabela 5.7, indica-se a utilização de $L = 50$.

Tabela 5.7: Relação entre os parâmetros *binary* e *L* e as características do cenário (densidade local e carga de mensagens na rede)

Carga de Mensagens na Rede	Densidade 0		Densidade 1		Densidade 2	
	<i>binary</i>	<i>L</i>	<i>binary</i>	<i>L</i>	<i>binary</i>	<i>L</i>
$\leq 100\%$	<i>true</i>	$\lceil L = b * 2 \rceil$	<i>true</i>	$\lceil L = b/12 \rceil$	<i>binary</i>	$L = b$
$> 100\%$	<i>true</i>	$\lceil L = b/20 \rceil$	<i>true</i>	$\lceil L = b/20 \rceil$	<i>binary</i>	$\lceil L = b/20 \rceil$

5.3.3 MaxProp

A Tabela 5.8 sintetiza os valores dos protocolo MaxProp em suas diversas combinações de densidade e ocupação de *buffer*, onde *MaxSize* corresponde ao tamanho do vetor *delivery likelihood* e α o valor associado a este parâmetro.

Analisando os resultados de todas as simulações deste protocolo observou-se a ocorrência do parâmetro *MaxSize* = 75 em 73,40%, e em apenas 13,33% com *MaxSize* = 50, seu valor padrão. Demonstrando, portanto, a necessidade de um acréscimo no tamanho do vetor de probabilidades. O aumento deste vetor representa uma maior alocação da quantidade de posições para armazenamento das probabilidades (o que não gera grande adição de espaço de armazenamento por ser um vetor de índices). Esse impacto mostrou uma análise interessante uma vez que é percebido que o valor de $\alpha > 1$ (onde $\alpha = 1$ é o padrão) está presente em mais de 90% das simulações, o que demonstra que deve-se dar maior relevância aos últimos encontros, mantendo assim os valores de probabilidade de entrega mais atualizados.

Tabela 5.8: Percentual de mensagens entregues do MaxProp - Configuração com melhor desempenho

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	MaxSize	75	75	75	75	75	75
	α	9	9	9	9	9	9
	λ	12,20%	98,00%	75,20%	90,00%	100,00%	81,20%
50B (250 e 500 Mensagens)	MaxSize	25	75	50	75	75	75
	α	0.01	9	1	9	9	9
	λ	7,72%	96,00%	58,00%	89,10%	100,00%	74,40%
100B (50, 250, 500 e 1.000 Mensagens)	MaxSize	50	75	50	75	75	75
	α	500	9	0.01	9	9	9
	λ	8,56%	96,75%	60,05%	89,35%	100,00%	75,90%
200B (500 e 1.000 Mensagens)	MaxSize	75	75	500	75	75	75
	α	10	9	0.1	9	9	9
	λ	6,93%	96,50%	52,66%	89,00%	100,00%	74,00%
500B (250 Mensagens)	MaxSize	75	75	100	75	250	75
	α	7	9	0.01	9	500	0.01
	λ	8,10%	97,00%	51,20%	89,40%	97,80%	74,40%
2000B (1.000 Mensagens)	MaxSize	75	75	25	75	100	1000
	α	20	9	0.01	C9	100	0.1
	λ	5,60%	97,00%	37,70%	61,60%	72,80%	53,20%

λ - corresponde ao percentual de mensagens entregues pela configuração.

A Tabela 5.9 demonstra o quão pequena foi a oscilação deste protocolo em grande parte das simulações, sendo possível notar uma maior oscilação a partir do momento em que a capacidade de armazenamento se torna menor, a partir da carga de mensagens na rede igual a 200B.

Tabela 5.9: Percentual de deficiência da configuração padrão e com pior desempenho para mensagens entregues do MaxProp

Capacidade de Armazenamento por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B, e 20B (50 Mensagens)	padrão	(0,00%)	(0,61%)	(0,16%)	(0,00%)	(0,07%)	(0,33%)
	pior	(0,61%)	(3,48%)	(0,80%)	(12,17%)	(2,79%)	(0,36%)
50B (250 e 500 Mensagens)	padrão	(0,00%)	(4,31%)	(1,47%)	(0,45%)	(2,34%)	(4,37%)
	pior	(0,44%)	(20,00%)	(12,32%)	(9,32%)	(4,88%)	(7,04%)
100B (50, 250, 500 e 1.000 Mensagens)	padrão	(1,12%)	(14,24%)	(4,89%)	(5,14%)	(4,93%)	(2,05%)
	pior	(3,86%)	(28,29%)	(15,34%)	(9,12%)	(5,23%)	(6,90%)
200B (500 e 1.000 Mensagens)	padrão	(0,55%)	(34,54%)	(1,06%)	(3,03%)	(5,13%)	(3,89%)
	pior	(3,15%)	(43,29%)	(2,23%)	(4,64%)	(8,88%)	(3,14%)
500B (250 Mensagens)	padrão	(0,75%)	(25,00%)	(7,49%)	(2,19%)	(7,14%)	(4,88%)
	pior	(2,27%)	(49,34%)	(17,23%)	(6,17%)	(19,42%)	(6,47%)
2000B (1.000 Mensagens)	padrão	(0,90%)	(22,03%)	(7,88%)	(2,65%)	(3,04%)	(5,05%)
	pior	(4,67%)	(66,67%)	(22,27%)	(9,61%)	(37,50%)	(7,26%)

Relaciona-se na Tabela 5.10 os valores equivalentes para os parâmetros *MaxSize*, que representa o tamanho do vetor *delivery likelihood* e α . Nota-se que quanto maior a densidade da rede, maior o valor de α , indicando maior relevância aos últimos encontros em cenários mais densos.

Tabela 5.10: Relação entre os parâmetros *MaxSize* e α e as características do cenário (densidade local e carga de mensagens na rede)

Carga de Mensagens na Rede	Densidade 0		Densidade 1		Densidade 2	
	<i>MaxSize</i>	α	<i>MaxSize</i>	α	<i>MaxSize</i>	α
$\leq 100\%$	50	1	75	9	75	50
$> 100\%$	75	9	75	9	75	100

5.3.4 RAPID

O protocolo RAPID teve sua variação baseada em relação às suas métricas e não em parâmetros específicos. Como indicação recebida pela lista de discussão do simulador The ONE, foi adotada a métrica *minimizing average delay*, que visa minimizar o atraso

médio, como a configuração padrão. Sendo assim, sob a ótica do percentual de mensagens entregues era esperado que a métrica *minimizing missed deadlines* resultasse em um melhor desempenho, como veio a se confirmar, ocorrendo em 63,43% das simulações. Isso ocorre uma vez que essa métrica objetivamente tenta minimizar o número de mensagens perdidas por expiração.

É interessante notar a partir da Tabela 5.12, que houve uma pequena oscilação em relação a configuração considerada como padrão, porém em relação à métrica mais conservadora adotada por este protocolo (a saber, *minimizing maximum delay*¹), se manteve sempre como a de pior desempenho, apresentando oscilações significativas.

Tabela 5.11: Percentual de mensagens entregues do RAPID - Configuração com melhor desempenho

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	λ	<i>Minimized Missed Deadlines</i> 10,60%	99,80%	34,40%	80,00%	100,00%	77,20%
50B (250 e 500 Mensagens)	λ	7,85%	99,30%	26,60%	83,70%	100,00%	71,60%
100B (50, 250, 500 e 1.000 Mensagens)	λ	7,86%	98,55%	51,25%	82,95%	100,00%	72,80%
200B (500 e 1.000 Mensagens)	λ	6,50%	98,50%	46,30%	83,70%	99,80%	74,50%
500B (250 Mensagens)	λ	7,85%	98,80%	50,40%	83,60%	74,00%	71,40%
2000B (1.000 Mensagens)	λ	6,15%	98,60%	37,00%	56,40%	51,20%	61,60%

λ - corresponde ao percentual de mensagens entregues pela configuração.

¹Onde uma mensagem somente será replicada quando seu atraso é estimado como o máximo em relação às demais mensagens que se encontram no *buffer*.

