

JOSÉ AÉLIO DE OLIVEIRA JÚNIOR

**ANYCASTING COM BALANCEAMENTO DE
CARGA EM DTNs USANDO CAMPOS
MAGNÉTICOS VIRTUAIS**

Tese apresentada ao Programa de Pós-Graduação em
Informática da Pontifícia Universidade Católica do
Paraná como requisito parcial para obtenção do título
de Doutor em Informática.

CURITIBA

2015

JOSÉ AÉLIO DE OLIVEIRA JÚNIOR

**ANYCASTING COM BALANCEAMENTO DE
CARGA EM DTNs USANDO CAMPOS
MAGNÉTICOS VIRTUAIS**

Tese apresentada ao Programa de Pós-Graduação em
Informática da Pontifícia Universidade Católica do
Paraná como requisito parcial para obtenção do título
de Doutor em Informática.

Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Luiz Augusto de Paula Lima Jr.

CURITIBA

2015

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

O48a
2015
Oliveira Júnior, José Aélío de
Anycasting com balanceamento de carga em DTNs usando campos magnéticos virtuais / José Aélío de Oliveira Júnior ; orientador, Luiz Augusto de Paula Lima Jr. – 2015.
ix, 107 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná,
Curitiba, 2015
Bibliografia: f. 101-107

1. Roteadores (Redes de computação). 2. Campos magnéticos. 3. Informática. I. Lima Júnior, Luiz Augusto de Paula. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título.



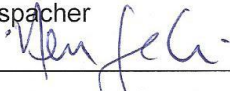
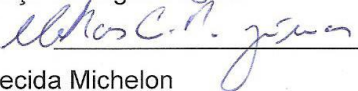
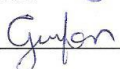
CDD 20. ed. – 004



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE DEFESA DE TESE DE DOUTORADO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO
DEFESA DE TESE DE DOUTORADO Nº 034/2015

Ao 1º dia de Outubro de 2015 realizou-se a sessão pública de Defesa da Tese de Doutorado intitulada **“Anycasting com Balanceamento de Carga em DTNs Usando Campos Magnéticos Virtuais”** apresentada pelo aluno **José Aélío de Oliveira Junior** como requisito parcial para a obtenção do título de Doutor em Informática, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Augusto de Paula Lima Junior PUCPR (Orientador)	 _____ (assinatura)	<u>aprovado</u> (aprov/reprov.)
Prof. Dr. Alcides Calsavara PUCPR	 _____ APROVADO	
Prof. Dr. Henri Frederico Eberspacher PUCPR	 _____ aprovação	
Prof. Dr. Methanias Colaço Rodrigues Junior UFS	 _____ APROVADO	
Prof.ª Dr.ª Gisane Aparecida Michelson UNICENTRO	 _____ Aprovado	

Conforme as normas regimentais do PPGIa e da PUCPR, o trabalho apresentado foi considerado aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.


Prof.ª Dr.ª Andreia Malucelli
Coordenadora do PPGIa
Coordenadora do Programa de Pós-Graduação em Informática.



Agradecimentos

Dedico este trabalho à minha família que sempre me apoiou e me incentivou durante todos os momentos difíceis. Em especial a minha mãe Crizélia, a minha irmã Cristiane, a minha noiva Rafaela e ao meu padrasto José Augusto (*in memoriam*).

Ao meu orientador Dr. Luiz Augusto de Paula Lima Junior, pela motivação, pelos ensinamentos, pela paciência e confiança depositada em mim durante todo este período.

Aos membros da Banca pelas contribuições.

A todos que fazem parte do PPGIA, especialmente, aos professores do grupo de sistemas distribuídos.

Finalmente, agradeço a Deus por ter me dado força para a conclusão de mais esta etapa.

Sumário

Agradecimentos	i
Sumário	ii
Lista de Figuras	iv
Lista de Tabelas	vii
Resumo	viii
Abstract	ix
Introdução	1
1.1. Motivação	3
1.2. Objetivos	4
1.3. Organização	4
Capítulo 2 - Fundamentação Teórica	6
2.1. ANYCASTING	6
2.2. DTN	9
2.2.1. Arquitetura	10
2.2.2. Tipos de Contatos	12
2.2.3. Aplicações de DTN	12
2.2.4. Roteamento em DTN	16
2.3. Campos Magnéticos Virtuais	19
2.3.1. <i>QuickPath</i>	22
2.4. Resumo e Conclusão	23
Capítulo 3 - Trabalhos Relacionados	25
3.1. <i>Spray and Wait</i>	25
3.2. <i>PRoPHET</i>	27
3.3. <i>PRoPHETv2</i>	28
3.4. <i>NECTAR - Neighborhood Contact History</i>	30
3.5. <i>MaxProp</i>	30
3.6. Meeting Visit	31
3.7. <i>Core-aided</i>	31
3.8. Algoritmos Genéticos	32
3.9. Resumo e Conclusão	33
Capítulo 4 – Anycasting em DTNs usando Campos Magnéticos Virtuais	37
4.1. Adaptação de Campos Magnéticos para DTN – Visão Geral	37
4.2. Cálculo da Força de Atração	39
4.3. <i>Heartbeat</i>	44
4.4. Roteamento da Mensagem	54
4.5. Balanceamento de Carga	62
4.6. Controle de Réplicas	67
4.7. Conclusão	68
Capítulo 5 – Ambiente de Simulação e Resultados Obtidos	70
5.1. Ambiente de Simulação	70
5.1.1. Cenário da Simulação	72

5.1.2. Parametrização	73
5.2. Análise dos resultados com DTN <i>anycast</i>	74
5.3. Comparativo com outros algoritmos da literatura.....	80
5.4. Análise e comparativo dos resultados com a utilização de <i>throwboxes</i>	85
5.5. Conclusão	95
Capítulo 6 - Conclusão	96
6.1. Relevância do Estudo	96
6.2. Contribuições da Pesquisa.....	98
6.3. Trabalhos Futuros.....	99
Referências Bibliográficas	101

Lista de Figuras

Figura 1 - <i>Unicast</i> [42]	7
Figura 2 - <i>Multicast</i> [42].....	7
Figura 3 - <i>Anycast</i> [42]	8
Figura 4 – DNS com uso de <i>Anycast</i>	9
Figura 5- Camada de agregação [7].....	11
Figura 6 - Técnica <i>store-and-forward</i> [8].....	11
Figura 7 - Condor [10].....	13
Figura 8 - DieselNet [12].....	14
Figura 9 - Internet Interplanetária [13]	14
Figura 10 - Seaweb [32]	16
Figura 11 - Roteamento epidêmico [7].....	17
Figura 12 - Roteamento baseado no controle do movimento do nodo [7]	19
Figura 13 - <i>Campos Magnéticos Virtuais</i> [20]	20
Figura 14 - Planos Magnéticos em Redes <i>Overlay</i> [21].....	21
Figura 15 - Grafo de Vizinhança (NG) e Tabela de Campos Magnéticos (MFT) [21].....	23
Figura 16 - Atraso de entrega em função do número de nodos M e raio de alcance K. [23] ...	26
Figura 17 - Percentual de mensagens entregues. [24]	28
Figura 18 – Quantidade de mensagens repassadas. [24]	28
Figura 19 – Percentual de entrega e <i>overhead</i> de mensagens [30].....	29
Figura 20 – Calculando o menor custo com <i>Delivery likelihood</i>	31
Figura 21 - Probabilidade de entrega com os algoritmos baseado em GA[31].....	32
Figura 22 - Um grafo NG é composto de k ligado sub-redes Vi	39
Figura 23 – Sub-rede composta pelos os nodos 1 e 2.....	41
Figura 24 - Nodo 3 conecta-se a sub-rede	42
Figura 25 - A mensagem mc é atraída para o nodo 2	42
Figura 26 - Nodo 1 recebendo a mensagem mc	43
Figura 27 - Nodo 1 desconecta-se da sub-rede.....	43
Figura 28 - Diminuição da força dos nodos 2 e 3.....	44

Figura 29 - Linha do tempo do alarme no estado NPC	47
Figura 30 - Linha do tempo do alarme no estado PC	49
Figura 31 – Sub-rede conexa contendo pelo menos um nodo em PC	51
Figura 32 - Mensagem de dados é atraída para o nodo em PC mais forte	51
Figura 33 - Aumento da força dos nodos em NPC sem ultrapassarem a $Máx\{\sigma p, \forall p \in Vi \cap PC\}$	52
Figura 34 – Ajuste proporcional da força dos nodos em NPC devido ao novo limite.....	53
Figura 35 – Diminuição da força dos nodos em NPC	53
Figura 36 – A NGt e a Tabela de Campos Magnéticos (MFT).....	57
Figura 37 – Mensagem de dados é atraída para o nodo em PC mais forte	58
Figura 38 – Sub-rede com dois nodos em PC	59
Figura 39 – Mensagem entregue a p' em virtude da atenuação da força de p	60
Figura 40 – Sub-rede conexa sem nenhum nodo em PC	61
Figura 41 – Roteamento de mensagens em sub-rede sem nenhum nodo em PC	62
Figura 42 – Todos os nodos da sub-rede conexa.....	64
Figura 43 – Mensagem atraída e sendo processada para o nodo em PC mais forte.....	65
Figura 44 – Nodo em PC mais forte reduz a força atração	66
Figura 45 – Balanceamento de carga.....	67
Figura 46 - Simulador executando com 100 nodos, utilizando o algoritmo <i>Spray and Wait</i> e modelo <i>map-based movement</i> [34].....	71
Figura 47 - Modelo <i>Random waypoint</i> com 200 nodos com raio de alcance de 250m [34]	71
Figura 48 - Comparação com 30 nodos em PC com apenas uma cópia da mensagem	75
Figura 49 - Comparação com 6 nodos em PC com apenas uma cópia da mensagem	76
Figura 50 - Comparação com 1 nodo em PC com apenas uma cópia da mensagem.....	77
Figura 51 - Comparando com diferentes quantidades de nodos em PC com alcance de 80 metros	77
Figura 52 - Comparando com diferentes quantidades de nodos em PC com alcance de 50 metros	78
Figura 53 - Comparando com diferentes quantidades de nodos em PC com alcance de 30 metros	78
Figura 54 - Comparação de taxas de entrega de mensagens para a faixa de 80 metros	81
Figura 55 - Comparação de taxas de entrega de mensagens para a faixa de 50 metros	82

Figura 56 - Comparação de taxas de entrega de mensagens para a faixa de 30 metros	82
Figura 57 - <i>Throwboxes</i> interligando a parte central da cidade de <i>Helsinki</i>	86
Figura 58 - Comparação de taxas de entrega de mensagens para a faixa de 80 metros	87
Figura 59 - Comparação de taxas de entrega de mensagens para a faixa de 50 metros	87
Figura 60 - Comparação de taxas de entrega de mensagens para a faixa de 30 metros	88

Lista de Tabelas

Tabela 1: Comparativo entre algoritmos de roteamento	35
Tabela 2: Mensagens abortadas e entregues em relação à HB	74
Tabela 3: Transmissões de mensagens e entregues em relação à força de atração	74
Tabela 4: Atraso médio na entrega das mensagens	79
Tabela 5: Mensagens descartadas devido ao <i>buffer overflow</i>	79
Tabela 6: Média de Saltos por Mensagem.....	79
Tabela 7: Tempo Médio de uma Mensagem nos <i>Buffers</i> (segundos).....	80
Tabela 8: Tráfego Médio em <i>Kilobyte</i> por Nodo com Transmissões <i>Heartbeat</i>	80
Tabela 9: Probabilidade de Entrega.....	83
Tabela 10: Tempo médio de uma mensagem nos <i>buffers</i>	83
Tabela 11: Mensagens descartadas devido ao <i>buffer overflow</i>	84
Tabela 12: Atraso médio na entrega das mensagens	84
Tabela 13: Tráfego gerado pelas transmissões de mensagens.....	85
Tabela 14: Percentual de mensagens entregues com o raio de 80 metros	89
Tabela 15: Percentual de mensagens entregues com o raio de 50 metros	90
Tabela 16: Percentual de mensagens entregues com o raio de 30 metros	91
Tabela 17: ANOVA.....	91
Tabela 18: Atraso médio na entrega das mensagens com o raio de 80 metros	92
Tabela 19: Atraso médio na entrega das mensagens com o raio de 50 metros	93
Tabela 20: Atraso médio na entrega das mensagens com o raio de 30 metros	94
Tabela 21: Mensagens descartadas devido ao <i>buffer overflow</i>	95

Resumo

O triunfo da Internet nas últimas décadas se deve ao conjunto de protocolos designado TCP/IP. No entanto, em cenários com longos atrasos e frequentes desconexões, os protocolos TCP/IP não funcionam bem e novos protocolos são necessários. As redes que consideram estes cenários foram denominadas Redes Tolerantes a Atrasos (DTN - *Delay Tolerant Networks*) e um dos seus principais desafios é o repasse das mensagens, pois é preciso determinar rotas sem o estabelecimento de um caminho fim-a-fim. Cada nodo da rede desempenha o papel de roteador, armazenando e encaminhando as mensagens nodo a nodo desde a origem até o destino. Vários protocolos para encaminhamento de mensagens têm sido propostos. Todavia, as propostas existentes na literatura são, em sua maior parte, elaboradas para atender ao problema de entrega de mensagens a um único destinatário específico na rede (*unicasting*). No entanto, em alguns cenários do mundo real, existe a necessidade de se entregar uma mensagem a pelo menos um nodo dentro de um grupo de destinatários possíveis, o que recebe o nome de *anycasting*. Contudo, os principais trabalhos encontrados na literatura investigam DTN *anycast* do tipo determinísticas. No entanto, o presente trabalho explora uma abordagem inédita no roteamento para DTNs *anycast* do tipo estocásticas com suporte a balanceamento de carga. Para tal, este trabalho apresenta uma adaptação do algoritmo de campos magnéticos virtuais para rotear mensagens *anycast* em DTNs. O algoritmo realiza roteamento por meio de mensagem de atração em vez da abordagem tradicional de encaminhamento de mensagens. Foi possível analisar o desempenho do algoritmo tanto no contexto de *anycast* e *unicast* (que é um caso particular de *anycast*) Neste último, foi possível compará-lo com algoritmos como *Epidemic*, *PRoPHET*, *PRoPHET V2* e *Spray and Wait*. O algoritmo de campos magnéticos virtuais definido apresentou um percentual de entrega de mensagens superior em todos os cenários analisados, quando comparado com *Epidemic*, *PRoPHET*, *PRoPHET V2* e superior na maior parte dos cenários analisados, quando comparado com o *Spray and Wait*. Quando aplicado no contexto *anycast*, o seu desempenho é ainda melhor, conforme esperado.

Palavras-Chave: DTN; *anycast*; campos magnéticos virtuais; roteamento.

Abstract

The triumph of the Internet in recent decades is due to the set of protocols called TCP/IP. However, in scenarios with long delays and frequent disconnections, the TCP/IP protocols do not work well and new protocols are needed. Networks that consider these scenarios were called DTN (Delay Tolerant Networks) and one of its main challenges is the transfer of messages, it is necessary to determine routes without establishing an end-to-end path. Each network node plays the role of router, storing and forwarding messages node to node from the origin to the destination. Several protocols for message routing have been proposed. However, existing proposals in the literature are, for the most part, designed to attend the delivery problem messages to a single specific recipient in the network (unicasting). However, in some real-world scenarios, there is a need to deliver a message to at least one node in a group of possible recipients, which is called anycasting. However, the main studies in the literature investigating DTN anycast the deterministic type. However, this paper explores a unique approach to routing anycast DTNs the stochastic type with support for load balancing. To this end, this paper presents an adaptation of the Virtual Magnetic Fields algorithm for routing anycast messages in DTNs. The algorithm performs routing through message attraction rather than the traditional approach of message forwarding. It was possible to analyze the performance of the algorithm both in the context of anycast and unicast (which is a particular case of anycast). In the latter, it was possible to compare it with algorithms such as Epidemic, Prophet, Prophet V2 and Spray and Wait. The virtual magnetic field algorithm presented a higher message delivery rate in all scenarios analyzed when compared to Epidemic, Prophet, Prophet V2 and superior in most scenarios analyzed, compared with the spray and Wait. When applied to the anycast context, its performance is even better, as expected.

Keywords: DTN; anycast; virtual magnetic fields; routing.

Capítulo 1

Introdução

O triunfo da Internet nas últimas décadas se deve ao conjunto de protocolos designado TCP/IP. O modelo TCP/IP foi planejado para operar de forma autônoma da tecnologia de sub-rede que existisse. Todos os dispositivos de sub-redes que compõem a Internet utilizam estes protocolos para roteamento e garantia de entrega de mensagens [1].

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. Exemplos de tais ambientes são: comunicações entre dispositivos móveis, comunicações sem fio, comunicações rurais, comunicações em campo de batalha, comunicações interplanetárias, comunicações submarinas, comunicações entre dispositivos com restrições de energia, dentre outros [2].

Estes ambientes, considerados “desafiadores”, possuem em comum a dificuldade de manter uma comunicação fim-a-fim, além da alta latência e altas taxas de perda de pacotes.

Devido a estas características, as redes que consideram estes aspectos foram denominadas Redes Tolerantes a Atrasos (DTN - *Delay Tolerant Networks*) [3].

Redes Tolerantes a Atrasos são redes sem fio onde desconexões são tão frequentes que, na maioria do tempo, não existe um caminho completo entre uma origem e um destino [4].

Em virtude das peculiaridades e das diversas mudanças nos ambientes DTNs, vários protocolos para encaminhamento de mensagens têm sido propostos. Todavia, as propostas existentes na literatura são, em sua maior parte, elaboradas para atender ao problema de entrega de mensagens a um único destinatário específico na rede.

Mas em alguns cenários, existem situações onde a possibilidade de vários nodos da rede estejam aptos a processar um determinado tipo mensagem são cada vez maiores como, por exemplo, em redes de transporte urbanas onde mensagens de emergências poderiam ser disparadas por um veículo (nodo) na rede com a finalidade de obter ajuda de equipes especializadas como, por exemplo, carro de bombeiros (nodo) ou uma ambulância (nodo) que estariam aptos a processá-la e não somente um veículo (nodo) específico na rede.

Vale ressaltar que a estratégia citada acima, difere do modelo conhecido como *multicast* [48] pelo fato de que somente um único destino receberá a mensagem ao contrário do conceito de *multicast* que um determinado grupo de nodos receberá uma mesma mensagem. Todavia, a estratégia supracitada se equivale ao conceito denominado de *anycast* [41], cujo princípio baseia-se na entrega de uma mensagem a um determinado membro pertencente a um determinado grupo. No exemplo citado, poderia ser considerado um destinatário qualquer carro de bombeiros (nodo) ou ambulância (nodo) disponível na rede.

Pensando nesse cenário, esta tese de doutorado realizou diversas adaptações no conceito de *Campos Magnéticos Virtuais*[20] com o objetivo de torná-lo aderente ao método de endereçamento *anycast* e por consequência ao método *unicast* [49] em DTNs estocásticas (*unicast* é um caso particular de *multicast*). Neste modelo, nodos da rede competem por mensagens “atraindo-as” para si com diferentes forças de atração.

Durante a fase de adaptação do modelo, além de novas definições sobre aumento e diminuição da força, foram definidas regras de roteamento e propagação da força. Foi criado um novo mecanismo de notificação denominado *heartbeat* com a finalidade de notificar os nodos de uma sub-rede a respeito da existência de nodos aptos a processar um determinado tipo de mensagem, fazendo com que a mensagem seja atraída para o mesmo pelo caminho mais rápido, permitindo que a rota de uma mensagem a um possível destino possa ser alterada dinamicamente em virtude da mobilidade dos nodos, sendo essa uma característica recorrente nesse tipo de rede.

O novo algoritmo de notificação viabilizou de forma implícita, um mecanismo de *balanceamento de carga* para DTNs *anycast* do tipo estocástica, permitindo que um nodo aumente ou reduza a sua probabilidade de receber mensagens baseado em limitações físicas e lógicas como, por exemplo, limitações de energia, armazenamento, processamento, dentre outros. Isto é feito pelo ajuste da sua própria força, fazendo com que as mensagens sejam atraídas ou não para outro nodo disponível na rede.

A ideia é, portanto, fazer uso da adaptação do conceito de campos magnéticos virtuais para encaminhar mensagens em DTNs *anycast* e *unicast* do tipo estocástica e avaliar o desempenho do novo método.

1.1. Motivação

A motivação desse trabalho se concentra em problemas do mundo real, onde em diversos tipos de cenários de comunicação o protocolo TCP e outros existentes possuem dificuldades em manter uma comunicação fim-a-fim – DTNs. A dificuldade se agrava mais quando existem diversos tipos de redes envolvidas, dificultando a interoperabilidade entre essas redes e conseqüentemente restringindo a utilização de serviços disponíveis nas mesmas.

As principais pesquisas na área de DTN se concentram principalmente nos algoritmos de roteamento das mensagens até um único destino específico na rede, denominado *unicast*. Mas existem cenários onde não teríamos um único destino e sim vários destinos potenciais capazes de processar um determinado tipo de mensagem na rede. Este conceito é denominado *anycast*.

O conceito *anycast* em DTNs pode ser aplicado em diversos cenários. Além do cenário de emergência citado anteriormente, o mesmo poderia ser utilizado, por exemplo, em aplicações de táxi, onde uma pessoa poderia enviar uma mensagem solicitando um táxi e a mensagem seria atraída para o táxi mais próximo dentre os táxis existentes naquela região.

Observou-se que, na área de roteamento em DTNs, existem lacunas de conhecimento, em particular, no encaminhamento *anycast* em DTNs do tipo estocásticas. Essa lacuna existe pelo fato dos principais trabalhos direcionarem seus esforços para o encaminhamento de mensagens de uma origem para um único destino. Por outro lado, o uso de *Campos Magnéticos Virtuais* ainda não foi explorado no contexto específico de DTNs.

Diante disso, esse trabalho realizou uma pesquisa das peculiaridades das DTNs aplicando o conceito de *Campos Magnéticos Virtuais* para o roteamento de mensagens no cenário *anycast*. Esse novo modelo de roteamento foi estudado em rede de transportes urbanos para fins de avaliação em um cenário realístico, porém sem perda de generalidade. O princípio de funcionamento do algoritmo poderá, em trabalhos futuros, ser evidentemente também aplicado a outros cenários como rede de transportes (rodoviária, ferroviária e aérea),

rede de sensores que monitoram a vida selvagem, redes em campo de batalha (redes militares), submarinas, redes interplanetárias, dentre outras.

1.2. Objetivos

O objetivo principal desta tese é o desenvolvimento e a avaliação de um novo método de roteamento de mensagens para DTNs *anycast* do tipo estocásticas com suporte a balanceamento de carga, através da adaptação do paradigma de *Campos Magnéticos Virtuais*. Este método não considera aspectos de segurança, portanto, fora do escopo deste trabalho.

Como objetivos específicos que o trabalho atingiu podemos mencionar:

- Definição de um método para roteamento de mensagens para DTNs *anycast* e *unicast* do tipo estocásticas;
- Definição de novas as regras para aumento e diminuição da força de atração do método proposto;
- Criação de um algoritmo de notificação e limitação da força denominado *Heartbeat*;
- Definição de novas regras para propagação da força de atração no método proposto;
- Definição de novas regras para o roteamento das mensagens no método proposto;
- Definição do mecanismo de balanceamento de carga para DTNs *anycast* no método proposto;
- Implementação do novo método no simulador *The ONE*;
- Simulação e análise dos dados obtidos no contexto de DTN *anycast* do tipo estocástica;
- Simulação e análise dos dados obtidos no contexto de DTN *unicast* do tipo estocástica comparando com os algoritmos *Epidemic*, *Prophet*, *Prophet v2* e *Spray and Wait*.

1.3. Organização

Esta tese está organizada em 5 capítulos. O Capítulo 2 descreve a fundamentação teórica, abordando *anycast*, conceito de DTNs e aplicações e por fim, campos magnéticos

virtuais. O Capítulo 3 apresenta os trabalhos que são correlatos a esta tese. O Capítulo 4 descreve o novo método de roteamento de mensagens para DTNs *anycast* do tipo estocásticas com balanceamento de carga. O Capítulo 5 apresenta e discute os principais resultados obtidos. E por fim, são apresentadas as conclusões e trabalhos futuros.

Capítulo 2 - Fundamentação Teórica

Neste capítulo são apresentados os principais conceitos que foram utilizados no desenvolvimento deste trabalho. Na seção 2.1 é abordado o método de encaminhamento de mensagens *anycast*. A seção 2.2 apresenta os fundamentos sobre as DTNs e suas aplicações. A seção 2.3 trata do conceito original de campos magnéticos virtuais que foi posteriormente adaptado para DTNs no capítulo 4. Finalmente, a seção 2.4 conclui este capítulo.

2.1. ANYCASTING

Nas comunicações de redes de computadores atuais, existem varias formas de encaminhamento de mensagens, onde as mais conhecidas são:

- *Unicast*: onde uma mensagem é encaminhada a um único nodo da rede;
- *Broadcast* [50]: onde uma mensagem é encaminhada para todos os nodos da rede;
- *Multicast*: onde uma mensagem é encaminhada a um determinado grupo de nodos na rede;
- *Anycast*: onde uma mensagem é encaminhada a um nodo pertencente a um determinado grupo de nodos na rede.

A forma de encaminhamento *anycast*, foi concebida inicialmente com o objetivo de ser utilizada na camada de rede através da RFC 1546 [41] em 1993. O paradigma de encaminhamento *anycast* compartilha características tanto com o *multicast* e o *unicast*. Sendo semelhante ao *multicast*, o paradigma *anycast* consiste em grupos de destinos, mas com a semântica que somente um destino em um determinado grupo *anycast* será escolhido para receber uma determinada mensagem. Após eleito o nodo de destino se torna semelhante ao

unicast, onde um remetente que se comunica com um grupo *anycast* normalmente interage com um destino escolhido a partir do grupo *anycast*.

Na definição de *anycast*, uma mensagem é considerada entregue, se a mesma for entregue a 1 elemento pertencente a um grupo.

Com pode ser observado nas figuras 1, 2 e 3 apresentam respectivamente as formas de encaminhamento *unicast*, *multicast* e *anycast*.

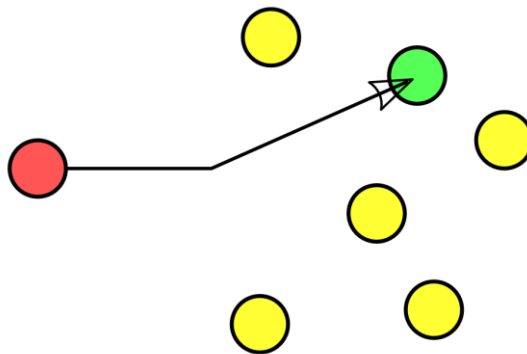


Figura 1 - Unicast [42]

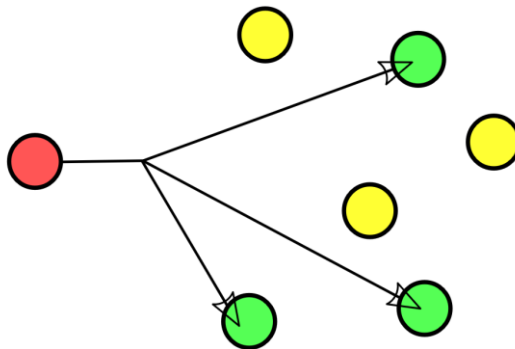


Figura 2 - Multicast [42]

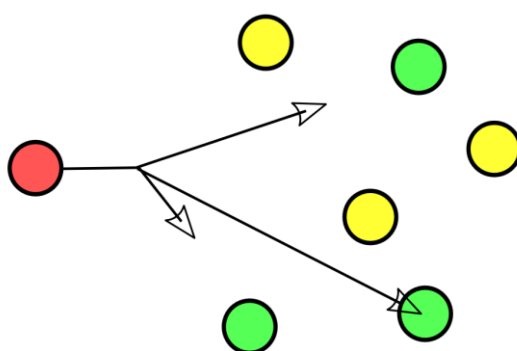


Figura 3 - Anycast [42]

No entanto, logo se percebeu que o conceito de *anycast* poderia ser utilizado na camada de aplicação com o intuito de aumentar a confiabilidade, disponibilidade e balanceamento de carga de serviços como, por exemplo, servidores DNS.

Na figura 4 pode ser observado um exemplo do uso de DNS com *anycast*. Um cliente DNS, configurado com o endereço *IP Anycast DNS* de 10.10.10.10, é mostrado realizando uma resolução de DNS que contém três servidores de nomes usando o mesmo endereço *IP anycast*.

O cliente poderia utilizar qualquer um dos três servidores DNS mostrados na figura citada. No entanto, se o algoritmo de roteamento seguir pelo caminho mais curto irá encaminhar os pacotes através de roteador R1 devido à topologia de roteamento. Caso o roteador R1 ou o servidor A falharem, os pacotes do cliente DNS serão automaticamente redirecionados para o servidor DNS mais próximo via os roteadores R2 e R3, e assim por diante. Além disso, o percurso para o servidor A, seria removido a partir das tabelas de encaminhamento, impedindo assim a utilização desse servidor de nome. Quando o roteador R1 e o servidor A restabelecerem seu funcionamento normalmente, o percurso será restabelecido novamente.

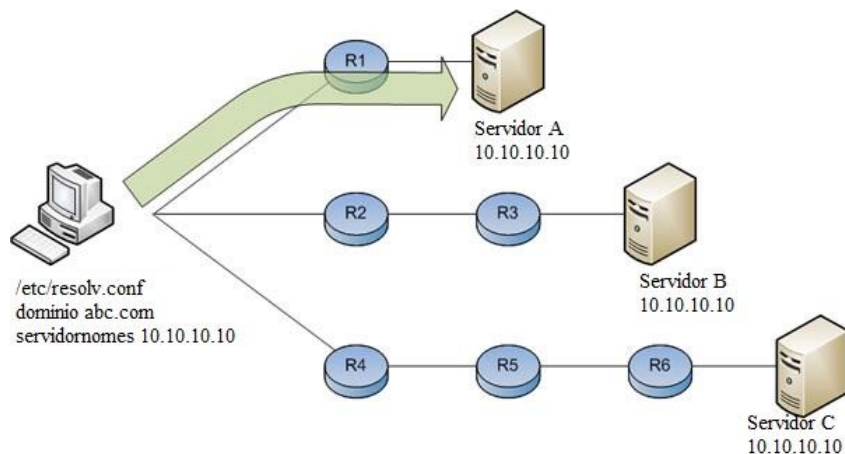


Figura 4 – DNS com uso de Anycast

A utilização de *anycast* não fica restrita somente a DNS, podendo o mesmo ser utilizado em servidores web e demais outros serviços que possam se enquadrar nessa forma de encaminhamento.

Contudo, tanto a definição da RFC 1546 como os exemplos citados acima se referem ao uso principalmente em redes conexas que possuem sua topologia previamente conhecida. Todavia, no contexto de redes móveis com frequentes desconexões conhecidas como DTNs, abordada na seção 2.2, onde a topologia da rede não é conhecida, se torna um grande desafio na implementação deste conceito.

A falta de informações dos nodos existentes na rede e se os mesmos são alcançáveis, além das dificuldades em relação a definições de grupos e criação de rotas em uma rede com frequentes mudanças de topologia dificultam a utilização de *anycast*.

2.2. DTN

O triunfo da Internet nas últimas décadas se deve ao conjunto de protocolos designado TCP/IP. O modelo TCP/IP foi planejado para operar de forma autônoma da tecnologia de sub-rede. Todos os dispositivos de sub-redes que compõem a Internet utilizam estes protocolos para roteamento e garantia de entrega de mensagens [1].

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. Exemplos de tais ambientes são: comunicações entre dispositivos móveis, comunicações sem fio, comunicações rurais, comunicações em campo de

batalha, comunicações interplanetárias, comunicações submarinas, comunicações entre dispositivos com restrições de energia, dentre outros [2].

Estes ambientes, considerados desafiadores, possuem em comum a dificuldade de manter uma comunicação fim-a-fim, além da baixa latência e altas taxas de perda de pacotes.

Devido a estas características, as redes que consideram estes aspectos foram denominadas Redes Tolerantes a Atrasos (DTN - *Delay Tolerant Networks*) [3].

Redes Tolerantes a Atrasos são redes sem fio onde desconexões são tão frequentes que, na maioria do tempo, não existe um caminho completo entre uma origem e um destino [4].

Durante a década de 90 a DARPA (*Defense Advanced Research Projects Agency*) investiu recursos para que a NASA e outros institutos de pesquisa desenvolvessem uma proposta para a IPN (*Interplanetary Networking*). Sendo que um dos principais desafios encontrados pelos pesquisadores era desenvolver um conjunto de tecnologias que fossem capazes de lidar com os significativos atrasos e perdas de pacotes comuns nas comunicações espaciais.

Logo se percebeu que algumas das soluções desenvolvidas no projeto IPN poderiam ser aplicadas em ambientes terrestres ou sendo mais específico em redes terrestres. Sendo que em 2002 o pesquisador Kevin Fall iniciou as primeiras adaptações para o ambiente terrestre onde o se originou pela primeira vez o termo DTN.

Nesse mesmo ano o IRTF (*Internet Research Task Force*) criou um grupo denominado DTNRG (*Delay Tolerant Network Research Group*) com a finalidade de aplicar os conceitos de DTN em redes terrestres, onde em 2007 o DTNRG publicou as RFC 5050 [5] e RFC 4838 [6], sendo que a primeira citada descreve a especificação do protocolo de agregação e a última define a arquitetura DTN.

2.2.1. Arquitetura

A dificuldade existente em diversos tipos de ambientes de comunicação em manter uma comunicação fim-a-fim e atenuada pela existência de diferentes tipos de redes envolvidas, dificultando a interoperabilidade entre essas redes motivou a criação de uma nova camada denominada *Bundle Layer* (Camada de Agregação) [7].

A camada de agregação é uma *overlay* (sobrecamada) que fica situada abaixo da camada de aplicação com a finalidade de garantir a interoperabilidade com os vários de tipos

de rede devido a mesma está acima da camada de transporte como pode ser observado na figura 5 [7].

