

UNIVERSIDADE FEDERAL FLUMINENSE  
INSTITUTO DE COMPUTAÇÃO  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

**Diego Passos**

## **Métricas de Roteamento para Redes em Malha Sem Fio**

Niterói  
2007

**Diego Passos**

# **Métricas de Roteamento para Redes em Malha Sem Fio**

**Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.**

Orientador: Célio Vinicius Neves de Albuquerque

Niterói  
2007

**Diego Passos**

# **Métricas de Roteamento para Redes em Malha Sem Fio**

**Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.**

Aprovado em Julho de 2007

## **BANCA EXAMINADORA**

---

Prof. CÉLIO VINICIUS NEVES DE ALBUQUERQUE, Ph.D.  
Orientador  
UFF

---

Prof. JULIUS LEITE, Ph.D.  
UFF

---

Profa. LÚCIA M. A. DRUMMOND, D.Sc.  
UFF

Niterói  
2007

# RESUMO

Redes em malha sem fio são redes de múltiplos saltos, auto-configuráveis e de baixo custo. Este trabalho apresenta um estudo sobre as principais métricas de roteamento utilizadas neste tipo de rede, apontando suas virtudes e limitações. São também propostas, implementadas e analisadas três métricas alternativas.

Para analisar o desempenho das métricas propostas, medidas comparativas foram realizadas com a métrica *Expected Transmission Count* utilizando-se as redes do projeto ReMesh. Os resultados obtidos mostram que, em certos casos, o desempenho da rede com as métricas propostas foi melhorado em termos de estabilidade de rotas, taxa de perda de pacotes, atraso e vazão.

## **Palavras Chave:**

Redes em Malha, Métricas de Roteamento, Protocolos de Roteamento

# **ABSTRACT**

Wireless mesh networks are low-cost self-configurable multihop networks. This work presents a study about the main routing metrics used in this kind of network, pointing out their virtues and limitations. It also proposes, implements and analyze three alternative metrics.

To evaluate the performance of the proposed metrics, comparative measurements with the Expected Transmission Count were made. The obtained results show that, in some cases, the network performance with the proposed metrics has been improved in terms of routing stability, packet loss rates, delays and throughput.

**Keywords:**

Mesh Networks, Routing Metrics, Routing Protocols

## **LISTA DE ACRÔNIMOS**

|        |                                                |
|--------|------------------------------------------------|
| AODV:  | Ad-hoc On-Demand Distance Vector               |
| DSR:   | Dynamic Source Routing                         |
| ENT:   | Effective Number of Transmissions              |
| ETX:   | Expected Transmission Count                    |
| ETT:   | Expected Transmission Time                     |
| HSLs:  | Hazy Sighted Link State                        |
| mETX:  | Modified Expected Transmission Count           |
| ML:    | Minimum Loss                                   |
| MLAC:  | Minimum Loss with Additive Cost                |
| MLURP: | Minimum Loss with Unicast Rate Probing         |
| OLSR:  | Optimized Link State Routing                   |
| RTT:   | Round-Trip Time                                |
| TC:    | Topology Control                               |
| TCP:   | Transmission Control Protocol                  |
| WCETT: | Weighted Cumulative Expected Transmission Time |

À minha família.

## AGRADECIMENTOS

À Universidade Federal Fluminense,  
Ao Professor Célio Vinicius Neves de Albuquerque,  
pela sua constante orientação,  
Aos Professores Luiz Cláudio Schara Magalhães e  
Débora Christina Muchaluat Saade,  
pelas oportunidades e conselhos,  
A todos os professores do Instituto de Computação,  
dos quais tive o privilégio de ser aluno,  
A Douglas Vidal, Jairo Duarte e Rafael do Valle,  
por sua ajuda inestimável neste trabalho,  
A Fernanda Gonçalves de Oliveira,  
por sua paciência e companhia,  
Aos meus pais,  
por sua infinita dedicação e por tudo o que me  
proveram.



# SUMÁRIO

|                                                                               |           |
|-------------------------------------------------------------------------------|-----------|
| <b>CAPÍTULO 1 - INTRODUÇÃO</b>                                                | <b>9</b>  |
| <b>CAPÍTULO 2 - REDES EM MALHA</b>                                            | <b>11</b> |
| 2.1 Características . . . . .                                                 | 11        |
| 2.2 Implementações de Redes em Malha . . . . .                                | 12        |
| 2.3 Dificuldades na Escolha de Rotas . . . . .                                | 13        |
| <b>CAPÍTULO 3 - MÉTRICAS DE ROTEAMENTO ATUAIS</b>                             | <b>16</b> |
| 3.1 Número de Saltos . . . . .                                                | 17        |
| 3.2 A Métrica <i>Expected Transmission Count</i> . . . . .                    | 18        |
| 3.3 A Métrica <i>Expected Transmission Time</i> . . . . .                     | 21        |
| 3.4 A Métrica <i>Modified Expected Transmission Count</i> . . . . .           | 23        |
| 3.5 A Métrica <i>Effective Number of Transmissions</i> . . . . .              | 23        |
| 3.6 A Métrica <i>Weighted Cumulative Expected Transmission Time</i> . . . . . | 25        |
| <b>CAPÍTULO 4 - NOVAS PROPOSTAS</b>                                           | <b>28</b> |
| 4.1 A Métrica <i>Minimum Loss</i> . . . . .                                   | 28        |
| 4.1.1 Definições . . . . .                                                    | 28        |
| 4.1.2 Implementação . . . . .                                                 | 29        |
| 4.1.3 Críticas à Proposta Apresentada . . . . .                               | 30        |
| 4.2 A Métrica ML com um Fator Aditivo . . . . .                               | 31        |
| 4.3 A Métrica ML com Medidas Mais Precisas . . . . .                          | 32        |
| 4.3.1 Dificuldades de Implementação . . . . .                                 | 33        |

|                                             |           |
|---------------------------------------------|-----------|
| <b>CAPÍTULO 5 - AVALIAÇÃO DO DESEMPENHO</b> | <b>35</b> |
| 5.1 Topologia Externa . . . . .             | 35        |
| 5.2 Topologias Internas . . . . .           | 35        |
| 5.2.1 Rede Interna A . . . . .              | 36        |
| 5.2.2 Rede Interna B . . . . .              | 36        |
| 5.3 Testes com a Métrica ML . . . . .       | 37        |
| 5.3.1 Perda de Pacotes . . . . .            | 38        |
| 5.3.2 Atraso . . . . .                      | 39        |
| 5.3.3 Vazão . . . . .                       | 41        |
| 5.3.4 Estabilidade das Rotas . . . . .      | 42        |
| 5.4 Testes com a Métrica MLAC . . . . .     | 42        |
| 5.4.1 Perda de Pacotes . . . . .            | 43        |
| 5.4.2 Atraso . . . . .                      | 44        |
| 5.4.3 Vazão . . . . .                       | 44        |
| 5.5 Testes com a Métrica MLURP . . . . .    | 46        |
| 5.5.1 Perda de Pacotes . . . . .            | 46        |
| 5.5.2 Atraso . . . . .                      | 47        |
| 5.5.3 Vazão . . . . .                       | 47        |
| <b>CAPÍTULO 6 - CONCLUSÃO</b>               | <b>49</b> |
| 6.1 Trabalhos Futuros . . . . .             | 50        |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS</b>           | <b>51</b> |

## LISTA DE FIGURAS

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Figura 1: Exemplo de topologia de uma rede em malha. . . . .                                                | 11 |
| Figura 2: Exemplo de árvore gerada com base na métrica de menor número de saltos. . . . .                   | 17 |
| Figura 3: Exemplo de uma escolha ruim feita pelo ETX. . . . .                                               | 20 |
| Figura 4: Valores de incremento para algumas janelas. . . . .                                               | 21 |
| Figura 5: Exemplo de topologia com auto-interferência. . . . .                                              | 25 |
| Figura 6: Exemplo de topologia onde o algoritmo de Dijkstra falha com a métrica WCETT. . . . .              | 26 |
| Figura 7: Topologia externa utilizada nos testes. . . . .                                                   | 36 |
| Figura 8: Rede Interna A. Primeira topologia interna utilizada nos testes. . . . .                          | 36 |
| Figura 9: Rede Interna B. Segunda Topologia interna utilizada nos testes. . . . .                           | 37 |
| Figura 10: Perda de pacotes na topologia interna A. Comparativo entre ETX e ML. . . . .                     | 38 |
| Figura 11: Perda de pacotes na topologia externa. Comparativo entre ETX e ML. . . . .                       | 39 |
| Figura 12: Atraso na topologia interna A. Comparativo entre ETX e ML. . . . .                               | 39 |
| Figura 13: Atraso na topologia externa. Comparativo entre ETX e ML. . . . .                                 | 40 |
| Figura 14: Vazão na topologia interna A. Comparativo entre ETX e ML. . . . .                                | 41 |
| Figura 15: Vazão na topologia externa. Comparativo entre ETX e ML. . . . .                                  | 42 |
| Figura 16: Perda de pacotes com a métrica MLAC, variando $\lambda$ para os valores 0, 0,3, 0,7 e 1. . . . . | 43 |
| Figura 17: RTT médio com a métrica MLAC, variando $\lambda$ para os valores 0, 0,3, 0,7 e 1. . . . .        | 44 |
| Figura 18: Vazão média com a métrica MLAC, variando $\lambda$ para os valores 0, 0,3, 0,7 e 1. . . . .      | 45 |

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| Figura 19: Perda de pacotes na rede interna A. Comparativo entre as métricas ML e MLURP. . . . . | 46 |
| Figura 20: Atraso na rede interna A. Comparativo entre as métricas ML e MLURP. .                 | 47 |
| Figura 21: Vazão na rede interna A. Comparativo entre as métricas ML e MLURP. . .                | 48 |

# LISTA DE TABELAS

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| Tabela 1: Estatística da qualidade dos enlaces de uma rede em malha sem fio durante um dia. . . . . | 14 |
| Tabela 2: Valores de vazão (em kbps) obtidos nos testes com a métrica MLAC. . . .                   | 45 |

## CAPÍTULO 1 - INTRODUÇÃO

Redes *ad hoc* são redes sem fio de múltiplos saltos, que têm como características serem auto-configuráveis e suportarem mobilidade. Este tipo de rede se opõe às redes sem fio estruturadas, onde existem pontos de acesso fixos que intermediam toda comunicação dos nós clientes.