Tabela 5.12: Percentual de deficiência da configuração padrão e com pior desempenho para mensagens entregues do RAPID

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B, e 20B (50 Mensagens)	padrão	(0,95%)	(0,61%)	(3,69%)	(9,78%)	(0,00%)	(9,66%)
	pior	(24,71%)	(3,48%)	(51,47%)	(62,90%)	(40,45%)	(59,20%)
50B (250 e 500 Mensagens)	padrão	(4,57%)	(4,31%)	(0,00%)	(3,59%)	(0,81%)	(10,84%)
	pior	(38,11%)	(21,00%)	(75,67%)	(84,77%)	(131,48%)	(146,90%)
100B (50, 250, 500 e 1.000 Mensagens)	padrão	(3,15%)	(1,24%)	(1,18%)	(5,40%)	(0,81%)	(10,47%)
	pior	(31,66%)	(28,29%)	(71,40%)	(79,93%)	(99,20%)	(117,64%)
200B (500 e 1.000 Mensagens)	padrão	(3,33%)	(4,54%)	(0,54%)	(5,42%)	(0,81%)	(10,23%)
	pior	(29,17%)	(23,29%)	(88,98%)	(92,41%)	(139,32%)	(158,55%)
500B (250 Mensagens)	padrão	(4,67%)	(4,65%)	(0,00%)	(2,71%)	(0,00%)	(10,87%)
	pior	(41,44%)	(24,34%)	(70,61%)	(74,48%)	(70,51%)	(124,82%)
2000B (1.000 Mensagens)	padrão	(2,50%)	(1,03%)	(0,00%)	(0,00%)	(0,00%)	(0,00%)
	pior	(28,12%)	(34,39%)	(60,87%)	(26,70%)	(22,55%)	(61,59%)

É relevante citar que cada uma das métricas do protocolo RAPID tem um objetivo bem claro, e neste momento traçamos um comparativo entre todas as métricas sob a ótica da taxa de entrega de mensagens.

Para este protocolo suprimimos a tabela de configuração de parâmetros uma vez que em todas as simulações a métrica *minimizing missed deadlines* obteve melhor desempenho.

5.3.5 EBR

A célula em destaque da Tabela 5.13 representa o valor de $\alpha = 0,50$, referente ao peso dado ao CWC do último encontro. Já o $L = 20$ equivale ao número de cópias geradas igual a vinte.

A partir dos dados apresentados na Tabela 5.13 é possível notar a presença na maioria das simulações da configuração de melhor desempenho cujo número de cópias foi igual a vinte ($L = 20$), reafirmando a necessidade de se gerar cópias e inserir um número tão relevante quanto possível. Como afirmado pelo artigo de proposta deste protocolo [40], o valor de $\alpha = 0,85$ esteve presente em 68,89% das simulações, o que

reafirma a necessidade de um maior peso a ser atribuído aos últimos encontros.

Tabela 5.13: Percentual de mensagens entregues do EBR - Configuração com melhor desempenho

Carga de Mensagens da Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	α	0,50	0,85	0,85	0,85	0,85	0,85
	L	20	20	20	20	20	20
	λ	11,00%	96,00%	61,40%	89,60%	84,00%	79,60%
50B (250 e 500 Mensagens)	α	0,85	0,50	0,85	0,85	0,85	0,50
	L	20	20	20	20	20	20
	λ	7,20%	95,25%	48,65%	87,70%	84,30%	73,70%
100B (50, (250, 500 e 1.000 Mensagens)	α	0,85	0,85	0,85	0,85	0,85	0,50
	L	20	20	20	20	20	20
	λ	7,89%	95,50%	50,45%	88,35%	83,55%	75,30%
200B (500 e 1.000 Mensagens)	α	0,85	0,85	0,85	0,85	0,85	0,50
	L	20	20	20	20	20	20
	λ	6,47%	95,25%	44,75%	88,00%	82,50%	74,00%
500B (250 Mensagens)	α	0,85	0,85	0,50	0,85	0,85	0,50
	L	20	20	20	20	20	20
	λ	7,65%	95,50%	48,90%	87,80%	85,00%	73,60%
2000B (1.000 Mensagens)	α	0,85	0,85	0,50	0,50	0,85	0,50
	L	20	20	20	5	20	20
	λ	5,80%	96,00%	35,30%	71,20%	70,20%	62,40%

λ - corresponde ao percentual de mensagens entregues pela configuração.

As maiores oscilações em relação a configuração padrão e a de pior desempenho são notadas no cenário de simulação ZebraNet com densidade igual a oito (D 8), como pode ser visualizado na Tabela 5.14 onde o número de cópias disponível na rede se mostrou de suma importância. Isso ocorre devido ao aumento da densidade, e consequentemente a recorrência de encontros, onde o fator número de cópias se torna ainda mais importante.

Tabela 5.14: Percentual de deficiência da configuração padrão e com pior desempenho para mensagens entregues do EBR

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B, e 20B (50 Mensagens)	padrão	(1,85%)	(1,05%)	(6,60%)	(0,45%)	(23,53%)	(0,51%)
	pior	(13,40%)	(7,87%)	(19,92%)	(2,75%)	(77,97%)	(4,19%)
50B (250 e 500 Mensagens)	padrão	(0,70%)	(2,42%)	(6,92%)	(0,34%)	(20,26%)	(0,68%)
	pior	(6,67%)	(7,93%)	(23,01%)	(2,45%)	(63,37%)	(4,69%)
100B (50, 250, 500 e 1.000 Mensagens)	padrão	(1,15%)	(1,73%)	(6,84%)	(0,40%)	(20,74%)	(0,67%)
	pior	(9,43%)	(7,61%)	(21,57%)	(2,38%)	(65,94%)	(4,29%)
200B (500 e 1.000 Mensagens)	padrão	(1,41%)	(1,87%)	(6,42%)	(0,46%)	(20,09%)	(0,82%)
	pior	(8,38%)	(7,32%)	(22,94%)	(2,44%)	(61,45%)	(4,08%)
500B (250 Mensagens)	padrão	(0,00%)	(2,14%)	(5,39%)	(0,23%)	(19,05%)	(0,55%)
	pior	(6,25%)	(7,91%)	(18,12%)	(2,33%)	(63,46%)	(4,84%)
2000B (1.000 Mensagens)	padrão	(0,00%)	(1,59%)	(1,15%)	(1,14%)	(11,08%)	(1,96%)
	pior	(5,45%)	(7,26%)	(6,65%)	(2,01%)	(40,40%)	(3,31%)

A Tabela 5.15 exibe os parâmetros correlacionados para as características de densidade e ocupação de *buffer*.

Tabela 5.15: Relação entre os parâmetros L e α e as características do cenário (densidade local e carga de mensagens na rede)

Carga de Mensagens na Rede	Densidade 0		Densidade 1		Densidade 2	
	L	α	L	α	L	α
$\leq 100\%$	20	0.50	20	0.50	20	0.85
$> 100\%$	20	0.85	11	0.85	11	0.85

5.3.6 NECTAR

Como já citado, o protocolo de roteamento NECTAR proposto em [44, 46] foi implementado no simulador utilizado por este trabalho, o The ONE, como forma de contribuição aos pesquisadores da área. Seu código-fonte encontra-se no Anexo B deste trabalho.

A partir da Tabela 5.16 notamos os altos valores de entrega praticados pelo NEC-

TAR. Esse protocolo não utilizou de variações em seus valores de parâmetros, o que afirma sua relevante capacidade de adaptação ao contexto em que se encontra utilizando sua seleção de funcionamento característico de sua classificação híbrida (funcionando ora em modo epidêmico, ora em modo probabilístico). Essa observação fica mais clara na sub-seção 5.3.7.

Tabela 5.16: Percentual de mensagens entregues do NECTAR - Configuração padrão (única)

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	λ	20,00%	98,00%	84,00%	90,00%	100,00%	80,00%
50B (250 e 500 Mensagens)	λ	14,00%	96,05%	60,00%	90,00%	98,00%	75,01%
100B (50,250, 500 e 1.000 Mensagens)	λ	20,30%	98,00%	84,00%	91,00%	100,00%	80,00%
200B (500 e 1.000 Mensagens)	λ	13,00%	97,00%	51,00%	90,00%	95,00%	75,00%
500B (250 Mensagens)	λ	13,06%	96,00%	53,00%	87,60%	73,00%	71,00%
2000B (1.000 Mensagens)	λ	10,02%	97,00%	27,00%	59,90%	52,00%	51,45%

λ - corresponde ao percentual de mensagens entregues pela configuração padrão do NECTAR.

5.3.7 Ocorrência de protocolos em relação à entrega

Mesmo não sendo o enfoque principal deste trabalho, foi realizada uma análise sobre a ocorrência dos protocolos em relação aos melhores resultados sob cada ambiente simulado. A partir da Tabela 5.17 é possível verificar o percentual de mensagens entregues por aquele protocolo que melhor se comportou naquele contexto. Por exemplo, com densidade igual a zero (D 0) e carga de mensagens na rede igual a 5B (célula em destaque), o NECTAR foi o protocolo que obteve maior percentual de entrega de mensagens em relação a todos os protocolos, com o total de 20,00% das mensagens entregues ao destino. Já a Tabela 5.18 resume o percentual total de vezes que cada um

dos protocolos apareceu como o de melhor resultado².