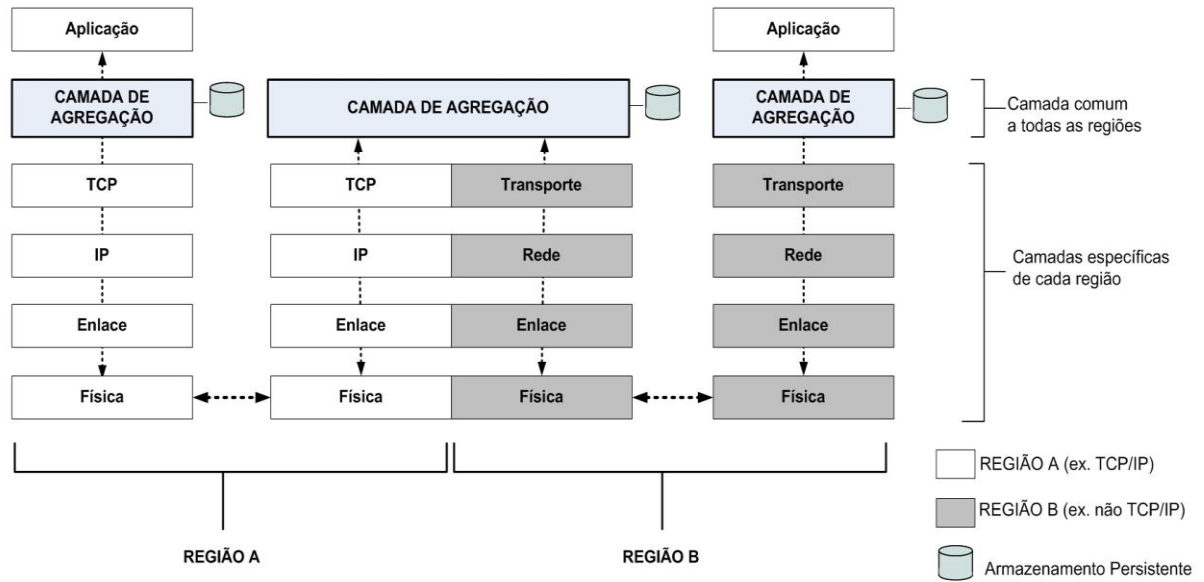


Figura 5- Camada de agregação [7]

Para que essa interoperabilidade fosse possível a arquitetura DTN utiliza a técnica conhecida como comutação de mensagens acompanhada do armazenamento persistente dos dados pelos nodos e envio ao próximo nodo (*store-and-forward*), ou seja, primeiro a mensagem é recebida por completa e armazenada para, em seguida, ser enviada ao próximo nodo, que pode ou não ser o destino [3]. Como pode ser visto na figura 6.



Figura 6 - Técnica *store-and-forward* [8]

Na literatura existem diversos trabalhos que esmiúçam a arquitetura DTN e suas técnicas sendo essa arquitetura já completamente definida nas RFC 5050 e RFC 4838 citadas

anteriormente, onde nesse trabalho não tem a intenção de propor nenhuma modificação em sua definição e seu funcionamento, devido ao simulador que se pretende utilizar o torna transparente para o propósito final dessa pesquisa.

2.2.2. Tipos de Contatos

Um contato corresponde a uma ocasião favorável para os nodos trocarem mensagens. Em DTNs devido a diversos fatores como, por exemplo, mobilidade não se assume que todos os nodos sejam alcançáveis e podem ser contatados a qualquer momento. Dessa forma os contatos em DTN foram classificados em cinco tipos que são: Contatos persistentes, sob demanda, programados, oportunistas e previsíveis [9].

- Contatos persistentes: São aqueles contatos que estão sempre disponíveis;
- Contatos sob demanda: São os tipos de contatos que precisam ser instanciados e após funcionam como contatos persistentes até serem encerrados;
- Contatos programados: São aqueles tipos de contatos que são programados previamente com o horário e duração do contato. Normalmente um pré-requisito para esse tipo de contato é que haja uma sincronização de tempo na rede para que a troca de informação seja possível;
- Contatos oportunistas: São os contatos que ocorrem através de encontros não programados entre nodos;
- Contatos previsíveis: São os contatos que fazem previsões sobre o horário e a duração dos contatos baseados em dados históricos.

2.2.3. Aplicações de DTN

Ao longo dos anos instituições de pesquisa, indústria e demais outros seguimentos começaram a utilizar os conceitos de DTN com a finalidade de desenvolverem projetos que antes não eram possíveis, seja por falta de uma infraestrutura mínima de comunicação ou pelo baixo poder executivo da região. Em seguida serão apresentados alguns projetos desenvolvidos que utilizam o conceito de DTN.

Condor (*Command and Control On-the-move Network Digital Over-the-horizon Relay*) é um projeto de cunho militar que tem como objetivo aumentar a confiabilidade na comunicação em cenários de campos de batalha onde na falha na comunicação por satélite as mensagens ficam armazenadas podendo ser encaminhadas em uma nova oportunidade de conexão como visto na figura 7 [10].

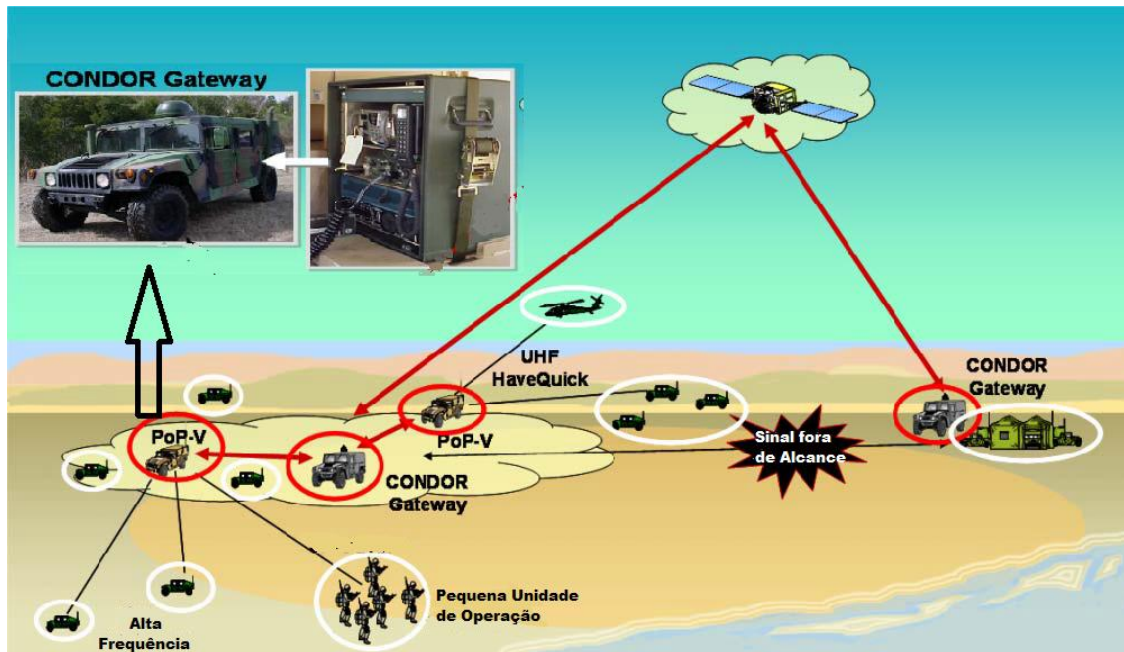


Figura 7 - Condor [10]

TierStore é um projeto desenvolvido pelo grupo TIER (*Technology and Infrastructure for Emerging Regions*) pertencente a Universidade da Califórnia em Berkeley que tem como objetivo levar tecnologia da informação a países subdesenvolvidos. O TierStore oferece recursos como web, email, dentre outros através da criação de centrais de dados com acesso à Internet para prover um armazenamento confiável e permanente deixando de forma a esconder a complexidade da conectividade intermitente existente naquela região [11].

DieselNet é um projeto criado pela Universidade de *Massachusetts Amherst*, nos Estados Unidos e tem como finalidade oferecer serviços de mobilidade aos passageiros do transporte urbano. Foram implantados em 40 ônibus computadores, antenas de radiofrequência e GPS (*Global Positioning System*) que de forma periódica registra a localização de cada veículo. Em alguns prédios foram colocados *throwboxes* que são nodos

fixos com uma maior capacidade de armazenamento e processamento que foram instalados em prédios ao longo dos itinerários dos ônibus como visto na figura 8[12].



Figura 8 - DieselNet [12]

NASA Tests First Deep-Space Internet – Em novembro de 2008 a NASA conseguiu enviar pela internet pela primeira vez imagens no espaço. Os dados transmitidos entre uma base na Terra e uma nave a 32 milhões de quilômetros de distância da Terra levaram 3 minutos e 20 segundos para percorrer todo o trajeto utilizando DTN como visto na figura 9 [13].

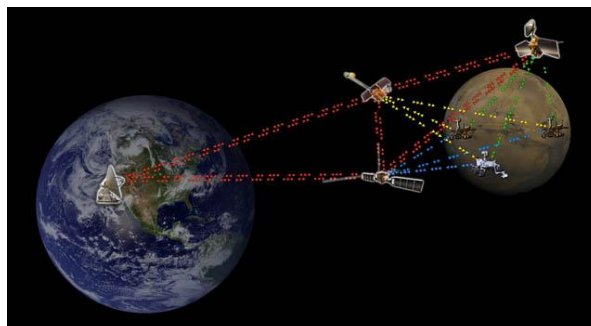


Figura 9 - Internet Interplanetária [13]

DakNet, cujo nome deriva da palavra Hindi para “post” ou “postal”, combina um meio físico de transportes com transferência de dados sem fio para estender a conectividade com a Internet. São utilizados veículos como motocicletas e ônibus como pontos de acesso, levando comunicação a áreas rurais. Nas vilas existem pontos locais denominados quiosques, quando um veículo entra no raio de alcance de um quiosque, automaticamente recebe e transmite os dados armazenados de locais. Ao se deslocar para uma região com acesso à Internet, os pontos de acesso transmitem as informações coletadas e armazenam outras informações recebidas da Internet para adiante repassá-las aos quiosques [14].

ZebraNet, projeto realizado pela Universidade de Princeton, dos Estados Unidos, se propõe a fazer o monitoramento da vida selvagem de zebras utilizando sensores colocados em colares. Esses colares são compostos por dispositivos de posicionamento global (GPS) que obtém periodicamente as coordenadas da posição da zebra e armazena no próprio colar. As informações são enviadas a estação-base que corresponde a um carro conduzido por pesquisadores, sendo capazes de conectar-se e coletar as informações armazenadas nos colares. Com o intuito de aumentar a probabilidade de entregas das mensagens coletas pelos colares, os mesmos podem trocar informações entre si, aumentando a chance de uma mensagem a ser entregue a uma estação-base [18].

Seaweb, desenvolvido por pesquisadores da NPS – Naval Postgraduate School em Monterey, Califórnia. Esta rede é constituída por vários nodos ancorados ao fundo da baía, e que se comunicam entre si através de modems de telesemar. Viabilizando a comunicação entre diversos nodos móveis como submarinos, dentre outros [16]. Como visto na figura 10.

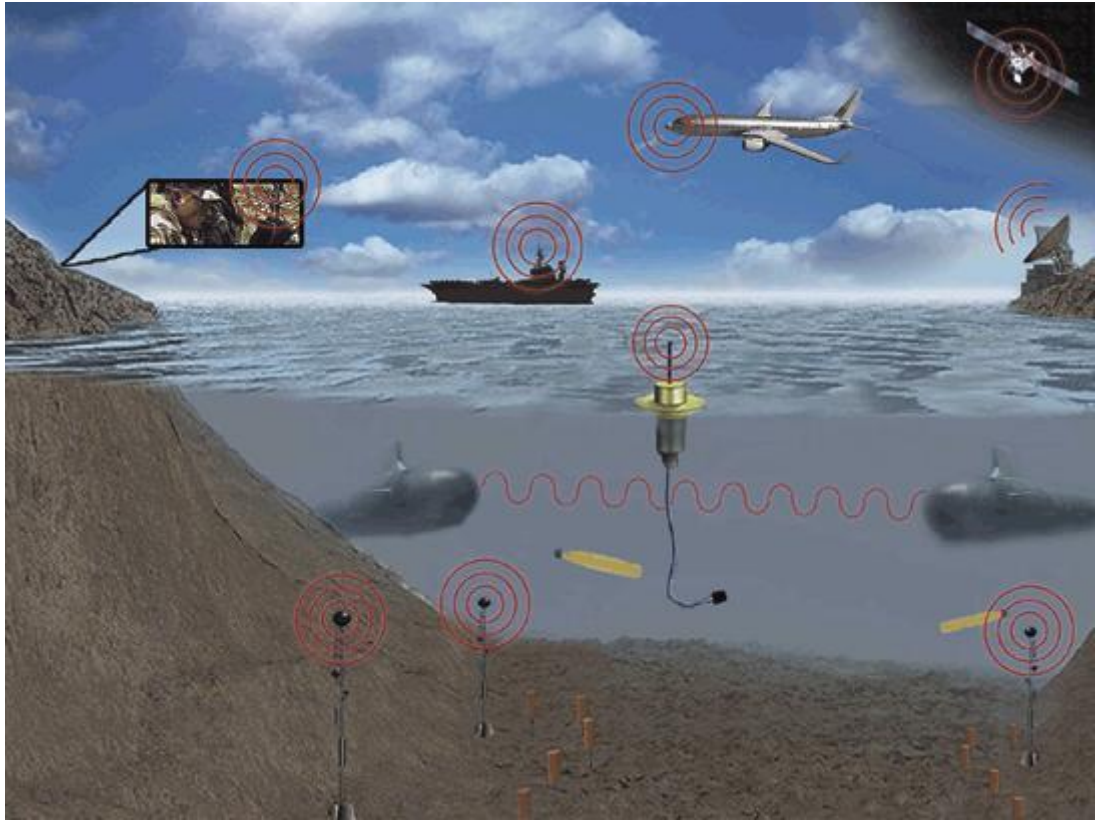


Figura 10 - Seaweb [32]

Como base na literatura pode-se afirmar que a aplicação do conceito DTN no mundo real é viável onde possuem outras aplicações práticas como por exemplos: Drive-thru Internet [15], e TurtleNet [17].

2.2.4. Roteamento em DTN

Um dos principais desafios no conceito de DTN é o roteamento, pois é necessário desenvolver protocolos que sejam capazes de lidar com os problemas dos atrasos e das frequentes desconexões causados pela mobilidade e a fatores ambientais, já que os algoritmos de roteamento para redes convencionais não estão aptos para tratar de forma eficiente à transmissão de dados em DTNs [7].

As DTNs são divididas em determinísticas e estocásticas [4].

Mesmo sendo dinâmicas as DTNs do tipo determinística possui previamente o conhecimento de quando ocorrem os contatos da rede ou, em alguns casos da topologia da rede em qualquer instante de tempo [4].

Nas redes do tipo estocásticas assume-se que o comportamento da rede não é conhecido. Em alguns casos o nodo não possui nenhuma informação histórica da rede o que dificulta a decisão do repasse da mensagem [4].

Na literatura foram propostos uma diversidade de algoritmos para redes estocásticas onde foram classificados em quatro grupos: roteamento epidêmico, roteamento baseado em estimativa, roteamento baseado em modelo e por último roteamento baseado em controle do movimento do nodo [7].

Roteamento epidêmico [19] foi o primeiro algoritmo de roteamento proposto para DTNs onde o funcionamento é semelhante a uma doença epidêmica: enquanto há espaço em buffer as mensagens são repassadas aos nodos encontrados como pode ser observado na figura 11.

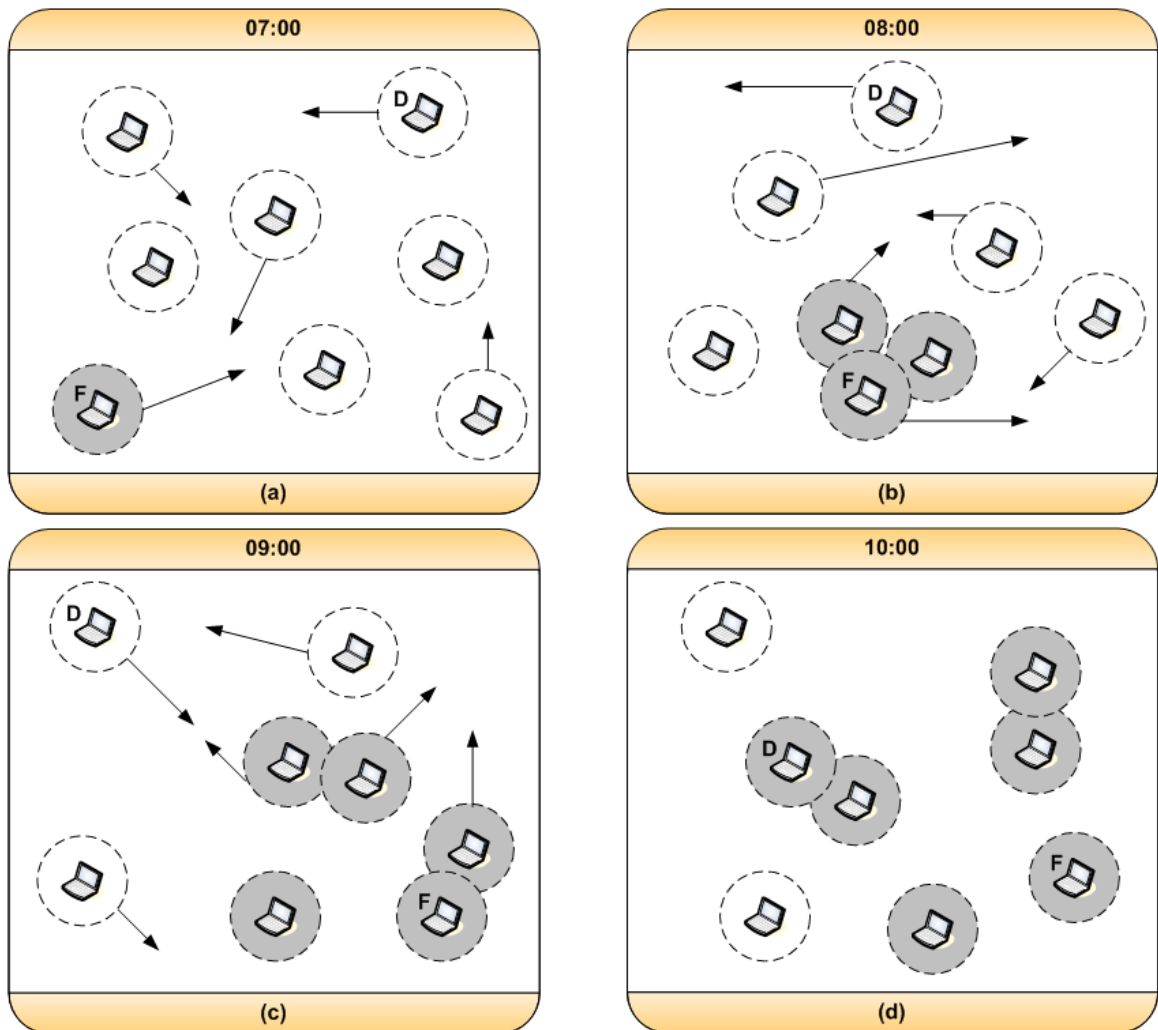


Figura 11 - Roteamento epidêmico [7]

Roteamento baseado em estimativa o nodo toma com base um dado histórico para decidir sobre o repasse ou não de uma mensagem nodo contatado naquele momento diferentemente do epidêmico que sempre replica a mensagem para o nodo vizinho no momento do contato [7].

Roteamento baseado em modelo ao contrário dos anteriores se baseia em modelos que podem ser utilizados como base para o repasse da mensagem. Como por exemplo, a movimentação de veículos (nodos) na trajetória de uma estrada onde um nodo em sentido contrário pode ter ou não mais chances da entrega de uma mensagem [7].

Roteamento baseado em controle do movimento do nodo controlam a mobilidade de alguns nodos da DTN para aumentar a capacidade de transmissão da rede. Como por exemplo, em uma rede composta por três camadas. Sendo a primeira camada composta por nodos DTN fixos sem limitações de capacidade de armazenamento, conectividade e energia. Na segunda camada teríamos os nodos móveis com grande capacidade de armazenamento e que são capazes de se comunicar com os nodos da primeira camada e também com os sensores (nodos) localizados na terceira camada. Mesmo não conhecendo a mobilidade dos nodos da segunda camada se utiliza dessa movimentação para entregar os dados coletados pelos sensores para a primeira camada [7]. Como pode ser observado na figura 12.

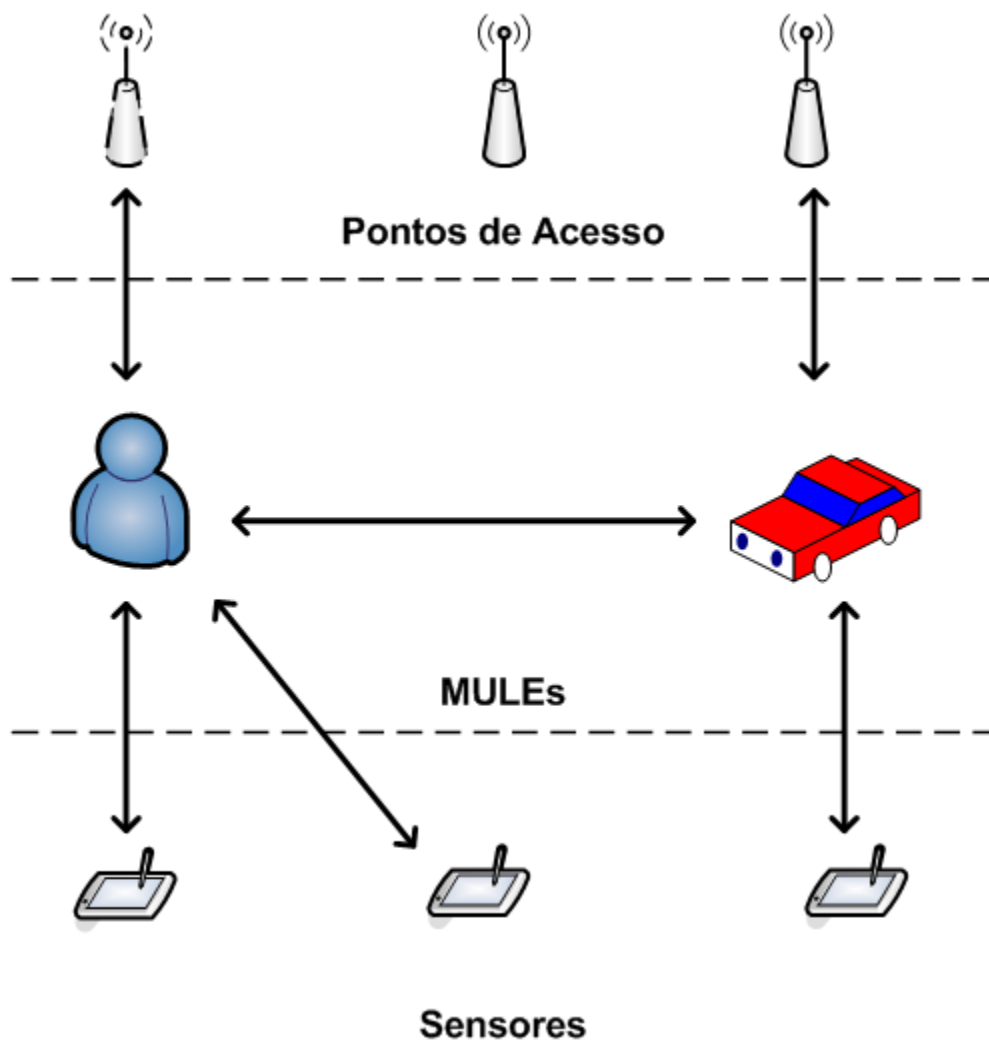


Figura 12 - Roteamento baseado no controle do movimento do nó [7]

Diversos trabalhos na literatura são derivados dos conceitos citados acima e são abordados na seção 3 como trabalhos relacionados.

2.3. Campos Magnéticos Virtuais

No contexto de DTN os principais algoritmos de roteamento da literatura focam no encaminhamento de uma mensagem partindo de um nó origem para um único nó destino. Em situações paliativas onde se faz uma adaptação do cenário para trabalhar com múltiplas entregas, mas em todos os casos encontrados continuam utilizando algum algoritmo que o foco é 1 para 1.

O algoritmo escolhido é denominado *Campos Magnéticos Virtuais* [20] que tem em sua essência o princípio de N destinos possíveis capazes de processar uma determinada mensagem inserida na rede, sendo que não existe nenhum estudo na literatura da aplicação do mesmo para o ambiente de DTN do tipo estocástica, foco dessa pesquisa. O algoritmo tem sido utilizado em diversas outras áreas de pesquisas, a exemplo [51] [52] [53].

O algoritmo de campos magnéticos se baseia em força de atração de uma mensagem chegar ao seu destino, cada nodo possui pesos que representam a força de atração, quanto mais forte essa atração mais chance que uma mensagem seja atraída para esse nodo, isso acontecendo de forma recursiva na rede até chegar ao seu destino [20]. Como demonstrado na figura 13.

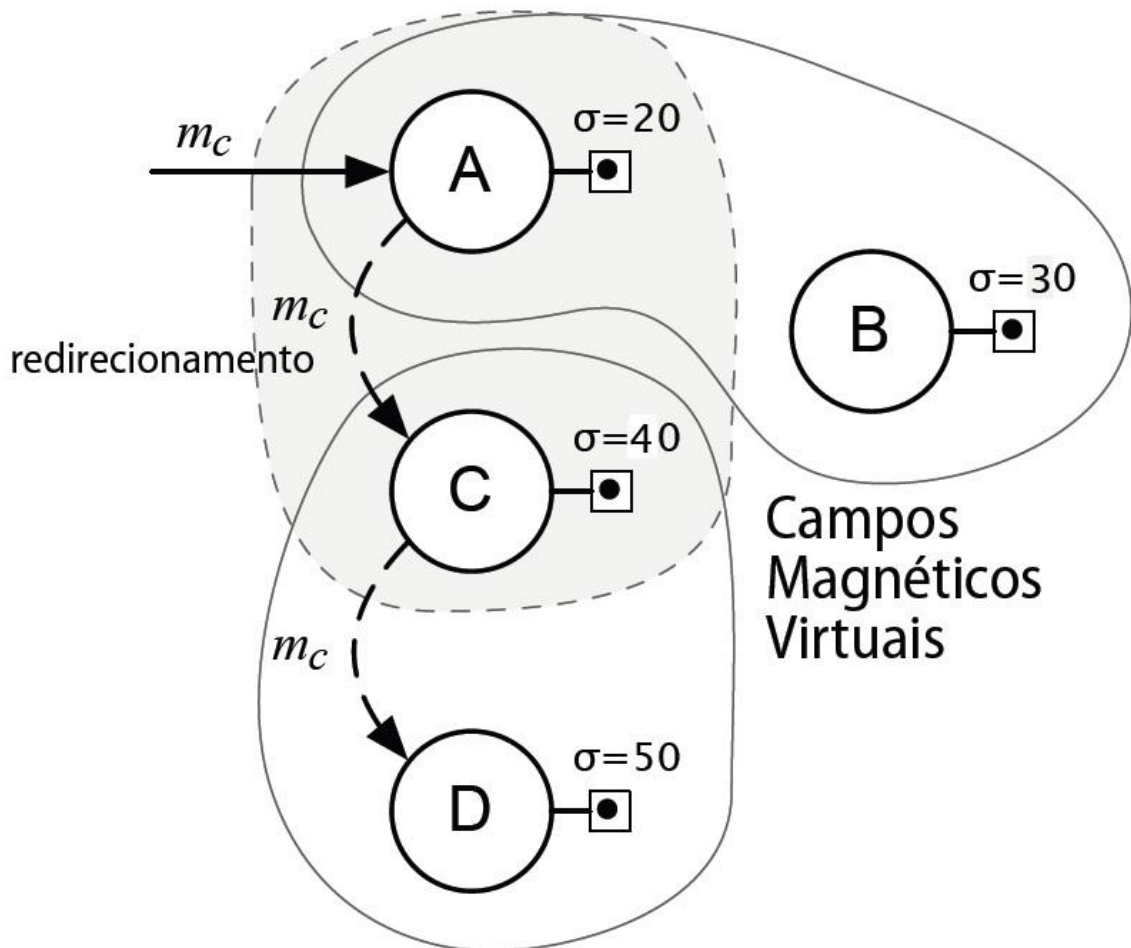


Figura 13 - *Campos Magnéticos Virtuais* [20]

Nesse cenário representado na figura 13, uma mensagem m_c de um tipo c é recebida pelo nodo A. Onde consideramos que todos os nodos são possíveis destinatários da mensagem m_c . O nodo A sofre atração magnética dos nodos B e C e o nodo C sofre atração magnética do nodo D. A força de atração em cada nodo é representada por σ . Nesse contexto, quando a m_c for recebida pelo nodo A será atraída para o nodo C, devido ao nodo que possui maior força de atração (indireta) sobre o A é o nodo D. Após a chegada ao nodo C a mensagem é atraída para o nodo D, que por não sofrer atração de nenhum outro nodo será responsável em processar a mensagem [20].

Da mesma forma como as DTNs o conceito de *Campos Magnéticos Virtuais* utiliza do conceito de rede *overlay* que representam melhor a visão da aplicação sobre os nodos com a finalidade de atender os requisitos de topologia orientados a fatores dinâmicos relacionados à semântica do sistema. Sendo que cada plano da figura 14 representa um tipo de classe magnética sobre a topologia física direcionando as conexões dos nodos de uma mesma classe, garantido atração sem mutualidade obrigatória [21].

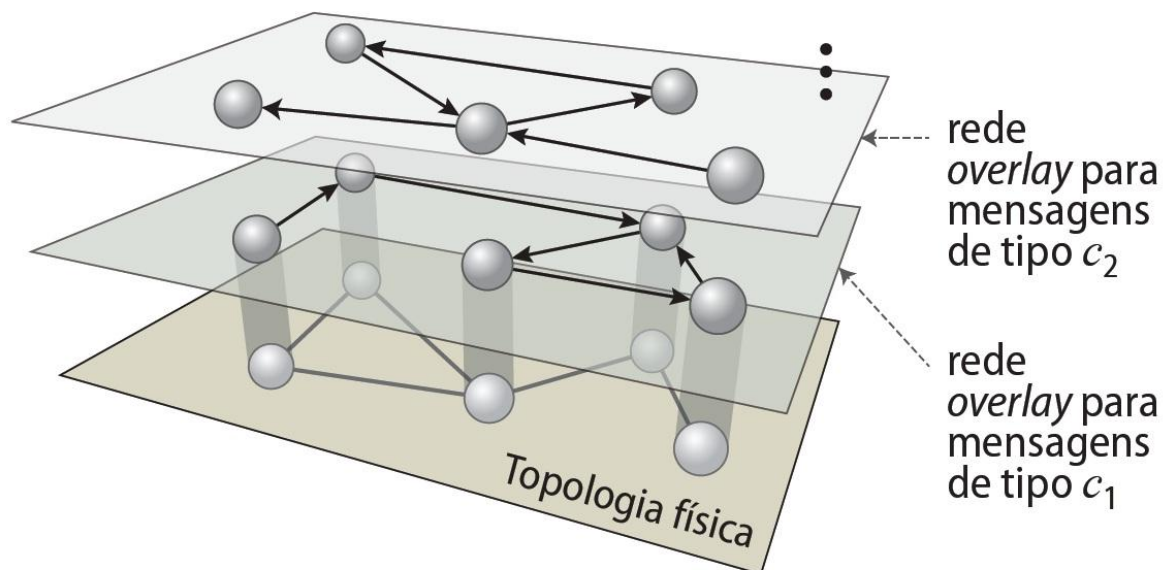


Figura 14 - Planos Magnéticos em Redes *Overlay* [21]

Para que os nodos se mantivessem atualizados sobre a força de atração dos outros nodos foram criados dois algoritmos com a finalidade de atualizar e propagar os valores de força de atração. São eles: *QuickPath* [20] e o *ShortPath* [22], sendo que o primeiro busca

caminhos com menor tempo para entrega das mensagens, já o segundo procura caminhos com o menor número de saltos para entrega de mensagens.

Em virtude da mobilidade dos nodos e das frequentes desconexões, essa pesquisa optou por utilizar o algoritmo *QuickPath* por supor que o menor tempo para entrega das mensagens de atualização possa aumentar a probabilidade do uso de *Campos Magnéticos Virtuais* para o contexto de DTN estocásticas.

2.3.1. *QuickPath*

No algoritmo *QuickPath* inicialmente alguns parâmetros básicos são definidos [21]:

- Cada nodo possui um identificador único i ;
- Cada nodo possui uma lista dos nodos vizinhos diretamente influenciados por ele e uma tabela de nodos cujos campos magnéticos de uma determinada classe c afetam i , chamada Tabela de Campos Magnéticos (MFT_c);
- Cada entrada de MFT_c é composta por origem, sendo o identificador do nodo que o influencia; pivô, sendo o identificador do nodo mais forte (direta ou indiretamente); força, sendo a força do pivô;
- O grafo representando as influências magnéticas para uma determinada classe c é uma rede overlay chamada de Grafo de Vizinhaça (NG_c).

Para exemplificar, pode ser visto na figura 15 um exemplo de um Grafo de Vizinhaça juntamente com suas Tabelas de Campos Magnéticos.

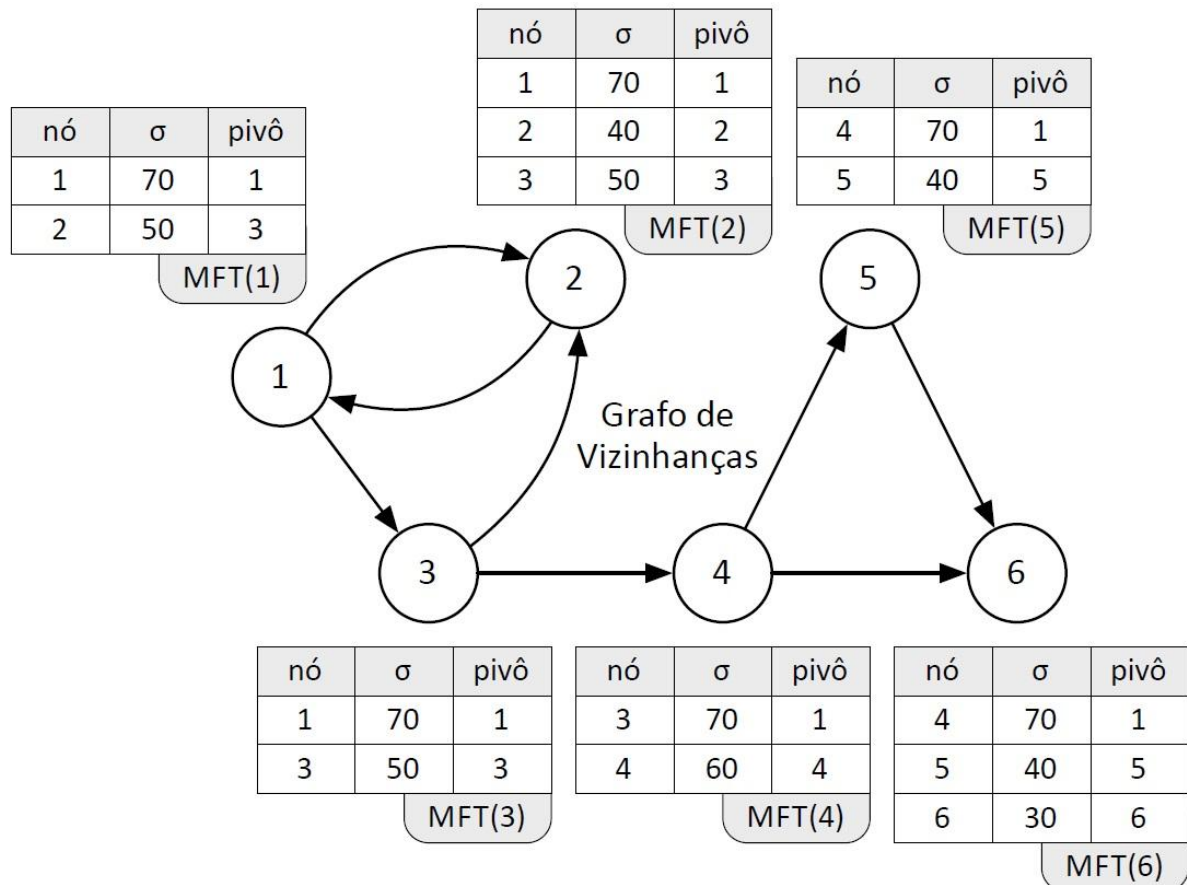


Figura 15 - Grafo de Vizinhança (NG) e Tabela de Campos Magnéticos (MFT) [21]

O princípio básico para o disparo de propagação de mensagens no algoritmo *QuickPath* ocorre quando ocorre eventos que modifiquem a força do nodo mais forte ou modifiquem o pivô percebido por um nodo [21].