Em uma rede estruturada, se um nó cliente deseja enviar uma informação a um outro cliente, os pacotes deverão ser transmitidos diretamente a um ponto de acesso, que por sua vez irá redirecioná-los ao nó de destino. Já em uma rede *ad hoc* não há necessidade desta infraestrutura, já que a transmissão entre dois clientes se dá de forma direta, ou através de múltiplos saltos, onde os nós intermediários são outros clientes da mesma rede, atuando como roteadores encaminhando os pacotes até o destino final.

Claramente as redes *ad hoc* têm custo menor que as estruturadas, já que não é necessária a utilização de pontos de acesso. Além disso, uma rede *ad hoc* é altamente escalável, já que cada cliente é potencialmente um roteador de pacotes para outros nós. Por outro lado, o roteamento neste tipo de rede não é trivial, já que não existe *a priori* um conjunto de nós com a missão de encaminhar pacotes, fazendo com que seja necessário que cada cliente descubra as rotas até o seu destino. Existe ainda a característica da mobilidade, que faz com que as rotas sejam dinâmicas, uma vez que os nós estão potencialmente em movimento.

Neste cenário, surge um novo conceito de rede: as redes em malha ou *mesh*. As redes em malha utilizam comunicação sem fio e de múltiplos saltos, como as redes *ad hoc*. Porém, em redes *mesh*, existe um subconjunto de nós que não apresenta mobilidade. Estes nós atuam como roteadores para os nós clientes, que podem ser móveis. A idéia é que cada roteador da rede tenha diversos enlaces sem fio com alguns outros nós, de forma que a topologia obtida seja uma malha de enlaces. A grande vantagem desta abordagem é o fato de que cada nó tem vários caminhos alternativos para um mesmo destino. A escalabilidade das redes *ad hoc* continua valendo, já que adicionar um novo ponto à malha só faz com que o número de caminhos alternativos cresça.

Embora a mobilidade neste tipo de rede seja menor, a necessidade de protocolos de roteamento continua, já que ela trabalha com múltiplos saltos, além de precisar se adaptar automaticamente à adição de novos nós. Dada a semelhança com as redes *ad hoc*, houve uma utilização

natural de protocolos de roteamento *ad hoc* em redes *mesh*. No entanto, várias particularidades tornam o desenvolvimento de protocolos de roteamento *mesh* desejável.

O objetivo deste projeto é definir as características de uma rede em malha e usá-las para classificar uma rota neste tipo de rede. Em outras palavras, deseja-se estabelecer uma métrica de roteamento, de forma a tirar o máximo proveito das características de uma rede *mesh*. Esta métrica, que tem como objetivo minimizar a perda de pacotes, será chamada de *Minimum Loss* (ML). Além dela, outras duas variações serão propostas ao longo deste trabalho.

Este trabalho está organizado da seguinte maneira: no Capítulo 2 é feita uma análise das características das redes em malha, bem como as principais dificuldades encontradas na tarefa de criar um protocolo de roteamento *mesh*. São mostrados também alguns projetos de implementação deste tipo de rede. No Capítulo 3, é feita uma descrição de algumas métricas que vêm sendo utilizadas neste tipo de rede, com ênfase na métrica ETX [6]. No Capítulo 4 a métrica ML e suas variações são propostas. No Capítulo 5 são mostrados resultados dos testes de desempenho. Finalmente, no Capítulo 6 é feita uma breve conclusão seguida de idéias para trabalhos futuros.

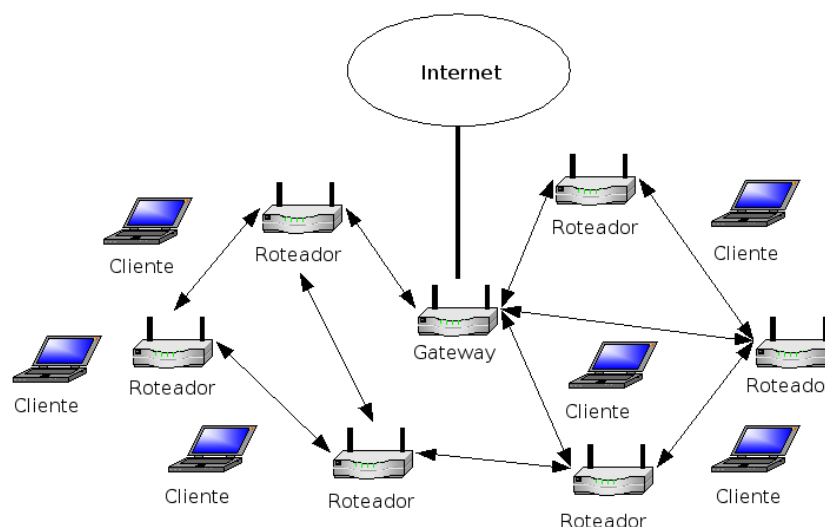
## CAPÍTULO 2 - REDES EM MALHA

Ao longo do Capítulo 1 foi realizada uma breve descrição das redes em malha, na qual algumas das principais características destas redes foram apresentadas. Este capítulo abordará de forma mais profunda o tema, discutindo não só características mas também aplicações e implementações das redes *mesh*.

### 2.1 CARACTERÍSTICAS

Segundo [1], uma rede em malha é formada por um conjunto de nós que pode ser dividido em dois sub-conjuntos: roteadores *mesh* e clientes *mesh*. Enquanto no primeiro grupo, os nós são praticamente estáticos, no segundo pode existir tanto dinamismo e mobilidade quanto esperado em uma rede *ad hoc* tradicional.

Os roteadores *mesh* formam o *backbone* da rede, que pode ser integrada a outras redes através de um ou mais nós específicos atuando como *gateways*. Já os clientes podem ter acesso de forma sem fio, ou conectados diretamente aos roteadores por uma interface *ethernet*, por exemplo. A Figura 1 ilustra uma possível topologia *mesh*.



**Figura 1: Exemplo de topologia de uma rede em malha.**

Nós clientes poderiam, a princípio, ser utilizados como roteadores encaminhando tráfego



de outros nós. No entanto, a maior parte das implementações de redes em malha não explora esta possibilidade. Uma razão para isto é o fato de que, em geral, um roteador *mesh* apresenta configurações específicas, determinadas pelo projeto de cada rede. Como exemplo disso, pode-se citar o protocolo de roteamento. Soluções *mesh* necessitam de roteamento dinâmico e, por isso, tais protocolos são implementados nos roteadores. Porém, para permitir um acesso transparente ao usuário, as soluções não devem exigir ou supor que os clientes utilizem o protocolo. Um outro motivo é a própria classificação dos nós dentro da rede, que atribui apenas aos roteadores a função de encaminhamento de tráfego.

Outra característica marcante das redes em malha é a auto-configuração. A rede deve ser capaz de se adaptar a mudanças causadas pela adição, retirada ou falha de certos nós, por exemplo. Esta característica provê às redes *mesh* três qualidades importantes: desenvolvimento incremental, baixo custo de instalação e tolerância a falhas [3].

A implantação de uma rede em malha pode ser feita de maneira bastante gradual. Roteadores *mesh* podem ser instalados conforme a necessidade e a disponibilidade de recursos. Desta forma, o investimento inicial necessário pode ser consideravelmente reduzido. Outro aspecto em que as redes em malha apresentam baixo custo se refere a instalação de cada nó. A razão para isto é a auto-suficiência dos pontos *mesh*. Apenas um conjunto reduzido de roteadores necessita de uma conexão direta a uma infra-estrutura cabeada.

Em relação à tolerância a falhas, as redes em malha são bastante confiáveis. Quanto maior o número de roteadores *mesh* formando o *backbone*, maior a probabilidade de que existam caminhos alternativos entre cada par de nós da rede. Desta forma, não existe um ponto central de falha e, com isso, os danos causados pelo mal funcionamento de um roteador são bastante localizados. Mesmo o *gateway*, ponto crítico da rede, pode ser replicado. Ou seja, mais de um ponto pode apresentar esta funcionalidade, fazendo com que a rede apresente ainda mais confiabilidade.