Tabela 5.17: Percentual de mensagens entregues para o protocolo com melhor desempenho

Carga de Mensagens na Rede por Densidade		ZebraNet	WDM	ZebraNet	DieselNet	ZebraNet	DieselNet
		D 0	D 1	D 2	D 4	D 8	D 9
5B, 10B e 20B (50 Mensagens)	λ	NECTAR 20,00%	RAPID 99,00%	NECTAR 84,00%	NECTAR, SnW* e MaxProp 90,00%	NECTAR, SnW*, PRoPHET RAPID, MaxProp 100,00%	SnW* MaxProp 100%
50B (250 e 500 Mensagens)	λ	NECTAR 14,00%	RAPID 99,30%	NECTAR 60,00%	NECTAR 90,00%	PRoPHET, RAPID MaxPROP 100%	NECTAR 75,01%
100B (50, 250, 500 e 1.000 Mensagens)	λ	NECTAR 20,30%	RAPID 98,55%	NECTAR 84,00%	NECTAR 91,00%	NECTAR, PRoPHET RAPID, MaxProp 100%	NECTAR 80,00%
200B (500 e 1.000 Mensagens)	λ	NECTAR 13,00%	RAPID 98,50%	MaxProp 52,66%	NECTAR 90,00%	MaxProp 100%	NECTAR 75,00%
500B (250 Mensagens)	λ	NECTAR 13,06%	RAPID 98,80%	NECTAR 53,00%	EBR 87,80%	SnW* 99,20%	MaxProp 74,40%
2000B (1.000 Mensagens)	λ	NECTAR 10,02%	RAPID 98,60%	MaxProp 37,70%	EBR 71,20%	SnW* 84,00%	EBR 62,40%

λ - corresponde ao percentual de mensagens entregues pela configuração.

* - SnW equivale ao protocolo *Spray and Wait*, abreviado por falta de espaço.

Podemos destacar que o protocolo que melhor se adaptou aos ambientes, obtendo assim o melhor resultado em relação à taxa de mensagens entregues foi o NECTAR. Lembramos ainda que esse protocolo foi simulado somente em sua configuração padrão, o que demonstra a sua maior habilidade de percepção de contexto em relação aos demais protocolos.

²Somando-se os valores de ocorrência dos protocolos em termos percentuais nota-se que ultrapassa o valor total de 100%, isso acontece por conta do empate ocorrido em alguns momentos

Tabela 5.18: Percentual de aparição de cada protocolo para mensagens entregues como melhor desempenho

Protocolo	Porcentagem de ocorrência
PRoPHET	9,37%
<i>Spray and Wait</i>	15,62%
MaxProp	28,12%
RAPID	28,12%
EBR	9,37%
NECTAR	59,32%

5.3.8 Análise sobre a deficiência dos protocolos

A Tabela 5.19 mensura a dimensão da oscilação que as diversas configurações dos protocolos avaliados apresentaram durante as simulações, exibindo, por cenário, os piores resultados da configuração padrão e da configuração que obteve o pior desempenho. Os valores representam, em termos percentuais, o quanto as respectivas configurações deveriam melhorar para se igualar à configuração de cada protocolo que obteve o melhor desempenho. Optamos por expor somente a taxa de mensagens entregues e a latência média, uma vez que os resultados das demais métricas avaliadas não acrescentaram informações significativas.

Tabela 5.19: Percentual de deficiência dos protocolos

Protocolo	Configuração	Percentual de Mensagens Entregues			Latência Média		
		DieselNet	WDM	ZebraNet	DieselNet	WDM	ZebraNet
<i>Spray and Wait</i>	Padrão	(4,10%)	(3,12%)	(24,59%)	(8,09%)	(851,01%)	(468,64%)
	Pior Desempenho	(177,04%)	(66,67%)	(506,58%)	(116,11%)	(3.161,28%)	(3.411,42%)
PRoPHET	Padrão	(10,82%)	(66,19%)	(41,37%)	(33,55%)	(94,78%)	(83,74%)
	Pior Desempenho	(36,04%)	(73,94%)	(100,00%)	(41,42%)	(181,64%)	(257,57%)
MaxProp	Padrão	(5,14%)	(25,00%)	(8,88%)	(5,93%)	(67,17%)	(27,36%)
	Pior Desempenho	(9,61%)	(66,67%)	(37,50%)	(10,68%)	(68,40%)	(39,61%)
RAPID	Padrão	(10,87%)	(18,23%)	(21,30%)	(12,14%)	(11,03%)	(42,26%)
	Pior Desempenho	(158,55%)	(83,73%)	(234,34%)	(93,69%)	(81,38%)	(375,48%)
EBR	Padrão	(1,96%)	(6,38%)	(30,00%)	(3,74%)	(37,13%)	(52,81%)
	Pior Desempenho	(4,84%)	(8,70%)	(105,26%)	(15,00%)	(138,38%)	(134,49%)

Embora o protocolo *Spray and Wait* tenha demonstrado a maior oscilação, esse comportamento está relacionado a simulações cujas configurações o parâmetro L tenha assumido o valor 1. Nesses casos, independente do valor associado ao parâmetro *binary*, o protocolo *Spray and Wait* comporta-se como o protocolo DD (*Direct Delivery*), ou seja, a estação de origem produz uma única cópia da mensagem que somente será encaminhada quando e se a estação de destino for encontrada. Essa observação ratifica que as características das DTNs requerem o envio de mais de uma cópia da mesma mensagem, de forma a otimizar todas as métricas. Podemos, ainda, destacar que os valores de latência média expressivos dos cenários WDM e ZebraNet, informados na Tabela 5.19, foram extraídos de um conjunto reduzido de simulações, cuja característica principal do contexto avaliado era a alta densidade (densidade igual a, respectivamente, 15 e 8) e uma taxa de ocupação da área de armazenamento menor ou igual a 100%. Nesse cenário com restrições mínimas, configurações com L reduzido obtiveram uma latência média significativamente superior em função da quantidade reduzida de cópias das mensagens na rede. Como praticamente inexitem restrições, quanto maior o valor de L , menor a latência, entretanto esse resultado implica em um comportamento similar ao do protocolo *Epidemic* [57], que pôde ser comprovado ao analisarmos a relação entre mensagens entregues e mensagens transmitidas. No entanto, deve-se ressaltar que o protocolo *Spray and Wait* obteve uma colocação melhor que o PRoPHET e o EBR, como

pode ser visto na Tabela 5.18.

Não obstante o protocolo P_{Ro}PHET ter apresentado, em média, oscilações inferiores ao protocolo *Spray and Wait*, coube a ele o pior desempenho em termos do percentual de mensagens entregues, entre outras métricas. O algoritmo que efetua cálculo da métrica *Delivery Predictability* tem se mostrado ineficiente quando comparado com outros protocolos [1, 46, 41], apresentando resultados muitas vezes inexpressivos. Podemos observar que o algoritmo do protocolo P_{Ro}PHET baseia-se na quantidade de unidades de tempo para efetuar o envelhecimento de rotas, enquanto que o protocolo MaxProp efetua o envelhecimento à medida que estabelece novos contatos. Logo, em cenários com grande mobilidade, o protocolo MaxProp demonstra uma capacidade superior de manter suas rotas atualizadas, evitando o envio de mensagens por rotas obsoletas. Oliveira e Albuquerque em [46] afirmam que, em alguns cenários de grande mobilidade, o P_{Ro}PHET mostra-se incapaz de lidar com o dinamismo da rede, o que implica em uma convergência lenta da tabela de rotas e, em redes veiculares, à medida que a quantidade de nós e a quantidade de mensagens aumenta, a eficiência é reduzida [41].

A análise das informações presentes na Tabela 5.19 nos permite concluir também que o protocolo MaxProp apresentou, junto com o protocolo RAPID e o EBR, uma das menores oscilações entre os protocolos avaliados. Tanto o protocolo MaxProp quanto o protocolo RAPID aparecem em segundo lugar quando avaliados sob a métrica de melhor desempenho para mensagens entregues, conforme a Tabela 5.18 exposta anteriormente. Atribuímos ao MaxProp seus relevantes resultados principalmente por seu algoritmo de cálculo de custo que mesmo utilizando os valores padrão para cada parâmetro mostra a habilidade de priorizar o envio de mensagens através de nós contactados mais recentemente e de envelhecer rapidamente as demais rotas, reduzindo o tempo de convergência da tabela de roteamento e reduzindo assim sua latência. E ao RAPID por sua métrica preocupada com a perda de mensagens por expiração.