Sendo verdadeira essa condição em um nodo, ele dispara uma mensagem de propagação notificando aos nodos vizinhos sobre a alteração.

No caso de um nodo receber uma mesma mensagem a mesma é descartada evitando mensagens repetidas na rede. Essa técnica permite que mensagens de aplicação utilizem o caminho com menor tempo como é feito para as mensagens de propagação [21].

2.4. Resumo e Conclusão

Neste capítulo foram apresentados os conceitos e as tecnologias básicas que foram utilizadas para o desenvolvimento deste trabalho.

O principal problema estudado neste trabalho é o roteamento de mensagens para DTNs *anycast* (e *unicast*) do tipo estocástica com suporte a balanceamento de carga.

Os principais estudos sobre DTNs *unicast* ou *anycast* [19] [23] [24] [30] [25] [31] partem do princípio que o destino de uma mensagem seja definido no momento de sua geração. No entanto, a proposta deste estudo não considera necessariamente um destinatário específico (a não ser no caso de *unicast*), mas sim grupos de destinatários potenciais definidos pelo tipo (ou “classe”) da mensagem. Desta forma, a mensagem será atraída até seu destino que corresponde a um dos nodos capazes de processá-la. Nós intermediários farão o roteamento até o destino mais adequado.

Assim sendo, a proposta tem como premissa a adaptação do conceito de campos magnéticos virtuais para DTNs *anycast* do tipo estocásticos com balanceamento de carga (estabelecendo distinções entre os destinatários potenciais), conceitos esses abordados nesse capítulo.

O próximo capítulo aborda os trabalhos existentes na literatura que mais se relacionam a esta proposta de tese, com suas vantagens e desvantagens.

Capítulo 3 - Trabalhos Relacionados

Neste capítulo são descritos os trabalhos relacionados a esta tese. São considerados os trabalhos de roteamento para DTNs, os quais serviram de base para o estudo e que foram utilizados para comparação com o novo método desenvolvido. As Seções 3.1 a 3.3 abordam, respectivamente, os algoritmos *Spray-and-Wait*, *PRoPHET* e *PRoPHET V2* utilizados na simulação para fins de comparação. As Seções 3.4 a 3.7 apresentam, respectivamente, os algoritmos *NECTAR*, *Meeting Visit* e *Core-aided* sendo a seção 3.8 aborda a utilização de Algoritmos Genéticos. Não foram utilizados na comparação devido a não serem os mais referenciados na literatura e em alguns casos por limitações nas descrições dos funcionamentos dos algoritmos que não permitiram uma comparação fidedigna. No entanto, são aqui apresentados para fins de conhecimento. A seção 3.9 conclui este capítulo.

3.1. *Spray and Wait*

Spyropoulos, Psounis e Raghavendra apresentam um algoritmo de roteamento denominado *Spray and Wait* [23] para DTN do tipo estocásticas onde as mudanças topológicas não são conhecidas.

O algoritmo *Spray and Wait* propõe uma abordagem para controlar o *flooding* limitando o número de cópias da mensagem na rede. Para cada mensagem, um nodo pode copiá-la na rede L vezes, onde L é configurado. Assim que todas as cópias forem distribuídas, cada cópia pode ser entregue somente de forma direta ao destino.

O roteamento é realizado em duas fases:

- *Spray*: Para cada mensagem gerada no nodo de origem são gerados L cópias desta mensagem nos quais são repassados individualmente para os próximos $L-1$ nodos vizinhos.
- *Wait*: Se o nodo de destino não for encontrado então cada nodo L que contem cópias das mensagens encaminhará somente para o nodo destino (*Direct Transmission*).

O algoritmo *Spray and Wait* adota um conceito no repasse da mensagem denominado de *spraying binary*. O nodo de origem possui uma variável C que controla o número de cópias de uma mensagem. Quando um nodo possui $C > 1$ e estabelece contato com um nodo, ele repassa $C/2$ das mensagens para o outro nodo ficando com a metade das cópias e assim sucessivamente até que o número de cópias seja igual a 1. Então ativa o roteamento *Direct Transmission*.

Na fase de *spray* as mensagens são encaminhadas de acordo com as seguintes regras [23]:

- cada nodo possui um *summary vector* com a identificação de todas as mensagens que ele armazena; quando dois nodos fazem contato, eles trocam seus vetores e verificam quais mensagens têm em comum;
- quando um nodo carrega uma cópia da mensagem e possui $C > 1$ réplicas da mensagem para encaminhar, então ele a repassa $C/2$ de suas réplicas para o outro nodo;
- quando um nodo tem apenas uma réplica da mensagem para encaminhar, então ele pode somente utilizar a regra *wait*.

Foram realizadas simulações com os seguintes algoritmos: *Epidemic*, *Randomized flooding*, *Utility-based flooding*, *Spray and Wait*. Os resultados demonstraram que o algoritmo *Spray and Wait* apresentou, uma probabilidade de entrega superior aos outros algoritmos como visto na figura 16 [23].

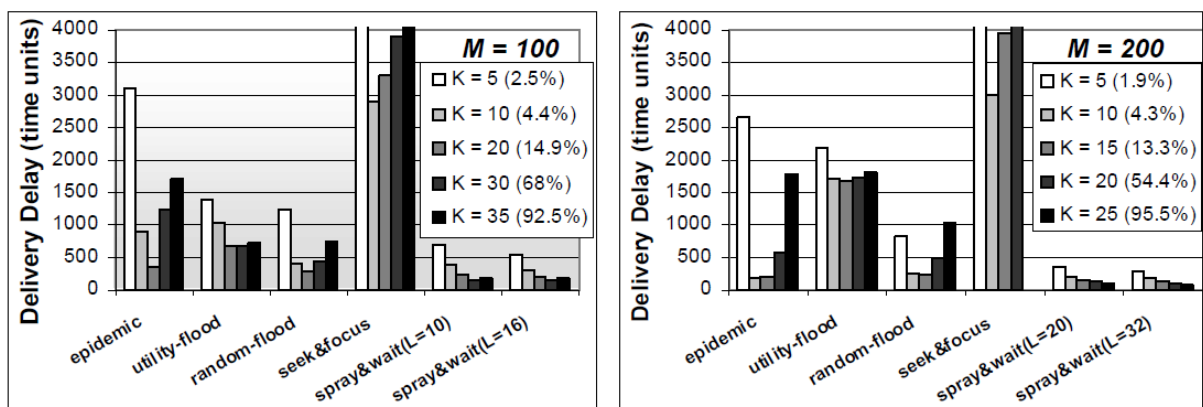


Figura 16 - Atraso de entrega em função do número de nodos M e raio de alcance K . [23]

3.2. *PRoPHET*

Lindgren, Doria e Schelén apresentam um algoritmo de roteamento denominado *PRoPHET* (*Probabilistic ROutingProtocol using History of Encounters and Transitivity*) [24] para DTN do tipo estocásticas onde as mudanças topológicas não são conhecidas.

O algoritmo *PRoPHET* utiliza a métrica probabilística para indicar a capacidade de um nodo entregar uma mensagem ao outro nodo com a finalidade de ficar mais perto do nodo de destino. Essa métrica era chamada previsibilidade de entrega.

Primeiramente calculava o histórico de contatos dos nodos ao longo do tempo. Quando dois nodos se encontravam eram trocadas listas com informações que identificam as mensagens armazenadas em cada nodo, sendo que o algoritmo *PRoPHET* possui um campo adicional referente a probabilidade do nodo A tem de entregar a um destino conhecido B.

Existem duas maneiras de alterar o valor da probabilidade: de forma direta, sempre que A e B se encontra os dois aumentam a probabilidade de entrega para ambos os lados. E a forma transitiva quando o nodo A tem contato frequente com o nodo B e o B tem contato frequente com o nodo C, mesmo que o nodo A não tenha contato direto com C sua probabilidade de entrega com relação ao nodo C é aumentada devido ao seu contato com B.

O cálculo da probabilidade de entrega do nodo a para o nodo b é representado por $P_{(a,b)}$, sendo realizado pela equação 1.1, onde $P_{(a,b)anterior}$ é o valor de $P_{(a,b)}$ anterior ao novo contato entre a e b , $P_{inicial}$ é uma constante de inicialização definida previamente, sendo realizado esse cálculo a cada contato entre dois nodos.

$$P_{(a,b)} = P_{(a,b)anterior} + (1 - P_{(a,b)anterior}) \times P_{inicial} \quad (1.1)$$

O valor da probabilidade de entrega cai quando dois nodos param de se encontrar armazenado em uma variável k denominada constante de envelhecimento.

Apesar do algoritmo se preocupar com restrições de armazenamentos nos nodos, ele não leva em consideração as limitações de largura de banda podendo em alguns casos aumentar o atraso na entrega da mensagem.

Durante as simulações realizadas em [24] foram obtidos a média de 5 rodadas de simulações, onde foram comparadas com o algoritmo epidêmico.

Os resultados apresentaram um melhor percentual de entrega com uma diminuição na quantidade de mensagens repassadas como pode ser visto nas figuras 17 e 18.

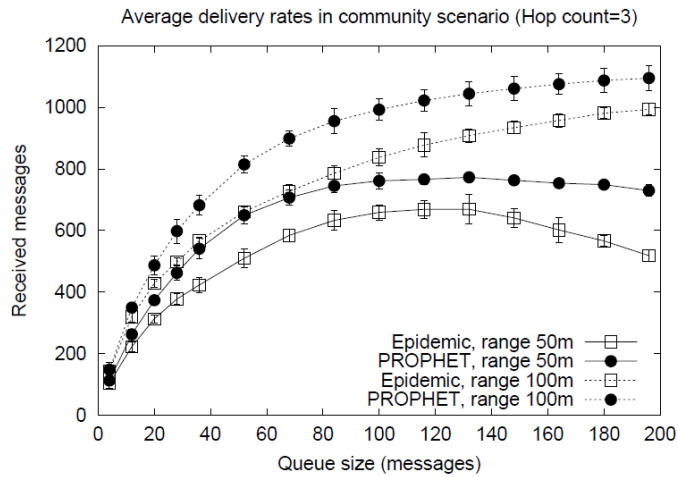


Figura 17 - Percentual de mensagens entregues. [24]

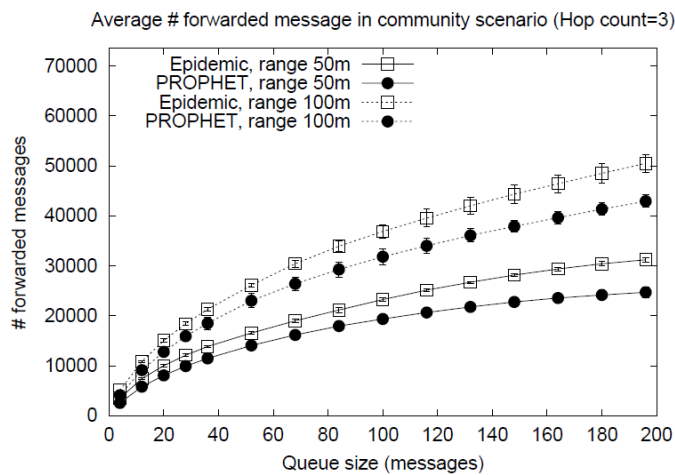


Figura 18 – Quantidade de mensagens repassadas. [24]

3.3. *PRoPHETv2*

Grasic, Davies, Lindgren e Doria apresentam uma variação no algoritmo de roteamento denominado *PRoPHET* onde eles denominaram de *PRoPHETv2* [30].

Os pesquisadores identificaram que em alguns cenários, o algoritmo aumentava de forma exponencial sua probabilidade de entrega devido ao problema que eles intitularam de problema no estacionamento lotado, onde eles fazem uma analogia de pessoas (nodos) em um estacionamento de um supermercado juntamente com a intermitência de conexão no local devido a uma má qualidade de conexão do wireless utilizado.

Durante a investigação percebeu-se que devido à má conexão os nodos se conectam várias vezes e devido a estarem concentrados em um mesmo local percebeu-se que o algoritmo *PRoPHET* considerava os restabelecimentos das conexões como novos encontros e consequentemente aumentando a probabilidade de entrega, gerando assim uma falsa probabilidade de entrega.

Como proposta de solução foi proposto uma modificação na fórmula que leva em consideração o curto espaço de tempo ocorrido entre dois nodos, fazendo que nesses casos a sua probabilidade de entregas aumente de forma mais lenta.

Durante os experimentos realizados em [30], o tempo de simulação foi definido para uma semana. O número de nodos foi de 500, e um total de 17000 mensagens foram enviadas. O espaço de buffer foi limitado a 100 MB por nodo e a taxa de transmissão foi de 100 kbit/s.

Como podemos observar na figura 19, o *PRoPHETv2* obteve desempenho melhor do que epidêmico e *PRoPHET* neste cenário. Da mesma forma, a taxa de entrega do *PRoPHETv2* era significativamente maior do que para *Spray and Wait*, mas também com uma maior sobrecarga. A conclusão que os autores chegaram é que com as mudanças se torna viável a utilização de um algoritmo de características epidêmicas em um cenário restrito [30].

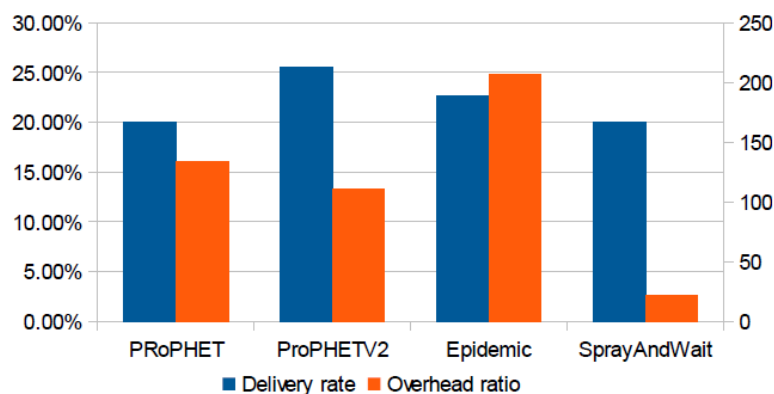


Figura 19 – Percentual de entrega e *overhead* de mensagens [30]

Os algoritmos supracitados *Spray and Wait*, *PRoPHET* e *PRoPHET V2*, foram escolhidos para fins de comparação devido as suas características na redução de inundação, ou sejam partilham do mesmo propósito deste trabalho, além de serem considerados por diversos trabalhos na literatura como objetos de comparação como novas propostas de roteamento. O epidêmico (citado na seção 2.2.4) foi utilizado com a finalidade de avaliar o

método proposto com um algoritmo sem restrições de *flood*. Sendo que os demais algoritmos clássicos encontrados não possuem em sua grande parte informações integras que permitam uma implementação fidedigna para fins de comparação.

Portanto, apesar das importantes contribuições para a ciência, as mesmas serão descritas a partir desse momento de forma resumida.

3.4. NECTAR - Neighborhood Contact History

O algoritmo *NECTAR* [25] utiliza o conceito de índice de vizinhança, considerando que os nodos movimentam-se de forma que existe certa probabilidade que vizinhos possam ser reencontrados. Esse é denominado *Neighborhood Index*.

O *Neighborhood Index* se baseia no fundamento da heurística entre os nodos que se movimentam em cenários estocásticos onde existe uma probabilidade de que dois nodos que já se encontraram possam se encontrar novamente.

Políticas de descarte de mensagens são desenvolvidas e um mecanismo de replicação de mensagens é desenvolvido.

Na política de descarte é priorizado o descarte de mensagens mais antigas e mensagens que tiveram altos índices de cópias repassadas.

Em [25] foram realizadas simulações comparando com os algoritmos epidêmicos e *PRoPHET*, onde os resultados demonstraram um melhor desempenho em relação às taxas de entrega nos cenários simulados e um desempenho inferior nas transmissões com o *PRoPHET* quando simulado em cenários com maior capacidade de armazenamento.

3.5. MaxProp

O *MaxProp* [26] é um algoritmo do tipo probabilístico que faz uso do histórico de encontros dos nodos para otimizar o roteamento das mensagens.

Possui um vetor denominado *delivery likelihood* que nada mais é que o custo para alcançar cada destinatário conhecido.

Quando dois nodos se encontram eles trocam os vetores possibilitando o cálculo da menor rota para cada destinatário como visto na figura 20.

O algoritmo tem o foco voltado para as DTNs determinísticas onde os contatos do tipo previsíveis são mais encontrados.

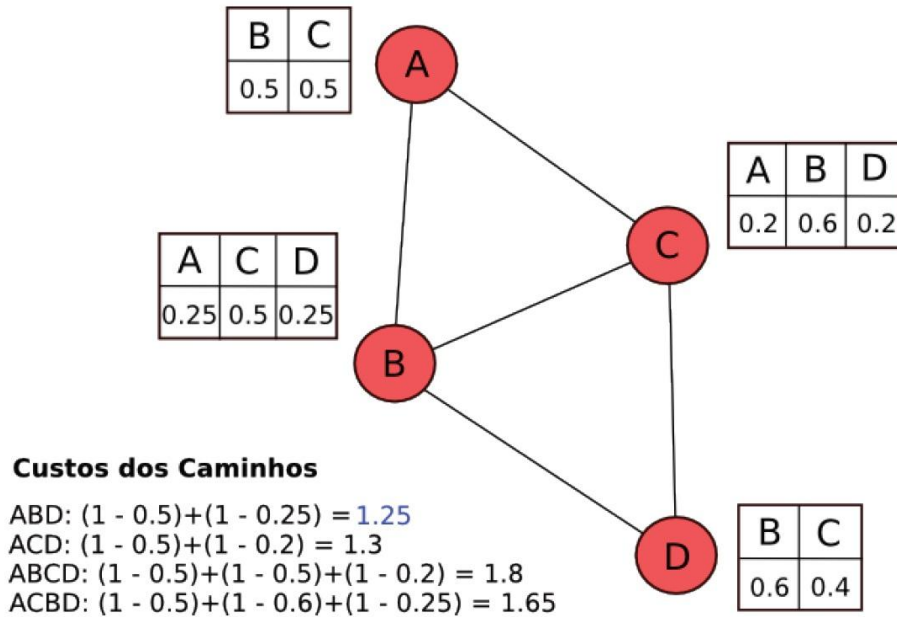


Figura 20 – Calculando o menor custo com *Delivery likelihood*

3.6. Meeting Visit

O *Meeting Visit* [27] é um protocolo que foi baseado no *PRoPHET*, o qual tem como objetivo aprender a frequência dos encontros entre nodos e seu histórico em visitas a certas regiões geográficas.

O algoritmo baseia-se do princípio que a probabilidade de um nodo passar por uma determinada região está fortemente ligada à quantidade de vezes que esse nodo tenha passado pela mesma.

Basicamente quando dois nodos se encontram são trocadas as listas e em seguida os nodos selecionam na lista as mensagens que eles não possuem e retiram aquelas cuja probabilidade de entrega seja mais alta no outro nodo.

3.7. Core-aided

O *Core-aided* [28] foi desenvolvido focando dois tipos de nodos em uma DTN: Os nodos com recursos ilimitados e os nodos com limitações.

Os nodos que não possui limitações são denominados de core nodos. Esses nodos quando em contato com os nodos que possuem limitações são capazes de receber parcialmente ou totalmente as mensagens dos nodos com limitações.

O algoritmo adota duas estratégias sendo que a primeira o core realiza cópias das mensagens dos nodos que possuem limitações.

A segunda estratégia está mais associada à limitação do tamanho do *buffer* dos nodos em que nos casos de algum nodo estiver com limitação do *buffer* o core ao invés de copiar a mensagem ele move a mesma desafogando o nodo.

3.8. Algoritmos Genéticos

Em [31] foi realizado um estudo da aplicação do uso de GA (algoritmos genéticos) em DTNs *Anycast* determinísticas.

Foram realizadas simulações combinando com o algoritmo de *Dijkstra* para encontrar o menor caminho e com o algoritmo de entrega mais rápida (*Earliest Delivery*) no qual calcula a rota na qual a mensagem será entregue o mais rápido possível.

O estudo em si não foi comparado com nenhum algoritmo clássico da literatura, ficando restritas as comparações entre o algoritmo de *Dijkstra (SP)*, algoritmo genético (GA1) e o algoritmo genético que possui o menor caminho com entrada (GA1 entrada SP). No entanto, a utilização do algoritmo genético foi construída com o foco em DTNs *anycast* determinísticas. Os resultados concluíram pela viabilidade do uso de algoritmo genético nesse tipo de rede como pode ser visto na figura 21.

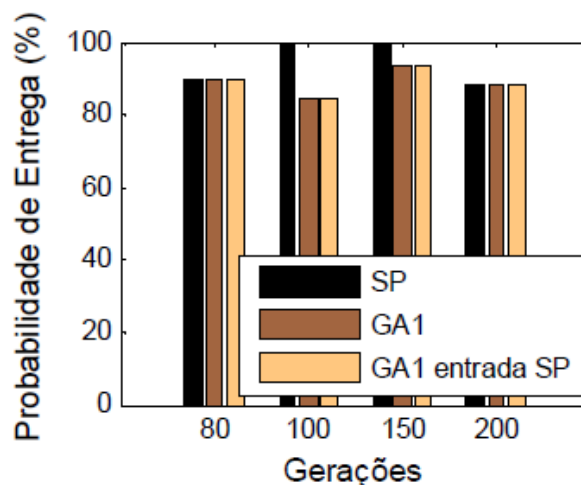


Figura 21 - Probabilidade de entrega com os algoritmos baseado em GA[31]

3.9. Resumo e Conclusão

Foram abordados neste capítulo os principais algoritmos citados na literatura, dentre eles o *PRoPHET*, *PRoPHET V2*, *Spray and Wait* e Epidêmico (citado na seção 2.2.4). Estes são os mais utilizados para fins de comparação com outras propostas na literatura [9] [25] [36] [45] [46] [47].

Para fins de conhecimento, foram abordados outros trabalhos da literatura como, por exemplo, *NECTAR*, *MaxProp*, *Meeting Visit*, *Core-aided e Algoritmos Genéticos*.

Na Tabela 1 é apresentado um comparativo entre os algoritmos descritos neste capítulo que teve como a base para a sua construção a tabela em [36]. As seguintes características foram comparadas:

- Gerenciamento do descarte por *overflow*: Se o algoritmo considera descarte e qual o procedimento adotado neste caso;
- Informações armazenadas: quais informações são armazenadas no nodo;
- Informações trocadas na rede: informações que são trocadas quando ocorre o contato;
- Controle de réplicas: se o algoritmo faz controle de réplicas;
- Redes estocásticas: se o algoritmo suporta redes estocásticas;
- Controle de repasse: se o algoritmo faz algum tipo de controle para evitar o *flood*;
- *Anycast*: se o algoritmo dá suporte ao método de encaminhamento *anycast*.

Em relação às DTNs *anycast*, os principais algoritmos encontrados na literatura direcionam seus estudos para redes determinísticas a exemplo de [31] [39]. Contudo, os raros trabalhos encontrados com DTN *anycast* do tipo estocástica a exemplo de [40] concentram os seus esforços em propostas de modelos para formação de grupos e definição de receptores,

sendo predominantes na literatura os algoritmos que se baseiam no método de endereçamento *unicast*.

Observou-se também que o conceito de campos magnéticos nunca foi utilizado para DTNs *anycast* e *unicast* do tipo estocásticas, foco desta pesquisa.

Baseado nesse levantamento, percebeu-se a oportunidade de contribuir com a área de DTN *anycast* e *unicast* do tipo estocásticas, por meio do desenvolvimento nesse estudo de um novo método de roteamento para DTNs *anycast* e *unicast* estocásticas com suporte a balanceamento de carga através da adaptação do conceito de campos magnéticos. Este método será apresentado no próximo capítulo.

Tabela 1: Comparativo entre algoritmos de roteamento

Algoritmo	Gerenciamento do descarte por overflow	Informações armazenadas	Informações trocadas na rede	Controle de réplicas	Redes estocásticas	Controle de repasse	<i>Anycast</i>
Epidêmico	Não considera	Lista dos nodos que se conectou	<i>Summary vector</i>	Não	Sim	Não	Não
<i>Spray-and-Wait</i>	Não considera	Nenhuma	<i>Summary vector</i>	Sim	Sim	Não	Não
<i>Prophet</i>	Usa FIFO para descarte	Tabela com a probabilidade de todos os nodos	<i>Summary vector</i>	Não	Sim	Sim	Não
<i>NECTAR</i>	Remove as mensagens mais antigas e as mais repassadas	Tabela <i>Neighborhood Index</i>	Tabela <i>Neighborhood Index</i>	Não	Sim	Sim	Não
<i>Maxprop</i>	<i>Buffer</i> limitado para mensagens recebidas	Tabela com a probabilidade de todos os nodos	Tabela com a probabilidade de contato	Não	SIM, mas possui foco em determinísticas	Sim	Não

Algoritmo	Gerenciamento do descarte por overflow	Informações armazenadas	Informações trocadas na rede	Controle de réplicas	Redes estocásticas	Controle de repasse	<i>Anycast</i>
<i>Meeting Visit</i>	<i>Buffer</i> limitado para mensagens recebidas e remove com menos probabilidade	Tabela com a probabilidade de todas as mensagens	<i>Summary vector</i> juntamente com a probabilidade entrega	Não	Sim	Sim	Não
<i>Core-aided</i>	Não considera	Não	Troca de informações sobre disponibilidade de <i>buffer</i>	Permite a cópia como também mover uma mensagem	Sim	O core se limita a fazer as cópias dos nodos com limitações	Não
Algoritmos Genéticos	Evita o repasse, mas não considera as mensagens criadas pelo próprio nodo	Armazena informações sobre rotas e <i>buffers</i> dos nodos na rede	Informações sobre rotas da rede, grupos e <i>buffer</i> do nodo	Não	Não	Sim	Sim
Campos Magnéticos Virtuais	Evita o repasse, mas mensagens criadas pelo próprio nodo são descartadas	Tabela de Campos Magnéticos	Mensagens <i>heartbeat</i>	Sim	Sim	Sim	Sim

Capítulo 4 – Anycasting em DTNs usando Campos Magnéticos Virtuais

Neste capítulo é apresentada a proposta da tese, a qual aborda basicamente a descrição e o funcionamento do método de encaminhamento de mensagens em DTNs.

As seções do Capítulo 4 encontram-se estruturadas na seguinte ordem: a seção 4.1 apresenta as características gerais do método proposto; a seção 4.2 descreve o processo de ganho e perda da força de atração; a seção 4.3 descreve, em detalhes, o funcionamento do algoritmo responsável em notificar a existência de um nodo apto a processar uma mensagem na sub-rede, limitar a força e realizar atualizações na MFT; a seção 4.4 descreve, em detalhes, o processo de roteamento de uma mensagem; a seção 4.5 descreve, em detalhes, o processo de balanceamento de carga; a seção 4.6 descreve o mecanismo de controle de réplicas e, por fim, a seção 4.7 apresenta as considerações finais.

É importante ressaltar que aspectos de segurança não fazem parte do escopo deste, tendo como principal foco deste trabalho o roteamento de mensagens em DTNs estocásticas.

4.1. Adaptação de Campos Magnéticos para DTN – Visão Geral

Em DTNs, nodos estabelecem e perdem conectividade com outros nodos sucessivamente durante a sua operação. A conectividade é estabelecida ou perdida quando os nodos conseguem ou não trocar mensagens diretamente, respectivamente. Esta variação da conectividade é devida a vários fatores como mobilidade dos nodos, fatores ambientais, ou características dos próprios nodos como nível de energia, dentre outros.

Portanto, considerando que nodos transportam mensagens que devem ser encaminhadas com o intuito de chegarem ao seu destino, o principal desafio de um algoritmo de roteamento para DTN é se uma mensagem deve ou não ser repassada para outro nodo quando estes estabelecem um contato.

O método de encaminhamento de mensagem em DTN definido neste capítulo, baseia-se na utilização de *Campos Magnéticos Virtuais*, o que possibilita o repasse de uma mensagem através do conceito de atração.

Na literatura, os principais trabalhos de roteamento em DTNs utilizam o método de endereçamento *unicast*, onde o destino para uma mensagem é definido no momento da sua geração, ou seja, o destino da mensagem é determinado no momento em que a mensagem é criada.

O nosso foco, no entanto, é *anycasting*, onde o destino para uma mensagem é qualquer elemento de um determinado grupo de nodos. *Anycasting* pode se reduzir a *unicasting* se o grupo de nodos destinatário contiver apenas um nodo.

Em *anycasting*, o grupo de destino da mensagem é determinado no momento em que a mensagem é criada. No contexto de CMVs (Campos Magnéticos Virtuais) para DTNs, a definição do grupo está associada ao tipo de mensagem que um nodo pode processar. Em outras palavras, quanto mais tipos um nodo é capaz de processar, a mais grupos o nodo pertence.

A decisão do repasse de uma mensagem é baseada na força de atração que um nodo possui para diferentes tipos de mensagens.

O caminho que uma mensagem utiliza para atingir o seu destino é determinado por um algoritmo distribuído que monta uma tabela de campos magnéticos semelhante a utilizada pelo algoritmo *QuickPath*.

O algoritmo atualiza o caminho dinamicamente durante o percurso de uma mensagem, uma vez que existe uma considerável possibilidade de que um nodo de destino se desconecte da sub-rede ou que a topologia da mesma mude antes da mensagem chegar a um potencial destino final.

O algoritmo original para campos magnéticos virtuais considera que todos os nodos de uma rede são potencialmente capazes de processar qualquer tipo de mensagem.

No entanto, no contexto de DTNs esta suposição não é adequada. Por exemplo, embora uma mensagem de emergência de tipo C (m_C) só pode ser processada por veículos da polícia, um ônibus também poderia transportar esta mensagem com a única intenção de repassá-la a um dos nodos aptos a processá-la. Isto é semelhante ao conceito de *anycast*, com a diferença de se restringir aqui ao nível da aplicação, característica essa comum nos trabalhos de roteamento para DTNs.

Para permitir o *anycasting* com campos magnéticos virtuais foram realizadas algumas adaptações do modelo original descrito em [20], a fim de permitir seu uso em DTNs conforme descrito nas seções seguintes.

4.2. Cálculo da Força de Atração

Seja $NG_t = (V, E_t)$ o grafo que representa o estado da DTN em um instante t arbitrário. V é o conjunto de nodos e $E_t \subseteq V \times V$ o conjunto de conexões entre nodos em um dado instante t . Neste instante, $V = \bigsqcup_{1 \leq i \leq k} V_i$, sendo cada V_i o conjunto de nodos de cada sub-rede conexa de NG_t tal que, $V_i \cap V_j = \emptyset, \forall i \neq j$ como representado na figura 22.

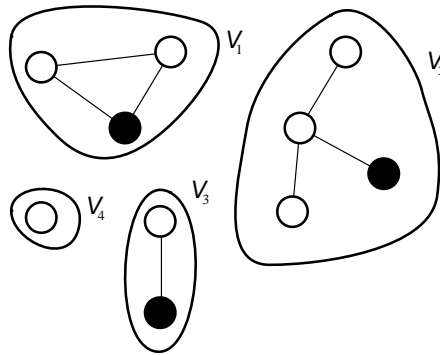


Figura 22 - Um grafo NG é composto de k ligadas sub-redes V_i

Seja $x \in V$, $S(x)$ representa a força do nodo. Seja $P_C \subseteq V$ o conjunto de nodos que são capazes de processar as mensagens do tipo C, e seja $NP_C = V - P_C$ (o conjunto de nodos que não são capazes de processar mensagens do tipo C).

Devido à mobilidade dos nodos e o fato de nem todos os nodos serem capazes de processar um tipo de mensagem específica, a força de atração é definida como se segue:

a) Um nodo $n \in NP_C$ terá valor inicial da sua força de atração de 0 (zero) para mensagens de tipo C. A fim de aumentar a sua força de atração, ele precisará estar conectado a um subgrafo conexo da rede que contém pelo menos um nodo pertencente a P_C . Seja V_i o conjunto de nós desta sub-rede conexa. Quando essa conexão é estabelecida, a força de

atração de n começa a aumentar gradualmente com o tempo desde que se mantenha conectado a uma sub-rede conexa contendo pelo menos um nodo em P_C . No entanto, a força de n permanecerá sempre abaixo da maior $\sigma(p)$ existente em P_C . Ou seja, $S(n) < Máx\{S(p), \rho \hat{V}_i P_C\}$, evitando assim que mensagens sejam atraídas para um nodo em NP_C .

b) Quando não houver mais na sub-rede conexa de um nodo n nenhum outro nodo em P_C , então n começa a diminuir gradualmente a sua força em intervalos de tempo pré-definidos. Deste modo, os nodos que recentemente fizeram parte de uma sub-rede contendo um destino potencial ocioso (i.e., com maior força de atração), terão maior força de atração do que aqueles que pertenciam a uma sub-rede dessas há mais tempo ou que se conectaram a sub-redes com nodos mais sobrecarregados.

Alguns trabalhos como o descrito em [25], consideram que os nodos que encontraram mais recentemente um nodo de destino possuam mais chances de reencontrá-lo em um futuro próximo. No entanto, a abordagem proposta neste trabalho leva também em consideração o tempo de conectividade que o nodo permaneceu em uma sub-rede contendo pelo menos um nodo em P_C . O princípio é que nas sub-redes onde os nodos passam mais tempo conectados como, por exemplo, em shopping, empresas, universidades, congestionamento de veículos, dentre outros, exista uma possibilidade maior de ocorrer o contato com um possível destino em virtude do tempo que permaneceu conectado.

No protocolo CMV, cada nodo $n \in NP_C$ monitora (ver na seção 4.3) o tempo que um nodo $x \in P_C$ está direta ou indiretamente conectado a ele, que é chamado de força de atração.