Por todas estas características, várias aplicações para este tipo de rede têm sido desenvolvidas. Entre elas, pode-se citar o acesso à Internet, especialmente em programas de inclusão digital, e a utilização como infra-estrutura para aplicações ubíquas.

## **2.2 IMPLEMENTAÇÕES DE REDES EM MALHA**

As implementações de redes em malha podem ser divididas em dois grupos: as soluções abertas e as soluções proprietárias. As soluções abertas englobam projetos de pesquisa e redes comunitárias que, em geral, tem foco no baixo custo de implantação deste tipo de rede. Por

outro lado, as soluções proprietárias ainda apresentam custos muito altos, dificultando o acesso do usuário final a esta tecnologia.

Nas soluções abertas, o *hardware* utilizado para os roteadores *mesh* costuma ser construído a partir de roteadores sem fio comerciais ou adaptando-se PCs para a tarefa. Em termos de protocolos de roteamento, são utilizados protocolos tradicionais de redes *ad hoc*, como o OLSR (*Optimized Link State Routing*) [4], AODV (*Ad-hoc On-Demand Distance Vector*) [15, 14] e DSR (*Dynamic Source Routing*) [9], além de variações destes, propostas em redes de pesquisa, como o Srcr [2] e o HSLS (*Hazy Sighted Link State*) [18].

O OLSR é um protocolo pró-ativo, baseado em estado de enlaces, sendo utilizado pelos projetos VMesh [23] e ReMesh [17, 13, 11]. Já o AODV é reativo e baseia-se em vetor de distâncias. O projeto *MeshCluster* [16] utiliza o AODV-ST (*AODV-Spanning Tree*), uma versão modificada do AODV, que emprega alguns conceitos do DSR.

O projeto RoofNet [2] utiliza o Srcr, um protocolo híbrido que combina informações de estado de enlaces com a técnica de descoberta de rotas do DSR. Finalmente, o HSLS é um protocolo desenvolvido pelo projeto CUWin [7]. Este protocolo, também baseado em estado de enlaces, tem por objetivo minimizar o custo da manutenção de uma visão consistente da rede. Mais detalhes sobre estes protocolos podem ser encontrados em [1].

Já em relação às soluções proprietárias, em geral são utilizados *hardware* e *software* fechados, desenvolvidos especificamente para redes em malha, como no caso das empresas Cisco [19] e Nortel [20].

### 2.3 DIFICULDADES NA ESCOLHA DE ROTAS

Como discutido na Seção 2.1, as redes em malha são uma tecnologia bastante promissora. No entanto, segundo [1], para desenvolverem todo o seu potencial como produto, as redes *mesh* necessitam de pesquisa em áreas relacionadas a todas as camadas da pilha TCP/IP.

Especificamente na área de roteamento, o desenvolvimento de novos protocolos e métricas é fundamental. A adoção de protocolos de roteamento *ad hoc* em redes em malha, embora possível, traz uma série de inconvenientes. Como exemplo, pode-se citar a grande quantidade de pacotes de controle utilizada por tais protocolos. O dinamismo de uma rede *ad hoc* torna necessária a constante avaliação da topologia da rede. Por outro lado, nas redes *mesh* a topologia do *backbone* é praticamente estática. Desta forma, um protocolo de roteamento *mesh* deve ser mais estável e menos custoso à rede.

Em relação às métricas, mesmo em redes *ad hoc*, as soluções existentes enfrentam problemas. Uma das razões é a dificuldade na modelagem deste tipo de sistema, devido a fatores como:

- A variabilidade das taxas de transmissão: os nós podem alterar dinamicamente sua taxa de transmissão, fazendo com que a capacidade de processamento de pacotes varie ao longo do tempo.
- A sensibilidade a interferências externas: perda de pacotes e aumento significativo no atraso podem ocorrer mesmo em roteadores não sobrecarregados, devido a falhas na propagação do sinal de transmissão.

Uma outra consequência deste segundo item é a grande variação de qualidade dos enlaces sem fio. A título de ilustração, a Tabela 1 mostra a média, o desvio padrão, o valor máximo e o valor mínimo obtidos pelo monitoramento da qualidade dos enlaces de uma rede em malha (Rede Interna A, apresentada no Capítulo 5), ao longo de um dia.

**Tabela 1: Estatística da qualidade dos enlaces de uma rede em malha sem fio durante um dia.**

| <b>Enlace</b> | <b>Média</b>  | <b>Mínimo</b> | <b>Máximo</b> | <b>Desvio Padrão</b> |
|---------------|---------------|---------------|---------------|----------------------|
| <b>L1</b>     | <b>9.40</b>   | <b>1.05</b>   | <b>71.30</b>  | <b>7.03</b>          |
| <b>L2</b>     | <b>1.06</b>   | <b>1.00</b>   | <b>1.97</b>   | <b>0.07</b>          |
| <b>L3</b>     | <b>1.12</b>   | <b>1.00</b>   | <b>51.00</b>  | <b>2.07</b>          |
| <b>L4</b>     | <b>10.09</b>  | <b>1.00</b>   | <b>53.12</b>  | <b>8.02</b>          |
| <b>L5</b>     | <b>90.91</b>  | <b>1.00</b>   | <b>451.56</b> | <b>72.11</b>         |
| <b>L6</b>     | <b>1.07</b>   | <b>1.00</b>   | <b>2.21</b>   | <b>0.09</b>          |
| <b>L7</b>     | <b>1.13</b>   | <b>1.00</b>   | <b>13.42</b>  | <b>0.17</b>          |
| <b>L8</b>     | <b>2.40</b>   | <b>1.00</b>   | <b>104.04</b> | <b>4.08</b>          |
| <b>L9</b>     | <b>199.60</b> | <b>1.00</b>   | <b>451.56</b> | <b>180.58</b>        |
| <b>L10</b>    | <b>1.02</b>   | <b>1.00</b>   | <b>1.32</b>   | <b>0.03</b>          |
| <b>L11</b>    | <b>1.07</b>   | <b>1.00</b>   | <b>1.39</b>   | <b>0.06</b>          |
| <b>L12</b>    | <b>1.01</b>   | <b>1.00</b>   | <b>1.24</b>   | <b>0.03</b>          |
| <b>L13</b>    | <b>1.06</b>   | <b>1.00</b>   | <b>2.28</b>   | <b>0.09</b>          |
| <b>L14</b>    | <b>1.05</b>   | <b>1.00</b>   | <b>68.45</b>  | <b>0.42</b>          |
| <b>L15</b>    | <b>1.04</b>   | <b>1.00</b>   | <b>30.44</b>  | <b>0.19</b>          |
| <b>L16</b>    | <b>1.20</b>   | <b>1.00</b>   | <b>451.56</b> | <b>4.19</b>          |
| <b>L17</b>    | <b>6.10</b>   | <b>1.00</b>   | <b>51.00</b>  | <b>3.54</b>          |
| <b>L18</b>    | <b>1.10</b>   | <b>1.00</b>   | <b>141.67</b> | <b>2.16</b>          |
| <b>L19</b>    | <b>2.25</b>   | <b>1.00</b>   | <b>106.25</b> | <b>2.17</b>          |
| <b>L20</b>    | <b>8.21</b>   | <b>1.05</b>   | <b>425.00</b> | <b>6.58</b>          |

Os valores apresentados são relativos a métrica ETX, abordada em detalhes no Capítulo 3. Como será mostrado posteriormente, valores desta métrica acima de 10 indicam um enlace de qualidade bastante ruim. Nota-se, no entanto, que mesmo enlaces cuja média ao longo do dia chegou a 90 ou mesmo 199, em algum instante apresentaram qualidade 1, o melhor valor possível nesta métrica. Nota-se, portanto, que uma avaliação coerente dos enlaces neste tipo de rede é bastante difícil.

Além disso, a avaliação de uma rota como um todo apresenta ainda outros problemas. Por se tratar de um meio de transmissão compartilhado, o envio de dados realizado por um nó afeta todos os seus vizinhos. Ou seja, é necessário que a métrica avalie não só os enlaces de um caminho individualmente, mas também como a interação entre eles se dará.

Todos estes fatores explicam as dificuldades encontradas no roteamento nas rede em malha e justificam as motivações deste trabalho.

## CAPÍTULO 3 - MÉTRICAS DE ROTEAMENTO ATUAIS

Como comentado no Capítulo 1, uma métrica de roteamento pode ser definida como uma maneira de classificar rotas dentro de uma rede, de forma que seja possível decidir o melhor caminho entre dois nós. Em geral, esta tarefa é dividida em duas partes:

1. avaliar os enlaces individualmente e atribuir pesos correspondentes;
2. determinar uma função de composição do peso total de um caminho.

A primeira parte consiste em determinar uma ou mais características de interesse no enlace e como as mesmas podem ser obtidas. Por exemplo, uma característica interessante poderia ser a banda disponível no enlace. Uma maneira de realizar esta verificação seria através de medidas ativas. Ou seja, introduzir tráfego artificial nos enlaces. No entanto, tal abordagem iria interferir no tráfego normal da rede, possivelmente prejudicando os usuários. Outras técnicas menos agressivas também podem ser utilizadas, como a de *packet pair probing*, explicada em maiores detalhes na Seção 3.3. Assim, a viabilidade da implementação se mostra um fator fundamental na proposta de uma métrica.