5.3.9 Análise de ocorrência da configuração padrão

Tabela 5.20: Percentual da configuração padrão com melhor desempenho

Protocolo	Carga de Mensagens na Rede	DieselNet	WDM	ZebraNet	Média Parcial	Média Global
SprayAndWait	≤ 100%	50,00%	46,67%	0,00%	32,23%	26,69%
	> 100%	0,00%	52,00%	11,43%	21,14%	
PRoPHET	≤ 100%	25,00%	11,11%	7,14%	14,42%	7,21%
	> 100%	0,00%	0,00%	0,00%	0,00%	
MaxProp	≤ 100%	100,00%	88,89%	66,67%	85,19%	65,14%
	> 100%	40,00%	86,67%	8,57%	45,08%	
RAPID	≤ 100%	0,00%	38,32%	14,28%	17,70%	40,60%
	> 100%	56,69%	87,33%	46,49%	63,50%	
EBR	≤ 100%	0,00%	46,67%	5,26%	17,31%	18,19%
	> 100%	0,00%	52,00%	5,26%	19,08%	

A Tabela 5.20 exibe o percentual de vezes que a configuração padrão de cada um dos protocolos avaliados apresentou o melhor desempenho. Os dados demonstram a inabilidade dos protocolos, principalmente do PRoPHET, do EBR e do *Spray and Wait*, para lidar com os diversos contextos que se formaram ao longo dos cenários selecionados. Coube à configuração padrão do protocolo MaxProp um desempenho superior nesta comparação, mas ainda assim no cenário ZebraNet, com o capacidade de armazenamento acima de 100,0%, a média foi de apenas 8,57%. Já as configurações padrão do PRoPHET, do EBR e do *Spray and Wait* exibiram resultados irrisórios, corroborando a necessidade de se promover o desenvolvimento de mecanismos adaptativos a contextos para protocolos parametrizáveis.

Para a avaliação do protocolo RAPID houve a comparação entre suas três métricas disponíveis, e mesmo não sendo a intenção deste protocolo servir a todos os propósitos com uma única métrica, ele se mostrou com um valor significativo de aparições, se comparado aos demais protocolos. Já o protocolo EBR, assim como o PRoPHET, não obteve um grande número de ocorrências em sua configuração padrão, porém há de se destacar que a oscilação entre a configuração padrão e a configuração de melhor desempenho em relação ao percentual de mensagens entregues exposta pela Tabela 5.19, não foi tão expressiva para o protocolo EBR em relação a maioria dos cenários.

As Figuras 5.4 e 5.5 demonstram a média percentual de mensagens entregues pelos protocolos avaliados com a configuração padrão, cujos valores foram agrupados por carga de mensagens na rede menor ou igual 100% (onde o nó tem capacidade de armazenar todas as mensagens geradas inicialmente na rede) e maior que 100% (onde o nó não tem capacidade de armazenar todas as mensagens geradas inicialmente na rede), e identificados por Carga de Mensagens na Rede $\leq 100\%$ e por Carga de Mensagens na Rede $> 100\%$. O desempenho de todos os protocolos abordados são também comparados ao protocolo Epidemic [57]. A intenção dessa comparação é utilizar o protocolo Epidemic como um limiar de desempenho aos demais protocolos, exatamente por não haver nenhuma restrição, *a priori*, do próprio protocolo, somente àquelas oferecidas pelas próprias características da rede, como a limitação de espaço de armazenamento. Foram selecionados apenas alguns cenários cujas densidades são iguais a 1 (WDM), 4 (ZebraNet) e 9 (DieselNet), que encontram-se identificados, respectivamente, por D 1, D 4 e D 9. As barras verticais representam o intervalo de confiança de 95%.

Apesar do protocolo NECTAR ter apresentado em geral a maior ocorrência como o protocolo de melhor desempenho, conforme visto na Tabela 5.18, é possível observar a partir das Figuras 5.4 e 5.5 a oscilação dos protocolos avaliados conforme os cenários e suas características de densidade da rede e carga de mensagens na rede. Sendo assim, nos cenários aqui analisados, coube aos protocolos PRoPHET e ao EBR a alternância como o protocolo de pior desempenho. Onde, para Carga de Mensagens na Rede $\leq 100\%$, temos o EBR ocorrendo como àquele de pior desempenho, e quando a Carga de Mensagens na Rede é $> 100\%$, o PRoPHET destacando-se como o protocolo de pior desempenho.

Analisando separadamente o desempenho do protocolo Epidemic em relação aos demais, é possível perceber, que este protocolo tem um desempenho superior aos demais enquanto há espaço de armazenamento suficientemente grande para o armazenamento de todas as mensagens da rede ($\leq 100\%$), como pode ser observado na Figura 5.4. Da mesma forma, quando este espaço de torna restrito, seu desempenho decai de forma brusca, aparecendo como o protocolo de pior de desempenho para os cenários ZebraNet e DieselNet, como pode ser visto na Figura 5.4. Estes resultados são conhecidos da literatura [36] [6] e reforçam a necessidade de que protocolos de roteamento em DTNs realizem ponderações sobre o encaminhamento e replicação de suas mensagens, levando em consideração as diversas características dos contextos e dos próprios nós (*e.g.* espaço de armazenamento, energia disponível, taxa de transferência etc).

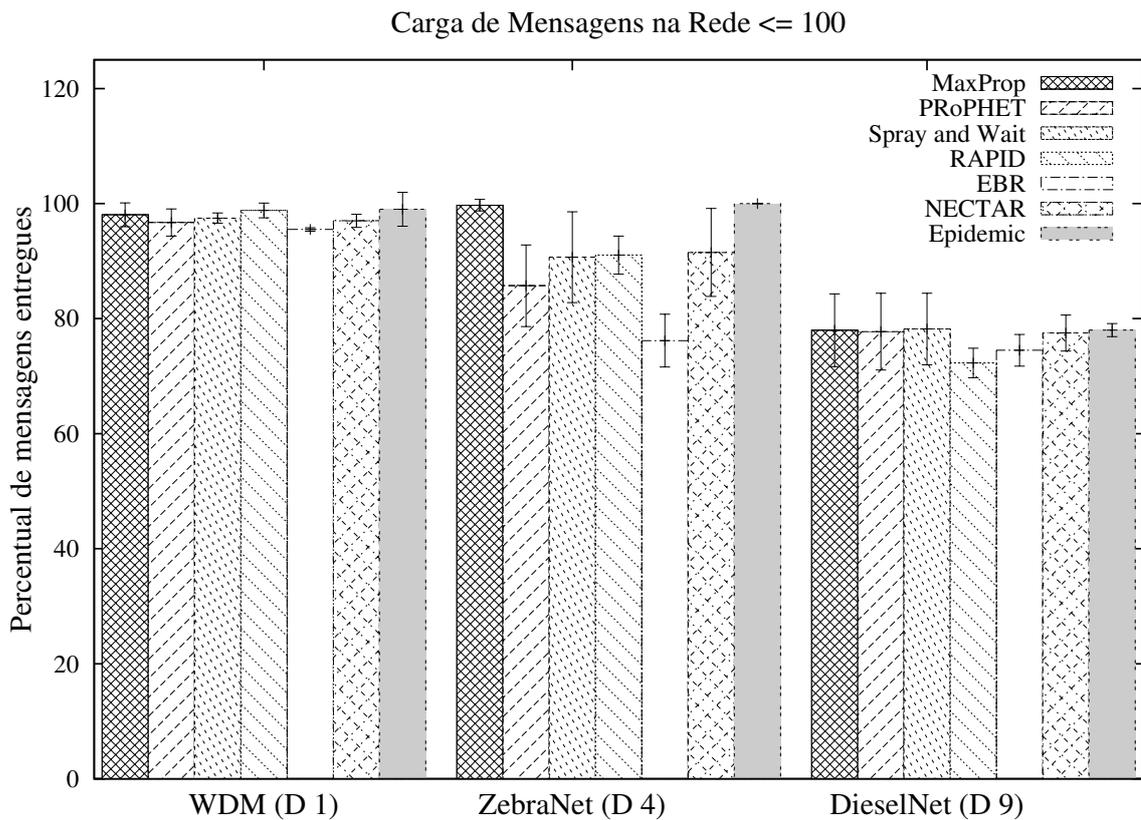


Figura 5.4: Carga de Mensagens na Rede $\leq 100\%$

Na Figura 5.4, para Carga de Mensagens na Rede $\leq 100\%$, nota-se o destaque do protocolo MaxProp no cenário ZebraNet, porém uma aproximação dos resultados de todos os protocolos para o cenário WDM. Já o protocolo EBR apresenta uma grande oscilação no cenário ZebraNet em relação aos demais protocolos neste cenário irrestrito (Carga de Mensagens na Rede $\leq 100\%$), consequência de sua limitação no número de cópias de mensagens e conservadora política de encaminhamento de cópias.

Já na Figura 5.5, cuja Carga de Mensagens na Rede é $> 100\%$, destaca-se o detrimento do desempenho do PRoPHET em relação aos demais protocolos avaliados, assim como é possível notar a melhora de desempenho do protocolo EBR, uma vez que as restrições impostas pelo contexto demonstram que sua política conservadora agora se torna mais coerente. É relevante ainda observar que os protocolos em maior destaque são o NECTAR, MaxProp e RAPID.