Os nodos n alteram o valor de sua força de atração, aumentando ou diminuindo, à medida que um nodo x conecta-se ou desconecta-se da sub-rede em cada período de tempo.

A cada nova conexão direta ou indireta com um nodo x , o nodo n aumenta a sua força de atração enquanto conectado em um intervalo de tempo registrado em seu relógio local. E a cada desconexão do nodo x , o nodo n diminui sua força de atração em um intervalo de tempo registrado em seu relógio local.

Portanto, a força de atração é uma função que considera o tempo que um nodo em NP_C permaneceu conectado ou desconectado a um nodo em P_C .

Para entendermos melhor a aplicação do cálculo da força citada acima, consideremos um cenário básico composto de 3 nodos. Além disso:

- somente o nodo 1 processa mensagens do tipo C (i.e., $P_C = \{1\}$);
- os nodos 2 e 3 não processam mensagens do tipo C (i.e., $NP_C = \{2,3\}$).

Inicialmente existem três nodos, sendo que os nodos 1 e 2 fazem parte de uma sub-rede (embora o nodo 1 seja o único apto a receber e processar mensagens da classe C). O nodo 3 não faz para as sub-rede citada, porém está com a custódia de uma mensagem m_c (Figura 23).

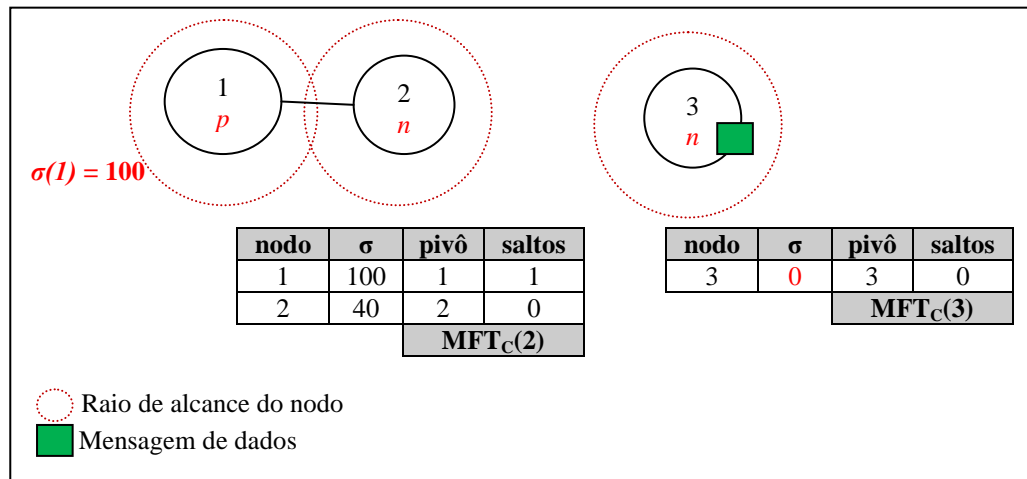


Figura 23 – Sub-rede composta pelos os nodos 1 e 2

Algum tempo depois, o nodo 3 se locomoveu no sentido da sub-rede e, em um determinado momento, foi conectada a sub-rede. Percebe-se que nesse intervalo de tempo a força de atração do nodo 2 aumentou baseado em um intervalo de tempo que esteve conectado a uma sub-rede que era contemplada com um nodo pertencente a P_C (Figura 24).

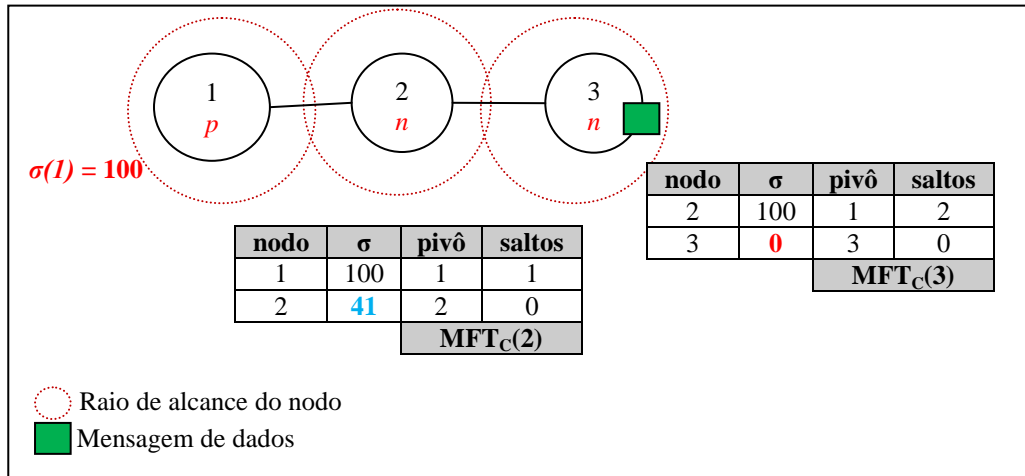


Figura 24 - Nodo 3 conecta-se a sub-rede

Como pode ser observado, o nodo 2 sofre influência magnética do nodo 1 e o nodo 3 sofre influência do nodo 2. No exemplo, a mensagem m_c é direcionada ao nodo 2, pois o nodo com maior força que exerce influência (indireta) sobre o nodo 3 é o nodo 1 como pode ser visto na (Figura 25).

Ao chegar ao nodo 2, a mensagem m_c será finalmente atraída para o nodo 1, pois é o nodo com maior força. O nodo 1, finalmente processará a mensagem m_c .

Note que, ao longo do tempo, as forças dos nodos 2 e 3 aumentaram, como pode ser visto na (Figura 26).

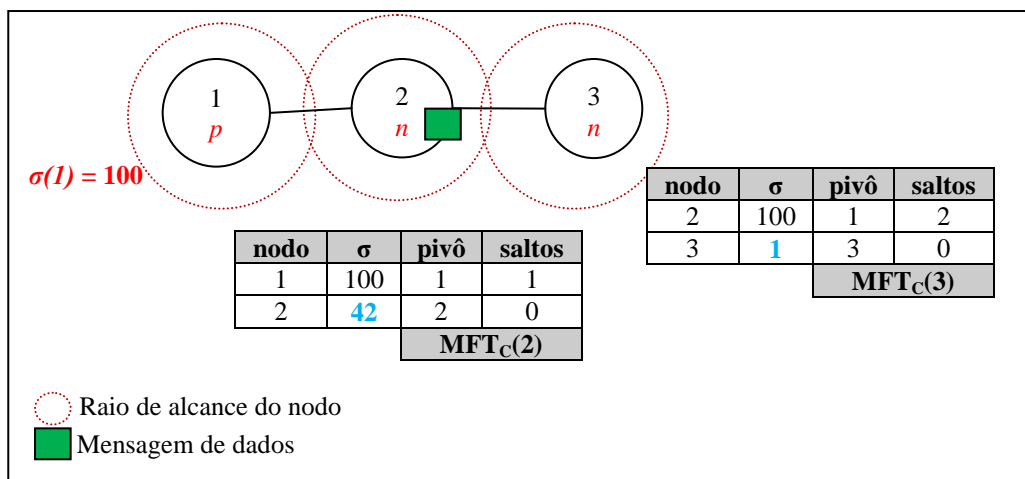


Figura 25 - A mensagem m_c é atraída para o nodo 2

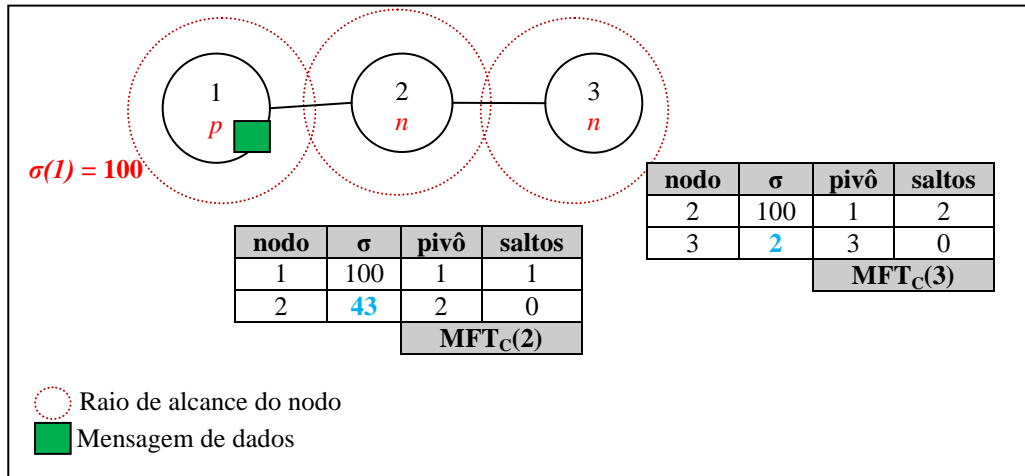


Figura 26 - Nodo 1 recebendo a mensagem m_c

Algum tempo depois, em virtude da mobilidade dos nodos, o nodo 1 desconecta-se da sub-rede, ou seja, a partir desse momento o nodo 3 passar a sofrer influência direta do nodo 2 como pode ser visto na $MFT_C(3)$ na (Figura 27).

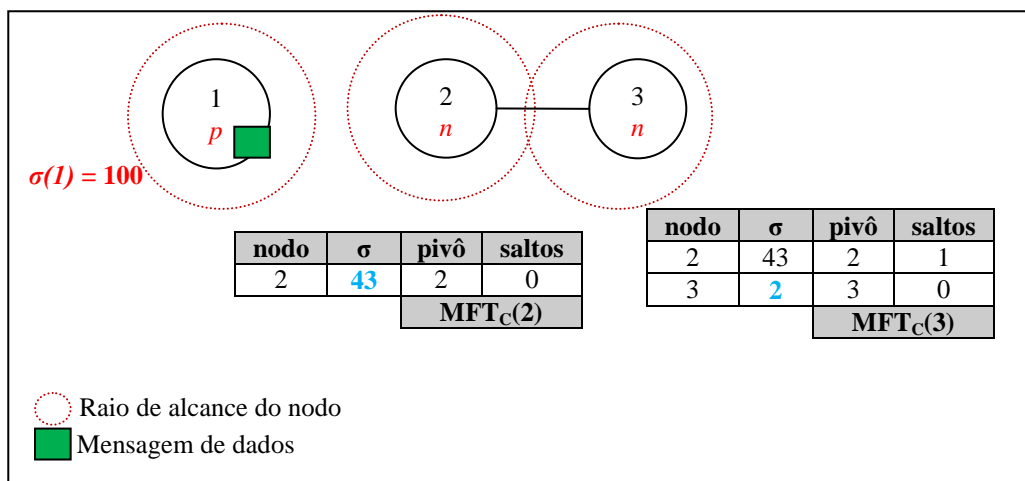


Figura 27 - Nodo 1 desconecta-se da sub-rede

Note que, ao longo do tempo, as forças dos nodos 2 e 3 diminuiram em virtude da inexistência de um nodo pertencente a P_C na sub-rede, como pode ser visto na (Figura 28).

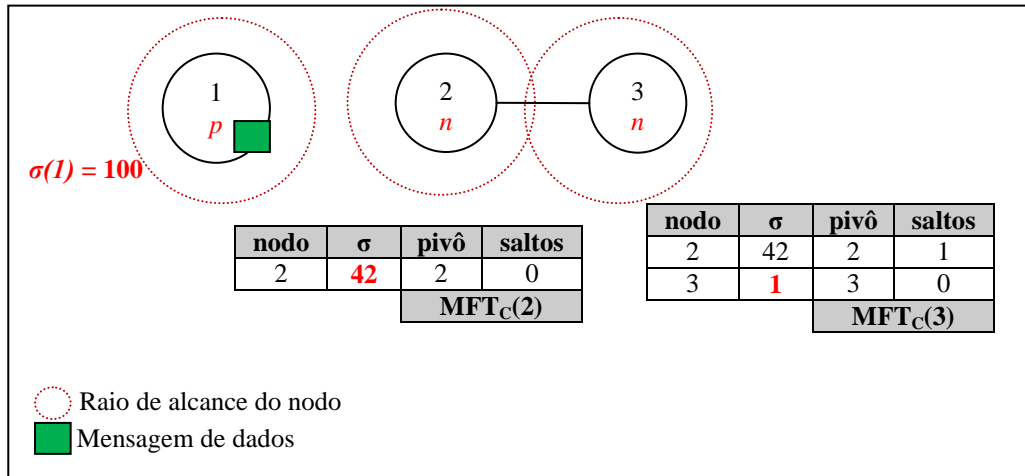


Figura 28 - Diminuição da força dos nodos 2 e 3

4.3. Heartbeat

Conforme descritos na seção 4.2, os nodos que não podem processar mensagens do tipo C aumentam e diminuem a sua força atração se estiverem conectados direta ou indiretamente a um nodo que processa mensagens do tipo C.

Portanto, é necessário utilizar um algoritmo de notificação que informe aos nodos em NP_C sobre a existência de nodos $p \in P_C$ na sub-rede conexa ao qual pertencem. Para atender essa finalidade foi criado o algoritmo denominando *Heartbeat*.

O algoritmo *heartbeat* usa mensagens de controle de forma a que os nodos em NP_C saibam o que fazer com suas forças de atração com o passar do tempo. Mensagens de *Heartbeat* também são usadas para limitar as forças de atração de nodos em NP_C com base na regra definida na seção 4.2. Outra finalidade do algoritmo *heartbeat*, é permitir que os nodos em NP_C , identifiquem o nodo mais forte em P_C , permitindo assim que os nodos em NP_C atualizem a entrada na MFT correspondente ao nodo pertencente a P_C com maior força.

Portanto, nodos pertencentes a P_C enviam essas mensagens de *Heartbeat* a fim de comunicar a sua presença, as suas forças de atração e possibilitam que nodos em NP_C saibam qual o caminho até o nodo mais forte em P_C .

O algoritmo *Heartbeat* é apresentado no Algoritmo 1. Este algoritmo tem as seguintes estruturas e variáveis:

- x : o identificador do nodo que está executando o algoritmo;
- $N(x)$ é o conjunto de vizinhos diretos do nodo x ;
- $set_alarm(t)$: faz com que um evento *When* seja acionado dentro de t unidades de tempo;
- $\sigma(y)$: força de atração do nodo cujo identificador é y ;
- σ_{limit} : força máxima permitida para o nodo (em NP_C) em um dado instante;
- *origin*: a identificação do nodo adjacente que repassou a mensagem *heartbeat*;
- τ_1 : intervalo de tempo para gerar *heartbeats*;
- τ_2 : intervalo de tempo para aumentar e diminuir a força de atração;
- *adjustStrength* ($\sigma_{limit}, \sigma(x), \sigma(y)$): ajusta a força proporcionalmente ao novo limite (σ_{limit} , = limite antigo; $\sigma(x)$ = força; $\sigma(y)$ = novo limite);
- *increaseStrength* ($\sigma(x)$): função genérica para aumentar a força de atração enquanto receber mensagens *heartbeats* (até σ_{limit});
- *decreaseStrength* ($\sigma(x)$): função genérica para diminuir a força de atração quando cessa o recebimento de *heartbeats*.
- *updateMaxPcMFT* ($hb(\sigma(y))$): atualiza a entrada da MFT correspondente ao nodo com maior força.

Algoritmo 1 – Algoritmo *Heartbeat*

<p>ESTADOS = {NP,P}</p> <p>NP</p> <p><i>Spontaneously</i> begin $\sigma_{limit} := 0;$ set_alarm(τ_2); end</p> <p><i>When</i> begin $\sigma(x) := decreaseStrength(\sigma_{limit});$ set_alarm(τ_2); end</p> <p><i>Receiving $hb(s(y))$</i> begin set_alarm(τ_2); // $t_2 > t_1$ if $\sigma(y) \neq \sigma_{limit}$ then $\sigma(x) := adjustStrength(\sigma_{limit},$ $\sigma(x), \sigma(y));$ $\sigma_{limit} := \sigma(y);$ else $\sigma(x) := increaseStrength(\sigma_{limit});$ endif $updateMaxPcMFT(s(y));$ send ($hb(s(y))$) to N(x) - {origin}; end</p>	<p>P</p> <p><i>Spontaneously</i> begin send ($hb(s(x))$) to N(x); set_alarm(τ_1); end</p> <p><i>When</i> begin $updateMaxPcMFT(s(x));$ send ($hb(s(x))$) to N(x); set_alarm(τ_1); end</p> <p><i>Receiving $hb(s(y))$</i> begin if $s(y) > s(x)$ then $updateMaxPcMFT(s(y));$ send ($hb(s(y))$) to N(x) - {origin}; set_alarm($\tau_1 + 1$); endif end</p>
--	--

O algoritmo *Heartbeat* admite apenas dois estados que são NP_C e P_C , sendo executado correspondentemente ao conjunto que o nodo executante pertence.

No estado NP_C , quando um nodo é inicializado, é atribuído o valor zero para o limite da sua força e o alarme é configurado para um intervalo τ_2 de tempo.

Se nesse intervalo, o nodo receber uma mensagem *heartbeat*, o alarme é configurado para um intervalo τ_2 de tempo e em seguida é verificado se a força de atração contida no *heartbeat* recebido é diferente do limite da força de atração, caso seja, o nodo em NP_C ajusta proporcionalmente a força ao novo limite através da função *adjusteStrength*. Em seguida, o nodo ajusta o limite máximo que sua força de atração pode alcançar (σ_{limit}).

No entanto, caso a força recebida no *heartbeat* seja igual a sua própria força de atração, a força é incrementada através da função *increaseStrength*, sendo essa responsável em realizar o incremento da força da atração respeitando o limite máximo permitido naquele.

Após o fim da execução da condição de seleção supracitada, a função *updateMaxPcMFT* é chamada com a finalidade de atualizar a entrada da MFT correspondente ao nodo com a maior força em P_C . Por fim, o nodo repassa a mensagem de *heartbeat* para todos os nodos adjacentes, com exceção do nodo que lhe repassou a mensagem e aguarda o recebimento do próximo *heartbeat*.

Contudo, caso não receba um *heartbeat*, ou seja, não existe nenhum nodo em P_C conectado à sub-rede, a força de atração do nodo começa a ser decrementada através da função *decreaseStrength* em um intervalo τ_2 de tempos até que ocorra o recebimento de uma mensagem *heartbeat*, ou seja, até que um nodo P_C se conecte a sub-rede.

A linha do tempo referente ao funcionamento do alarme no estado NP_C pode ser observada na (Figura 29). Note que toda vez que um alarme é agendado (operação *set_alarm*), o anterior é cancelado. O alarme só será disparado, se o nodo não receber nenhuma uma mensagem de *heartbeat* (*hb*) no intervalo τ_2 . Se receber, o alarme será adiado.

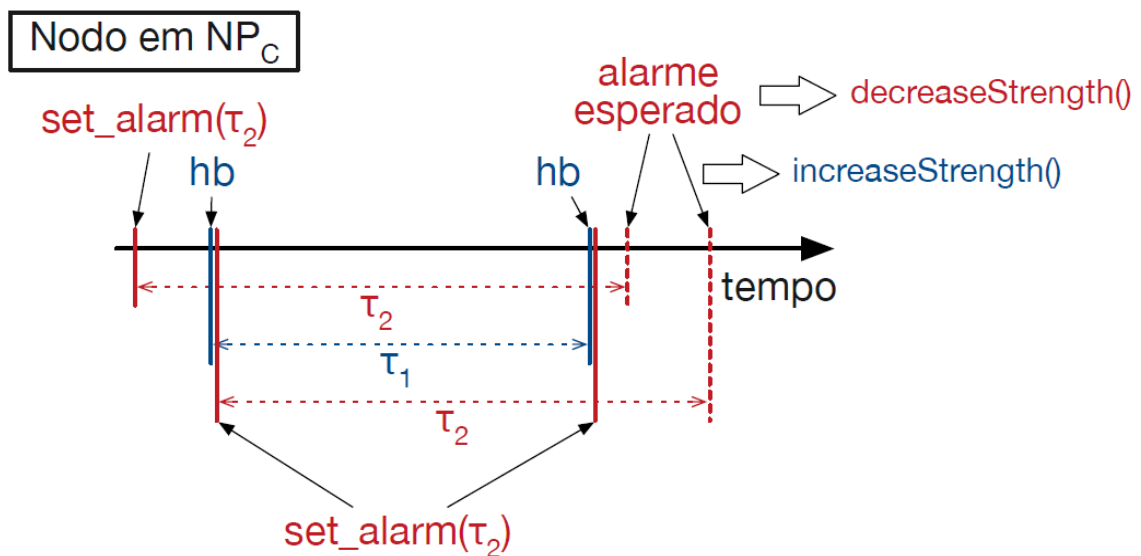


Figura 29 - Linha do tempo do alarme no estado NP_C

No estado P_C , os nodos em P_C são os responsáveis pela geração da mensagem *heartbeat*. Quando um nodo é inicializado, ele transmite um mensagem de *heartbeat* contendo

a sua própria força para os nodos adjacentes e configura o alarme para um intervalo τ_1 de tempo.

No entanto, caso o mesmo receba um *heartbeat* enviado por outro nodo em P_C , o nodo verifica se a força de atração contida no *heartbeat* recebido é maior que a sua, e caso seja, a função *updateMaxPcMFT* é chamada com a finalidade de atualizar a entrada da MFT correspondente ao nodo com a maior força em P_C , e em seguida o nodo repassa a mensagem de *heartbeat* para todos os nodos adjacentes, com exceção do nodo que lhe repassou. Por fim, o alarme é configurado para o intervalo $\tau_1 + 1$ de tempo e aguarda o próximo *heartbeat*. É necessário um intervalo maior que τ_1 para que o nodo possa voltar a repassar o seu próprio *heartbeat* caso não receba *heartbeats* de nodos mais fortes.

Assim, caso não receba uma nova mensagem *heartbeat* no tempo configurado no alarme, a função *updateMaxPcMFT* é chamada com a finalidade de atualizar a entrada da MFT com a sua própria força. Em seguida, o nodo começa a enviar o seu próprio *heartbeat* para os nodos adjacentes e em seguida configura o alarme para o intervalo τ_1 de tempo.

A linha do tempo referente ao funcionamento do alarme no estado P_C está ilustrada na (Figura 30). No exemplo, o nodo repassa o *heartbeat* de outro nodo mais forte, e o alarme é sempre adiado.

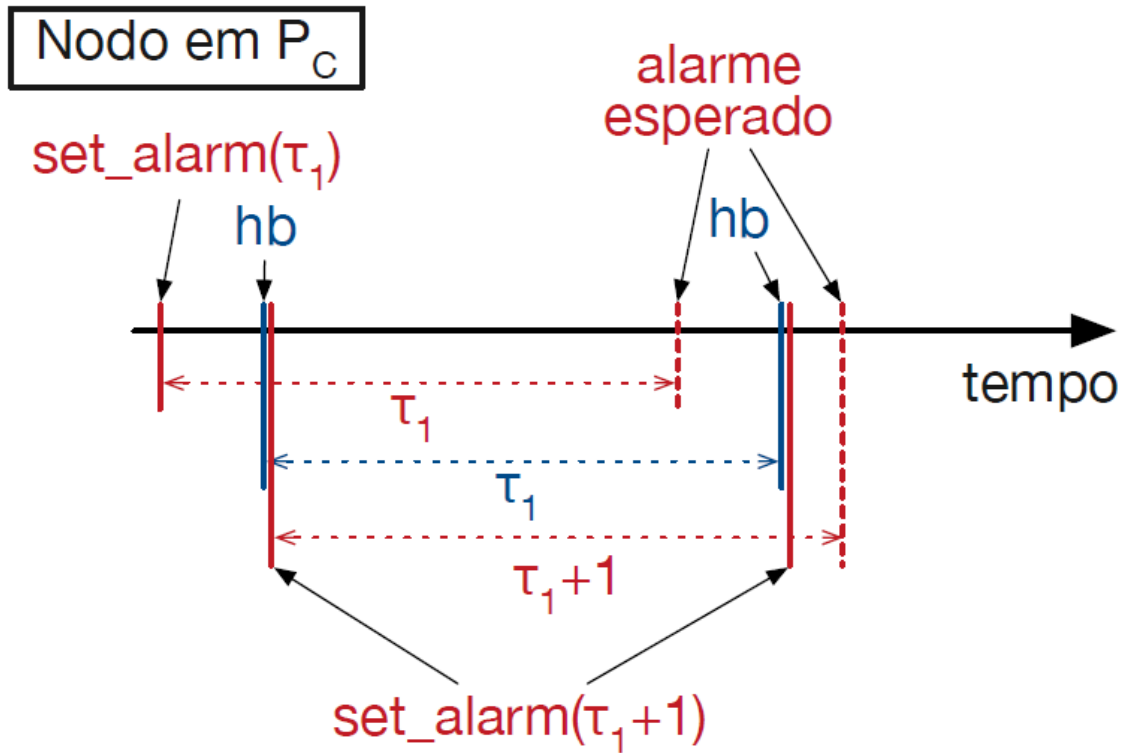


Figura 30 - Linha do tempo do alarme no estado P_C

Em ambos os estados, se o nodo em P_C ou em NP_C receber uma mensagem em duplicidade a mesma é descartada. Sendo que a primeira recebida indica que o caminho percorrido por ela é o mais rápido e conseqüentemente o nodo que a repassou que terá a influência sobre o nodo que a recebeu, conforme previsto no algoritmo original do *QuickPath*.

No algoritmo *QuickPath*, o princípio básico para o disparo de propagação de mensagens ocorre quando há eventos que modifiquem a força do nodo mais forte ou modifiquem o pivô percebido por um nodo. Ocorrendo esta condição em um nodo, ele dispara uma mensagem de propagação notificando aos nodos vizinhos sobre a alteração. Estes últimos, por sua vez, verificam a mesma condição para decidirem se dão ou não sequência à propagação.

Contudo, em DTNs, existem algumas peculiaridades que são a mobilidade dos nodos que geram frequentes desconexões, limitações de *buffers*, consumo excessivo de energia dos dispositivos em virtude do número excessivo de transmissões, dentre outros. Conseqüentemente, a topologia da sub-rede tende a mudar constantemente, havendo assim a necessidade de um número excessivo de mensagens de propagação caso seja utilizado o *QuickPath* em seu conceito original no contexto de DTNs, devido às constantes alterações de

força e/ou pivô. Assim, foram realizadas algumas mudanças no algoritmo original com o intuito de diminuir o número de mensagens de propagação e que atendesse às características de mobilidade inerentes a DTNs e que, ao mesmo tempo, conseguisse montar um caminho de forma dinâmica à medida que a topologia se altera durante o trajeto da mensagem de dados até seu destino.

Portanto, em virtude das limitações descritas acima, a propagação da força realizada pelo *heartbeat* ocorre somente com o objetivo de informar aos nodos em NP_C qual nodo em P_C possui a maior força, a que distância se encontra e o caminho mais curto para chegar ao mesmo. Ou seja, um dos objetivos do *heartbeat* é prover as MFTs dos nodos em NP_C o caminho até o nodo em P_C com maior força.

Para entendermos melhor a propagação supracita, o aumento e diminuição das forças e limitação das forças em nodos NP_C pelo *heartbeat*, consideremos um cenário básico. Além disso:

- $p, p' \in P_C$ (p e p' são capazes de processar mensagens do tipo C).
- $n_i \in NP_C$ (n_i é um nodo em NG_t que não é capaz de processar mensagens)
- $c \in NP_C$ (c é um nodo com a custódia da mensagem)

Na (Figura 31), é representado um cenário de uma sub-rede conexa contendo pelo menos um nodo em P_C . Como pode ser observado na $MFT_C(c)$, o nodo c que possui a custódia de uma mensagem m_c sofre influência do nodo n_l , pois o nodo com maior força que exerce influência (indireta) sobre o nodo n_l é o nodo p , essa influência indireta ocorreu em virtude do nodo n_l ser o responsável em repassar a mensagem *heartbeat* enviado por p e não por n_l ser vizinho direto de p .

Todavia, podemos observar na própria $MFT_C(c)$, que a mesma não possui nenhuma informação relacionada ao nodo p' , devido ao próprio nodo p' constatar que a força de atração contida no *heartbeat* enviado por p é maior que a sua, e conseqüentemente começa a repassar o *heartbeat* recebido ao invés do seu próprio.

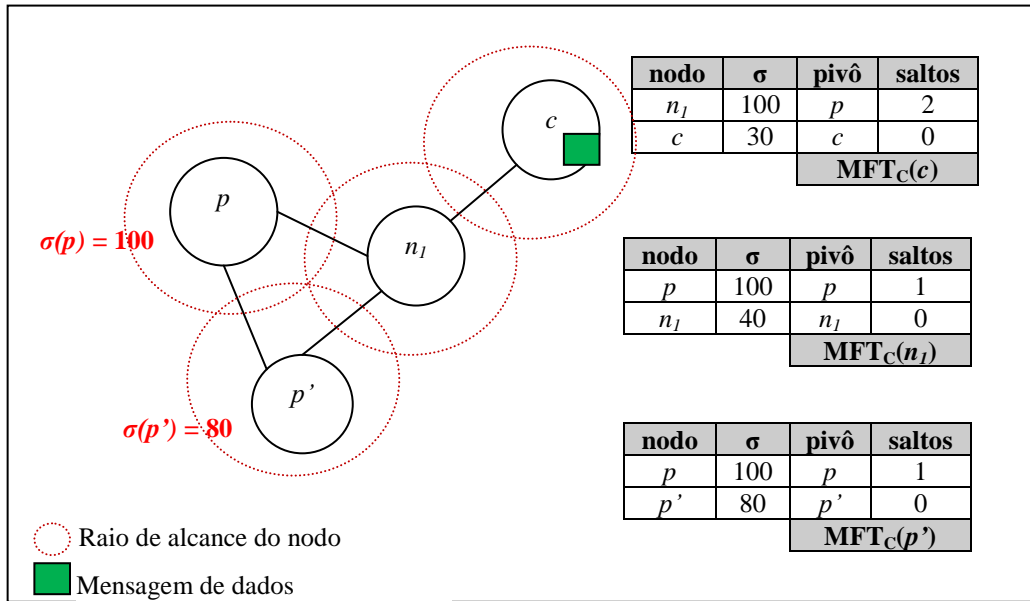


Figura 31 – Sub-rede conexa contendo pelo menos um nodo em P_c

Como pode ser observado, o nodo n_1 sofre influência magnética do nodo p e o nodo c sofre influência do nodo n_1 . No exemplo, a mensagem m_c é direcionada ao nodo n_1 , pois o nodo com maior força que exerce influência (indireta) sobre o nodo c é o nodo p . Ao chegar ao nodo n_1 , a mensagem m_c será finalmente atraída para o nodo p , pois é o nodo com maior força. O nodo p , finalmente processará a mensagem m_c como pode ser visto na (Figura 32).

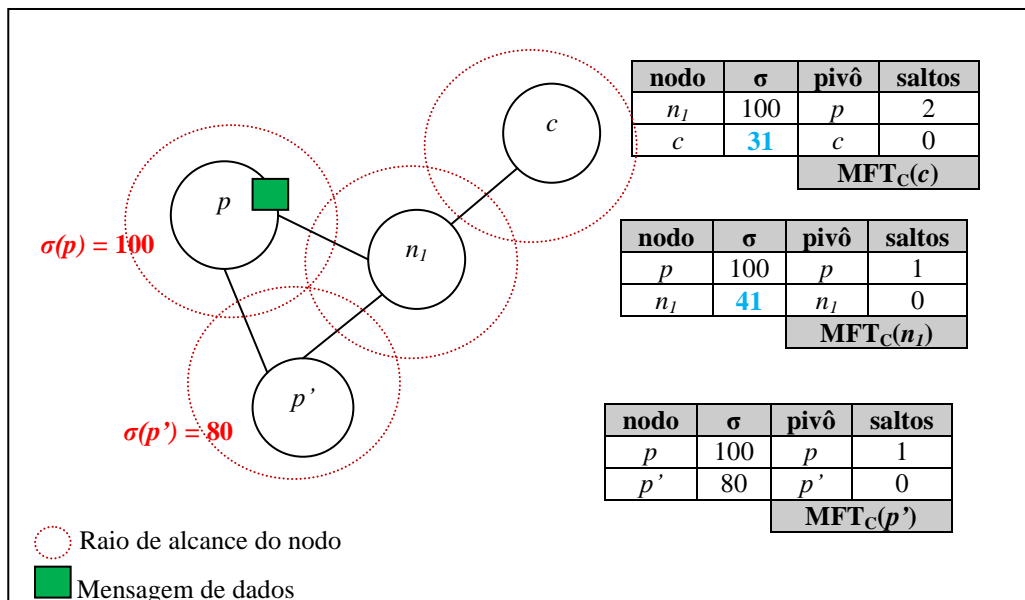


Figura 32 - Mensagem de dados é atraída para o nodo em P_c mais forte

Note que, ao longo do tempo, as forças dos nodos n_I e c aumentaram as suas forças sem ultrapassarem a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$, como pode ser visto na (Figura 33).

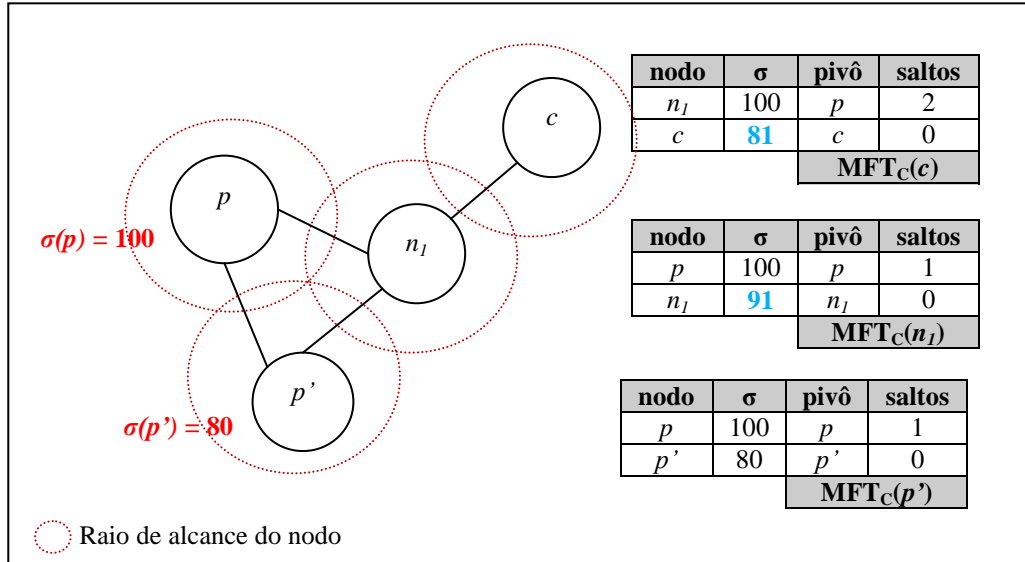


Figura 33 - Aumento da força dos nodos em NP_C sem ultrapassarem a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$

Algum tempo depois, em virtude da mobilidade dos nodos, o nodo p desconecta-se da sub-rede, ou seja, o nodo p' passa a ser o nodo mais forte em P_C , consequentemente o mesmo começa a enviar o seu *heartbeat*. Com o recebimento do novo *heartbeat*, os nodos em NP_C ajustam as suas forças proporcionalmente ao novo limite imposto por p' , como pode ser visto nas MFT_C(n_I) e MFT_C(c) na (Figura 34).