Em relação à segunda parte, o objetivo é criar uma função que atribua um custo a um caminho considerando os pesos individuais de cada enlace, além de outras informações relevantes. Uma função de composição bastante simples é a soma. Ou seja, o peso de cada enlace da rota em questão é somado, obtendo-se assim o custo total. Outros critérios podem ser utilizados também, como o nível de energia disponível, o que é relevante no caso de dispositivos móveis. Em geral, as funções de composição são criadas com o objetivo de representar alguma grandeza que tenha relação direta com o desempenho, como latência, por exemplo. Desta forma, o melhor caminho passa a ser aquele que minimiza ou maximiza esta grandeza.

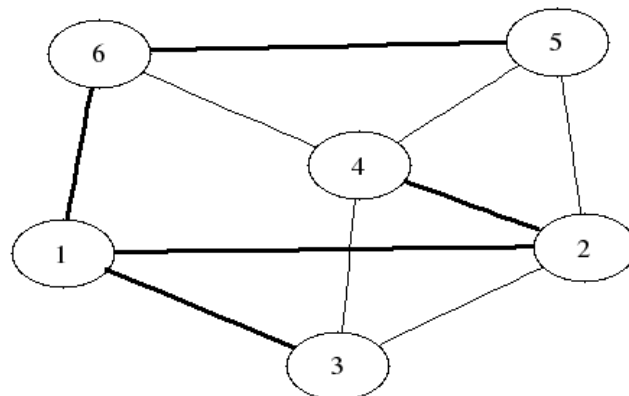
Definindo os critérios destes dois itens, é possível utilizar algoritmos de caminho mínimo clássicos, como o algoritmo de Dijkstra [5]. Entretanto, algumas métricas se utilizam de funções de composição que tornam a utilização de tais algoritmos não-determinística, como será visto a seguir.

Nas próximas seções serão apresentadas algumas das principais propostas atuais em termos de métricas de roteamento para redes sem fio de múltiplos saltos.

### 3.1 NÚMERO DE SALTOS

Esta métrica tem como objetivo minimizar o número de saltos em cada rota. De um outro ponto de vista, pode-se dizer que ela atribui a todos os enlaces o mesmo peso, sem verificar qualquer característica dos mesmos. A única informação necessária para a sua implementação é a da vizinhança de cada nó, o que é facilmente obtido através do envio periódico de pacotes de *broadcast*.

No protocolo OLSR [4] por exemplo, tais pacotes de controle são chamados de *Hello*. Ao receber um pacote deste tipo, um nó pode afirmar que o nó de origem do *Hello* é seu vizinho. A periodicidade de envios permite também que se detecte mudanças na topologia. Além disso, como em algoritmos de estado de enlace (caso do OLSR) os nós precisam ter uma visão completa da topologia, é necessário que todos os nós da rede disseminem a informação da sua vizinhança. No OLSR isto é feito através dos pacotes de *Topology Control* (TC).



**Figura 2: Exemplo de árvore gerada com base na métrica de menor número de saltos.**

A Figura 2 mostra um exemplo desta métrica. As linhas em negrito mostram uma possível árvore formada pelas rotas escolhidas com base na métrica, do nó 1 a todos os demais. Nota-se que, independentemente das condições do enlace, a métrica dá preferência à rota direta entre os nós 1 e 2, por exemplo, mesmo havendo vários outros caminhos alternativos possivelmente melhores em termos de perda de pacotes e vazão.

Se por um lado esta abordagem é bastante simples, levando a baixo *overhead* do sistema e da rede, por outro ela claramente deixa de levar em consideração aspectos fundamentais na sua avaliação. Em redes sem fio, enlaces são afetados por vários fatores, como distância entre os nós, obstáculos e interferências. Desta forma, esta métrica em geral apresenta um desempenho ruim, já que tende a escolher enlaces longos demais, levando a grande perda de pacotes e baixa

vazão.

Para tentar contornar este problema, algumas implementações se utilizam de um limiar de qualidade aceitável. Os enlaces são avaliados por algum critério, como o número de pacotes de controle recebidos com sucesso em uma determinada janela de tempo, e antes de aplicar o algoritmo de caminho mínimo, enlaces com qualidade muito ruim são retirados do grafo. No entanto, as informações coletadas sobre os enlaces não são utilizadas para atribuir pesos às arestas.

Quando utilizada em redes em malha, esta métrica faz com que as rotas se mantenham praticamente estáticas. Como a posição dos nós não se altera com muita frequência, as vizinhanças se mantêm, fazendo com que o algoritmo tenda a escolher sempre os mesmos caminhos. A exceção acontece quando existem enlaces cuja qualidade varia em torno do limiar de aceitação. Neste caso, estes enlaces serão retirados e recolocados no grafo constantemente, podendo causar diferenças nas rotas obtidas.

### 3.2 A MÉTRICA EXPECTED TRANSMISSION COUNT

A métrica *Expected Transmission Count* (ETX), proposta em [6], tem por objetivo aumentar a vazão conseguida na rede. Para isto, os autores sugerem escolher rotas que diminuam o número total de transmissões no nível de enlace, ao longo do caminho.

Dada a probabilidade  $P_{ab}$  de sucesso na transmissão de um pacote pelo enlace entre dois nós  $a$  e  $b$ , o número médio de transmissões necessárias para que este envio ocorra é  $\frac{1}{P_{ab}}$ . Assim, define-se o ETX de um enlace como o inverso da probabilidade de sucesso na transmissão de um pacote através deste enlace. Ou seja:

$$ETX_{ab} = \frac{1}{P_{ab}}. \quad (3.1)$$

Em uma rede com três nós,  $a$ ,  $b$  e  $c$ , se são necessárias duas transmissões para obter sucesso no enlace  $a \rightarrow b$  e três no enlace  $b \rightarrow c$ , então, para um pacote ser enviado com sucesso de  $a$  até  $c$ , passando por  $b$ , são necessárias ao todo 5 transmissões de nível 2. Seguindo este raciocínio, define-se o ETX de um caminho formado por  $n$  enlaces como:

$$ETX_n = \sum_{i=0}^{n-1} \frac{1}{P_{a_i a_{i+1}}}, \quad (3.2)$$

onde  $P_{a_i a_{i+1}}$  denota a probabilidade de sucesso na transmissão de um pacote entre os nós  $a_i$  e

$a_{i+1}$ .

Pelas definições, fica claro que é necessário obter as probabilidades de sucesso dos enlaces para calcular o ETX. Para inferir tais valores, os autores propõem a utilização de pacotes de controle enviados em *broadcast*. Pelo padrão 802.11, tais pacotes não sofrem retransmissão em nível 2, fazendo com que seja possível estimar a probabilidade de sucesso através da seguinte fórmula:

$$P_{ab} = \frac{s_{ab} \cdot s_{ba}}{w^2}, \quad (3.3)$$

onde  $w$  é o tamanho de uma janela de pacotes, e  $s_{ab}$  e  $s_{ba}$  são, respectivamente, o número de pacotes de controle de  $b$  recebidos por  $a$  e o número de pacotes de controle de  $a$  recebidos por  $b$ . Esta janela representa o estado dos  $w$  últimos pacotes que deveriam ter sido recebidos. Nela um pacote pode assumir dois estados: recebido ou não recebido.

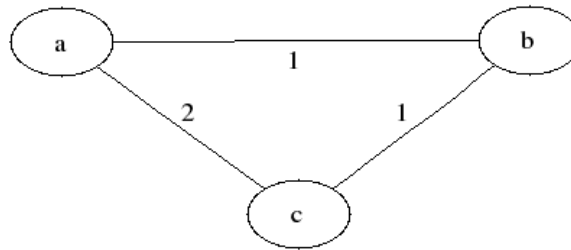
Desta forma, o algoritmo de roteamento precisa apenas enviar pacotes de controle periódicos e fazer a estatística de recebimentos de cada vizinho. Por se tratar de um sistema assíncrono, os pacotes devem apresentar um número de sequência para que seja possível detectar perdas entre dois recebimentos. Além disso, deve-se estipular uma validade para estas mensagens, de forma que a métrica de um enlace decaia caso não sejam recebidos novos pacotes durante um certo intervalo de tempo.

Porém, realizando apenas este procedimento, cada nó  $a$  teria a princípio somente a informação da probabilidade de receber um pacote enviado por um vizinho  $b$  (ou seja,  $\frac{s_{ab}}{w}$ ). Contudo, isto é diferente da probabilidade de sucesso na transmissão de um pacote, pois é necessário também que o reconhecimento (*ack*) seja recebido pelo transmissor (o que no modelo proposto ocorre com uma probabilidade  $\frac{s_{ba}}{w}$ ). Assim, é preciso que cada nó envie para os seus vizinhos o estado de suas janelas, de forma que todos obtenham as informações necessárias para o cálculo do ETX.

Como o significado desta métrica é o número médio de transmissões em nível 2 necessárias para enviar um pacote entre dois nós, o algoritmo de roteamento deve escolher como caminho mínimo aquele com menor valor de ETX dentre todos os possíveis.