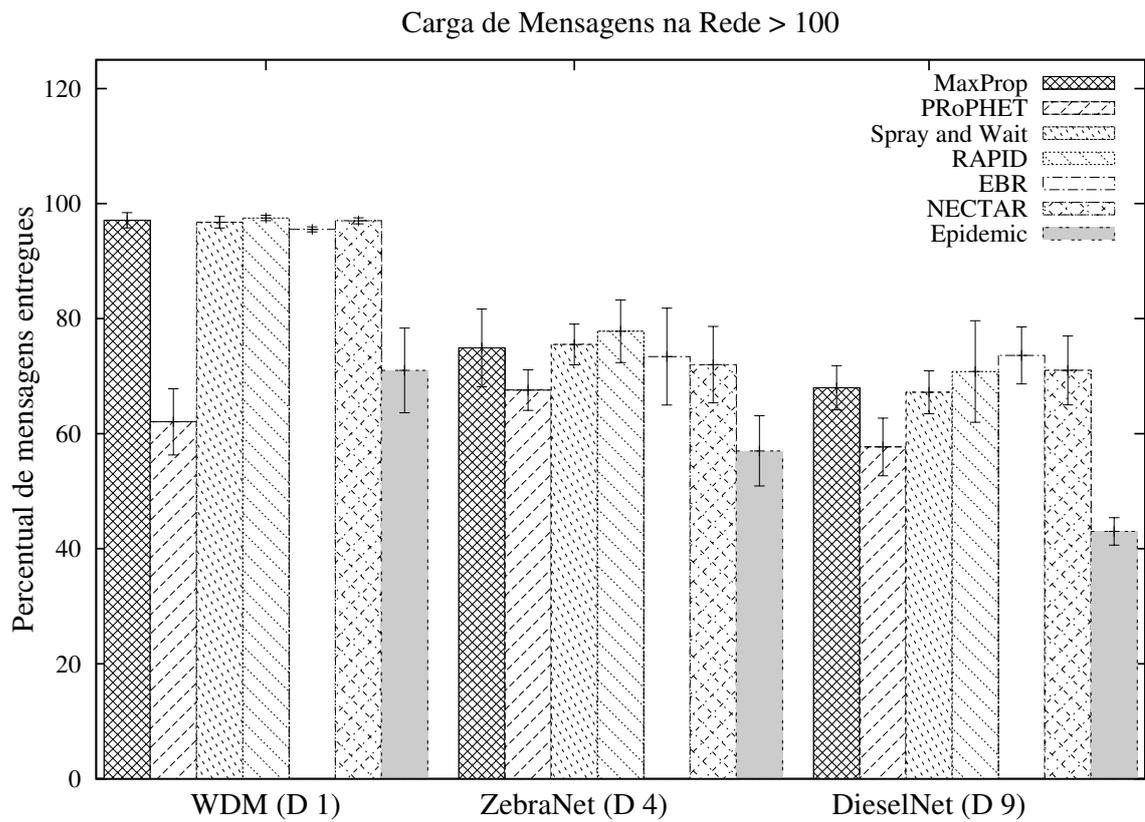


Figura 5.5: Carga de Mensagens na Rede > 100%

5.4 Conclusão

Este capítulo apresentou os resultados e análises dos experimentos realizados por este trabalho. Foi possível mensurar o ganho e a deficiência de cada um dos protocolos como esperado. Confirmando assim a necessidade de adaptação a contextos pelos protocolos de roteamento em DTNs de forma a ampliar sua abrangência e relevância aos diversos cenários contrastantes que podem ser formar.

Capítulo 6

Considerações Finais

Foram abordados por este trabalho tanto os conceitos básicos ao funcionamento de DTNs quanto as mais relevantes características necessárias para a avaliação de protocolos de roteamento em DTNs. Também foram expostos os principais tipos de contato existentes no ambiente de Redes Tolerantes a Atraso e Desconexões, os cenários formados e os ambiente de aplicação de DTNs mais citados na literatura. Este trabalho ainda abordou, como foco principal, uma grande gama de protocolos de roteamento de extrema relevância atualmente para esta área de pesquisa, classificando-os conforme uma taxonomia que facilitou o entendimento de cada um deles. Foi ainda mensurado, a partir de simulações, o impacto de diversificadas configurações de cenários para as métricas de taxa de entrega de mensagens e latência de cada um dos protocolos selecionados.

Acredita-se que a partir de tal trabalho esteja confirmada a necessidade de uma modelagem mais ampla dos protocolos de roteamento para DTNs, uma vez que foi demonstrada que há a necessidade de protocolos sensíveis ao contexto em que se encontram. Foi confirmada a necessidade de que protocolos de roteamento devem observar as características tanto locais ao nó (*e.g.* espaço de armazenamento disponível, potência de transmissão, tempo de contato, tempo entre contatos, energia restante etc.) quanto da rede como um todo, para que assim possam tomar decisões mais precisas sobre o ajuste dinâmico dos valores de seus parâmetros a fim de melhorar o desempenho em relação às métricas. Protocolos que apresentam alguma forma de adaptação a contexto, como é o caso do protocolo NECTAR, se mostram muito mais eficazes que os demais. No caso do protocolo NECTAR, este apresentou melhor desempenho que os demais protocolos em 59,32% das simulações. Se comparado o desempenho do protocolo NECTAR com aqueles que apresentaram menor poder de adaptação (que foi o caso do PROPHET e do EBR), temos que o protocolo NECTAR foi 49,95% melhor que

estes. Pode-se citar ainda que, conforme ajustes realizados nos valores dos parâmetros de cada protocolo, estes obtiveram ganhos de mais de 100% em relação à configuração padrão.

Pretende-se estender este trabalho com a proposta de um protocolo de roteamento para DTNs que seja sensível a contexto e tenha como foco aplicações da crescente área de redes sociais sem fio, redes estas formadas principalmente por dispositivos móveis como: *smartphones*, *notebooks* etc. Como principal motivação para tal continuidade se tem os resultados expostos, onde a necessidade de adaptação se faz clara a partir do ganho observado com a mudança dos valores associados a cada parâmetro dos protocolos de roteamento DTN. Para que este desenvolvimento seja possível, tem-se como etapa inicial a análise do ambiente de redes sociais e geração ou adaptação de um modelo de movimentação com suas características particulares. Em um segundo momento pretende-se que seja realizado o desenvolvimento ou adaptação de um protocolo de roteamento DTN que seja sensível às características de cada contexto em que se encontra. Após tais etapas, acreditamos ser possível propor melhores técnicas e diretrizes para protocolos de roteamento em DTNs, assim como um protocolo que apresente desempenho satisfatório independentemente do ambiente em que seja aplicado.

Referências Bibliográficas

- [1] BALASUBRAMANIAN, A., LEVINE, B., AND VENKATARAMANI, A. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.* 37 (August 2007), 373–384.
- [2] BALASUBRAMANIAN, A., LEVINE, B. N., AND VENKATARAMANI, A. Enabling Interactive Web Applications in Hybrid Networks. In *Proc. ACM Mobicom* (September 2008), pp. 70–80.
- [3] BANERJEE, N., CORNER, M., TOWSLEY, D., AND LEVINE, B. N. CRAWDAD trace umass/diesel/transfer/fall2007 (v. 2008-09-14). Downloaded from <http://crawdad.cs.dartmouth.edu/umass/diesel/transfer/fall2007>, Sept. 2008.
- [4] BANERJEE, N., CORNER, M., TOWSLEY, D., AND LEVINE, B. N. CRAWDAD trace umass/diesel/transfer/fall2007 (v. 2008-09-14). Downloaded from <http://crawdad.cs.dartmouth.edu/umass/diesel/transfer/fall2007>, Sept. 2008.
- [5] BANERJEE, N., CORNER, M. D., AND LEVINE, B. N. Design and Field Experimentation of an Energy-Efficient Architecture for DTN Throwboxes. *IEEE/ACM Transactions on Networking* 18, 2 (April 2010), 554–567.
- [6] BURGESS, J., GALLAGHER, B., JENSEN, D., AND LEVINE, B. N. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *In Proc. IEEE INFOCOM* (2006).
- [7] BURLEIGH, S., HOOKE, A., TORGERSON, L., FALL, K., CERF, V., DURST, B., SCOTT, K., AND WEISS, H. Delay-tolerant networking: an approach to interplanetary Internet. *IEEE Communications Magazine* 41, 6 (2003), 128–136.
- [8] BURNS, B., BROCK, O., AND LEVINE, B. N. MORA Routing and Capacity Building in Disruption-Tolerant Networks. *Elsevier Ad hoc Networks Journal* 6, 4 (June 2008), 600–620.
- [9] CERF, V., BURLEIGH, S., HOOKE, A., TORGERSON, L., DURST, R., SCOTT, K., FALL, K., AND WEISS, H. RFC 4838, Delay-Tolerant Networking Architecture. *IRTF DTN Research Group* (2007).
- [10] DEFRAWY, K. E., SOLIS, J., AND TSUDIK, G. Leveraging social contacts for message confidentiality in delay tolerant networks. *Computer Software and Applications Conference, Annual International 1* (2009), 271–279.
- [11] DEMMER, M., BREWER, E., FALL, K., HO, M., AND PATRA, R. Implementing delay tolerant networking. Tech. rep., 2003.
- [12] DTNRG. DTN research group. <http://www.dtnrg.org/>, 2006.