Portanto, a partir desse momento o nodo n_I passar a sofrer influência direta do nodo p' e o nodo c sofre influência (indireta) do nodo p' .

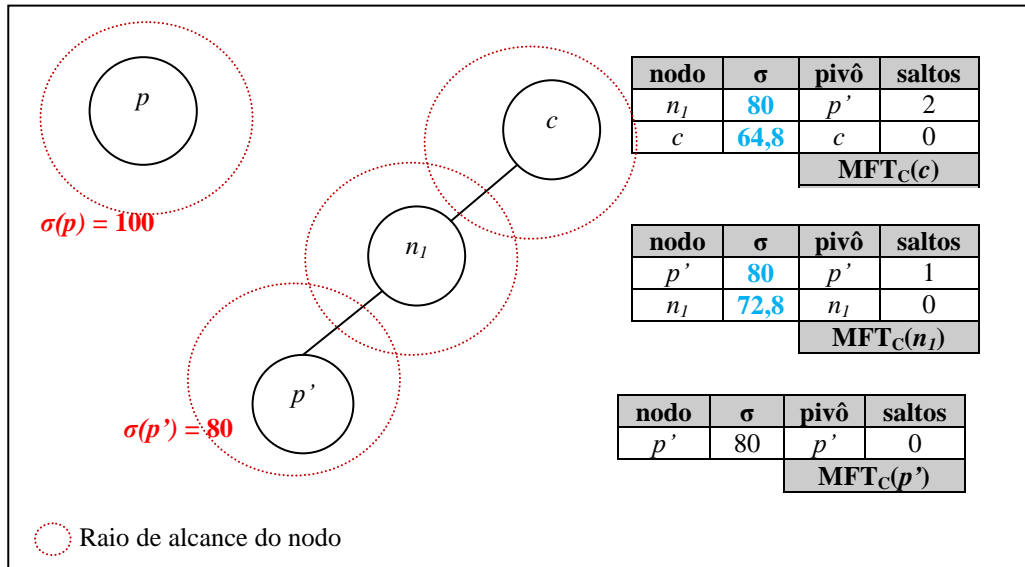


Figura 34 – Ajuste proporcional da força dos nodos em NP_C devido ao novo limite

Alguns tempo depois, o nodo p' desconecta-se da sub-rede, ou seja, a sub-rede passa a não ter mais nodos pertencentes em P_C , conseqüentemente cada nodo em NP_C exclui de sua MFT a entrada correspondente ao $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$. Portanto, as forças dos nodos n_1 e c diminuirão ao longo do tempo em virtude da inexistência de um nodo pertencente a P_C na sub-rede, como pode ser visto na (Figura 35).

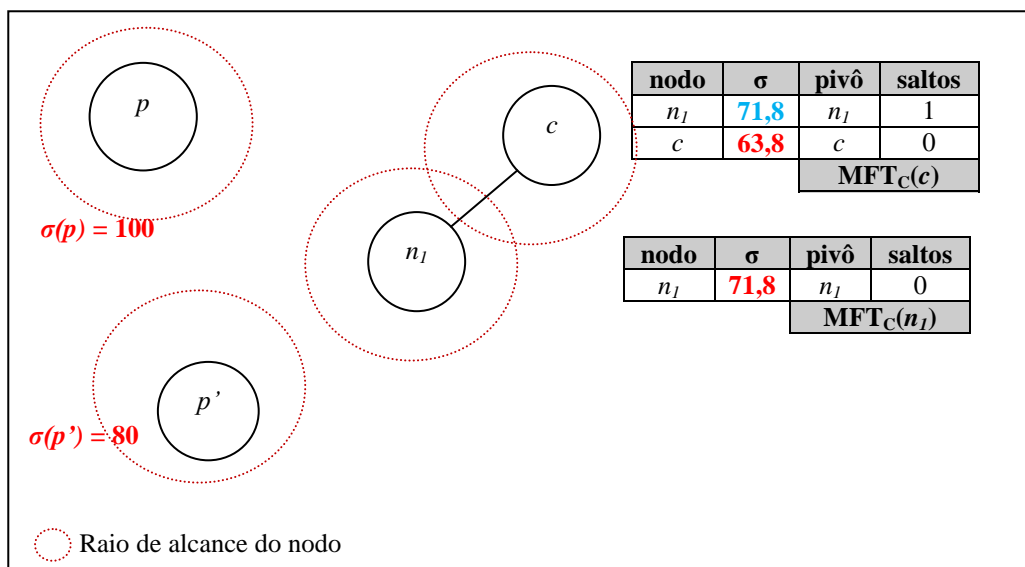


Figura 35 – Diminuição da força dos nodos em NP_C

No entanto, vale ressaltar que o objetivo do cenário demonstrado acima é simplesmente mostrar o funcionamento do algoritmo *Hearbeat* nas seguintes atribuições: propagação da força do nodo em P_C mais forte, ajuste da limitação da força dos nodos em NP_C e por fim, o aumento e diminuição da força. Sendo consideradas na seção 4.4, questões relacionadas a roteamento de mensagens de dados, entregas diretas e trocas de forças entre nodos em NP_C .

4.4. Roteamento da Mensagem

Uma das decisões mais difíceis e importantes em DTNs está relacionada ao repasse ou não da mensagem.

Quando utilizamos o termo repasse de uma mensagem, estamos se referindo a transmissão de uma mensagem de dados (aplicação) de um nodo para um nodo adjacente que não é o destino final da mensagem.

No entanto, quando utilizamos o termo entrega direta (*Direct Delivery*) de uma mensagem, estamos se referindo a transmissão de uma mensagem de dados de um nodo para um nodo adjacente que é o destino final da mensagem.

Com base na sua MFT, um nodo decide se faz um repasse, uma entrega direta ou se continua com a custódia da mensagem. Podendo levar em consideração a força e a distância que o outro nodo está localizado. Portanto, as entradas que compõem a tabela de campos magnéticos são fundamentais para essa decisão.

A criação da tabela de campos magnéticos é realizada no momento em que um nodo ingressa em uma sub-rede.

A inclusão, alteração ou exclusão dos nodos que compõem a MFT_C ocorre naturalmente em virtude da desconexão dos nodos adjacentes atuais e das novas conexões adjacentes causadas pela mobilidade dos nodos e conseqüentemente mudando a topologia da sub-rede. Sendo que os nodos em P_C e NP_C realizam procedimentos distintos conforme descritos abaixo

Quando um nodo p ingressa em uma sub-rede. Neste momento o nodo faz uma varredura nos nodos adjacentes e caso algum nodo adjacente possua um *heartbeat*, o mesmo é repassado ao nodo p que fará a atualização da MFT_C caso a força contida no *heartbeat* seja superior a $\sigma(p)$, conforme descrito na seção 4.3.

Contudo, quando um nodo n ingressa em uma sub-rede. Neste momento o nodo n faz uma varredura nos nodos adjacentes e caso algum nodo adjacente possua um *heartbeat*, o mesmo é repassado ao nodo n que fará a atualização da MFT_C conforme descrito na seção 4.3. Caso não tenha recebido o *heartbeat* ou o *heartbeat* recebido tenha como pivô um nodo em P_C não adjacente, o nodo n então verifica a existência de nodos em P_C adjacentes, caso existam, n seleciona o que possuir a maior força e atualiza na MFT_C . Por fim, caso não existam nodos em P_C , verifica-se a existência de nodos adjacentes pertencentes a adjacentes pertencentes a NP_C que possuam a força de atração maior que $\sigma(p)$. Caso existam, n seleciona o que possuir a maior força e o adiciona à MFT_C .

Portanto, a MFT pertencente a um nodo em P_C conterà no máximo duas entradas que são: a entrada correspondente à própria força e a correspondente ao nodo que possua a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$, se ambas forem pertencentes ao mesmo nodo permanecerá somente uma entrada.

No entanto, a MFT pertencente a um nodo em NP_C conterà no máximo três entradas que são: além da entrada correspondente à própria força, a entrada correspondente ao nodo que possua a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$ e caso o mesmo não seja adjacente, poderá também existir a entrada correspondente ao nodo em P_C adjacente que possua a maior força. Senão, se não existirem nodos em P_C , a MFT será composta pelo correspondente à própria força mais a entrada correspondente ao nodo em NP_C adjacente que possua a maior força e que a $\sigma(n) > \sigma(n')$.

Em virtude das limitações em DTNs citadas na seção 4.3, foram definidas regras de como e quando as forças são propagadas. Nos casos dos nodos pertencentes a P_C , somente o nodo mais forte faz a propagação através do algoritmo *heartbeat* como descrito na seção 4.3.

Em uma sub-rede conexa contendo pelo menos um nodo em P_C , a propagação da força somente é realizada pelo o nodo em P_C mais forte através do *heartbeat* com o intuito de reduzir a utilização dos recursos disponíveis. Sendo justificada pelas chances reais da entrega de uma mensagem a um possível destino existente na V_i .

Contudo, os nodos em NP_C não fazem a propagação da força e conseqüentemente também não propagam o pivô, se limitando apenas a informarem as suas forças aos nodos adjacentes no momento da conexão. A justificativa inicial se deve ao fato de não haver chances reais de entrega de uma mensagem a um destino final na V_i , isso se deve ao fato da sub-rede conexa em questão não possuir nodos em P_C . Portanto, seria um uso desnecessário

de recursos como energia, rede, *buffers*, dentre outros. Principalmente em cenários onde não existem nodos em P_C e nem mensagens a serem entregues.

Nos casos das atualizações da força de atração de um nodo adjacente $n' \in NP_C$ que compõe a MFT_C de n , é possível prever o aumento da sua força, devido ao fato do nodo n como também n' estarem recebendo a mensagem de *heartbeat*. Nos casos que não existam nodos em P_C , é possível estimar a força em virtude da execução do algoritmo *heartbeat* onde o tempo local de cada nodo passa na mesma velocidade para todos.

Com os dados atualizados na MFT , o nodo faz uma análise e realiza o repasse ou a entrega direta da mensagem. Essa análise considera, além da força, o número de saltos até um nodo em P_C . Isto permite, por exemplo, aplicar uma atenuação pelo número de saltos, e consequentemente isto impactaria na influência que um nodo em P_C tem sobre um determinado nodo em NP_C .

Onde poderíamos considerar o seguinte o exemplo da equação 1.2, sendo diretamente proporcional à força e inversamente proporcional à distância, dependendo da aplicação.

- se $\frac{\sigma(p)}{d(c,p)} > \frac{\sigma(p')}{d(c,p')}$ a mensagem é enviada a p
 - se $\frac{\sigma(p)}{d(c,p)} < \frac{\sigma(p')}{d(c,p')}$ a mensagem é enviada a p'
 - se $\frac{\sigma(p)}{d(c,p)} = \frac{\sigma(p')}{d(c,p')}$ a mensagem é enviada ao mais próximo
- (1.2)

No entanto, em virtude das diversas restrições impostas nas DTNs, o método foi proposto limitando a aplicação da equação 1.2, entre o nodo em P_C indireto mais forte com o nodo em P_C adjacente mais forte, ou seja, é analisada a viabilidade da entrega entre o nodo em P_C indireto mais forte com o nodo em P_C adjacente (*Direct Delivery*).

Contudo, o modelo pode ser facilmente adaptado para que seja viável a aplicação da equação com qualquer nodo em P_C . No entanto, essa modificação exigiria uma mudança na propagação da força realizada pelo algoritmo *Heartbeat*, onde todos os nodos em P_C seriam obrigados a enviarem seus próprios *heartbeat*, independentemente de possuírem uma força de atração inferior ao nodo em P_C mais forte. Tendo como consequência um aumento exponencial no número de mensagens *heartbeat*.

A (Figura 36) ilustra um exemplo de grafo juntamente com suas Tabelas de Campos Magnéticos. Conforme pode ser observado, a figura mostra uma sub-rede composta por 6 nodos, sendo $P_C = \{p\}$ e $NP_C = \{n_1, n_2, n_3, n_4, c\}$ e $\sigma(p)=100$ e c com a custódia de uma mensagem. Suponhamos que os nodos n_3, n_4 e c sejam adjacentes entre eles. Mesmo o nodo n_2 possuindo a força de atração maior que o nodo n_1 , o mesmo não é inserido na $MFT_C(n_1)$ conforme descrito anteriormente, ou seja, isso somente ocorrerá quando não existir nodos em P_C .

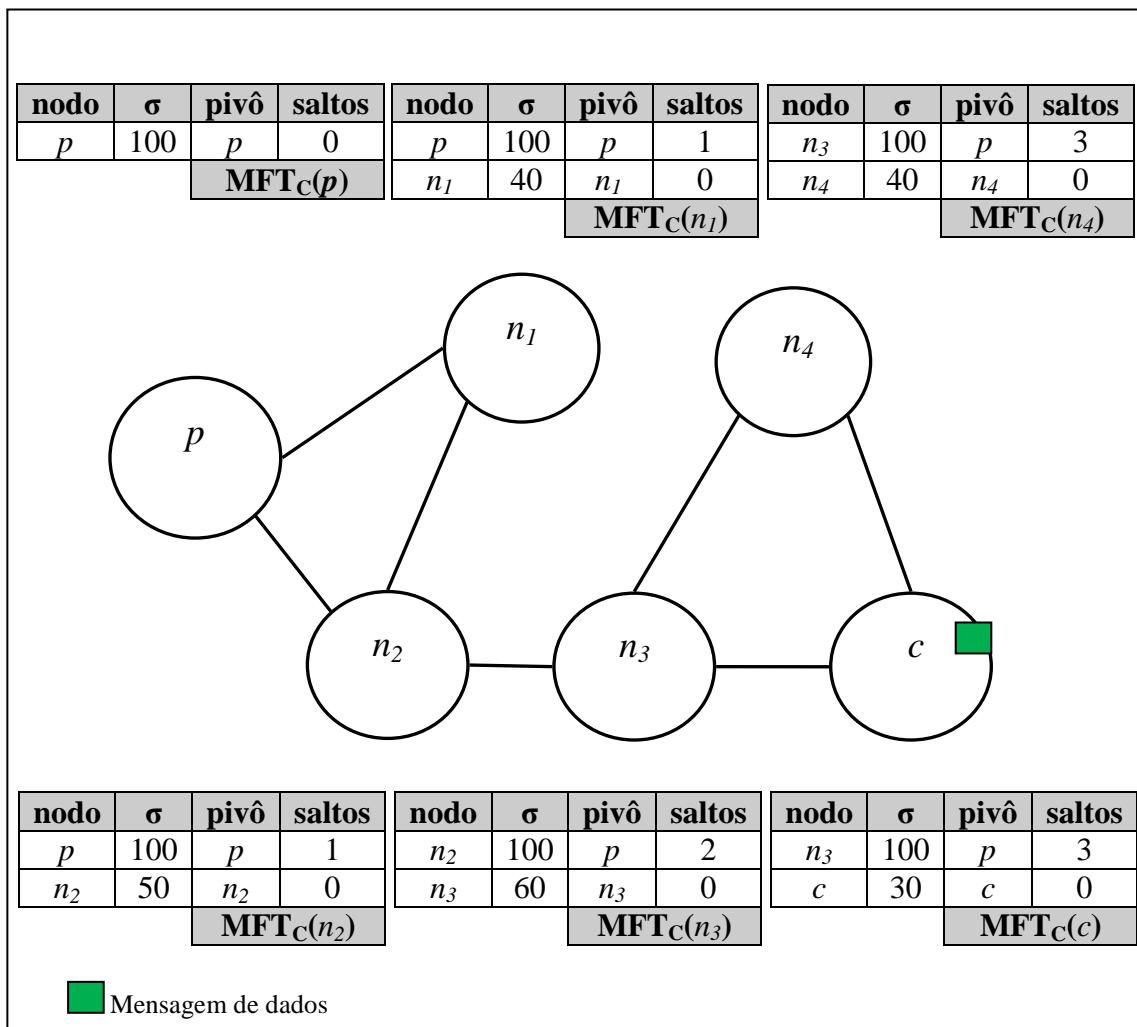


Figura 36 – A NG_t e a Tabela de Campos Magnéticos (MFT)

No cenário, a mensagem m_c é direcionada ao nodo n_3 , pois o nodo com maior força que exerce influência (indireta) sobre o nodo c é o nodo p . Ao chegar ao nodo n_3 , a mensagem m_c é direcionada ao nodo n_2 , pois o nodo com maior força que exerce influência (indireta)

sobre o nodo n_3 é o nodo p . Ao chegar ao nodo n_2 , a mensagem m_c será finalmente atraída para o nodo p , pois é o nodo com maior força. O nodo p , finalmente processará a mensagem m_c como pode ser visto na (Figura 37).

Note que, ao longo do tempo, as forças dos nodos em NP_C aumentaram as suas forças sem ultrapassarem a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$.

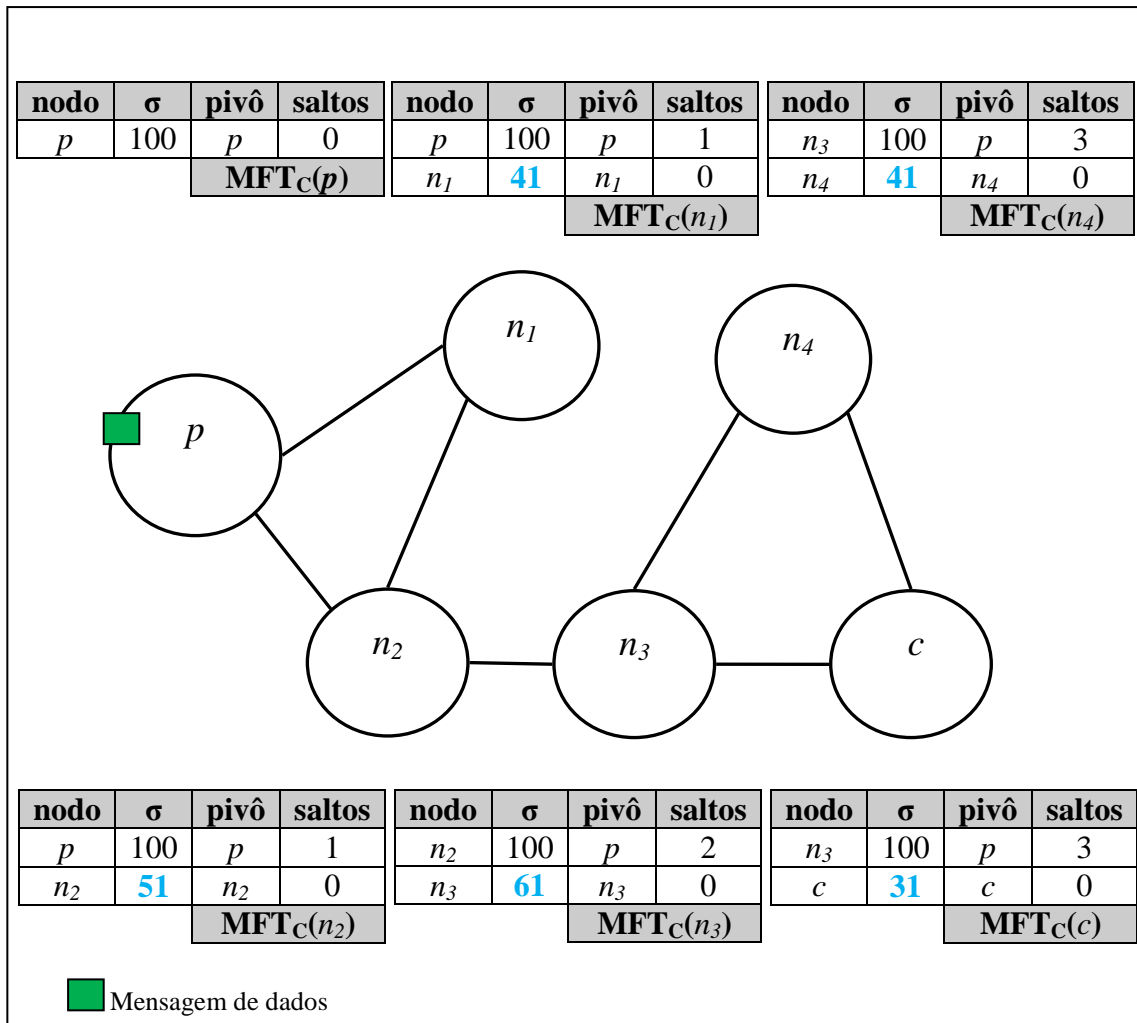


Figura 37 – Mensagem de dados é atraída para o nodo em P_C mais forte

Algum tempo depois, em virtude da mobilidade dos nodos, o nodo n_4 desconecta-se da sub-rede e um novo nodo $p' \in P_C$ com a $\sigma(p')=80$ conecta-se ao nodo c . Em seguida c gera uma nova mensagem de dados m_c . A partir deste momento, o nodo c continuará sofrendo influência do nodo em P_C com maior força, ou seja, p e também do nodo adjacente mais forte pertencente a P_C , ou seja, do nodo p' como pode ser visto na (Figura 38),

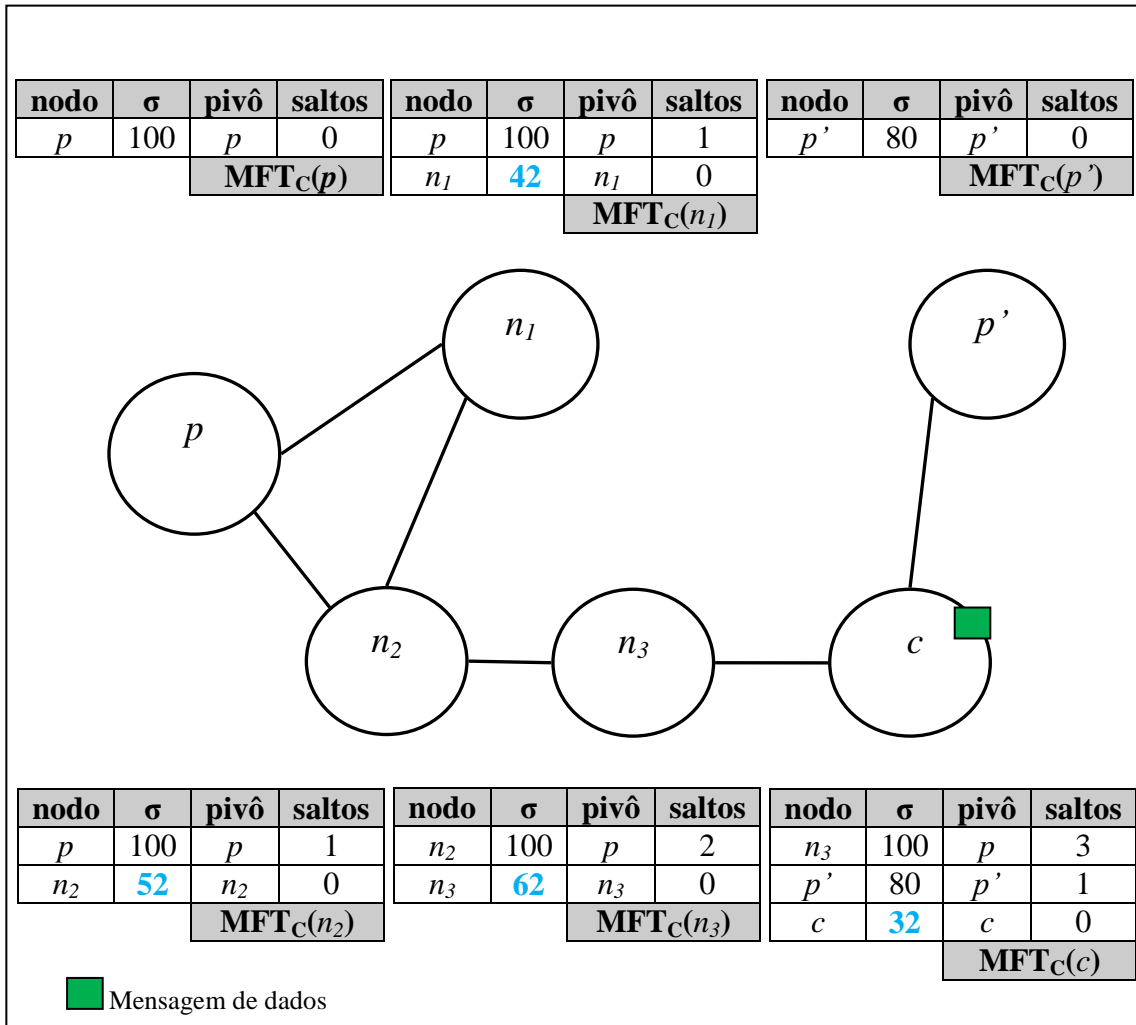


Figura 38 – Sub-rede com dois nodos em P_c

Como pode ser observado na $MFT_C(c)$, o nodo c sofre influência de dois nodos pertencentes a P_c . No entanto, o nodo n_3 que possui a maior força encontra-se a 3 saltos do nodo c , ao contrario do nodo p' que está a 1 salto, mas possui uma menor força de atração. Todavia, o modelo possibilita aplicar uma atenuação pelo número de saltos.

Como demonstração, após a aplicação da equação 1.2 pelo nodo c obteve como resultado para fins de decisão $\sigma(p')= 80$ e $\sigma(p)=33,33$, ou seja, com base na aplicação da equação a mensagem m_c é atraída para p' , ou seja, é realizada uma entrega direta como pode ser visto na (Figura 39).

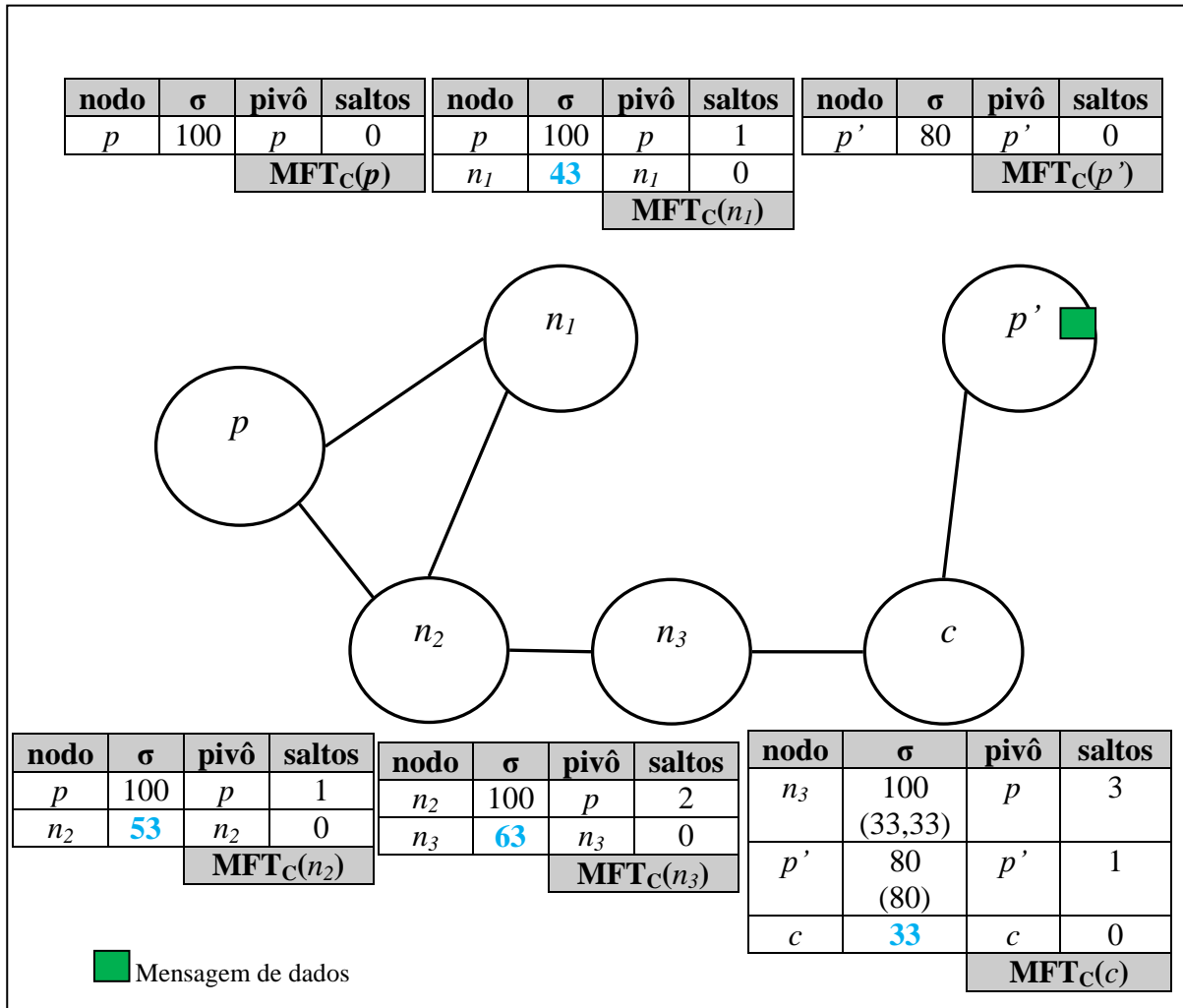


Figura 39 – Mensagem entregue a p' em virtude da atenuação da força de p

Algum tempo depois, os nodos p e p' desconectam-se da sub-rede, ou seja, a partir desse momento não existem nodos em P_C . Conforme citado anteriormente, caso não existam nodos em P_C , cada nodo em NP_C verifica a existência de nodos adjacentes pertencentes a NP_C que possuam a força de atração superior a sua própria força. Caso existam, n seleciona o que possuir a maior força e o adiciona à MFT_C . Em seguida n_1 gera uma nova mensagem de dados m_C como pode ser visto na (Figura 40).

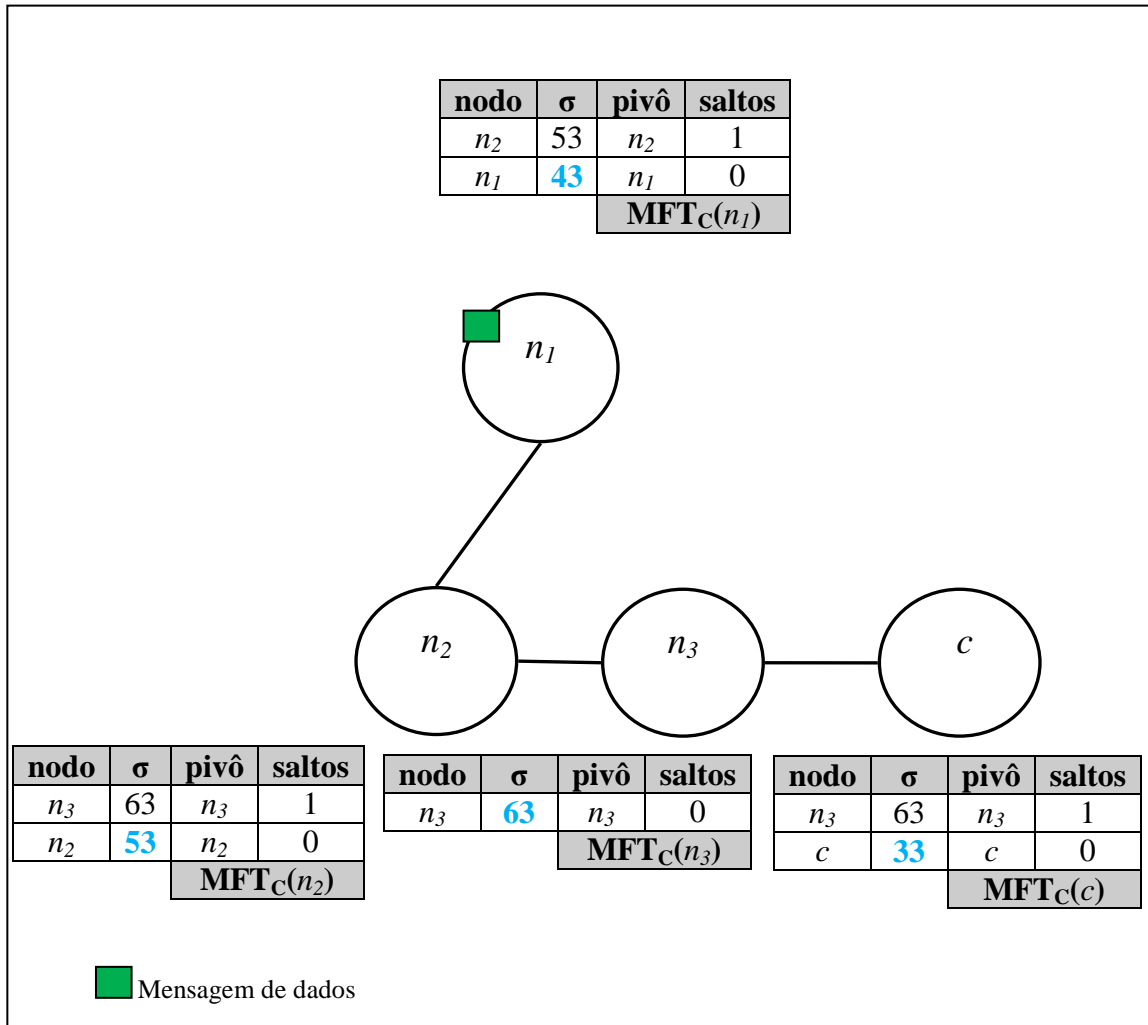


Figura 40 – Sub-rede conexa sem nenhum nodo em P_c

Portanto, a mensagem m_c é direcionada ao nodo n_2 , pois é o nodo com maior força que exerce influência direta sobre o nodo n_1 . Ao chegar ao nodo n_2 , a mensagem m_c é direcionada ao nodo n_3 , pois é o nodo com maior força que exerce influência direta sobre o nodo n_3 . Ao chegar ao nodo n_3 , o mesmo manterá a custódia da mensagem até um nodo com maior força atrair a mensagem para si. Como pode ser visto na (Figura 41).