Em [6] é mostrado que com esta abordagem o desempenho de uma rede sem fio de múltiplos saltos aumenta consideravelmente em relação à métrica de menor número de saltos. Mesmo assim, é fácil notar que este tipo de abordagem pode levar a escolhas ruins. Por exemplo, dada uma topologia como a da Figura 3, percebe-se que a métrica avalia que a rota  $a \rightarrow c$  é tão boa quanto a rota  $a \rightarrow b \rightarrow c$ . De fato, o protocolo escolherá a primeira opção, por dar preferência





**Figura 3: Exemplo de uma escolha ruim feita pelo ETX.**

ao caminho com menor número de saltos. Porém, o peso 2 no enlace direto representa uma taxa de perda de 50% dos pacotes, enquanto pelo caminho alternativo a probabilidade de perda seria 0.

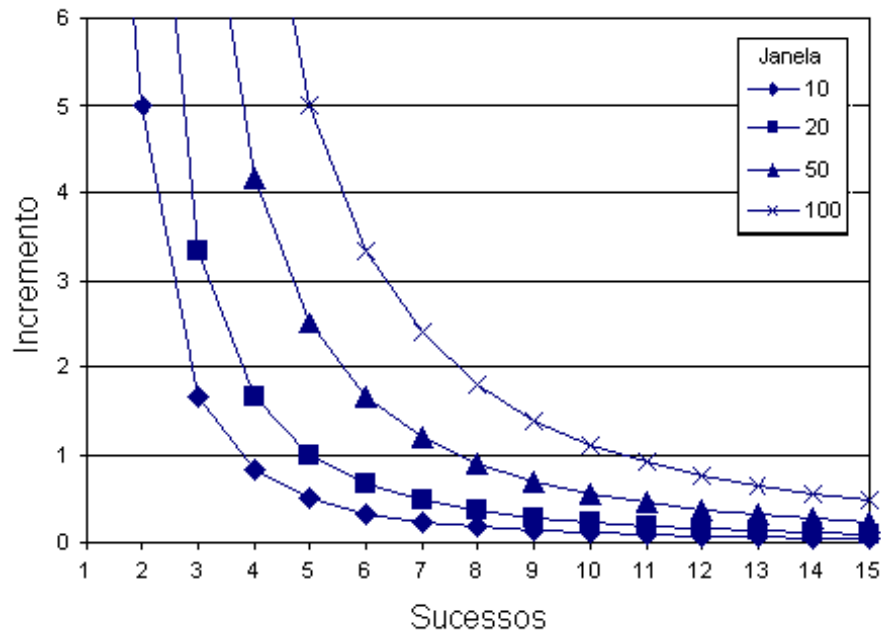
Segundo [6], em uma rede sem fio, existe uma tendência de queda exponencial da vazão em função do número de saltos. Por isso, dar preferência aos caminhos mais curtos é uma boa estratégia do ponto de vista da vazão. No entanto, uma taxa de perda de 50% é muito acentuada, levando à queda do desempenho da rede. Isto é especialmente verdadeiro numa transmissão confiável de dados, utilizando o protocolo TCP.

Esta deficiência pode ser vista sobre um outro enfoque. Para uma janela de tamanho  $w$ , o incremento no valor do ETX de um enlace por pacote perdido em função do número  $r$  de sucessos é dado por:

$$\delta = \frac{w}{r(r-1)}. \quad (3.4)$$

A Figura 4 mostra alguns valores de incremento para janelas de tamanho 10, 20, 50 e 100 pacotes. Este incremento é bem baixo para valores relativamente altos de  $r$ . Apenas para valores mais baixos de  $r$  a função assume valores próximos ou maiores do que 1, o peso mínimo de um enlace nesta métrica. Isto faz com que o ETX associe um custo muito alto à adição de um novo enlace a uma rota, já que são necessárias muitas perdas para que uma rota mais curta seja preterida por outra mais longa.

Um outro problema um pouco mais sutil desta métrica está no fato de que são utilizados pacotes de *broadcast* para inferir as probabilidades. Pelo padrão 802.11, estes pacotes são transmitidos na taxa mais baixa (de 1Mbps). Quanto mais baixa a taxa de transmissão utilizada, mais robusta é a comunicação e menor é a probabilidade de perda do pacote. Portanto, utilizando esta técnica, pode-se fazer uma estimativa muito otimista da qualidade de um enlace.



**Figura 4: Valores de incremento para algumas janelas.**

Por exemplo, suponha dois enlaces,  $i$  e  $j$ , cujos valores de ETX associados sejam 1. Isto significa que à taxa de transmissão de 1Mbps ambos não sofreram perdas. Porém, é possível que a uma taxa maior o enlace  $i$  comece a apresentar perdas, enquanto  $j$  não. Desta forma, a métrica avaliaria ambos da mesma forma, embora o enlace  $j$  seja melhor do que o enlace  $i$ .

### 3.3 A MÉTRICA EXPECTED TRANSMISSION TIME

A métrica *Expected Transmission Time* (ETT) [8] é uma extensão da métrica ETX que procura resolver o problema da superestimação das probabilidades de sucesso. Para isto, é considerada também a taxa de transmissão utilizada, de forma a avaliar com mais precisão a qualidade dos enlaces. A idéia é obter um valor que reflita o atraso do canal, combinando o ETX do enlace com a taxa de transmissão do nó. Desta forma, define-se o ETT de um enlace como:

$$ETT = ETX \cdot \frac{s}{t}, \quad (3.5)$$

onde  $s$  denota o tamanho em bits do pacote de controle utilizado para inferir a métrica ETX e  $t$  representa a taxa de transmissão de dados do canal.

A razão de  $s$  por  $t$  equivale ao tempo de envio de um pacote pelo enlace, enquanto o ETX representa o número médio de vezes que esta transmissão deve ser realizada para que se obtenha

um sucesso. Assim, o produto destas grandezas é uma estimativa da média de tempo efetivo que o pacote precisará para ser enviado. Analogamente ao ETX, o ETT de um caminho composto por  $n$  enlaces é dado pela soma dos valores de ETT de cada enlace. Desta forma, o melhor caminho até um destino será aquele com o menor valor de ETT.

O ETX de um enlace pode ser facilmente calculado através da técnica descrita na Seção 3.2, enquanto o tamanho do pacote de controle é definido pelo próprio protocolo. Porém, obter a taxa de transmissão de dados do canal é um pouco mais complicado, pois boa parte das interfaces de rede não disponibiliza esta informação através do *driver*.

Para contornar este problema, [8] propõe a utilização da técnica de *packet pair probing*. Ela consiste em estimar a taxa através do envio periódico de pares de pacotes: o primeiro é um pacote pequeno, enquanto o segundo é um pouco maior. Supondo que os envios sejam feitos imediatamente um após o outro, basta que o receptor calcule a razão entre o tamanho do segundo pacote e o intervalo entre os recebimentos.

Obviamente não se pode garantir que os envios serão realizados sem um intervalo entre eles. Além disso, é possível que haja retransmissões no nível de enlace, aumentando o tempo registrado pelo receptor. Por isso, os autores sugerem que seja utilizada uma janela de 10 tentativas, onde o intervalo estimado será o menor obtido. Testes mostram que isto é bastante eficiente para taxas relativamente baixas, porém não apresenta a mesma eficácia para as modulações mais altas.

De toda forma, esta proposta não resolve por completo o problema de superestimação do ETX, já que ele continua a ser medido através de pacotes de *broadcast*. Em [2], a implementação proposta contorna este problema, utilizando uma interface capaz de realizar transmissões *broadcast* em diferentes modulações. Periodicamente, são enviados pacotes em cada possível taxa de transmissão. Assim, o ETT de um enlace entre dois nós  $a$  e  $b$  será:

$$ETT_{ab} = \min \left( ETX_{ab}(i) \cdot \frac{S}{t_i} \right), \quad (3.6)$$

onde  $i$  representa cada modulação possível.

Esta abordagem, embora melhore consideravelmente a precisão da avaliação da qualidade dos enlaces, apresenta dois defeitos: é pouco portátil, por depender de uma característica que poucos dispositivos oferecem, e aumenta bastante o tráfego de controle gerado na rede pelo protocolo de roteamento. O problema da portabilidade poderia ser resolvido utilizando-se pacotes *unicast* ao invés de pacotes de *broadcast*. No entanto, isto aumentaria ainda mais o *overhead* na rede, comprometendo definitivamente a escalabilidade do protocolo.

### 3.4 A MÉTRICA MODIFIED EXPECTED TRANSMISSION COUNT

A definição da métrica ETX leva em consideração a probabilidade de sucesso no envio de um pacote no nível de enlace. Segundo [10], isto faz com que o ETX e todas as métricas baseadas nele não consigam captar a variabilidade do canal em escalas de tempo menores do que da transmissão de um pacote.

Em outras palavras, o ETX não consegue captar o que acontece no canal durante o intervalo de tempo de uma transmissão. Assim, uma maneira mais precisa de estimar a qualidade de um enlace seria avaliar a probabilidade de se ter um bit errado na transmissão de um pacote. Desta forma, a estatística levaria em conta escalas de tempo menores e, possivelmente, seria mais acurada. A métrica *Modified Expected Number of Transmissions* (mETX) [10] é definida para um enlace da seguinte forma:

$$mETX_{ab} = e^{(\mu + \frac{1}{2}\sigma^2)}, \quad (3.7)$$

onde  $\mu$  é a média de bits errados por pacote e  $\sigma$  é a variância.