- [13] DURST, R. C., FEIGHERY, P. D., AND SCOTT, K. L. Why not use the standard internet suite for the interplanetary internet. In *Interplanetary Internet Study Seminar, California Institute of Technology* (1999), Corporation.
- [14] EKMAN, F., KERÄNEN, A., KARVO, J., AND OTT, J. Working day movement model. In *Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models* (New York, NY, USA, 2008), MobilityModels '08, ACM, pp. 33–40.
- [15] FALL, K. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2003), ACM, pp. 27–34.
- [16] FMS. First Mile Solutions. <http://www.firstmilesolutions.com>, 2007.
- [17] GROSSGLAUSER, M., AND TSE, D. Mobility increases the capacity of ad-hoc wireless networks. *IEEE/ACM Transactions on Networking* 10 (2002), 477–486.
- [18] GUO, S., DERAKHSHANI, M., FALAKI, M. H., ISMAIL, U., LUK, R., OLIVER, E. A., RAHMAN, S. U., SETH, A., ZAHARIA, M. A., AND KESHAV, S. Design and implementation of the KioskNet system. *Computer Networks* (August 2010).
- [19] GUO, Z., WANG, B., AND CUI, J.-H. Prediction assisted single-copy routing in underwater delay tolerant networks. In *GLOBECOM* (2010), pp. 1–6.
- [20] HUANG, T.-K., LEE, C.-K., AND CHEN, L.-J. PROPHET+: An adaptive PROPHET-Based routing protocol for opportunistic network. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications* (Apr. 2010), IEEE, pp. 112–119.
- [21] HUI, P., CHAINTREAU, A., SCOTT, J., GASS, R., CROWCROFT, J., AND DIOT, C. Pocket switched networks and human mobility in conference environments. *ACM Press* (2005), 244–251.
- [22] IRTF. Internet research task force. <http://www.irtf.org/>, 2010.
- [23] JAIN, S., FALL, K., AND PATRA, R. Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.* 34 (August 2004), 145–158.
- [24] JOHNSON, D. B., AND MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In *MOBILE COMPUTING* (1996), Kluwer Academic Publishers, pp. 153–181.
- [25] JONES, E. P. C., LI, L., AND WARD, P. A. S. Practical routing in delay-tolerant networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (New York, NY, USA, 2005), WDTN '05, ACM, pp. 237–243.
- [26] JUANG, P., OKI, H., WANG, Y., MARTONOSI, M., PEH, L. S., AND RUBENSTEIN, D. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2002), ACM Press, pp. 96–107.

- [27] JUBIN, J., AND TORNOW, J. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE* 75(1): 21-32 (1987).
- [28] JUN, H., AMMAR, M. H., CORNER, M. D., AND ZEGURA, E. W. Hierarchical power management in disruption tolerant networks using traffic aware optimization. *Computer Communications* 32, 16 (October 2009), 1710–1723.
- [29] KARVO, J., AND OTT, J. Time scales and delay-tolerant routing protocols. In *CHANTS '08: Proceedings of the third ACM workshop on Challenged networks* (New York, NY, USA, 2008), ACM, pp. 33–40.
- [30] KERÄNEN, A., AND OTT, J. Increasing reality for DTN protocol simulations. Tech. rep., Helsinki University of Technology, July 2007.
- [31] KERÄNEN, A., OTT, J., AND KÄRKKÄINEN, T. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques* (New York, NY, USA, 2009), ICST.
- [32] LAKKAKORPI, J., PITKÄNEN, M., AND OTT, J. Adaptive routing in mobile opportunistic networks. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems* (New York, NY, USA, 2010), MSWIM '10, ACM, pp. 101–109.
- [33] LINDGREN, A., DIOT, C., AND SCOTT, J. Impact of communication infrastructure on forwarding in pocket switched networks. In *in Proc. (CHANTS (2006))*.
- [34] LINDGREN, A., AND DORIA, A. Probabilistic routing protocol for intermittently connected networks - draft-lindgren-dtnrg-prophet-03, 2007.
- [35] LINDGREN, A., DORIA, A., DAVIES, E., AND GRASIC, S. Probabilistic routing protocol for intermittently connected networks - draft-irtf-dtnrg-prophet-09, 2011.
- [36] LINDGREN, A., DORIA, A., AND SCHELÉN, O. Probabilistic Routing in Intermittently Connected Networks. *Lecture Notes in Computer Science* 3126 (January 2004), 239–254.
- [37] MAYER, C., AND WALDHORST, O. On the Impact of Graph Structure on Mobility in Opportunistic Mobile Networks. In *INFOCOM Workshop NetSciCom* (Apr. 2011), IEEE.
- [38] MISSEN, C., AND McNULTY, D. M. Widernet. <http://www.widernet.org>, 2000.
- [39] NAIDU, S., CHINTADA, S., SEN, M., AND RAGHAVAN, S. Challenges in deploying a delay tolerant network. In *Proceedings of the third ACM workshop on Challenged networks* (New York, NY, USA, 2008), CHANTS '08, ACM, pp. 65–72.
- [40] NELSON, S. C., BAKHT, M., AND KRAVETS, R. Encounter-Based routing in DTNs. In *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications* (Apr. 2009), IEEE, pp. 846–854.
- [41] NELSON, S. C., BAKHT, M., KRAVETS, R., AND HARRIS, III, A. F. Encounter: based routing in DTNs. *SIGMOBILE Mob. Comput. Commun. Rev.* 13 (June 2009), 56–59.

- [42] NUNES, C. M., DOTTI, F. L., AND OLIVEIRA, J. B. S. APRP-Group: Roteamento para redes DTN com repasse baseado em agrupamento de nodos por potencial de entrega. In *SBRC* (2010), pp. 451–464.
- [43] OLIVEIRA, C. T., MOREIRA, M. D. D., RUBINSTEIN, M. G., COSTA, L. H. M. K., AND DUARTE, O. C. M. B. *Redes tolerantes a atrasos e desconexões*, 2007.
- [44] OLIVEIRA, E., AND ALBUQUERQUE, C. Análise do protocolo NECTAR em cenário com mobilidade e frequentes interrupções. In *Simpósio Brasileiro de Redes de Computadores - SBRC09* (2009).
- [45] OLIVEIRA, E., SILVA, E., AND DE ALBUQUERQUE, C. V. N. Promovendo adaptação a contextos em DTNs. In *SBRC 2011* (may 2011).
- [46] OLIVEIRA, E. C. R., AND DE ALBUQUERQUE, C. V. N. NECTAR: a DTN routing protocol based on neighborhood contact history. In *SAC* (2009), pp. 40–46.
- [47] PENTLAND, A., FLETCHER, R., AND HASSON, A. DakNet: rethinking connectivity in developing nations. *Computer* 37, 1 (2004), 78–83.
- [48] RUTISHAUSER, M., PETKOV, V., BOICE, J., OBRACZKA, K., MANTEY, P., WILLIAMS, T., AND WILMERS, C. Carnivore: A disruption-tolerant system for studying wildlife. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on* (2010), pp. 1–8.
- [49] SCOTT, K., AND BURLEIGH, S. RFC 5050, Bundle Protocol Specification, 2007.
- [50] SCPS. Space communication protocol standards, 2011.
- [51] SETH, A., KROEKER, D., ZAHARIA, M., GUO, S., AND KESHAV, S. Low-cost communication for rural internet kiosks using mechanical backhaul. In *Proceedings of the 12th annual international conference on Mobile computing and networking* (New York, NY, USA, 2006), MobiCom '06, ACM, pp. 334–345.
- [52] SONG, L., AND KOTZ, D. F. Evaluating opportunistic routing protocols with large realistic contact traces. In *In Proc. ACM 2nd Workshop on Challenged Networks (CHANTS 07)*, pp 35-42 (2007).
- [53] SPACE, R. F., DELIVERY, C. F., AND BOOK, B. Consultative committee for space data systems, 1992.
- [54] SPYROPOULOS, T., MEMBER, S., PSOUNIS, K., AND RAGHAVENDRA, C. Single-copy routing in intermittently connected mobile networks. In *In Proceedings of IEEE SECON* (2004).
- [55] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (New York, NY, USA, 2005), WDTN '05, ACM, pp. 252–259.
- [56] SPYROPOULOS, T., RAIS, R. N., TURLETTI, T., OBRACZKA, K., AND VASILAKOS, A. Routing for disruption tolerant networks: taxonomy and design. *Wirel. Netw.* 16 (November 2010), 2349–2370.