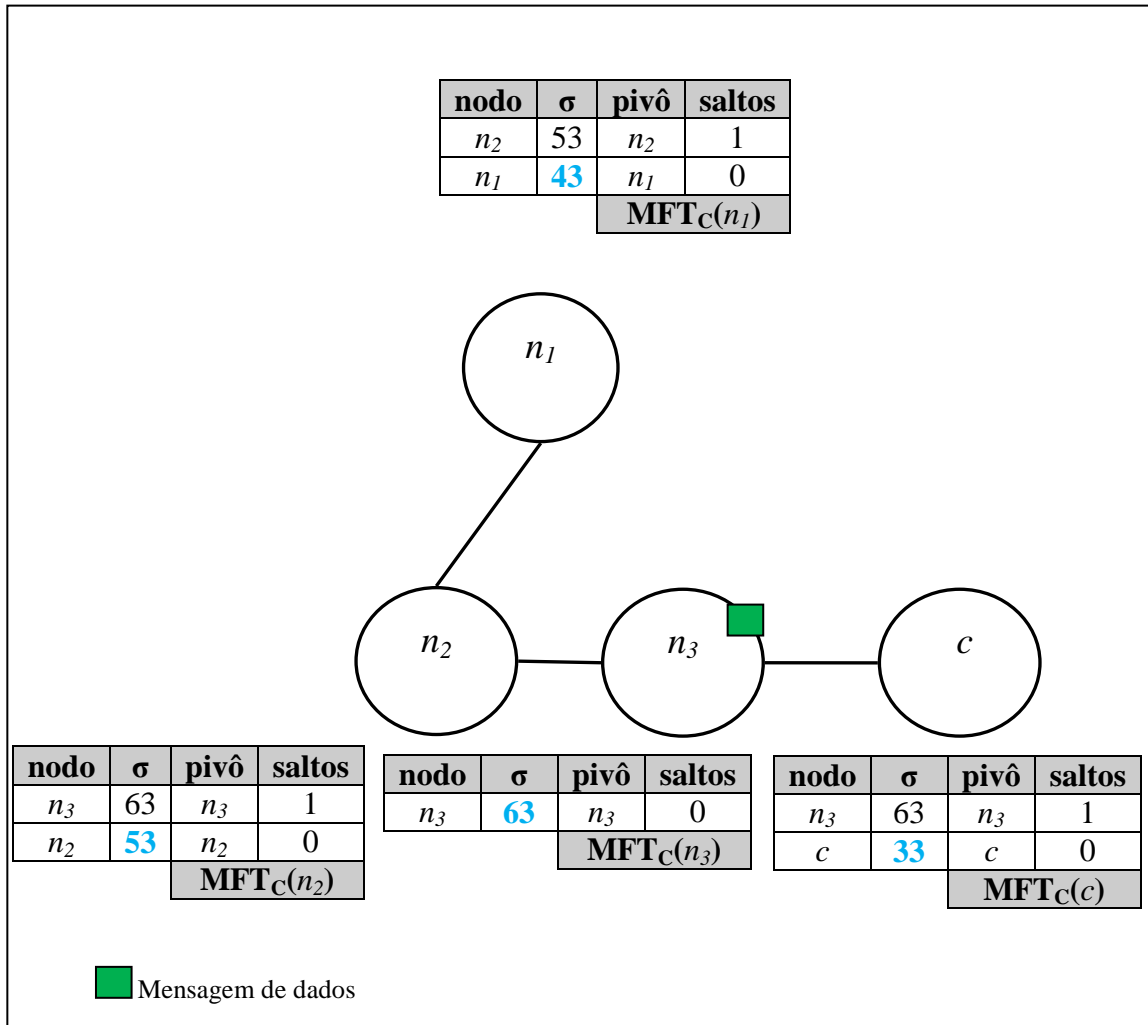


Figura 41 – Roteamento de mensagens em sub-rede sem nenhum nodo em P_c

4.5. Balanceamento de Carga

Como foi visto até o momento, percebemos que a utilização do *heartbeat* combinado com a propagação de força, fornece um caminho até o nodo capaz de processar a mensagem mais forte da sub-rede, salvo na entrega direta, utilizada principalmente em cenários onde existe uma mobilidade constante. A entrega direta visa equilibrar a distribuição de mensagens.

No entanto, em cenários com pouca mobilidade, mas com alta conectividade, os nodos mais fortes tendem a concentrar o recebimento das mensagens. Todavia, as mudanças no conceito de algoritmos magnéticos virtuais para DTNs trouxeram de forma implícita o balanceamento de carga, permitindo de forma flexível o critério de decaimento da força de

atração de um nodo em P_C que está concentrando o recebimento de mensagens, fazendo com que o fluxo de mensagens seja redirecionando para outro nodo em P_C que possua a maior força de atração. Em outras palavras, a força de atração de um nó em P_C pode ser determinada como sendo inversamente proporcional à sua sobrecarga em termos de uso de CPU, memória, capacidade de armazenamento, etc.

Como visto nas seções anteriores, o nodo pertencente a P_C envia mensagens *heartbeat* para os nodos na sub-redes com o intuito de informar a sua existência e ao mesmo tempo encaminhar o valor da sua força de atração, permitindo ao nodo em NP_C inferir qual nodo em P_C possui a maior força de atração naquele momento. Portanto, durante a execução, o nodo em P_C pode se sobrecarregar com o recebimento de mensagens, seja por baixa capacidade de processamento gerada por limitações de processador, memória, espaço de armazenamento, dentre outros, ou pela limitação de energia do dispositivo, característica essa inerente aos dispositivos móveis que são um dos principais meios de comunicação em DTNs. Nesta circunstância, o nodo em P_C pode reduzir sua força de atração até o ponto em que ela fique abaixo a de outro nodo em P_C existente, tornando o outro nodo em P_C o destinatário atual de mensagens, ou seja, fazendo balanceamento de carga. Uma vez superadas as limitações, o nodo em P_C pode recuperar sua força e voltar a atrair as mensagens para si mesmo.

Para entendermos melhor o funcionamento deste mecanismo, consideremos um cenário com uma sub-rede conexa composta de 5 nodos (i.e., $V_i = \{p, p', n_1, n_2, n_3\}$). Além disso:

- somente os nodos p e p' processam mensagens do tipo C;
- os nodos n_1, n_2 e n_3 não processam mensagens do tipo C.

Inicialmente, existem cinco nodos conectados (embora os nodos p e p' sejam os únicos aptos a receberem e processarem mensagens da classe C). Contudo, o nodo n_2 está com a custódia de quatro mensagens m_c (Figura 42).

Conforme descrito na seção 4.4, uma tabela magnética em um nodo em NP_C presente em uma sub-rede conexa contendo pelo menos um nodo em P_C conterà no máximo três entradas que são: a entrada correspondente à própria força, a entrada correspondente ao nodo que possua a $Máx\{\sigma(p), \forall p \in V_i \cap P_C\}$ e caso o mesmo não seja adjacente, poderá também existir a entrada correspondente ao nodo em P_C adjacente que possua a maior força.

Portanto, como pode ser observado na (Figura 42), o nodo n_2 sofre influência indireta do nodo p .

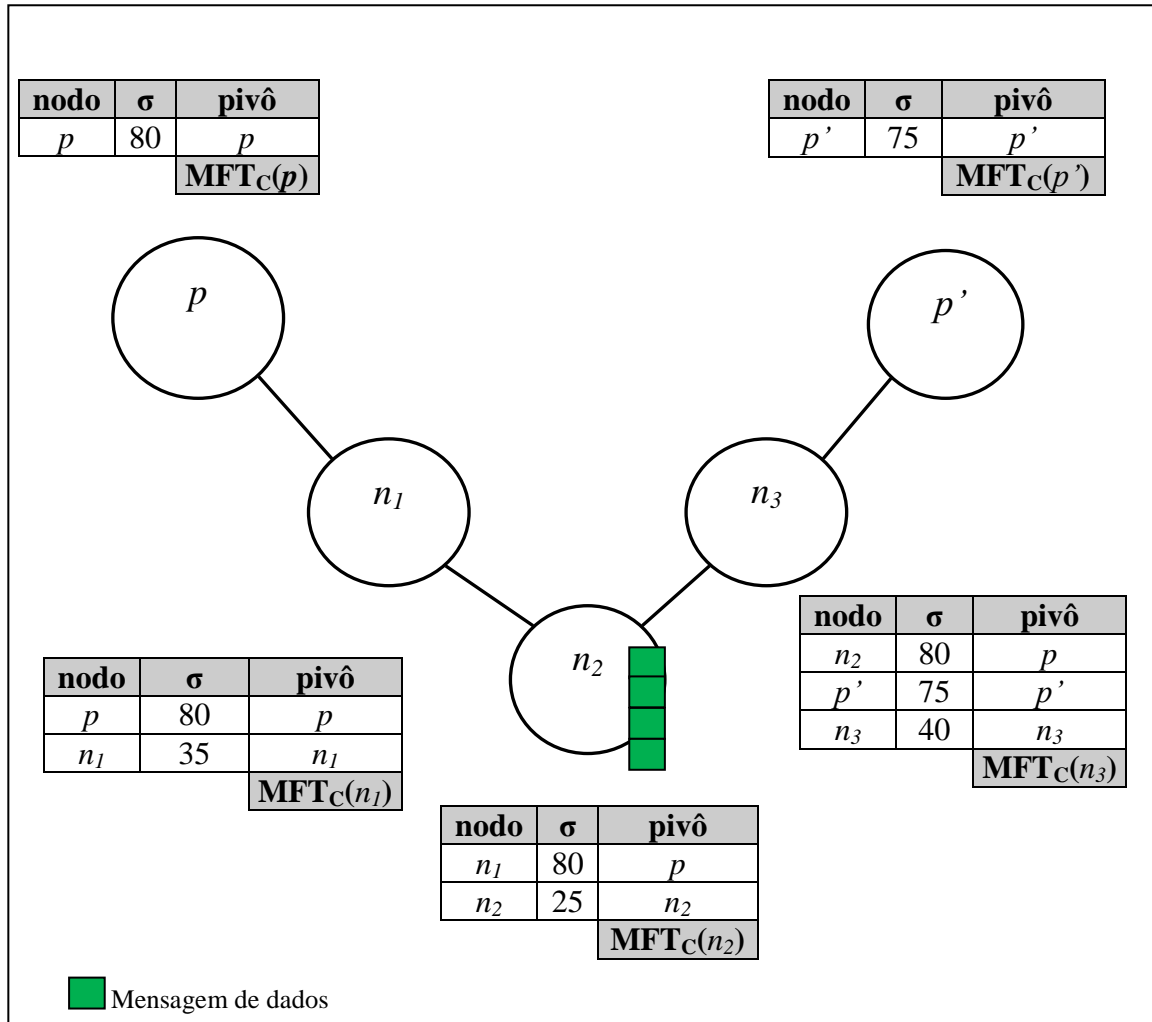


Figura 42 – Todos os nodos da sub-rede conexa

Com base nas influências descritas acima, uma das mensagens m_c é direcionada ao nodo n_1 , pois o nodo com maior força que exerce influência (indireta) sobre o nodo n_2 é o nodo p como pode ser visto na (Figura 43). Ao chegar ao nodo n_1 , a mensagem m_c será finalmente atraída para o nodo p , pois é o nodo com maior força. O nodo p , finalmente irá iniciar o processamento da mensagem m_c .

Ao longo do tempo, como a sub-rede contém nodos em P_C (i.e., $V_i \in P_C \setminus \mathcal{A}$), a força de atração dos nós $n \in V_i \in NP_C$ aumenta, como pode ser visto na Figura 43.

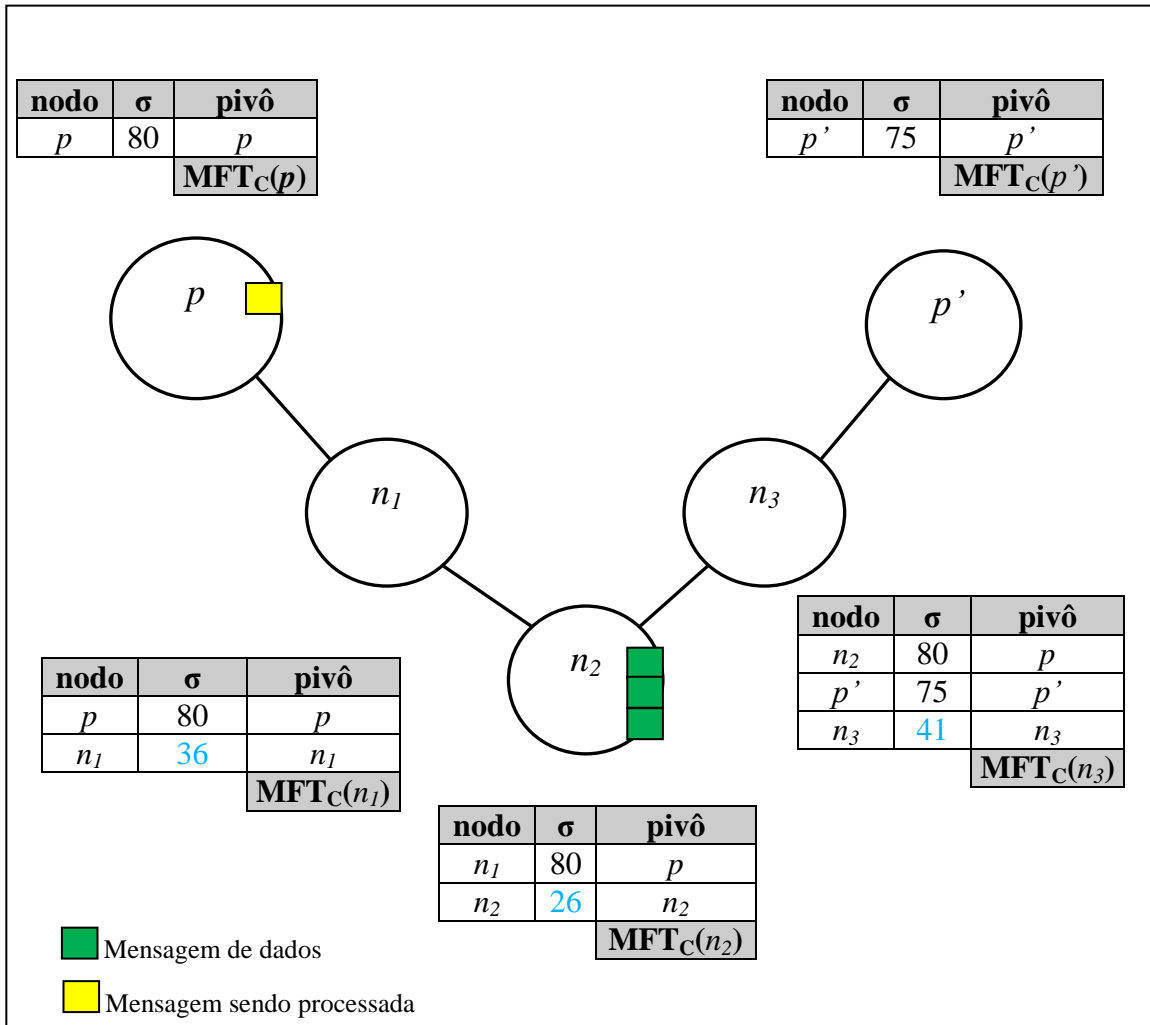


Figura 43 – Mensagem atraída e sendo processada para o nodo em P_C mais forte

Contudo, antes que as demais mensagens fossem atraídas ao nodo p , ocorreu uma limitação interna do nodo p que exigiu que o mesmo diminuísse sua força com o intuito de reduzir a quantidade de mensagens atraídas para si, alterando sua força de 80 para 74, ou seja, ficando abaixo da força de atração do nodo p' . Conseqüentemente o nodo p' cessa o repasse das mensagens *heartbeat* enviadas pelo nodo p e começa a enviar as suas próprias. Em consequência, as forças dos nodos em NP_C são reajustadas proporcionalmente ao novo limite definido na sub-rede, como pode ser visto na (Figura 44).

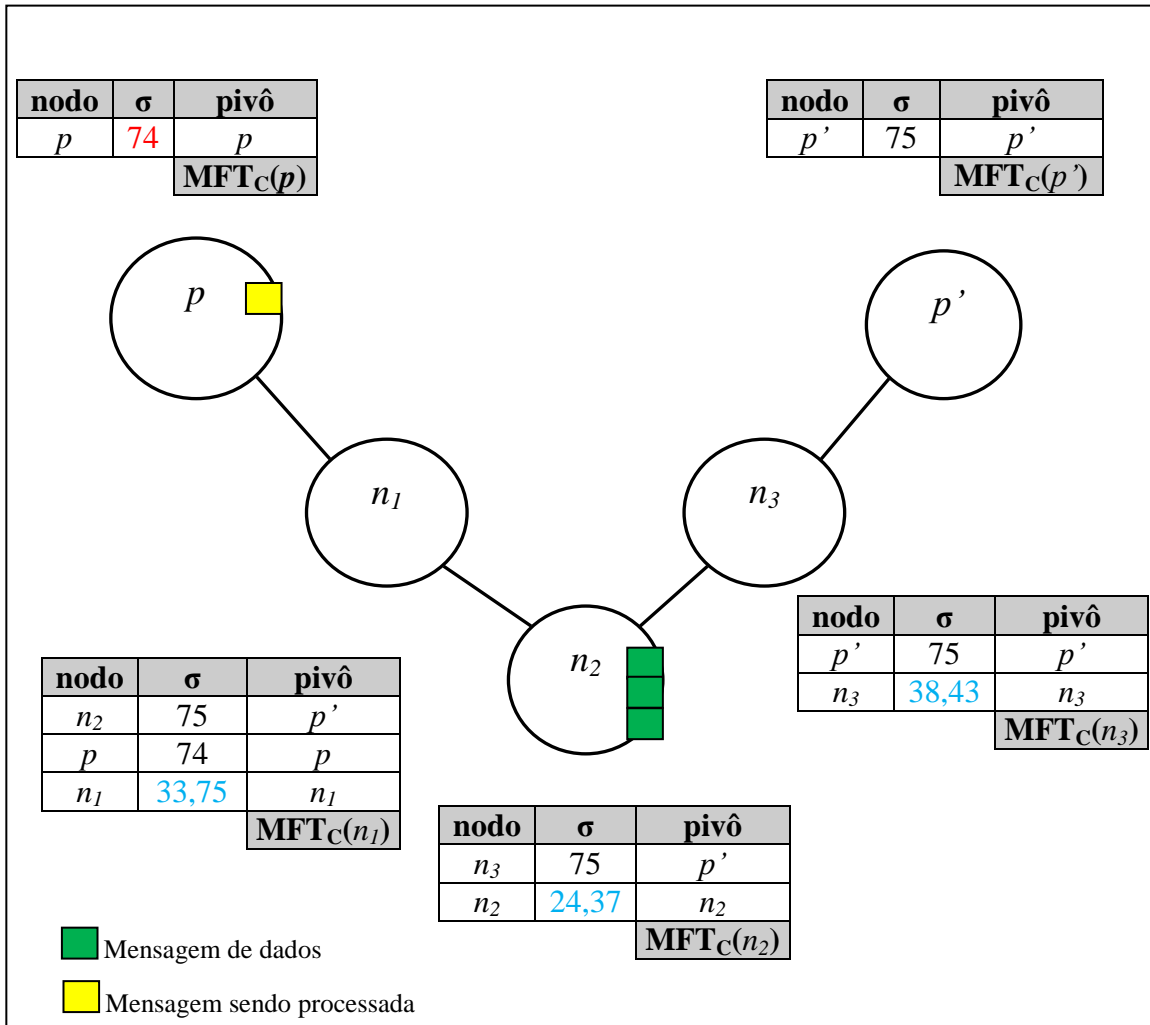


Figura 44 – Nodo em P_C mais forte reduz a força atração

Com base nas novas influências, uma das mensagens m_c é direcionada ao nodo n_3 , pois o nodo com maior força que exerce influência (indireta) sobre o nodo n_2 é o nodo p' (Figura 45). Ao chegar ao nodo n_3 , a mensagem m_c será finalmente atraída para o nodo p' , pois é o nodo com maior força. O nodo p' , finalmente irá iniciar o processamento da mensagem m_c . Portanto, como pode ser observado na (Figura 45), foi realizado o balanceamento de carga em relação às duas primeiras mensagens entregues a nodos pertencentes a P_C distintos.

Como descrito no início dessa seção, o decremento da força de um nodo em P_C pode ser implementada para as diversas situações como, por exemplo, quantidades de mensagens que estão sendo processadas, limitações de *hardware*, limitações de banda larga, etc. Ou seja, o intuito é que exista uma flexibilidade de um nodo que está sobrecarregado possa ajustar sua

força e por consequência realizar o balanceamento de mensagens de forma dinâmica. Um método detalhado de avaliação da sobrecarga de um nó e do impacto na sua força de atração, no entanto, é considerado fora do escopo deste trabalho.

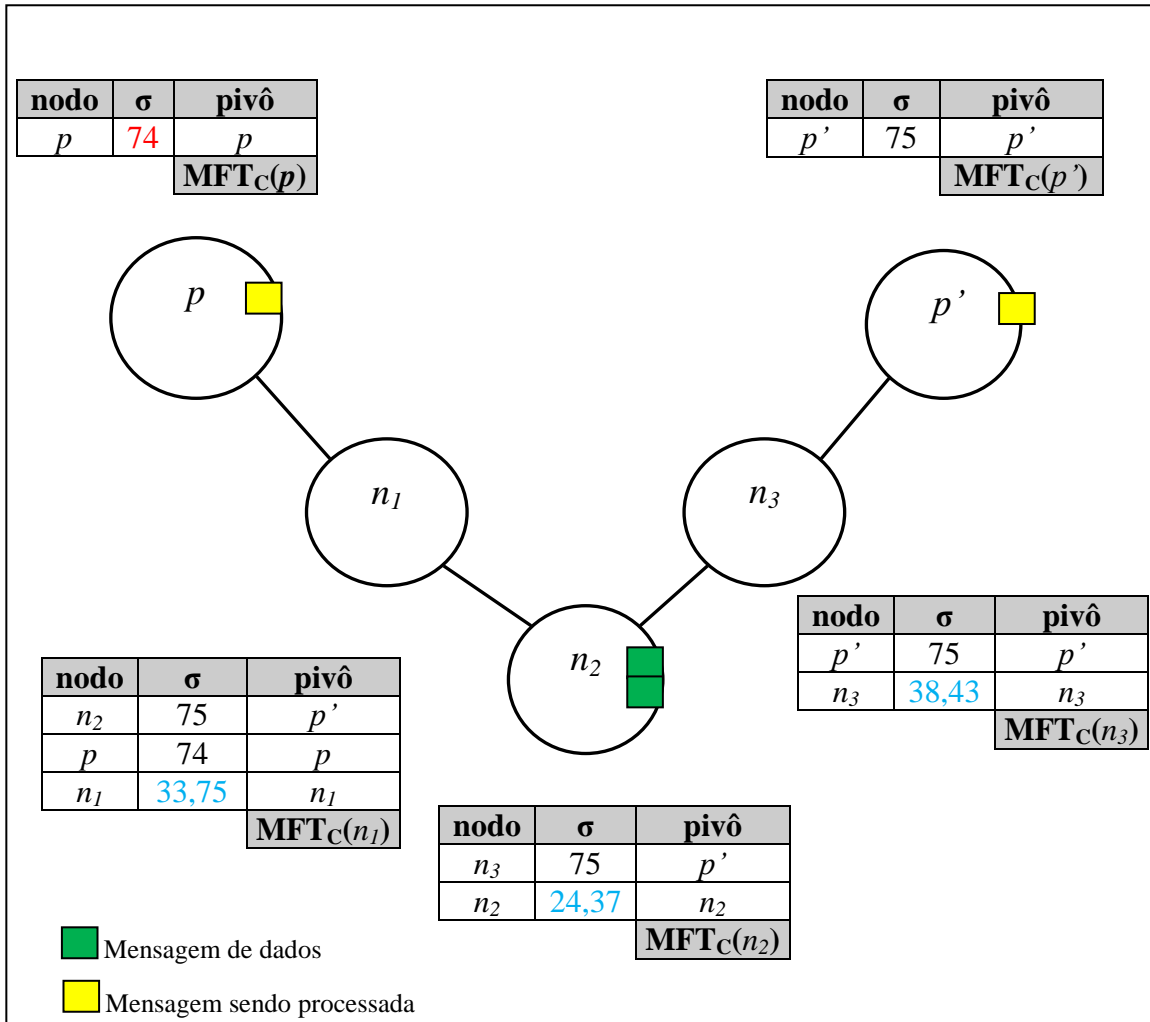


Figura 45 – Balanceamento de carga

4.6. Controle de Réplicas

Alguns algoritmos encontrados na literatura como *Epidemic*, *PRoPHET* e *PRoPHET V2*, não realizam controle de réplicas durante o processo de roteamento das mensagens. No entanto, essa falta de controle tem sido apontada como uma das principais causas relacionadas ao descarte de mensagens por *overflow*. Contudo, algoritmos como o *Spray and Wait*, por

exemplo, possuem mecanismos de controle de réplicas com o objetivo de se utilizar de maneira mais eficiente os recursos disponíveis.

No método proposto, foi desenvolvido um mecanismo de controle de réplicas de mensagens que permite controlar a quantidade máxima de cópias disponíveis na NG_t evitando a inundação da rede com mensagens replicadas.

O controle das réplicas fica sobre a responsabilidade do nodo que gerou a mensagem e somente o mesmo pode realizar a criação das cópias. Portanto, no momento da sua geração, o nodo responsável associa um contador contendo o número máximo de cópias permitidas. Durante o repasse de uma mensagem, o nodo responsável verifica se o contador criado para aquela mensagem é maior que 1, e caso seja, o mesmo repassa uma cópia da mensagem e decrementa o contador. Caso o contador possua o valor 1, ou seja, chegou no limite máximo permitido, o mesmo faz o repasse da mensagem e a elimina do *buffer*. Contudo, em caso de uma entrega direta a mensagem é entregue e excluída do *buffer* do nodo de origem independente do valor do contador associado àquela mensagem.

No contexto de *anycast* em DTNs, ou seja, quando existem múltiplos destinos possíveis se torna inevitável evitar a entrega duplicada de mensagens a mais de um nodo apto a processar um determinado tipo de mensagem.

As constantes desconexões não permitem que os nodos notifiquem em tempo hábil os demais nodos sobre o recebimento de uma mensagem ou a implementação de um mecanismo de confirmação de recebimento (*ack*) confiável.

4.7. Conclusão

A adaptação do algoritmo de campos magnéticos virtuais, apresentado neste capítulo, se baseia na força de atração exercida pelos nodos com o intuito de trazer para si mensagens de um determinado tipo. A cada nodo é atribuído um peso que representa a força de atração. Quanto mais forte essa atração, mais chance há de que uma mensagem seja atraída para esse nodo. As atrações ocorrem de forma recursiva na rede até chegar ao(s) nó(s) com maior(es) força(s).

Algumas adaptações permitiram a utilização do algoritmo proposto em DTNs *anycast* e *unicast* ambas do tipo estocástica. O algoritmo de notificação denominado *heartbeat*, mais as definições de propagação da força de atração, forneceram ao modelo proposto um

roteamento dinâmico capaz de adaptar a mobilidade dos nodos inerentes as DTNs. Tais mudanças foram essenciais para o conceito de balanceamento de carga, que ocorre de forma dinâmica em DTNs, sendo esta uma característica não encontrada em outros algoritmos da literatura.

Conforme descrito na seção 4.4, o modelo é flexível o bastante para permitir estendê-lo, sendo esse um dos trabalhos futuros a serem aprofundados.

A seguir serão apresentados os resultados de uma avaliação que corrobora a eficiência do modelo proposto.

Capítulo 5 – Ambiente de Simulação e Resultados Obtidos

As seções deste capítulo encontram-se estruturadas na seguinte ordem: a seção 5.1 apresenta o ambiente de simulação, ferramentas utilizadas, parâmetros e métricas utilizadas para avaliação; a seção 5.2 apresenta uma análise dos resultados referentes à utilização do algoritmo proposto para DTNs *anycast* estocásticas; a seção 5.3 apresenta uma análise comparativa dos resultados com outros algoritmos clássicos da literatura; a seção 5.4 apresenta, uma análise da aplicação de *throwboxes* comparando os resultados com outros algoritmos da literatura. e, por fim, a seção 5.5 apresenta as considerações finais.

5.1. Ambiente de Simulação

O simulador adotado para este trabalho foi o *The ONE (The Opportunistic Network Environment simulator)* [34] que é um simulador desenvolvido na linguagem JAVA [35] que foi concebido com o foco em DTNs estocásticas. Ele tem sido cada vez mais utilizado no meio acadêmico para realizar simulações em DTN, devido aos recursos disponíveis como, por exemplo, modelos de mobilidade, mapas, algoritmos de roteamento clássicos previamente implementados e a possibilidade da geração de resultados das principais métricas adotadas na literatura para a construção de tabela e gráficos mais elaborados como pode ser observado nas Figuras 46 e 47.

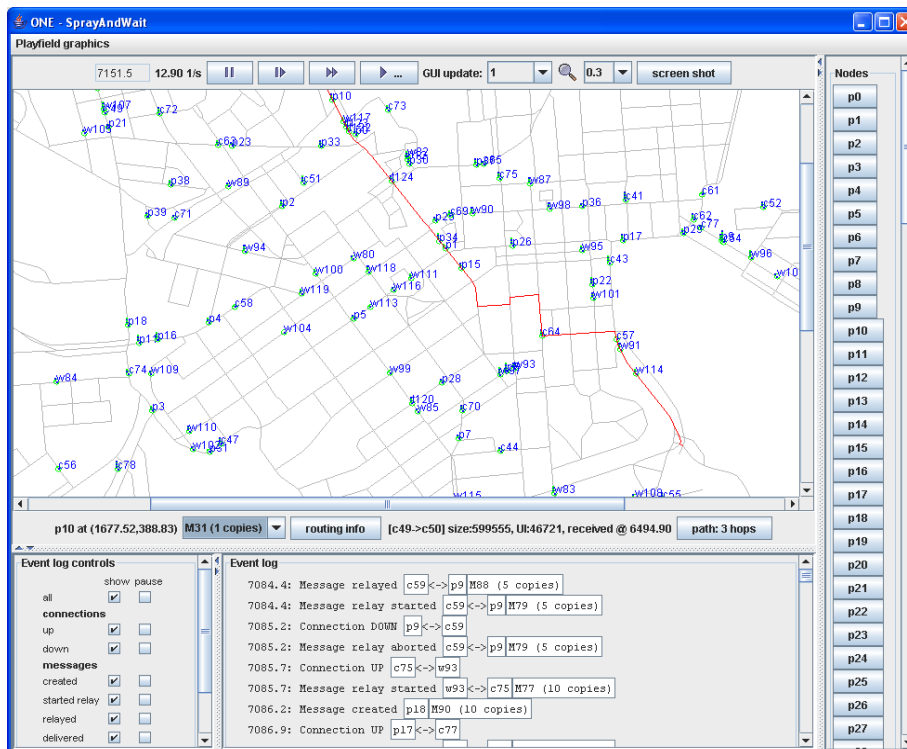


Figura 46 - Simulador executando com 100 nodos, utilizando o algoritmo *Spray and Wait* e modelo *map-based movement* [34]

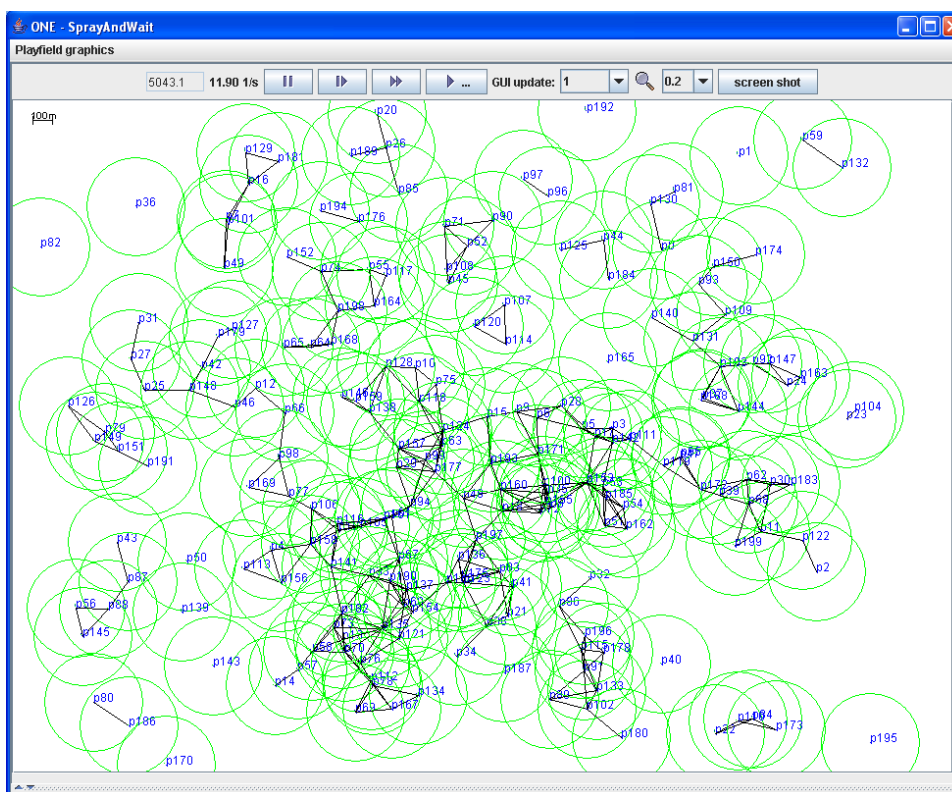


Figura 47 - Modelo *Random waypoint* com 200 nodos com raio de alcance de 250m [34]

5.1.1. Cenário da Simulação

Para definição do cenário, foram utilizados recursos existentes no simulador (ressaltando que não existe um cenário de consenso na literatura devido ao fato de que cada cenário apresenta características próprias e tem tipicamente objetivos distintos). Os parâmetros de simulação considerados foram definidos baseadas nas características do cenário investigado em combinação com características de trabalhos existentes na literatura [36][43][44][45].

- Área da simulação: Uma área retangular de 4500 x 3400 metros. O mapa que acompanha o simulador é uma representação da parte central da cidade de *Helsinki*, na Finlândia (esta escolha, no entanto, só foi feita por razões práticas, uma vez que o próprio método proposto é obviamente geral);
- Quantidade de nodos: 300;
- Tipos de nodos: Os nodos foram divididos em partes iguais entre pessoas com dispositivos móveis e veículos;
- Velocidade dos nodos: variando entre 0,5 m/s – 1,0 m/s para as pessoas e 2,7 m/s – 13,9 m/s para os automóveis;
- Capacidade de armazenamento: os veículos têm um *buffer* de 40MB e as pessoas, de 20MB;
- Taxa de transmissão entre pessoas e veículos de 2 Mbps;
- Modelo de mobilidade: *Shortest Path Map-Based Movement* [37], por se tratar de um modelo realístico que se baseia na utilização de informações extraídas de um mapa como os obstáculos para traçar o caminho mais curto, o que permite a flexibilidade de se definir grupos de nodos;
- Raio de alcance dos nós: o mesmo para todos os nós em cada simulação, mas variando entre 80, 50 e 30 metros em simulações diferentes.

Baseado em um estudo realizado em [36], um número de mensagens sugeridas para as simulações são de 5000 mensagens criadas durante a simulação em origens aleatórias e para pontos de destinos aleatórios. Esse mesmo estudo constatou que para os experimentos são necessários 55 horas simuladas no simulador para tornar o resultado da simulação o mais realista possível, sendo realizadas 7 rodadas de simulação para cada cenário.