Para estimar estes dois parâmetros, os autores sugerem a utilização de um pacote composto por uma seqüência de bits conhecida. Toda vez que fosse detectado um pacote com erros, os bits errados deveriam ser contabilizados na estatística, através da comparação do que foi recebido com o padrão conhecido.

Embora este seja um resultado teórico bastante interessante, sob o ponto de vista prático, esta métrica encontra sérios obstáculos. Em primeiro lugar, na maior parte das implementações, pacotes recebidos com erros na camada de enlace são descartados sem nenhum tipo de sinalização para as camadas superiores. Além disso, este tipo de verificação poderia representar uma grande complexidade para o sistema.

### 3.5 A MÉTRICA EFFECTIVE NUMBER OF TRANSMISSIONS

Os mesmos autores do mETX [10] sugerem ainda que as perdas de pacotes em um enlace sem fio ocorrem em rajadas. Por exemplo, se um canal tem uma probabilidade de perda de pacotes de 10%, é possível que, de cada 100 pacotes transmitidos, as 10 perdas esperadas ocorram em seqüência. Isto implica no aumento da probabilidade de perda de pacotes nas camadas superiores, como a camada de transporte, pois no 802.11 a camada de enlace tem um limite de tentativas de transmissão (por padrão, 7 vezes). Ou seja, caso as 7 retransmissões de

um *frame* sejam perdidas em rajada, o pacote será considerado definitivamente perdido.

Assim, a métrica *Effective Number of Transmissions* (ENT) [10] de um enlace é definida como:

$$ENT_{ab} = e^{(\mu + 2\delta\sigma^2)}, \quad (3.8)$$

sendo  $\mu$  e  $\sigma$ , respectivamente, a média e a variância de bits errados por pacote e  $\delta$  um parâmetro configurável, relacionado com a taxa de perda tolerável nas camadas acima da camada de enlace. Quanto maior o valor de  $\delta$ , menor é a tolerância a perdas.

A grandeza  $ENT_{ab}$  tem significado estatístico. Dentro do modelo proposto pelos autores, o ENT de um enlace está relacionado à quantidade de perdas em sequência que podem ocorrer dentro de uma rajada. Ao aumentar o valor de  $\delta$ , o modelo se torna cada vez mais pessimista, fazendo com que o tamanho esperado das rajadas também aumente.

Calculado o ENT de cada enlace, a atribuição de pesos é feita da seguinte forma:

$$W_i = \begin{cases} \infty, & \text{se } ENT_{ab} > M \\ ETX_{ab}, & \text{se } ENT_{ab} \leq M, \end{cases} \quad (3.9)$$

onde  $M$  é o limite de tentativas de retransmissão na camada de enlace.

O significado desta expressão é bem simples: se o ENT de um enlace é maior do que o número máximo de retransmissões, então espera-se que durante as rajadas de perda o pacote seja definitivamente perdido (em nível 2). Desta forma, enlaces com esta característica são descartados através da atribuição de pesos infinitos.

Se, por outro lado, o ENT é menor do que o limite de retransmissões, então o enlace é considerado seguro e, portanto, constará do grafo. A atribuição de pesos pode ser feita com qualquer métrica. Entretanto, os autores sugerem e utilizam em seu trabalho a métrica ETX.

Nota-se que o ENT é apenas um pré-processamento que tem por objetivo retirar enlaces cuja probabilidade de perda de pacotes em nível 2 leve a uma quantidade excessiva de perdas nas camadas superiores. Além disso, este pré-processamento diminui a complexidade do algoritmo de caminho mínimo aplicado, já que o número de arestas no grafo considerado é menor do que na rede real.

No entanto, os autores não fazem qualquer tipo de estudo sobre valores interessantes de  $\delta$  para diferentes aplicações. Na realidade, o valor ideal para este parâmetro é o maior possível, tal que o grafo obtido ainda seja conexo. Isto claramente varia de topologia para topologia,

além de ser um valor muito difícil de ser especificado.

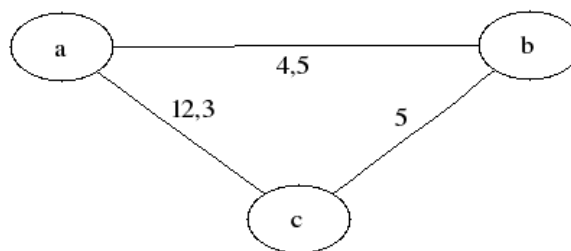
Além disso, a métrica ENT também se vale de estatísticas de bits errados por pacote, assim como a métrica mETX. Isso a torna tão impraticável quanto a primeira proposta, tendo-se em vista as dificuldades apontadas na Seção 3.4.

Uma alternativa seria utilizar a própria métrica ETX para realizar o pré-processamento sugerido. Ou seja, poderia-se definir um parâmetro ajustável  $\beta$ , tal que todo enlace que possua ETX maior do que  $\beta$  seja retirado do grafo. No entanto, esta abordagem apresenta um erro fundamental: em momento algum ela considera a variação da medida de ETX. Segundo [10], quanto maior a variação de um enlace, maior é a probabilidade de ele apresente rajadas de perdas longas. Desta maneira, seria necessário criar um modelo que levasse em conta este parâmetro.

Estes problemas tornam a implementação desta métrica improvável e subjetiva. Assim como o mETX, o ENT é um resultado teórico interessante, porém ainda sem aplicação prática.

### 3.6 A MÉTRICA WEIGHTED CUMULATIVE EXPECTED TRANSMISSION TIME

A métrica *Weighted Cumulative Expected Transmission Time* (WCETT), proposta em [8], é uma modificação da métrica ETT para redes em que são usados múltiplos rádios e diversidade de frequências. Neste tipo específico de rede, o protocolo de roteamento pode tirar proveito de rotas em que os enlaces apresentem alternância nas frequências utilizadas para evitar o problema de auto-interferência.



**Figura 5: Exemplo de topologia com auto-interferência.**

Na topologia apresentada na Figura 5, todos os enlaces se encontram na mesma frequência. Os pesos exibidos são os valores de ETT de cada enlace. Uma transmissão de dados do nó *a* ao nó *c* passará por *b*, já que  $4,5 + 5 < 12,3$ . No início da transmissão, o nó *a* enviará pacotes ao nó *b*, que por sua vez deverá enviá-los ao nó *c*. Porém, enquanto *b* estiver transmitindo, *a* não

poderá enviar mais informações, pois isso iria atrapalhar a comunicação em andamento. Desta forma, existe uma queda na vazão obtida na rede.

Porém, com a utilização de múltiplos rádios, esta situação pode ser evitada. Por exemplo, se na topologia apresentada os enlaces  $a \rightarrow b$  e  $b \rightarrow c$  estivessem em frequências que não interferissem uma com a outra, as transmissões de  $a$  para  $b$  e de  $b$  para  $c$  poderiam ocorrer de forma concorrente.

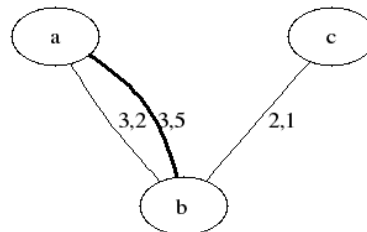
Nenhuma métrica apresentada anteriormente leva em consideração a utilização de múltiplas frequências. E é exatamente desta característica que o WCETT procura se aproveitar. Para isto, define-se o WCETT de um caminho composto por  $n$  enlaces como:

$$WCETT_n = (1 - \beta) \cdot \sum_{i=1}^n ETT_i + \beta \cdot \max(X_j), \text{ para } 1 \leq j \leq k \text{ e } 0 \leq \beta \leq 1, \quad (3.10)$$

onde  $k$  é o número de diferentes frequências disponíveis,  $X_j$  é o somatório dos valores de ETT de todos os enlaces na  $j$ -ésima frequência e  $\beta$  é um parâmetro ajustável.

O primeiro termo na fórmula é simplesmente a própria definição do ETT. Já o segundo tem por objetivo detectar o gargalo da rota em termos de auto-interferência, já que quanto mais bem distribuída pelas frequências disponíveis for uma rota, menor é a probabilidade de que um nó interfira nos demais. Desta forma, o parâmetro  $\beta$  faz o balanceamento entre a utilização da diversidade de frequências e o ETT global do caminho.

Nas comparações mostradas em [8], trabalhando com múltiplos rádios, o WCETT foi muito superior às métricas ETX e de menor número de saltos, demonstrando a importância de se levar em consideração a auto-interferência neste tipo de rede. No entanto, a definição acima não é a ideal, já que algoritmos clássicos para escolha de menor caminho, como o algoritmo de Dijkstra, nem sempre encontram a melhor solução.



**Figura 6: Exemplo de topologia onde o algoritmo de Dijkstra falha com a métrica WCETT.**

Na Figura 6, pode-se notar um exemplo em que isso ocorre. Os nós  $a$  e  $b$  estão conectados

por dois enlaces sem fio em canais diferentes. O enlace no canal 1 (representado pela linha mais fina) apresenta um ETT um pouco melhor do que o canal 2 (representado pela linha em negrito). Desta forma, o algoritmo de Dijkstra escolheria como melhor rota de  $a$  para  $c$  o caminho contendo os dois enlaces no canal 1, enquanto possivelmente a melhor opção seria escolher o enlace no canal 2 entre  $a$  e  $b$ , pois seriam utilizados dois canais diferentes, evitando assim a auto-interferência.