-
- [57] VAHDAT, A., AND BECKER, D. Epidemic routing for partially-connected ad hoc networks. Tech. rep., 2000.
- [58] WANG, Y., ZHANG, P., LIU, T., SADLER, C., AND MARTONOSI, M. CRAWDAD data set princeton/zebranet (v. 2007-02-14). Downloaded from <http://crawdad.cs.dartmouth.edu/princeton/zebranet>, Feb. 2007.
- [59] WANG, Y., ZHANG, P., LIU, T., SADLER, C., AND MARTONOSI, M. CRAWDAD trace set princeton/zebranet/movement (v. 2007-02-14). Downloaded from <http://crawdad.cs.dartmouth.edu/princeton/zebranet/movement>, Feb. 2007.
- [60] YOON, J., LIU, M., AND NOBLE, B. Random waypoint considered harmful, 2003.
- [61] ZARAFSHAN-ARAKI, M., AND CHIN, K.-W. TrainNet: a transport system for delivering non real-time data. *Computer Communications* (June 2010).

Anexos

ANEXO A – Contribuições no código-fonte do RAPID

```

/*
 * Copyright 2010 Institute of Telematics, Karlsruhe Institute of Technology (KIT)
 * Released under GPLv3. See LICENSE.txt for details.
 *
 * Christoph P. Mayer – mayer[at]kit[dot]edu
 * Wolfgang Heetfeld
 *
 * Modified by: Edelberto Franco Silva – esilva[at]ic[dot]uff[dot]br
 * Version 0.1.1 – released 11. Jan 2011
 */

// Include this line to future use
/** Router's setting namespace ({@value})*
public static final String RAPID_NS = "RapidRouter";

/* No using fixed value of metric in code. Actually this is sent of
 * settings file
 */
//private final UtilityAlgorithm ALGORITHM = UtilityAlgorithm.AVERAGE_DELAY;
public UtilityAlgorithm ALGORITHM;
public String ALGORITHM_COMPARE = new String();
public static final String ALGORITHM_S = "algorithm";
public static final String DEFAULT_ALGORITHM = "AVERAGE_DELAY";

/* In the construction is necessary compare the ALGORITHM
 * with to sent of settings file
 */
public RapidRouter(Settings settings) {
    super(settings);

    Settings rapidSettings = new Settings(RAPID_NS);
    if (rapidSettings.contains(ALGORITHM_S)) {
        //System.out.println("The algorithm parameter exists!");
        ALGORITHM_COMPARE = rapidSettings.getSetting(ALGORITHM_S);

        if (ALGORITHM_COMPARE.equals("MAXIMUM_DELAY")){
            ALGORITHM = UtilityAlgorithm.MAXIMUM_DELAY;
        } else {
            if (ALGORITHM_COMPARE.equals("MISSED_DEADLINES")) {
                ALGORITHM = UtilityAlgorithm.MISSED_DEADLINES;
            }
        }
    }
}

```

```
        } else {  
            ALGORITHM = UtilityAlgorithm.AVERAGE_DELAY;  
        }  
    }  
  
} else {  
    // No more using fixed metric  
    //ALGORITHM = DEFAULT_ALGORITHM;  
    ALGORITHM = UtilityAlgorithm.AVERAGE_DELAY;  
}
```

ANEXO B – Código-fonte do NECTAR

```

/*
 * Copyright 2008 TKK/ComNet
 * Released under GPLv3. See LICENSE.txt for details.
 */
package routing;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Map;
import java.util.Set;

import core.*;

/**
 * Implementation of NECTAR router as described in
 * <I>NECTAR: A DTN Routing Protocol Based on Neighborhood Contact History</I> by
 * Etienne C. R. de Oliveira , C  lio V. N. de Albuquerque.
 *
 * @author Edelberto Franco Silva
 * esilva [ at ] ic [ dot ] uff [ dot ] br
 * @version 1.0
 */
public class NectarRouter extends ActiveRouter {

    // PASSIVE_ACK
    /** IDs of the messages that are known to have reached the final destination */
    private Set<String> ackedMessageIds;
    /** Destination node. Here is possible to know */
    private Map<DTNHost, Integer> hopsToDest;
    /** List of nodes what the message have pass */
    private List <DTNHost> pathContains;
    private Map<DTNHost, List<DTNHost>> path;

    /** Neighborhood Indexes */
    private Map<DTNHost, Double> nidx;

```

```

private double NR_UPDATE = 0.5;
/* weight to transitive action */
private double weight = 5.0;
private double sigma;
/**
*
*/
public static final double MAX_SIGMA = 0.05;
/**
*
*/
public static final double MIN_SIGMA = 0.01;
/** last contact time */
private Map<DTNHost, Double> lastContactTime;

/** Percentage of buffer occupation is maximum of 90. If true, EP_DYN = EP_MAX */
public static final double EP_MAX_BUFFER = 90;
/** Buffer with more than 90% occupation */
public static final int EP_DYN_MIN = 1;
/** Buffer with less than 90% of occupation */
public static final int EP_DYN_MAX = 4;
/** to default */
public int EP_DYN = EP_DYN_MAX;

/**
* Constructor. Creates a new message router based on the settings in
* the given Settings object.
* @param s The settings object
*/
public NectarRouter(Settings s) {
    super(s);
    initNidx();
}

/**
* Copyconstructor.
* @param r The router prototype where setting values are copied from
*/
protected NectarRouter(NectarRouter r) {
    super(r);

    this.ackedMessageIds = new HashSet<String>();
    this.hopsToDest = new HashMap<DTNHost, Integer>();
    this.path = new HashMap<DTNHost, List<DTNHost>>();
    this.pathContains = new ArrayList<DTNHost>();
    this.lastContactTime = new HashMap<DTNHost, Double>();
    initNidx();
}

/**
* Initializes nidx hash
*/

```

```

private void initNidx () {
    this.nidx = new HashMap<DTNHost, Double>();
}

@Override
public void changedConnection(Connection con) {
    if (con.isUp()) {

        DTNHost otherHost = con.getOtherNode(getHost());
        DTNHost host = getHost();

        boolean openOrClose = verifyLastContact(otherHost);
        if (openOrClose) {

            addNodeOnPath(host, otherHost);

            if (con.isInitiator(getHost())) {
                /* initiator performs all the actions on behalf of the
                 * other node too (so that the meeting probs are updated
                 * for both before exchanging them) */
                MessageRouter mRouter = otherHost.getRouter();

                assert mRouter instanceof NectarRouter : "NectarRouter only works "+
                    "_with other routers of same type";
                NectarRouter otherRouter = (NectarRouter)mRouter;

                /* exchange ACKed message data */
                this.ackedMessageIds.addAll(otherRouter.ackedMessageIds);
                otherRouter.ackedMessageIds.addAll(this.ackedMessageIds);
                deleteAkedMessages();
                otherRouter.deleteAkedMessages();

                /* exchange and update Nidx */
                updateDeliveryNidxFor(otherHost);
                updateTransitiveNidx(otherHost);
            }
        }

        this.lastContactTime.put(otherHost, SimClock.getTime());
    }
}

/* calculate NR_Contact */
/**
 * 
 * @param otherHost
 * @return
 */
public boolean verifyLastContact(DTNHost otherHost){
    if (this.lastContactTime.get(otherHost) == null){
        return true;
    } else {
        Double now = SimClock.getTime();
        Double pass = this.lastContactTime.get(otherHost);
    }
}

```

```

        if ((now - pass) <= NR_UPDATE) {
            return false;
        } else {
            return true;
        }
    }
}

/**
 * Updates delivery nidx for a host.
 */
private void updateDeliveryNidxFor(DTNHost host) {
    Double now = SimClock.getTime();
    Double pass = this.lastContactTime.get(host);

    double diff=0;

    if (this.lastContactTime.get(host) == null){
        diff = now;
    } else {
        diff = now - pass;
    }

    if ((this.lastContactTime.get(host) == null) || (diff >= NR_UPDATE)) {
        double oldValue = getNidxFor(host);
        double hops = this.hopsToDest.get(host) + 1;
        double newValue = oldValue / hops;
        nidx.put(host, newValue);

    } else {

        double oldValue = getNidxFor(host);
        double hops = this.hopsToDest.get(host) + 1;
        double newValue = oldValue / Math.pow(hops * (diff / NR_UPDATE), sigma);
        nidx.put(host, newValue);
    }
}

/**
 * Updates transitive (A->B->C) delivery nidx.
 */
private void updateTransitiveNidx(DTNHost host) {
    MessageRouter otherRouter = host.getRouter();
    assert otherRouter instanceof NectarRouter : "NectarRouter only works " +
        " with other routers of same type";

    double pForHost = getNidxFor(host); // Nidx(a,b)
    Map<DTNHost, Double> othersNidx =
        ((NectarRouter)otherRouter).getDeliveryNidx();

    for (Map.Entry<DTNHost, Double> e : othersNidx.entrySet()) {
        if (e.getKey() == getHost()) {
            continue; // don't add yourself