5.1.2. Parametrização

O algoritmo proposto precisa ter configurados dois parâmetros: o intervalo de tempo em segundos para envio *heartbeat* e o intervalo utilizado para o aumento e para a diminuição da força atração. O intervalo de tempo foi considerado no formato de segundos por questões de padronização no uso do simulador não ficando o modelo restrito ao mesmo.

Para cada intervalo de tempo simulado foram realizadas 55 horas simuladas no simulador, sendo realizadas 7 rodadas de simulação para cada intervalo, e por fim comparando as médias dos intervalos simulados.

- Intervalo de tempo do *heartbeat*: foi observado que o intervalo de tempo de 5 segundos foi considerado suficiente para caracterizar o tempo que um nodo encontra-se no destino durante a atração de uma mensagem. Não se observou alterações significativas na quantidade de transmissões abortadas e nas taxas de entregas quando utilizado intervalos de 1 a 5 segundos. Sendo adotado o intervalo de 5 segundos em virtude de uma menor quantidade de mensagens *heartbeat* geradas nesse intervalo. Todavia, notou-se um crescimento gradual no número de transmissões de mensagens de dados abortadas quando simuladas com 6, 7, 8, 9 e 10 a medida que eram utilizados intervalos de tempo maiores como visto na tabela 2;
- Intervalo de tempo do aumento e diminuição da força: o intervalo de tempo para o aumento e diminuição da força. Conforme descrito no Algoritmo 1 (Heartbeat), o tempo de atualização da força deve ser maior que o tempo utilizado para gerar as mensagens *heartbeat*. Portanto, foram avaliados os intervalos 6, 7, 8, 9 e 10 segundos, onde o tempo de 6 segundos foi o que apresentou melhores taxas de entrega, sendo que os intervalos superiores a 6 apresentaram uma diminuição gradativa nas taxas de entrega e um aumento no número de transmissões de mensagens de dados, ocasionando assim o uso desnecessário da rede como visto na tabela 3. A cada atualização da força, foi utilizado 1 unidade de força para aumentar ou diminuir a força de atração.

Tabela 2: Mensagens abortadas e entregues em relação à HB

Intervalo de tempo HB	Mensagens abortadas	Percentual de mensagens entregues
1	3780	96,42%
2	3784	96,45%
3	3790	96,51%
4	3795	96,53%
5	3799	96,57%
6	5810	93,56%
7	6535	92,74%
8	7380	90,47%
9	9342	90,23%
10	11345	89,16%

Tabela 3: Transmissões de mensagens e entregues em relação à força de atração

Intervalo de tempo HB	Mensagens transmitidas	Percentual de mensagens entregues
6	18457	96,57%
7	20364	94,85%
8	22958	93,49%
9	23864	92,45%
10	25973	92,01%

5.2. Análise dos resultados com DTN *anycast*

Como esse projeto propõe a adaptação do conceito de campos magnéticos virtuais para DTNs *anycast*, foram realizadas simulações que avaliassem a aplicação do modelo proposto neste cenário. Para avaliarmos a possibilidade de múltiplos destinatários em potencial, foram realizadas simulações em cenários variando a quantidade de nodos em P_C . Foram realizadas simulações em cenários com P_C contendo 30, 6 e 1 nodos, representando, respectivamente 10%, 2% e 0,33%, dos nodos existentes. Em todos os cenários, foram analisados raios de alcance de 80, 50 e 30 metros. Nesse contexto, para melhor avaliar a eficiência do algoritmo, o controle de réplicas do algoritmo foi configurado para que garantisse a existência de somente uma única versão da mensagem na rede.

A primeira métrica analisada foi o percentual de mensagens entregues. A Figura 48 mostra o percentual de mensagens entregues comparando os cenários em que o raio de alcance dos nodos foram de 80, 50 e 30 metros, e onde 10% dos nodos eram capazes de processar uma mensagem. Percebeu-se que, mesmo com a faixa de 30m (o que implica baixa conectividade entre nodos), altas taxas de entrega de mensagens foram obtidas. No final da simulação, o percentual de entrega para os raios de alcance 80, 50 e 30 foram, respectivamente, 99,80%, 99,74% e 99,57%.

À medida que o tempo de simulação aumenta, a taxa de entrega aumenta em virtude do aumento no número de contatos entre nodos NP_C com P_C fazendo com que as mensagens sejam atraídas para um nodo pertencente a P_C .

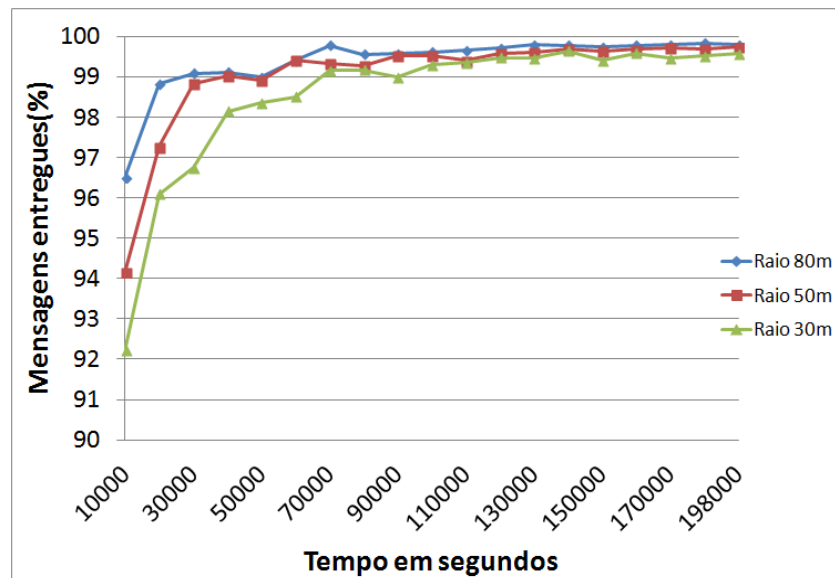


Figura 48 - Comparação com 30 nodos em P_C com apenas uma cópia da mensagem

A (Figura 49) mostra o percentual de mensagens entregues comparando cenários com raios de alcance de 80, 50 e 30 metros, onde apenas 2% dos nodos são capazes de processar uma mensagem. Diferentemente do cenário anterior, observa-se uma queda no desempenho no início da simulação em relação ao cenário anterior. Com a diminuição do número de nodos capazes de processar uma mensagem, o nodo leva mais tempo para se conectar diretamente ou indiretamente com um possível destino. No final da simulação, o percentual de entregas para os raios de alcance 80, 50 e 30 metros foram, respectivamente, 99,27%, 99,01% e 98,38%.

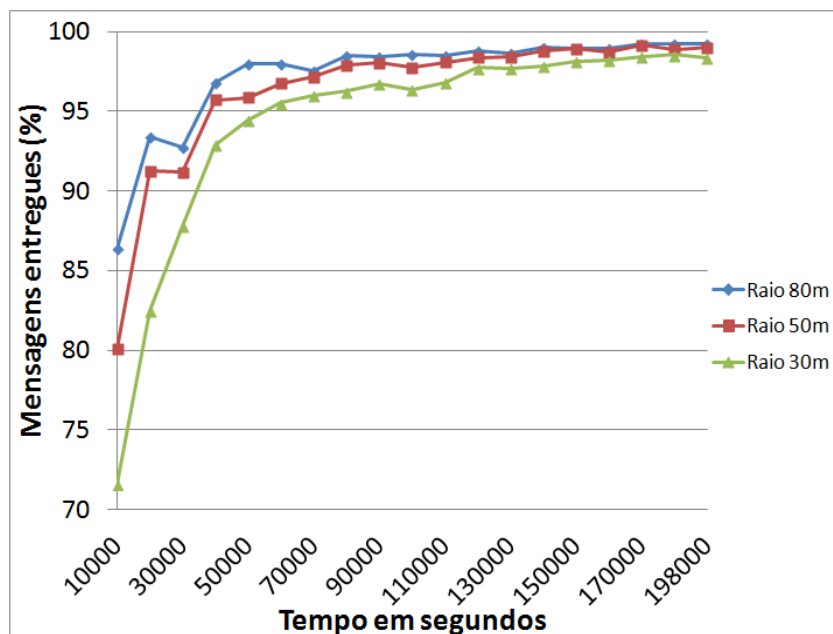


Figura 49 - Comparação com 6 nodos em P_C com apenas uma cópia da mensagem

A (Figura 50) mostra o percentual de mensagens entregues comparando cenários com raios de alcance 80, 50 e 30 metros, onde 0,33% dos nodos (ou seja, apenas um nodo) é capaz de processar uma mensagem. Seguindo a tendência de o cenário anterior, uma redução na taxa de entrega das mensagens ocorreu nos início da simulação, quando comparado com as simulações mostradas nas Figuras 48 e 49. A razão principal para esta diminuição do desempenho é por ser equivalente a *unicast* onde uma mensagem apenas pode ser entregue a um único destino específico, tendo como consequência um maior atraso na ocorrência de contatos entre os nodos em NP_C com o único nodo P_C existente. No entanto, o longo tempo de simulação e consequentemente o aumento no número de contatos, contribuíram para o resultado no final da simulação foram 96,57%, 94,96% e 86,54% percentuais de entregas com raios de alcance, respectivamente, 80, 50 e 30 metros.

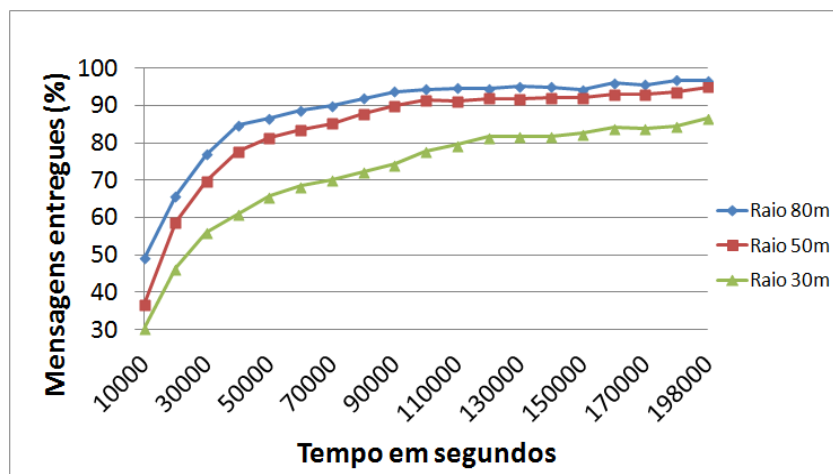


Figura 50 - Comparação com 1 nodo em P_C com apenas uma cópia da mensagem

O impacto causado pelo decréscimo no número de nodos em P_C , pode ser melhor observado quando os percentuais de entrega são comparados utilizando o mesmo raio de alcance com diferentes quantidades de nodos em P_C , como mostrado nas Figuras 51, 52 e 53. Nestes cenários foram utilizados raios de alcance respectivamente de 80, 50 e 30 metros, a fim de demonstrar o impacto em redes. Observa-se que, isto ocorre por causa da redução no número de nodos em P_C , que tem como consequência um número menor de contatos entre nodos P_C e NP_C , o que é agravado quando o raio de alcance diminui como pode ser visto na (Figura 53).

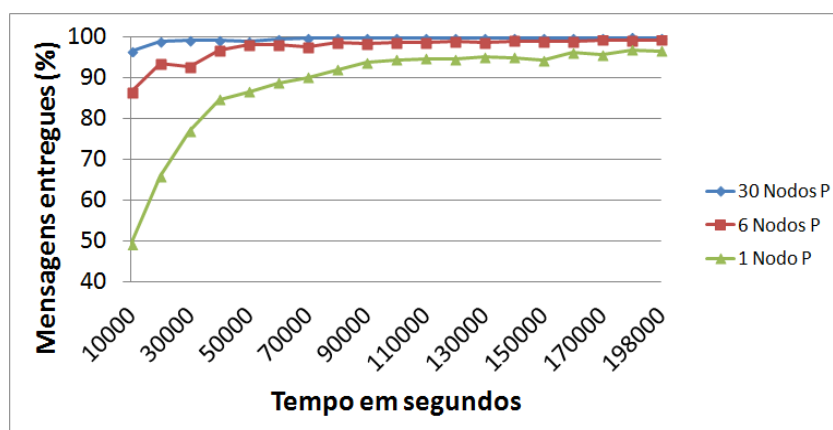


Figura 51 - Comparando com diferentes quantidades de nodos em P_C com alcance de 80 metros

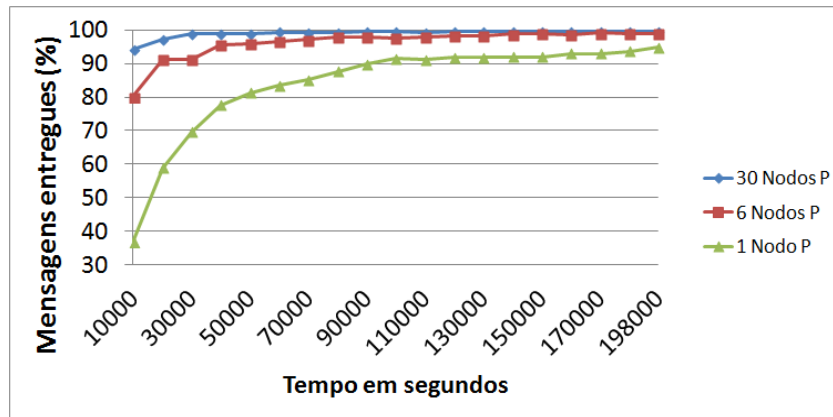


Figura 52 - Comparando com diferentes quantidades de nodos em P_C com alcance de 50 metros

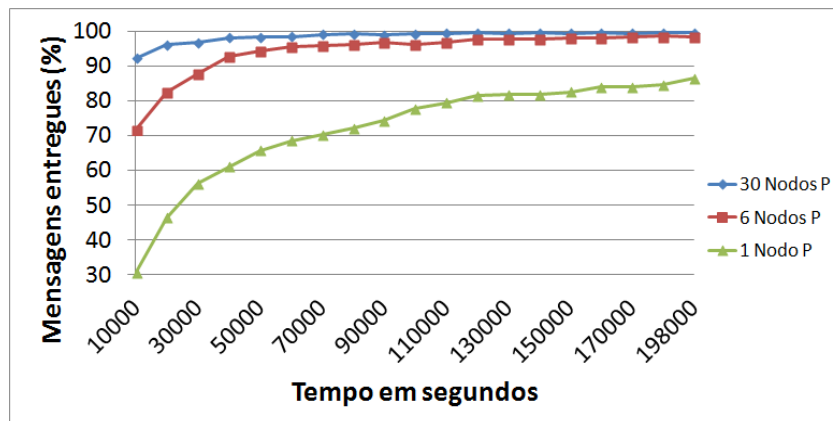


Figura 53 - Comparando com diferentes quantidades de nodos em P_C com alcance de 30 metros

Os atrasos médios das entregas de mensagens nos cenários das Figuras 48, 49 e 50 são mostrados na Tabela 4. O atraso médio é o tempo decorrido desde o momento em que a mensagem foi gerada ao momento em que é recebida em algum destino.

Observe que um pequeno número de nodos em P_C e a baixa conectividade causada por um baixo raio de alcance afetam negativamente os percentuais de entrega das mensagens.

Tabela 4: Atraso médio na entrega das mensagens

Número de nodos em P_C	Atraso médio (segundos)		
	Raio 80m	Raio 50m	Raio 30m
30	344	498	745
6	1411	1944	3000
1	6247	9479	18634

Em relação ao descarte de mensagens devido ao *buffer overflow*, não ocorreram com a utilização do algoritmo CMV como pode ser visto na Tabela 5.

Contudo, para que fosse possível alcançar essa eficiência na utilização dos *buffers*, dois fatores foram considerados fundamentais que são:

- O mecanismo de controle de réplicas: que tem como objetivo evitar o *flooding* de mensagens na rede;
- O conceito da atração do CMV: que possui como característica a atração (salto) de mensagens para o nodo mais forte, que por consequência diminui o tempo que uma mensagem permanece no *buffer* de um mesmo nodo como pode ser constatado nas Tabelas 6 e 7.

Tabela 5: Mensagens descartadas devido ao *buffer overflow*

Número de nodos em P_C	<i>Buffer Overflow</i>		
	Raio 80m	Raio 50m	Raio 30m
30	0	0	0
6	0	0	0
1	0	0	0

Tabela 6: Média de Saltos por Mensagem

Número de nodos em P_C	Número de saltos (<i>hops</i>)		
	Raio 80m	Raio 50m	Raio 30m
30	4,74	2,51	1,73
6	3,88	2,33	1,68
1	3,69	2,11	1,51

Tabela 7: Tempo Médio de uma Mensagem nos *Buffers*(segundos)

Número de nodos em P_C	Tempo Médio (segundos)		
	Raio 80m	Raio 50m	Raio 30m
30	88	213	441
6	297	772	1730
1	1693	4501	12503

Como descrito na seção 4.3, o modelo proposto possui um algoritmo de notificação utilizado pelos nodos em P_C para informar sua existência em uma sub-rede: o *heartbeat*. Com a finalidade de quantificar o tráfego gerado pelas mensagens *heartbeat* e com o intuito de verificar seu impacto no desempenho da rede, várias medições foram feitas. A média de todos os dados trafegados por nodo (P_C e NP_C) durante toda a simulação são apresentados na Tabela 8 e, com base na eficiência nos resultados dos percentuais de entrega obtidos, podemos considerar, pelo tempo da simulação e pelo número de bytes trafegados, os valores obtidos aceitáveis.

Tabela 8: Tráfego Médio em *Kilobyte* por Nodo com Transmissões *Heartbeat*

Número de nodos em P_C	Transmissões Médias <i>Heartbeat</i>		
	Raio 80(KB)	Raio 50(KB)	Raio 30(KB)
30	299,43	109,62	50,78
6	62,01	19,99	8,73
1	10,76	3,37	1,43

5.3. Comparativo com outros algoritmos da literatura

Com o objetivo de comparar CMV com os principais protocolos de roteamento DTN na literatura, definimos um cenário onde havia apenas um único nodo em P_C . Os algoritmos utilizados nas simulações para fins de comparação foram: *Epidemic*, *Prophet*, *Prophet V2* e *SNW (Spray and Wait)*. O *Prophet*, *Prophet V2* e *Spray and Wait* foram selecionados, pois possuem restrições da estratégia de *flooding*, assim como o CMV. Sendo que o *Epidemic* foi escolhido por ser juntamente com o *Prophet* e *Spray and Wait* os mais referenciados na literatura. CMV e *SNW* também foram comparados e configurados com controle de réplicas para 1 e 2 cópias. Na simulação do algoritmo de campos magnéticos, foi definida a existência

de somente um (*unicast*) nodo em P_C para fins de comparação igualitária com os demais algoritmos. No entanto, vale ressaltar que não foi possível a realização de simulações com trabalhos no contexto de DTN *anycast* estocástica devido ao fato de que os poucos trabalhos encontrados na literatura abordarem DTN *anycast* para redes *determinísticas*, a exemplo de [31] [39], sendo assim incompatíveis com o foco deste trabalho. Em DTN *anycast* do tipo estocástica descrito em [40], concentra os seus esforços em propostas de modelos para formação de grupos, definição de receptores e métricas, onde as informações relacionadas à roteamento são limitadas para fins de implementação e uma comparação mais fidedigna.

Os resultados apresentados nas Figuras 54, 55 e 56 mostram que as taxas de entrega de mensagens dos protocolos *Epidemic* e *Prophet* são semelhantes em todos os cenários simulados. O *Prophet V2*, que é uma variante do protocolo *Prophet*, tem um desempenho melhor do que os outros dois algoritmos, mas todos eles apresentam uma diminuição no percentual de entregas de mensagens como o decorrer da simulação. Isto acontece porque existe uma maior quantidade de mensagens recebidas pelos nodos, provocando *buffer overflow* e, portanto, maiores perdas de mensagens.

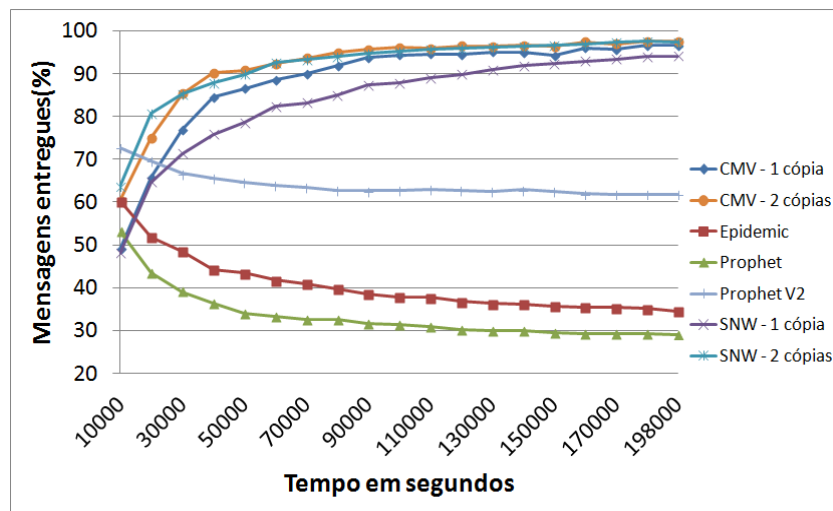


Figura 54 - Comparação de taxas de entrega de mensagens para a faixa de 80 metros

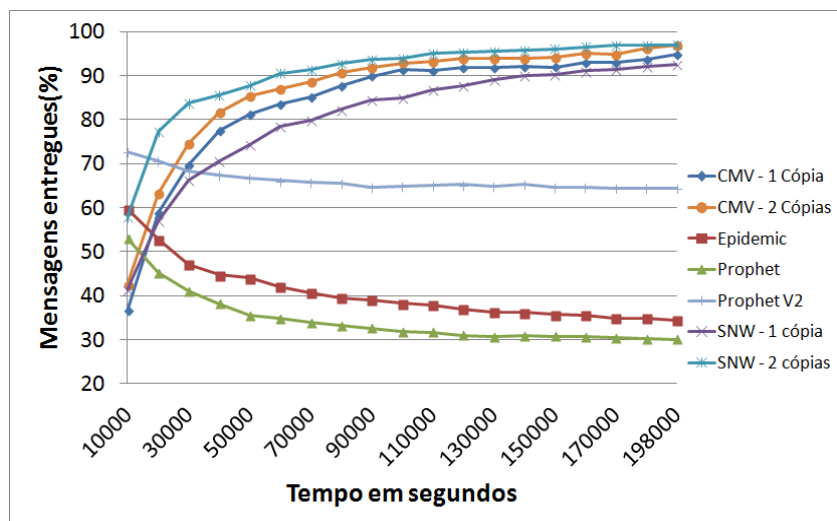


Figura 55 - Comparação de taxas de entrega de mensagens para a faixa de 50 metros

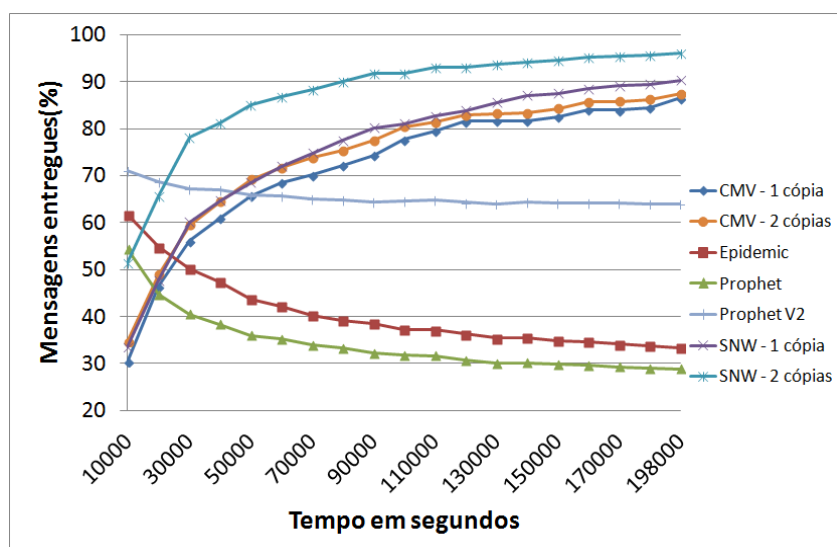


Figura 56 - Comparação de taxas de entrega de mensagens para a faixa de 30 metros

O percentual de entrega de mensagens no protocolo CMV é superior do que a dos protocolos *Epidemic*, *Prophet* e *Prophet V2* em todos os cenários simulados.

O CMV limita o número de cópias de mensagens, o que reduz significativamente o *buffer overflow*, mesmo quando a duração da simulação possui um tempo elevado. O protocolo *Spray and Wait* também faz isso. É por isso que seu percentual de entrega de mensagens é semelhante ao CMV, conforme detalhado na Tabela 9.

Tabela 9: Probabilidade de Entrega

Protocolos	Probabilidade de entrega(%)		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	96,57	94,96	86,54
CMV – 2 cópias	97,64	97,04	87,49
<i>Epidemic</i>	34,54	34,40	33,27
<i>PRoPHET</i>	29	30,06	28,84
<i>PRoPHET V2</i>	61,86	64,46	63,87
SNW – 1 cópia	94,09	92,53	90,29
SNW – 2 cópias	97,36	97,03	96

O mecanismo para controlar o número de cópias em CMV e SNW reduz significativamente *buffer overflow*. Também é possível observar que as mensagens no SNW percorrem um caminho mais curto, quando comparado com CMV. Consequentemente, SNW retém mais mensagens no *buffer*, como pode ser visto na Tabela 10. Em contraste, a combinação do controle de réplicas e o conceito de CMV faz com que ocorra *buffer overflow* menos frequentemente, embora mensagens possam percorrer caminhos mais longos a partir da origem até o destino, tal como mostrado na tabela 11.

Tabela 10: Tempo médio de uma mensagem nos buffers

Protocolos	Tempo médio de uma mensagem nos buffers (segundos)		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	1693	4501	12503
CMV – 2 cópias	1975	5798	15318
<i>Epidemic</i>	442	619	956
<i>PRoPHET</i>	440	586	859
<i>PRoPHET V2</i>	859	735	1106
SNW – 1 cópia	142696	142122	142667
SNW – 2 cópias	101296	108238	109292

Tabela 11: Mensagens descartadas devido ao *buffer overflow*

Protocolos	<i>Buffer Overflow</i>		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	0	0	0
CMV – 2 cópias	0	0	0
<i>Epidemic</i>	9103532	7282435	5131249
<i>PRoPHET</i>	5375269	4777731	3629252
<i>PRoPHET V2</i>	3960253	2897812	2146388
SNW – 1 cópia	231	228	237
SNW – 2 cópias	2547	2395	2437

Por outro lado, observa-se que o atraso na entrega das mensagens no protocolo CMV é maior do que nos protocolos *Epidemic*, *PRoPHET* e *PRoPHET V2*. No entanto, essa diferença diminui à medida que aumenta o raio de alcance do nodo. Isto acontece porque, no protocolo CMV, há um menor número de nodos que possuem cópias das mensagens, uma vez que a transferência é realizada apenas para os nodos que possuem uma maior força de atração. Isto causa um atraso mais longo para atingir o nodo de destino.

Em cenários muito esparsos, com raio de alcance de 30 metros, por exemplo, os atrasos são mais longos por causa de uma quantidade menor de contatos realizados como pode ser visto na Tabela 12.

Tabela 12: Atraso médio na entrega das mensagens

Protocolos	Atraso médio (segundos)		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	6247	9479	18634
CMV – 2 cópias	4232	7337	16848
<i>Epidemic</i>	1338	1561	2001
<i>PRoPHET</i>	933	1229	1533
<i>PRoPHET V2</i>	1062	1339	1816
SNW – 1 cópia	9629	11501	14753
SNW – 2 cópias	4446	5296	7215

Com o aumento do número de dispositivos de comunicação, e conseqüentemente o aumento do tráfego de dados, existe a necessidade de quantificar o tráfego gerado pela transmissão das mensagens de dados de cada protocolo. A Tabela 13 demonstra que os algoritmos sem controle de réplicas como *Epidemic*, *PRoPHET* e *PRoPHET V2* possuem um

tráfego de dados muito elevado quando comparados com os algoritmos com controle de réplicas como CMV e SNW. Uma das desvantagens do alto tráfego de dados é o aumento do consumo de energia.

Tabela 13: Tráfego gerado pelas transmissões de mensagens

Protocolos	Tráfego das mensagens (Megabyte)		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	21618	13896	12172
CMV – 2 cópias	25037	16023	13206
<i>Epidemic</i>	3902925	2840063	1895082
<i>PRoPHET</i>	4063283	3122087	2216858
<i>PRoPHET V2</i>	3459240	2613913	1894746
SNW – 1 cópia	4951	5228	6756
SNW – 2 cópias	10734	11438	15611

5.4. Análise e comparativo dos resultados com a utilização de *throwboxes*

Com base nos dados apresentados, observamos que o algoritmo de campos de magnéticos quando utilizado em cenários que possuem uma conectividade frequente permitiu um desempenho superior aos demais protocolos comparados. No entanto, quando há uma baixa conectividade, percebe-se uma queda no desempenho. Ao analisarmos a Figura 56 que utilizou o raio de alcance 30 metros, o algoritmo *spray and wait* obteve um resultado superior ao de campos magnéticos virtuais. Esta superioridade do *spray and wait* se deve principalmente ao seu bom desempenho em cenários abrangentes com alta mobilidade, onde a fase *wait* não leva em consideração a frequência dos contatos com os nodos que não são o destino final [23].

Com o intuito de avaliar outros cenários que permitam uma maior conectividade, mas que os nodos mantivessem o mesmo raio de alcance na comunicação entre si, foi avaliado um cenário utilizando *throwboxes*. *Throwboxes* são nodos estacionários sem limitações de energia com maior capacidade de processamento, armazenamento e que podem ser interligados permitindo “atalhos” de comunicação entre pontos na rede. A utilização de *throwboxes* já foi realizada com sucesso nos projetos [12] e [38], por exemplo. Assim, foram realizadas algumas alterações pontuais no cenário utilizado até o momento com o intuito de avaliar o uso de *throwboxes*.

Dentre as mudanças, foram inseridos 5 *throwboxes* em cinco pontos do mapa utilizado lembrando o formato de um X, onde em cada extremidade do X foi inserida uma *throwbox* e no centro do X uma *throwbox*, conforme ilustrado na Figura 57.

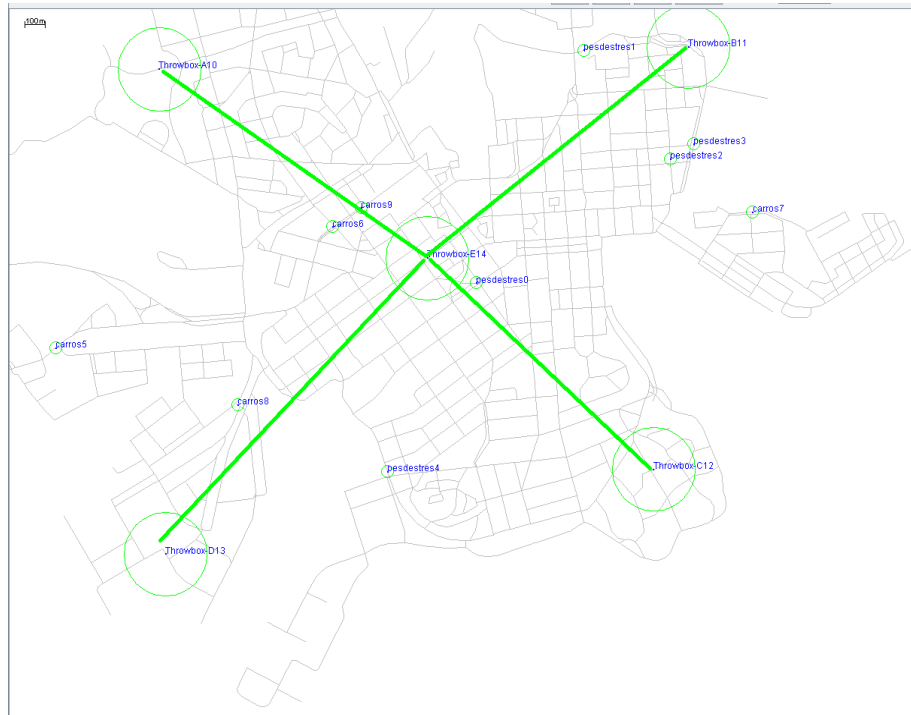


Figura 57 - Throwboxes interligando a parte central da cidade de Helsinki

As *throwboxes* das extremidades estão conectadas a *throwbox* do centro ficando assim todas interconectadas indiretamente entre elas.

Cada *throwbox* possui as seguintes características:

- Capacidade de armazenamento de 1024MB;
- Taxa de transmissão entre as *throwboxes* de 10Mbps;
- Em relação à comunicação entre uma *throwbox* e os veículos e pedestres, esta se restringe a um raio de alcance local de 200 metros e a uma taxa de transmissão de 4Mbps;
- O raio de alcance e a taxa de transmissão entre veículos e pedestres e demais características continuaram as mesmas do cenário descrito na seção 5.1.1.

Com base nos resultados obtidos nas simulações realizadas nas seções anteriores, foi constatado um desempenho visivelmente superior dos protocolos CMV (campos magnéticos virtuais) e SNW (*spray and wait*) em relação aos demais protocolos. Portanto, foram escolhidos os dois algoritmos citados para avaliar os seus desempenhos com a utilização de *throwboxes*.

Seguindo os mesmos critérios das simulações anteriores, foram realizadas simulações com os raios de alcance de 80, 50 e 30 metros e a utilização de controle de réplicas com 1 e 2 cópias da mensagem como pode ser visto nas Figuras 58, 59 e 60.