Isto ocorre porque a métrica WCETT não é isotônica [24] e o algoritmo de Dijkstra não garante a melhor solução neste caso. Ainda segundo [24], apenas algoritmos com complexidade exponencial podem ser utilizados nestes casos. Isto se torna um problema, pois o algoritmo de Dijkstra é utilizado por praticamente todas as implementações de protocolos de estado de enlaces. Desta maneira, a utilização da métrica WCETT requer adaptações profundas em protocolos já existentes. Mais que isso, a utilização de algoritmos de complexidade polinomial é desejável, afim de se prover escalabilidades aos protocolos. Uma alternativa seria utilizar protocolos baseados em vetor de distâncias. Entretanto, é inviável implementar o WCETT nestas condições [8], já que é necessário ter informações de todo o caminho para se tomar uma decisão.



## CAPÍTULO 4 - NOVAS PROPOSTAS

Este capítulo tem por objetivo propor alternativas para contornar alguns dos problemas apresentados pelas métricas discutidas anteriormente. Três propostas serão apresentadas: uma métrica multiplicativa, chamada *Minimum Loss* (ML), além de duas extensões. Os resultados de desempenho obtidos serão apresentados e discutidos no Capítulo 5.

### 4.1 A MÉTRICA MINIMUM LOSS

A métrica *Minimum Loss* se baseia em probabilidades de sucesso na transmissão de pacotes no nível de enlace. Como seu nome sugere, a idéia é encontrar rotas que minimizem a probabilidade de perda de um pacote.

#### 4.1.1 DEFINIÇÕES

O peso atribuído a cada enlace  $a \rightarrow b$  será a probabilidade  $P_{ab}$  de que um pacote seja transmitido com sucesso por ele de  $a$  para  $b$ . Como citado anteriormente, este evento pode ser interpretado como a intersecção de dois sub-eventos:

1. um pacote enviado por  $a$  é recebido corretamente por  $b$ ;
2. o *ack* referente ao pacote de dados é recebido corretamente por  $a$ .

Desta forma, a probabilidade  $P_{ab}$  é igual ao produto das probabilidades dos sub-eventos.

Uma vez atribuídos os pesos a todos os enlaces da rede, a melhor rota entre dois nós será aquela com a maior probabilidade de sucesso na transmissão de um pacote fim a fim. Em outras palavras, supondo que as transmissões de um pacote pelos enlaces de uma rota sejam eventos independentes, o melhor caminho entre dois pontos da rede é aquele que apresenta o maior produto dos pesos dos enlaces. Desta forma, define-se o valor  $ML_n$  correspondente ao custo total de uma rota composta por  $n$  enlaces como:

$$ML_n = \prod_{i=0}^{n-1} P_{a_i a_{i+1}}. \quad (4.1)$$

#### 4.1.2 IMPLEMENTAÇÃO

A implementação da métrica ML é bastante semelhante a da métrica ETX e suas variações (ETT e WCETT, por exemplo). A primeira etapa é obter as informações necessárias de cada enlace. Isto pode ser feito através de um conjunto de regras:

- cada nó da rede deverá enviar pacotes de controle periódicos em *broadcast*. Estes pacotes deverão conter um número de seqüência e uma indicação do seu tempo de validade. Este tempo deve obrigatoriamente ser maior do que o intervalo de envio dos pacotes;
- ao receber um pacote deste tipo, o nó deverá atualizar a janela de recebimentos do vizinho correspondente. Para isso, a origem do pacote deve ser identificada. O número de seqüência deve ser verificado para que seja possível identificar pacotes perdidos desde o último pacote de controle recebido;
- se um pacote atualmente na janela tiver seu tempo de validade expirado, ele deve ser imediatamente desconsiderado. Ou seja, a janela deve deslizar em uma posição.

Desta forma, cada nó da rede será capaz de inferir a probabilidade de sucesso da recepção de um pacote enviado por cada um de seus vizinhos. Basta calcular a razão de pacotes recebidos pelo total de pacotes da janela.

No entanto, para calcular o peso dos enlaces é preciso também obter a informação inversa. Cada nó necessita saber qual a probabilidade de um pacote transmitido por ele chegar a cada um dos seus vizinhos. Esta informação é obtida através do estado da janela dos vizinhos. Ou seja, é necessário que cada nó da rede divulgue periodicamente o estado da suas janelas. Assim, como explicado na Seção 4.1.1, basta que, para cada enlace, as duas probabilidades sejam multiplicadas para que se obtenha o peso.

Uma vez associados os pesos aos enlaces, a função de composição definida pela métrica é a multiplicação. Esta etapa da implementação é altamente dependente do tipo de protocolo. Em um protocolo de estado de enlaces, por exemplo, em algum momento é aplicado um algoritmo de caminho mínimo a um grafo que representa toda a topologia. Neste ponto, é aplicada a função de composição. Já em um protocolo baseado em vetor de distâncias, ao receber o vetor de um de seus vizinhos, o nó verifica se deve atualizar suas informações. Para isso, é aplicada a função de composição.

É interessante notar que a implementação da métrica ML difere da implementação da métrica ETX apenas pela função de composição utilizada. Enquanto na métrica ETX é utilizada

uma soma, na ML esta é substituída pela multiplicação. De fato, partindo da implementação de um protocolo que já utilize a métrica ETX, basta substituir o operador da soma pelo da multiplicação no local apropriado.

### 4.1.3 CRÍTICAS À PROPOSTA APRESENTADA

Existem ao menos duas críticas muito importantes a serem feitas sobre a métrica ML, como proposta nas seções 4.1.1 e 4.1.2:

1. Por ter uma implementação muito semelhante a da métrica ETX, a métrica ML incorre em alguns dos mesmos problemas apontados no Capítulo 3, como a imprecisão das estatísticas dos enlaces.
2. Por utilizar a multiplicação como função de composição, é possível encontrar casos em que a adição de novos enlaces a um caminho não altere o custo total.

Em relação ao primeiro item, os problemas advindos destes detalhes de implementação já foram discutidos no Capítulo 3. Porém, no caso da métrica ML, a falta de precisão nas medidas das probabilidades é menos preocupante. Isso porque, em geral, a métrica tenderá a escolher caminhos com mais enlaces, que por sua vez serão, provavelmente, mais curtos e de melhor qualidade. Já na métrica ETX, as rotas são mais curtas, passando por enlaces mais longos nos quais o problema se manifesta de forma mais acentuada.

De toda forma, esta é uma falha conceitual que deve ser corrigida. Na Seção 4.3, será proposta uma extensão que procura evitar este problema.

Quanto ao segundo item, suponha que existam duas rotas entre os nós  $a$  e  $b$  de uma rede: uma direta, por um enlace cujo valor de ML é 0,99, e outra indireta, passando por uma quantidade muito grande de enlaces perfeitos (10, por exemplo). Neste cenário, a métrica escolheria o caminho indireto. No entanto, a grande quantidade de nós intermediários nesta rota faria com que a vazão caísse excessivamente. Além disso, por causa da auto-interferência, mesmo a probabilidade de perda poderia ser maior no caminho indireto do que na rota direta.

Para contornar estes cenários, na Seção 4.2 será apresentada uma outra extensão da métrica ML.

## 4.2 A MÉTRICA ML COM UM FATOR ADITIVO

Uma característica desejável nas métricas de roteamento é que elas não permitam a utilização de valores identidade ou negativos em relação à função de composição como pesos dos enlaces. Em outras palavras, a adição de um novo enlace a uma rota deve obrigatoriamente alterar o peso desta rota, sendo esta alteração sempre no sentido oposto do custo mínimo possível.

Por exemplo, no caso da função de composição ser a soma escalar, a métrica não deve permitir a atribuição de pesos menores ou iguais a zero aos enlaces. Caso contrário, problemas de rotas excessivamente longas ou até mesmo de *loops* infinitos (no caso dos pesos negativos) podem ocorrer.

A proposta original da métrica ML apresenta esta deficiência. Os enlaces podem apresentar peso 1, que em relação a operação de multiplicação é o elemento identidade. Assim, é interessante realizar uma modificação na proposta para que os problemas decorrentes desta abordagem sejam evitados. Para isto, existem duas alternativas.

A primeira consiste em não permitir a utilização do valor 1 como peso dos enlaces. Porém, este valor tem um significado bastante claro na definição proposta. O peso 1 representa o enlace perfeito. Ou seja, aquele em que não ocorrem perdas de pacotes (ao menos dentro da janela considerada). Assim, se torna bastante complicado definir um outro valor coerente com este significado.

A segunda alternativa é atribuir um custo à adição de um novo enlace, independentemente do custo individual do mesmo. Desta forma, define-se a métrica *Minimum Loss with Additive Cost* (MLAC) de um caminho composto por  $n$  enlaces como:

$$MLAC_n = \prod_{i=0}^{n-1} (P_{a_i a_{i+1}} + \lambda), \quad (4.2)$$

onde  $\lambda$  é um parâmetro ajustável menor que 0, representando um custo. A definição dos pesos individuais de cada enlace permanece inalterada.

O significado de  $\lambda$  na expressão acima pode ser definido como o conjunto de vários fatores que influenciam no aumento da probabilidade de perda de pacotes na rede. Estes fatores, como a auto-interferência dos nós em uma rota ou o consumo de recursos da rede, aumentam em função do número de enlaces de um caminho e exercem grande influência na probabilidade de sucesso na transmissão de um pacote.