```

```

    }

    double pOld = getNidxFor(e.getKey()); // Nidx(a,c)
    double pNew = ((pOld * weight) + pForHost) / (weight + 1);
    nidx.put(e.getKey(), pNew);
}

}

/**
 *
 * @param host
 * @return
 */
public double getNidxFor(DTNHost host) {

    if (nidx.containsKey(host)) {
        return nidx.get(host);
    }
    else {
        return 0;
    }
}

/**
 * Deletes the messages from the message buffer that are known to be ACKed
 */
private void deleteAckedMessages() {
    for (String id : this.ackedMessageIds) {
        if (this.hasMessage(id) && !isSending(id)) {
            super.deleteMessage(id, false);
        }
    }
}

@Override
public Message messageTransferred(String id, DTNHost from) {
    Message m = super.messageTransferred(id, from);
    /* was this node the final recipient of the message? */
    if (isDeliveredMessage(m)) {
        this.ackedMessageIds.add(id);
    }

    return m;
}

@Override
public void deleteMessage(String id, boolean drop) {

    double maiorTX = 0;

    DTNHost host = this.getHost();
    Collection<Message> msgCollection = host.getMessageCollection();

```

```

        for (Message m : msgCollection) {
            double msgTX = m.getHopCount() + m.getReceiveTime();
            if (msgTX > maiorTX) {
                maiorTX = msgTX;
                id = m.getId();
            }
        }

        Message removed = removeFromMessages(id);

        if (removed == null) throw new SimError("no_message_for_id_" +
            id + "_to_remove_at_" + host);

        for (MessageListener m : host.getListeners()) {
            m.messageDeleted(removed, host, true);
        }
    }

    /**
     * Method is called just before a transfer is finalized
     * at {@link ActiveRouter#update()}. Nectar makes book keeping of the
     * delivered messages so their IDs are stored.
     * @param con The connection whose transfer was finalized
     */
    @Override
    protected void transferDone(Connection con) {
        Message m = con.getMessage();
        /* was the message delivered to the final recipient? */
        DTNHost otherHost = con.getOtherNode(getHost());
        if (m.getTo() == otherHost) {
            this.ackedMessageIds.add(m.getId()); // yes, add to ACKed messages
            super.deleteMessage(m.getId(), false); // delete from buffer
        }
    }

    /**
     *
     * @param host
     * @param node
     */
    public void addNodeOnPath(DTNHost host, DTNHost node) {
        this.pathContains.add(node);
        this.path.put(host, pathContains);

        if (!this.hopsToDest.containsKey(node)) {
            this.hopsToDest.put(node, 1);
            this.sigma = MAX_SIGMA;
            this.nidx.put(node, 1.0);
        } else {
            double nidxAux = this.nidx.get(node) + 1;
            this.nidx.put(node, nidxAux);
        }
    }

```

```

}

/**
 *
 * @param node
 * @return
 */
public DTNHost nodeOnPath(DTNHost node){
    if (this.pathContains.isEmpty()){
        return null;
    } else {
        if (this.pathContains.contains(node)){
            return node;
        } else {
            return null;
        }
    }
}

/* To knows if the contact node is the first hop to the destination */
/**
 *
 * @param dest
 * @param other
 * @return
 */
public DTNHost nodeFirstOnPath(DTNHost dest , DTNHost other){
    if (this.path.containsKey(dest)){
        List<DTNHost> hop = this.path.get(dest);
        if (hop.isEmpty()){
            return null;
        } else {
            return hop.get(0);
        }
    } else {
        return null;
    }
}

/**
 * Returns a map of this router's delivery nidx
 * @return a map of this router's delivery nidx
 */
private Map<DTNHost, Double> getDeliveryNidx() {
    return this.nidx;
}

@Override
public void update() {
    super.update();
    if (isTransferring() || !canStartTransfer()) {
        return; // transferring, don't try other connections yet
    }
}

```

```

// Try first the messages that can be delivered to final recipient
if (exchangeDeliverableMessages() != null) {
    return; // started a transfer, don't try others (yet)
}

tryOtherMessages();

}

/**
 * Tries to send all other messages to all connected hosts ordered by
 * their delivery probability
 * @return The return value of {@link #tryMessagesForConnected(List)}
 */
private Tuple<Message, Connection> tryOtherMessages() {
    List<Tuple<Message, Connection>> messages =
        new ArrayList<Tuple<Message, Connection>>();

    Collection<Message> msgCollection = getMessageCollection();

    /* for all connected hosts collect all messages that have a higher
       probability of delivery by the other host */
    for (Connection con : getConnections()) {
        /* the node isn't the final recipient.
         * Then, have the hops calculated to the destination?
         * No? exit
         * Yes? calculate the hops to destination (1 + hops_in_nodecontact_to_destination)
         */
        /* instance the other node */
        DTNHost otherHost = con.getOtherNode(getHost());
        DTNHost host = getHost();

        NectarRouter mRouter = (NectarRouter)otherHost.getRouter();

        if (mRouter.isTransferring()) {
            continue; // skip hosts that are transferring
        }

        for (Message m : msgCollection) {
            /* The end node (contacted) knows who is the destination = TRUE */
            if (mRouter.hopsToDest.containsKey(m.getTo()) &&
                this.hopsToDest.containsKey(m.getTo())){

                int hostHops = this.hopsToDest.get(m.getTo());
                int otherHops = mRouter.hopsToDest.get(m.getTo());

                if (hostHops > otherHops){
                    this.hopsToDest.put(m.getTo(), otherHops+1);
                    addNodeOnPath(m.getTo(), host);
                } else {
                    mRouter.hopsToDest.put(m.getTo(), hostHops+1);
                    addNodeOnPath(m.getTo(), mRouter.getHost());
                }
            }
        }
    }
}

```

```

    }
}
//epidemicActive(con);

Collection<Message> msgColl = host.getMessageCollection();

//for (Message msg : msgOther) {
for (Message msg : msgColl) {
    if (msg.getNrofHops() > 1) {
        // EP_DYN is MAX
        if (otherHost.getBufferOccupancy() < EP_MAX_BUFFER) {
            EP_DYN = EP_DYN_MAX;
        } else {
            EP_DYN = EP_DYN_MIN;
        }
        DTNHost whois = nodeFirstOnPath(msg.getTo(), otherHost);

        int hopsDest;
        if (this.hopsToDest.get(msg.getTo()) == null){
            hopsDest = 99999;
        } else {
            hopsDest = this.hopsToDest.get(msg.getTo());
        }

        if ((whois == otherHost) || (msg.getNrofHops() > (hopsDest - EP_DYN)))
    {

        msg.maxHops = msg.getNrofHops()-1;

        messages.add(new Tuple<Message, Connection>(msg,con));

    }

    if (mRouter.getNidxFor(msg.getTo()) > getNidxFor(msg.getTo())) {
        // the other node has higher probability of delivery
        messages.add(new Tuple<Message, Connection>(msg,con));
    }

}

}

if (messages.isEmpty()) {
    return null;
}

// sort the message-connection tuples
Collections.sort(messages, new TupleComparator());

return tryMessagesForConnected(messages);
}

/**
 * Comparator for Message-Connection-Tuples that orders the tuples by

```

```

    * their delivery probability by the host on the other side of the
    * connection (GRTRMax)
    */
private class TupleComparator implements Comparator
    <Tuple<Message, Connection>> {

    public int compare(Tuple<Message, Connection> tuple1,
        Tuple<Message, Connection> tuple2) {
        // delivery probability of tuple1's message with tuple1's connection
        double p1 = ((NectarRouter)tuple1.getValue().
            getOtherNode(getHost()).getRouter()).getNidxFor(
            tuple1.getKey().getTo());
        // -- tuple2...
        double p2 = ((NectarRouter)tuple2.getValue().
            getOtherNode(getHost()).getRouter()).getNidxFor(
            tuple2.getKey().getTo());

        // bigger probability should come first
        if (p2-p1 == 0) {
            /* equal probabilities -> let queue mode decide */
            return compareByQueueMode(tuple1.getKey(), tuple2.getKey());
        }
        else if (p2-p1 < 0) {
            return -1;
        }
        else {
            return 1;
        }
    }
}

@Override
public NectarRouter replicate() {
    return new NectarRouter(this);
}
}

```