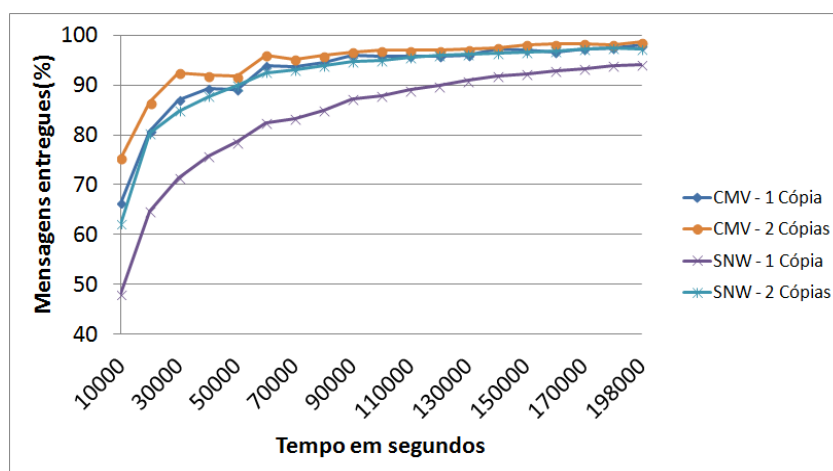


Figura 58 - Comparação de taxas de entrega de mensagens para a faixa de 80 metros

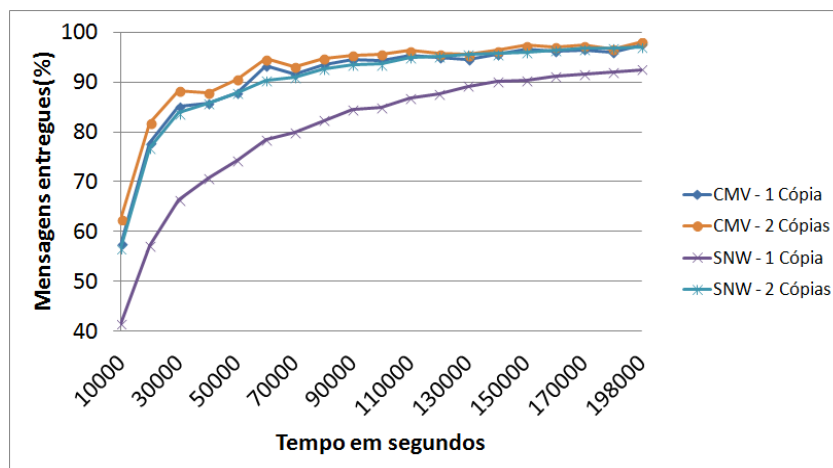


Figura 59 - Comparação de taxas de entrega de mensagens para a faixa de 50 metros

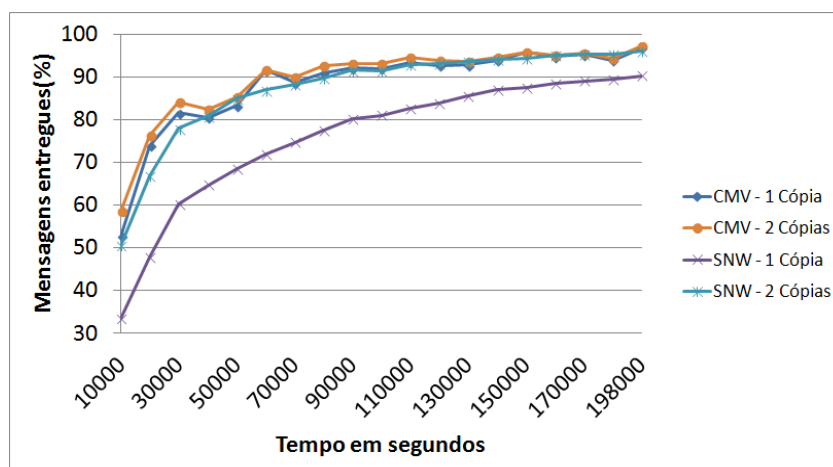


Figura 60 - Comparação de taxas de entrega de mensagens para a faixa de 30 metros

Os dados comprovam que, o aumento da conectividade proporcionada pela utilização das *throwboxes*, elevaram de forma significativa a eficiência na entrega das mensagens com o uso de campos magnéticos virtuais conforme pode ser observado nos dados das Tabelas 14, 15 e 16 utilizados para gerar os gráficos acima apresentados.

Se observarmos a Tabela 16, podemos perceber que o desempenho do CMV foi melhor do que o do protocolo *spray and wait* durante todo período da simulação, inclusive quando utilizado com o raio de alcance de 30 metros. No cenário avaliado anteriormente, o algoritmo *spray and wait* tinha obtidos dados superiores utilizado o raio de alcance de 30 metros. Este melhor desempenho se deu, principalmente em virtude do uso mais eficiente da conectividade pelo CMV, onde propiciou ao mesmo um alcance de atração maior, ao contrario do SNW, onde a fase *wait (Direct Delivery)* não permitiu a utilização de forma eficiente da conectividade propiciada pelas *throwboxes*.

Tabela 14: Percentual de mensagens entregues com o raio de 80 metros

Percentual de Mensagens Entregues				
Tempo(segundos)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 CÓPIAS
10000	66,41%	48,05%	75,39%	62,11%
20000	80,70%	64,72%	86,55%	80,27%
30000	87,01%	71,30%	92,47%	84,81%
40000	89,28%	75,80%	92,01%	87,72%
50000	89,09%	78,57%	91,74%	89,95%
60000	93,89%	82,39%	96,04%	92,53%
70000	93,65%	83,23%	95,21%	92,98%
80000	94,54%	84,99%	95,91%	93,96%
90000	96,01%	87,30%	96,66%	94,71%
100000	95,87%	87,84%	96,96%	94,93%
110000	95,75%	89,01%	97,02%	95,64%
120000	95,78%	89,86%	96,98%	95,91%
130000	96,01%	90,91%	97,18%	96,13%
140000	97,16%	91,84%	97,55%	96,32%
150000	97,01%	92,30%	98,15%	96,57%
160000	96,66%	92,91%	98,24%	96,76%
170000	97,32%	93,35%	98,37%	97,20%
180000	97,47%	93,91%	98,09%	97,36%
198000	97,99%	94,09%	98,70%	97,32%

Tabela 15: Percentual de mensagens entregues com o raio de 50 metros

Percentual de Mensagens Entregues				
Tempo(segundos)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 CÓPIAS
10000	57,42%	41,41%	62,50%	56,64%
20000	77,78%	57,12%	81,87%	76,76%
30000	85,06%	66,23%	88,31%	83,77%
40000	85,67%	70,66%	87,91%	85,77%
50000	87,84%	74,26%	90,56%	87,91%
60000	93,31%	78,48%	94,67%	90,44%
70000	91,59%	79,89%	93,15%	91,03%
80000	93,42%	82,30%	94,74%	92,64%
90000	94,45%	84,53%	95,41%	93,50%
100000	94,42%	84,91%	95,63%	93,61%
110000	95,32%	86,81%	96,31%	94,90%
120000	95,00%	87,68%	95,71%	95,16%
130000	94,48%	89,20%	95,53%	95,53%
140000	95,60%	90,08%	96,35%	95,68%
150000	96,72%	90,30%	97,37%	95,94%
160000	96,25%	91,10%	97,05%	96,47%
170000	96,51%	91,53%	97,41%	96,77%
180000	95,93%	92,07%	96,66%	96,79%
198000	97,52%	92,53%	98,05%	97,01%

Tabela 16: Percentual de mensagens entregues com o raio de 30 metros

Percentual de Mensagens Entregues				
Tempo(segundos)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 CÓPIAS
10000	52,73%	33,46%	58,59%	50,58%
20000	73,88%	47,76%	76,41%	66,80%
30000	81,56%	60,13%	84,03%	77,92%
40000	80,51%	64,68%	82,46%	80,88%
50000	83,31%	68,49%	85,34%	85,02%
60000	91,55%	71,91%	91,75%	86,93%
70000	88,64%	74,71%	90,08%	88,25%
80000	90,98%	77,47%	92,74%	89,76%
90000	92,07%	80,20%	93,20%	91,55%
100000	91,89%	81,05%	93,10%	91,46%
110000	93,48%	82,70%	94,54%	92,91%
120000	92,75%	83,88%	93,83%	93,14%
130000	92,83%	85,60%	93,64%	93,64%
140000	93,93%	87,10%	94,68%	94,09%
150000	95,74%	87,47%	95,89%	94,41%
160000	94,76%	88,45%	95,17%	95,15%
170000	95,27%	89,08%	95,50%	95,34%
180000	93,93%	89,43%	94,43%	95,43%
198000	96,75%	90,31%	97,16%	96,04%

Todavia, são necessárias evidências estatísticas suficientemente conclusivas para afirmar que o CMV possui desempenho melhor que o SNW com base nos resultados apresentados nas tabelas 14, 15 e 16.

Para isto, primeiramente, definimos um nível de significância de 0,05 em todo o experimento supracitado e foi utilizado o Action [54], para os cálculos. Neste contexto, o Teste de hipótese aplicado foi ANOVA[55], caracterizado como paramétrico. Para isso, verificou-se na tabela 17 que nas três amostras, os *p-value* são fortemente menores que o nível de significância adotado. Portanto, podemos afirmar que foram encontradas evidências estáticas para os resultados apresentados.

Tabela 17: ANOVA

Amostras	G.L.	Soma de Quadrados	Quadrado Médio	Estat. F	P-value
Tabela 14	1	718,4461289	718,4461289	7,12538737	0,0113
Tabela 15	1	982,2694737	982,2694737	7,129084073	0,0113
Tabela 16	1	1425,733753	1425,733753	8,187241523	0,007

Em relação ao atraso médio, foram levantadas a média de atraso por faixa de tempo da criação das mensagens. O CMV obteve um desempenho mais eficiente em relação ao SNW em todos os cenários simulados. Essa eficiência é mais evidente nos cenários com 1 cópia da mensagem, isso ocorre em virtude do CMV explorar o raio de alcance das *throwboxes* de forma mais eficiente do que o SNW com visto nas tabelas 18, 19 e 20.

Tabela 18: Atraso médio na entrega das mensagens com o raio de 80 metros

Atraso médio (segundos)				
Momento da criação (faixa)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 Cópias
0 - 10000	4204	11146	3045	5776
10000,01 - 20000	4617	10748	3005	5097
20000,01 - 30000	4594	13698	3046	6609
30000,01 - 40000	4298	11036	2933	5101
40000,01 - 50000	4008	12766	2954	4849
50000,01 - 60000	3875	12799	2751	5725
60000,01 - 70000	3849	12709	2825	4797
70000,01 - 80000	3579	13388	2604	6201
80000,01 - 90000	3263	10590	2495	4705
90000,01 - 100000	3823	10379	2678	4504
100000,01 - 110000	4530	10030	3153	4707
110000,01 - 120000	4721	9309	3236	5208
120000,01 - 130000	4425	9470	3847	4958
130000,01 - 140000	3988	8598	2933	4885
140000,01 - 150000	4325	7784	3114	4314
150000,01 - 160000	4238	8493	2839	4557
160000,01 - 170000	4648	6126	3158	3952
170000,01 - 180000	4331	5996	3117	4132
180000,01 - 190000	3730	4292	2917	3381

Tabela 19: Atraso médio na entrega das mensagens com o raio de 50 metros

Atraso médio (segundos)				
Momento da criação (faixa)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 Cópias
0 - 10000	5284	13276	4495	6967
10000,01 - 20000	4612	13224	3830	5812
20000,01 - 30000	5346	16449	3945	7559
30000,01 - 40000	5603	14230	4320	6647
40000,01 - 50000	4854	14488	3984	6117
50000,01 - 60000	4814	14757	3790	5556
60000,01 - 70000	5176	15240	4299	6065
70000,01 - 80000	4508	15467	3426	6350
80000,01 - 90000	4555	12186	3897	5671
90000,01 - 100000	4919	11517	3740	5186
100000,01 - 110000	5577	12199	4157	5630
110000,01 - 120000	5942	11355	4792	6328
120000,01 - 130000	7327	11274	5963	5748
130000,01 - 140000	5109	10263	4104	5072
140000,01 - 150000	5322	8874	4259	5408
150000,01 - 160000	5839	9916	4512	5212
160000,01 - 170000	6665	7038	5225	4190
170000,01 - 180000	6244	6681	4821	4534
180000,01 - 190000	4423	4465	3761	3791

Tabela 20: Atraso médio na entrega das mensagens com o raio de 30 metros

Atraso médio (segundos)				
Momento da criação (faixa)	CMV - 1 NÓ P	SNW - 1 Cópia	CMV - 1 NÓ P - 2 Cópias	SNW - 2 Cópias
0 - 10000	6163	18072	5658	8846
10000,01 - 20000	5879	18619	4885	8209
20000,01 - 30000	6964	20968	6353	8933
30000,01 - 40000	7843	17302	6889	8265
40000,01 - 50000	6296	19138	5800	7332
50000,01 - 60000	7052	18469	5974	7752
60000,01 - 70000	7368	19752	5831	8080
70000,01 - 80000	6318	18197	5753	8911
80000,01 - 90000	7971	15853	6691	8124
90000,01 - 100000	6560	15052	5764	7128
100000,01 - 110000	7396	15508	6318	7623
110000,01 - 120000	7773	15222	7144	7708
120000,01 - 130000	9038	15458	8555	8155
130000,01 - 140000	7668	13670	6835	7125
140000,01 - 150000	8052	11574	7194	7025
150000,01 - 160000	7440	12086	6485	7397
160000,01 - 170000	8808	8423	8195	5340
170000,01 - 180000	7613	7661	7365	6124
180000,01 - 190000	5070	4653	4933	4188

Em relação ao número de mensagens descartadas devido ao *buffer overflow*, o algoritmo *spray and wait* apresentou uma queda em relação à quantidade de mensagens descartas no cenário anterior. No caso do CMV, o algoritmo não descartou nenhuma mensagem em nenhum dos raios de alcance utilizados tanto com 1 ou com 2 cópias da mensagem, como pode ser observado na Tabela 21. Com isso, observamos que o algoritmo de campos magnéticos consegue, de forma eficiente, aproveitar o espaço de armazenamento distribuído disponível na rede.

Tabela 21: Mensagens descartadas devido ao *buffer overflow*

Protocolos	<i>Buffer Overflow</i>		
	Raio 80m	Raio 50m	Raio 30m
CMV – 1 cópia	0	0	0
SNW – 1 cópia	231	228	237
CMV – 2 cópias	0	0	0
SNW – 2 cópias	2481	2244	2177

5.5. Conclusão

Através das simulações, procuramos demonstrar que o algoritmo de campos magnéticos virtuais possui, em todos os casos, percentuais de mensagens entregues superiores aos dos protocolos *epidemic*, *PRoPHET* e *PRoPHET V2*, mantendo taxas de entrega superiores ao protocolo *Spray and Wait* na maioria dos cenários.

O algoritmo proposto apresentou uma melhor utilização dos *buffers*, evitando o descarte de mensagens em decorrência da ocorrência do *buffers overflow* quando comparado com os demais algoritmos.

Através da utilização de *throwboxes*, foi possível fazer uma avaliação do desempenho em consequência do aumento da conectividade gerada na rede. Neste caso, evidenciou-se um desempenho superior do algoritmo de campos magnéticos em relação ao *spray and wait* em todos os cenários simulados.

A avaliação feita por meio das simulações executadas, nos permite concluir que a adaptação do conceito de algoritmos magnéticos para *anycast* em DTNs estocásticas se mostrou viável e mesmo com resultados superiores. As taxas de entregas foram muito boas e o uso dos recursos disponíveis na rede foi econômico (sem onerar de forma significativa o armazenamento, tráfego da rede, dentre outros).

O balanceamento de carga só faz sentido em comunicação *anycast*. Porém, como nenhum dos protocolos da literatura lida com *anycasting* em DTNs estocásticas, uma análise comparativa não é viável. Desta forma, um estudo mais aprofundado (e.g., o ajuste fino de parâmetros) por meio de simulação desta característica do protocolo CMV é considerada trabalho futuro.

Capítulo 6 - Conclusão

Após apresentar os resultados obtidos nesta pesquisa, este capítulo posiciona a sua relevância no cenário estudado, demonstrando as principais contribuições e perspectivas de trabalhos futuros.

6.1. Relevância do Estudo

Esta pesquisa investigou o problema de roteamento de mensagens para DTNs *anycast* do tipo estocástica.

Na literatura, os principais trabalhos de roteamento em DTNs utilizam o método de endereçamento *unicast*, onde o destino para uma mensagem é definido no momento da sua geração, ou seja, o destino da mensagem é determinado no momento em que a mensagem é criada.

Mas em alguns cenários, existem situações onde a possibilidade de vários nodos da rede estejam aptos a processar um determinado tipo mensagem são cada vez maiores como, por exemplo, em redes de transporte urbanas onde mensagens de emergências poderiam ser disparadas por um veículo (nodo) na rede com a finalidade de obter ajuda de equipes especializadas como, por exemplo, carro de bombeiros (nodo) ou uma ambulância (nodo) que estariam aptos a processá-la e não somente um veículo (nodo) específico na rede, ou seja, *anycasting*.

Com base nessa necessidade e depois de um amplo levantamento bibliográfico, não foi encontrado um algoritmo de roteamento *anycast* com balanceamento de carga em DTNs do tipo estocásticas. Isso se deve ao fato de DTN *anycast* ser considerada uma área de investigação recente, mas que recentemente vem ganhando espaço em virtude da possibilidade de novos serviços/aplicações em um mercado cada vez mais crescente que é o de dispositivos móveis. Contudo, os poucos trabalhos encontrados na literatura direcionam seus estudos para DTNs *anycast* determinísticas, e os raros trabalhos encontrados com DTN *anycast* do tipo estocástica concentram os seus esforços em propostas de modelos para

formação de grupos e definições de receptores e métricas. Portanto, não foram encontrados trabalhos de roteamento *anycast* com suporte a balanceamento de carga em DTNs do tipo estocásticas em virtude principalmente a ser uma área em fase embrionária.

Todavia, neste tipo de roteamento, o destino para uma mensagem é qualquer elemento de um determinado grupo de nodos definido pelo tipo da mensagem. *Anycasting* pode se reduzir a *unicasting* se o grupo de nodos destinatário contiver apenas um nodo, sendo, portanto, mais genérico.

Este trabalho apresentou assim um novo método de roteamento *anycast* de mensagens em DTNs do tipo estocásticas através da adaptação do conceito de campos magnéticos virtuais, ainda com o subproduto do balanceamento de carga entre os destinatários potenciais.

Para que fosse possível a sua utilização, foram realizadas alterações relacionadas a nodos que estariam aptos a processar uma mensagem como também novas definições no processo de aumento e diminuição da força que se tornaram essenciais para a aplicação do algoritmo em DTNs.

Baseado na implementação de um algoritmo de notificação denominado *heartbeat*, foi possível utilizar o conceito de atração em cenários com frequentes desconexões ocasionadas pela mobilidade. Desta forma, foi possível atrair uma mensagem por uma rota dinâmica.

Ao mesmo tempo, o algoritmo de notificação viabilizou de forma implícita o suporte a balanceamento de carga em nível de aplicação. Por exemplo, um carro de polícia (nodo) sobrecarregado com diversos chamados pode reduzir sua força de atração de forma dinâmica, fazendo com que as mensagens sejam atraídas para outro veículo (nodo) apto a processá-las.

Os dados obtidos nas simulações no contexto de *anycast*, em cenários com grupos *anycast* que representavam 10% e 2% dos nodos existentes e raios de alcance dos nodos com 80, 50 e 30 metros, evidenciaram a viabilidade do uso do algoritmo. Nestes casos, o algoritmo apresentou percentuais de mensagens entregues superiores a 98%.

Em relação ao contexto *unicast*, foi possível compará-lo com algoritmos como *Epidemic*, *PRoPHET*, *PRoPHET V2* e *Spray and Wait* restringindo o conjunto de destinatários potenciais a apenas um nodo. O algoritmo de campos magnéticos virtuais definido apresentou um percentual de entrega de mensagens superior em todos os cenários analisados, quando comparado com *Epidemic*, *PRoPHET*, *PRoPHET V2* e superior na maior parte dos cenários analisados, quando comparado com o *Spray and Wait*. Esse desempenho superior deve-se principalmente ao conceito da atração, onde os nodos pertencentes a uma

sub-rede são capazes de identificar um destino em potencial em qualquer ponto da sub-rede em virtude do algoritmo *Heartbeat*, permitindo assim que a mensagem seja atraída rapidamente para um destino possível, diferentemente dos demais algoritmos citados que adotam outras estratégias de repasse.

Quando analisados, os dados obtidos relacionados ao descarte de mensagens em consequência do *overflow*, quando comparados o algoritmo proposto com os protocolos supracitados, evidenciou-se um desempenho bem acima dos demais algoritmos, principalmente com os algoritmos com características epidêmicas. Em todos os cenários com o algoritmo proposto, não ocorreram descartes de mensagens, demonstrando assim uma excelente utilização dos *buffers* disponíveis na rede. As principais causas para essa ótima eficiência na utilização dos buffers foram: o uso de um mecanismo de controle de réplicas, evitando assim estratégias de *flooding* que tendem a sobrecarregar a rede ao longo do tempo, e a utilização do conceito da atração do próprio algoritmo que evita a permanência das mensagens durante longo tempo em mesmo *buffer*. Ao contrario do *Spray and Wait*, que mesmo com controle de réplicas ocorrem *overflow* em virtude da segunda (*Wait*) fase do algoritmo, que tem como consequência uma maior permanência de uma mensagem em um mesmo nodo.

Essa pesquisa resultou em dois trabalhos apresentados¹ em eventos, os quais tiveram como objetivos obter um *feedback* que contribuísse para o direcionamento do mesmo e a comprovação da sua eficiência. Outros artigos com os resultados completos obtidos neste trabalho se seguirão (com base na utilização de *throwboxes*).

6.2. Contribuições da Pesquisa

A principal contribuição descrita nesta pesquisa, foi a apresentação e a constatação da viabilidade de um algoritmo de roteamento *anycast* (e *unicast*) para mensagens em DTNs, ambos do tipo estocástico com suporte a balanceamento de carga (no caso de *anycast*). Para

¹ Oliveira Júnior, J. A. ; Lima Junior, L. A. P. ; Andrade, G. E. ; Michellon, Gisane A . *Anycasting in DTNs using Virtual Magnetic Fields. In: The 11th Annual IEEE Consumer Communications & Networking Conference, 2014, Las Vegas. IEEE CCNC 2014 Conference Proceedings, 2014.*

Oliveira Júnior, J. A. ; Lima Junior, L. A. P. ; Roteamento *anycast* em DTNs usando campos magnéticos virtuais. Trabalho apresentado no 3º Congresso Sul Brasileiro de Iniciação Científica e Pós-Graduação, Curitiba, 2014.

isso, realizou-se uma adaptação no algoritmo de campos magnéticos virtuais com a finalidade de viabilizar o seu uso em DTNs *anycast* do tipo estocásticas.

Outra contribuição ocorreu durante processo de adaptação do algoritmo. A definição de um novo algoritmo de notificação denominado *heartbeat* foi necessária que além de fornecer um serviço de notificação para determinados tipos de nodos, permitiu a criação de uma rota dinâmica durante a atração da mensagem até um possível destino.

Ainda, como contribuição podemos citar o balanceamento de carga que ocorre de forma dinâmica em nível de aplicação viabilizado de forma implícita pelo algoritmo *heartbeat*.

Após a realização de extensivas simulações, comprovou-se a eficiência da proposta do algoritmo utilizado para DTNs *anycast* e *unicast* do tipo estocásticas em comparação com outros algoritmos da literatura.

Foram realizadas simulações com a utilização de *throwboxes*, recurso esse abordado em aplicações reais como em [12], mostrando um desempenho ainda superior ao algoritmo *Spray and Wait*, o qual foi utilizado para fins de comparação de desempenho. Isto também evidencia a flexibilidade do método proposto quando aplicado em cenários distintos.

Por fim, a revisão bibliográfica feita também corresponde a outra contribuição deste trabalho. Esta poderá ser utilizada para novas pesquisas e trabalhos relacionados nas áreas abordadas neste trabalho: DTNs, roteamento com *anycast*, balanceamento de carga e campos magnéticos virtuais.

6.3. Trabalhos Futuros

Destacam-se aqui algumas possibilidades que representam possíveis extensões desta pesquisa que valeriam uma investigação mais aprofundada:

- Realização de simulações adicionais em outros cenários e aplicações ainda não explorados.
- Estudo da aplicação de sensibilidade a contextos em cenários com baixa conectividade (onde observa-se uma diminuição no desempenho do algoritmo).
- Análise e verificação da viabilidade do uso de balanceamento de carga em cenários mais restritos com demandas de alto desempenho.

- Análise do impacto de estratégias de balanceamento de carga.
- Realização de um estudo mais aprofundado do uso de *throwboxes*.

Referências Bibliográficas

- [1] A. A. HANBALI, E. ALTMAN, AND P. NAIN, A Survey of TCP over Ad Hoc Networks, *IEEE Communications Surveys & Tutorials* (2005), vol. 7, IEEE Computer Society, pág. 22–36.
- [2] S. FARRELL, V. CAHILL, D. GERAGHTY, I. HUMPHREYS, AND P. MCDONALD, When TCP Breaks: Delay- and Disruption- Tolerant Networking. *IEEE Internet Computing* 10, 4 (2006), pág. 72–78.
- [3] K. FALL, A Delay-tolerant Network Architecture for Challenged Internets. Em *ACM SIGCOMM* (agosto de 2003), ACM Press, pág. 27–34.
- [4] Z. ZHANG, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials*, 2006, IEEE, pág. 24–37.
- [5] K. Scott, and S. Burleigh. IETF RFC 5050. Disponível em: <http://www.ietf.org/rfc/rfc5050.txt>. Acesso em: 15 ago. 2015.
- [6] V. Cerf. IETF RFC 4838. Disponível em: <http://www.ietf.org/rfc/rfc4838.txt>. Acesso em: 15 ago. 2015.
- [7] 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, in *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2007*, Belém, PA, 54p, Brasil.

- [8] GTA-UFRJ. Redes Tolerantes a Atrasos e Desconexões. Disponível em: http://www.gta.ufrj.br/grad/11_1/dtn/NotesImages/Topic4NotesImage2.jpg. Acesso em: 17 jun. 2015.
- [9] OLIVEIRA, E. C. Roteamento Adaptativo a Contextos para Redes Tolerantes a Atrasos e Desconexões, Tese de Doutorado do Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, 2011, Niterói, RJ, 173p, Brasil.
- [10] PARIKH, S.; DURST, R. C. Disruption tolerant networking for Marine Corps CONDOR. In: Military Communications Conference, 2005. MILCOM 2005. IEEE, Vol. 1, pág. 325-330.
- [11] DEMMER, M.; DU, B and BREWER. E. TierStore: a distributed filesystem for challenged networks in developing regions. In: 6th Conference on File and Storage Technologies, 2005. USENIX 2008. ACM.
- [12] UMass DieselNet, Disponível em <https://dome.cs.umass.edu//umassdieselnet>. Acesso em: 10 março de 2015.
- [13] NASA/JPL. NASA Tests First Deep-Space Internet. Disponível em: <http://www.nasa.gov/topics/technology/features/internet-20081118.html>. Acesso em: 11 mai. 2015.
- [14] Pentland, A. S.; Fletcher, R.; Hasson, A. “DakNet: Rethinking Connectivity in Developing Nations”. Computer, vol. 37-1, Janeiro 2004, pág. 78– 83.
- [15] Ott, J.; Kutscher, D. “A Modular Access Gateway for Managing Intermittent Connectivity in Vehicular Communications”. European Transactions on Telecommunications, vol. 17-2, Junho 2006, pág. 159–174.
- [16] Rice, J. Seaweb Acoustic Communication and Navigation Networks. Em International Conference on Underwater Acoustic Measurements: Technologies & Results, Grécia, 2005.

- [17] TurtleNet, Disponível em: <https://dome.cs.umass.edu//turtlenet>, Acesso em: 12 jan. 2015.
- [18] Juang, P.; Oki, H.; Wang, Y.; Martonosi, M.; Peh, L. S.; Rubenstein, D. “Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet”. In: 10th Annual Conference on Architectural Support for Programming Languages and Operating Systems, 2002, pág. 96–107.
- [19] VAHDAT, A.; BECKER, D. Epidemic Routing for Partially-Connected Ad Hoc Networks, Relatório Técnico, Duke University, 2000, 14p.
- [20] LIMA JR., L.; CALSAVARA, A. Autonomic Application-Level Message Delivery Using Virtual Magnetic Fields. *Journal of Network and Systems Management*, Springer, v. 18, n. 1, pág. 97-116, 2010.
- [21] GALPERIN, H. Composição de *Campos Magnéticos Virtuais* com Sistemas de Confiança e Reputação. Dissertação do Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, Curitiba-Brasil, 2011.
- [22] CALSAVARA, A.; LIMA JR., L. Routing Based on Message Attraction. In: IEEE. 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops. [S.l.], 2010. pág. 189-194.
- [23] SPYROPOULOS, T., PSOUNIS, K., and RAGHAVENDRA, C, Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 2008, pág. 77–90.
- [24] LINDGREN, A., DORIA, A. e SCHELÉN, O, Probabilistic routing in intermittently connected networks. *International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR)*, 2004, volume 7. Springer.

- [25] OLIVEIRA, E. C. R. DE; ALBUQUERQUE, C. V. N. NECTAR: A DTN Routing Protocol based on Neighborhood Contact History. In: 24th Annual ACM Symposium on Applied Computing, 2009, pág. 40–46.
- [26] BURGESS, J., GALLAGHER, B., JENSEN, D., LEVINE, B. N. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. Em: Proceedings of IEEE International Conference on Computer Communications (Infocom) (Abril 2006), pág. 1–11.
- [27] BURNS, B.; BROCK, O.; LEVINE, B. N. MV Routing and Capacity Building in Disruption Tolerant Networks. In: IEEE Infocom, 2005, pág. 398–408.
- [28] ABDULLA, M.; SIMON, R. The Impact of the Mobility Model on Delay Tolerant Networking Performance Analysis. In: 40th Annual Simulation Symposium, 2007, pág. 177–184.
- [29] ANDRADE, GIL EDUARDO DE ; LIMA JUNIOR, LUIZ AUGUSTO DE PAULA ; CALSAVARA, ALCIDES ; MAZIERO, CARLOS ALBERTO. Routing Protocol Based on the Position, Velocity, and Direction of the Nodes. In: IEEE 27th International Conference on Advanced Information Networking and Applications AINA/FINA, 2013, Barcelona, pág. 971–976.
- [30] GRASIC, S. DAVIES, E., LINDGREN, A. AND DORIA, A, The evolution of a DTN routing protocol - PRoPHETv2, in The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR), 2004.
- [31] SILVA, E. R. Algoritmo de Roteamento Anycast Baseado em Algoritmos Genéticos Para Redes Tolerantes a Atrasos e Desconexões, Tese de Doutorado do Programa de Pós-Graduação Em Engenharia Elétrica da UFU-MG, 2010, Uberlândia, RS, 165p, Brasil.
- [32] UNDERSEAWARFARE. The Official Magazine of the U.S. Submarine Force em: http://www.navy.mil/navydata/cno/n87/usw/issue_30/images/pic_undersea_acoustic.gif. Acesso em: 17 jun. 2015.

- [33] Unified Modeling Language, Disponível em: <http://www.omg.org/spec/UML/>, Acesso em: 12 ago. 2015.
- [34] SINDTN and CATDTN. The ONE. Disponível em: <http://www.netlab.tkk.fi/tutkimus/dtn/theone/>. Acesso em: 10 jan. 2015.
- [35] JAVA. Oracle Technology Network for Java Developers. Disponível em: <http://www.oracle.com/technetwork/java/index.html>. Acesso em: 11 jan. 2015.
- [36] NUNES, C. M. Roteamento Utilizando Potencial de Entrega em Redes Tolerantes a Atrasos, Tese de Doutorado do Programa de Pós-Graduação em Ciência da Computação da PUC-RS, 2010, Porto Alegre, RS, 149p, Brasil.
- [37] EKMAN, F.; KERÄNEN, A.; KARVO, J.; OTT, J. *Working Day Movement Model*. First ACM SIGMOBILE International Workshop on Mobility Models for Networking Research. 2008, pág. 33-40.
- [38] BANERJEE, N., CORNER, M., LEVINE, B., An Energy-Efficient Architecture for DTN Throwboxes, INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pág. 776–784, 2007.
- [39] XIAO, M.; HUANG, L.; LIU, A.; CHEN, W. Anycast routing in probabilistically contacted delay tolerant networks, In: International Conference on Communications and Mobile Computing, pág. 442-446, 2010.
- [40] GONG, Y., XIONG, Y. ZHANG,Q. ZHANGZ. WANG, W. and XU, Z. Anycast Routing in Delay Tolerant Networks. IEEE Globecom'06, San Francisco, CA, USA, Nov. 27-Dec. 1, 2006.
- [41] PARTRIDGE. C, MENDEZ, T. and MILLIKEN, W. IETF RFC 1546. Disponível em: <https://tools.ietf.org/html/rfc1546>. Acesso em: 15 ago. 2015.

- [42] Anycast. Disponível em: <https://pt.wikipedia.org/wiki/Anycast>. Acesso em: 17 jun. 2015.
- [43] KERÄNEN, A. OTT, J. and KÄRKKÄINEN, T. The ONE simulator for DTN protocol evaluation. Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Milan, ITA, 2009.
- [44] KAVEEVIVITTHAI S. OCHIAI H. and ESAKI, H. Message Deletion and Mobility Patterns for Efficient Message Delivery in DTNs. Proceedings: 8th IEEE international conference on pervasive computing and communications. 2010, pág. 760–763.
- [45] AZIZ, A. and YAMADA, S. A Handoff-based And Limited Flooding (HALF) Routing Protocol in Delay Tolerant Network (DTN). International Journal of Wireless & Mobile Networks (IJWMN) Vol. 4, No. 2, April 2012.
- [46] SILVA, R. F. ; CAMPOS, C. A. V. ; MORAES, L. F. M. . Caracterização da Mobilidade Veicular e o seu Impacto nas Redes Veiculares Tolerantes a Atrasos e Desconexões. Em: XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2010, Gramado (RS). 2010.
- [47] MOSCHETTO, D. A. ; ALMEIDA, R. A. P. ; GUARDIA, H. C. ; POMA, L. P. P. ; FREITAS, D. O. ; MARCONDES, C. A. C . Influência da topologia de mapas na capacidade de transmissão em DTN de pessoas. Em: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2012, Ouro Preto – MG.
- [48] DEERING, S. IETF RFC 1112. Disponível em: <https://tools.ietf.org/html/rfc1112>. Acesso em: 15 nov. 2015.
- [49] Unicast. Disponível em: <https://pt.wikipedia.org/wiki/Unicast>. Acesso em: 15 nov. 2015.
- [50] MOGUL, J. IETF RFC 919. Disponível em: <https://tools.ietf.org/html/rfc919>. Acesso em: 15 nov. 2015.

[51] PINHEIRO, R. Notificações de Eventos em Arquitetura baseada em Campos Magnéticos, Dissertação de Mestrado do Programa de Pós-Graduação em Informática da PUC-PR, 2011, Curitiba, PR, Brasil.

[52] ANDRADE, G. Protocolo de Roteamento Sensível ao Contexto: Posição, Velocidade e Sentido, Para Redes Tolerantes a Atrasos a Desconexões, Dissertação de Mestrado do Programa de Pós-Graduação em Informática da PUC-PR, 2010, Curitiba, PR, Brasil.

[53] GALPERIN, H. LIMA JR, Luiz Augusto. CALSAVARA, A., Applying Reputation to Virtual Magnetic Networks, IEEE 27th International Conference on Advanced Information Networking and Applications Workshops, pág. 831-836, 2013.

[54] Portal Action. Disponível em: <http://www.portalaction.com.br>. Acesso em: 15 nov. 2015.

[55] ANOVA. Disponível em: <http://www.portalaction.com.br/anova>. Acesso em: 15 nov. 2015.