Nota-se que quantificar tais fatores não é uma tarefa trivial. Na verdade, é provável que o

valor ideal de  $\lambda$  seja condicionado pela topologia da rede, bem como por outros fatores externos, como interferências de outras redes e obstáculos entre os nós. No Capítulo 5 serão mostrados resultados comparativos entre diferentes valores de  $\lambda$ .

Poderia-se propor ainda uma terceira solução para o problema apresentado. O intervalo dos valores possíveis para as probabilidades (que corresponde a  $[0; 1]$ ) poderia ser diminuído para  $[0; 0,5]$ , por exemplo. Para isso, basta que as probabilidades inferidas sejam divididas por 2. Assim, o produto entre os pesos de dois enlaces perfeitos, com peso 0,5, seria menor do que o valor máximo possível, solucionando o problema. Por outro lado, a adição do parâmetro  $\lambda$  à definição da métrica MLAC permite também a interpretação do fato de que as probabilidades utilizadas não são independentes, tornando o modelo proposto mais realista.

Outra observação importante em relação a definição da métrica MLAC está no fato de que a grandeza  $MLAC_n$  pode assumir valores negativos, caso  $\lambda$  apresente um valor absoluto alto. Isso contradiz a noção original de probabilidade da proposta da métrica ML, embora não apresente maiores consequências. Uma definição alternativa que contorna este problema pode ser feita da seguinte maneira:

$$MLAC_n = \left[ \prod_{i=0}^{n-1} \left( \frac{1}{P_{a_i a_{i+1}}} + \lambda \right) \right]^{-1}, \quad (4.3)$$

onde, neste caso, o parâmetro  $\lambda$  pode assumir qualquer valor positivo.

Como será comentado no Capítulo 5, esta última definição foi a utilizada na implementação do protocolo de testes.

### 4.3 A MÉTRICA ML COM MEDIDAS MAIS PRECISAS

Para que o modelo teórico de uma métrica seja propriamente aplicado, é fundamental que as informações obtidas de cada enlace sejam precisas. Como discutido na Seção 4.1 e no Capítulo 3, todas as métricas derivadas da métrica ETX apresentam o mesmo problema neste sentido: elas dependem da informação da probabilidade de sucesso dos enlaces, que em geral, é excessivamente otimista.

Algumas implementações particulares da métrica ETT procuram corrigir esta falha utilizando o envio *unicast* dos pacotes de controle relacionados a medida do ETX. Entretanto, esta prática leva a um grande *overhead* da rede, já que o mesmo pacote de controle precisa ser enviado para cada vizinho individualmente. Além disso, em geral, pacotes *unicast* podem sofrer retransmissões no nível de enlace, prejudicando ainda mais a medida.

A alternativa proposta nesta Seção consiste em continuar utilizando pacotes de *broadcast* para a avaliação das probabilidades. No entanto, deve-se fazer com que as transmissões em *broadcast* sejam realizadas na mesma modulação utilizada pelas comunicações *unicast*.

De fato, pode-se argumentar que esta proposta não é portátil, dado que a modulação utilizada para envios em *broadcast* é um parâmetro não necessariamente encontrado em todas as interfaces de rede. Além disso, mesmo quando disponível, alterar a modulação dos pacotes de *broadcast* não é uma tarefa trivial. No entanto, o objetivo desta proposta no contexto deste trabalho é tentar avaliar o impacto no desempenho causado pelos erros nas medidas de probabilidade feitas pelas métricas atuais.

#### 4.3.1 DIFICULDADES DE IMPLEMENTAÇÃO

Para os testes realizados neste trabalho foi feita uma implementação desta proposta. Durante este processo, pôde ser verificado que existem vários obstáculos, mesmo quando a configuração da modulação de *broadcast* está disponível.

Em primeiro lugar, os nós das redes de testes não apresentavam uma configuração fixa de modulação para transmissão *unicast*. Ao contrário, foi utilizado o algoritmo de seleção automática de taxa disponível na própria interface de acordo com o padrão 802.11. Desta forma, a primeira abordagem de implementação consistia em observar a taxa *unicast* escolhida pela interface e, sempre que houvesse uma mudança, alterar a taxa de *broadcast* para o novo valor.

Esta implementação se mostrou inviável rapidamente. O algoritmo de seleção automática de taxas utilizado tem a tendência de escolher as melhores taxas para a comunicação entre os vizinhos que estão transmitindo mais frequentemente. Isso porque a estatística utilizada por este algoritmo não diferencia pacotes pelo destino. Assim, destinos mais frequentes apresentam peso maior na decisão. Se tais destinos utilizarem enlaces muito bons, o algoritmo escolherá taxas altas, prejudicando a comunicação com os demais vizinhos.

Em termos do protocolo de roteamento, esta abordagem traz problemas em relação a descoberta da topologia. Utilizando pacotes de *broadcast* a taxas muito altas, alguns vizinhos deixam de ser detectados pelo protocolo, já que seus pacotes de controle deixarão de ser recebidos. Nos testes realizados, isto se mostrou um grande obstáculo, pois a topologia se mantinha sempre bipartida. Pequenos *clusters* se formavam, porém nenhum roteador conseguia obter as informações da topologia completa. Este fator torna impossível esta implementação.

Desta forma, foi proposto um segundo método que consistia em fixar tanto a taxa de transmissão *unicast* quanto a *broadcast* em um mesmo valor. Empiricamente, foi notado que a taxa

de 11Mbps poderia ser utilizada por todos os roteadores sem que o problema descrito anteriormente se manifestasse. Esta implementação traz o inconveniente de forçar a utilização de uma taxa não-ótima em todos os roteadores.

Uma terceira opção levantada foi uma mescla das duas primeiras. Ao mesmo tempo em que seria utilizada a seleção automática de taxas *unicast*, a taxa de transmissão em *broadcast* acompanharia a seleção automática da interface apenas até os 11Mbps. Ou seja, caso a taxa selecionada pelo algoritmo fosse menor ou igual a 11Mbps, a modulação para transmissões em *broadcast* seria configurada para o mesmo valor. Se, por outro lado, a taxa selecionada fosse maior do que 11Mbps, a implementação manteria a taxa fixada neste valor.

Assim como a primeira, esta opção foi descartada, tendo em vista que ela não resolvia o problema, embora amenizasse seus efeitos. Vários nós da rede freqüentemente utilizavam taxas de 36Mbps ou 54Mbps, modulações bem menos robustas do que o máximo definido para a taxa de *broadcast*. Optou-se, portanto, pela utilização da segunda proposta. Ou seja, fixar em todos os nós tanto a taxa de transmissão em *unicast* quanto em *broadcast* para 11Mbps. No Capítulo 5, quando serão exibidos os resultados obtidos, esta métrica será referenciada como *Minimum Loss with Unicast Rate Probing* (MLURP).

## CAPÍTULO 5 - AVALIAÇÃO DO DESEMPENHO

Para avaliar o desempenho das métricas propostas foi realizada uma série de testes utilizando a estrutura disponível no projeto ReMesh. Para isso, foram utilizadas três topologias distintas, uma externa e duas internas, formadas por roteadores *Linksys* WRT54G. O *firmware* original destes roteadores foi substituído por outro chamado OpenWrt [12], uma distribuição Linux bastante reduzida criada especificamente para este tipo de dispositivo.

A adoção de um *firmware* baseado em Linux permitiu a utilização de um protocolo de roteamento, bem como de diversas ferramentas de medição e monitoramento da rede, além de abrir um grande leque de configurações da interface de rede. As próximas seções descrevem em maiores detalhes as topologias utilizadas nos testes.

### 5.1 TOPOLOGIA EXTERNA

A topologia externa é formada por 5 roteadores colocados no topo de prédios de alunos, professores e funcionários da Universidade Federal Fluminense, além de um roteador colocado no topo de um dos prédios da Universidade, funcionando como *gateway* da rede. A Figura 7 mostra o posicionamento dos nós da rede, bem como a malha de enlaces formada entre eles.

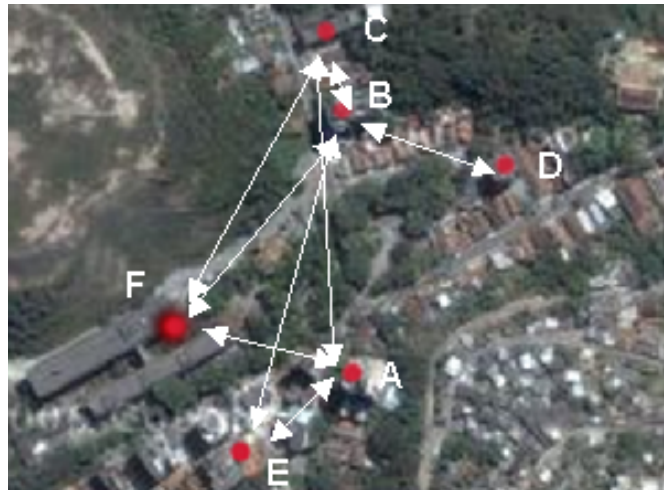
A antena original dos roteadores foi substituída por uma antena omni-direcional externa de 19dB ou, no caso do *gateway*, por uma antena direcional de 24dB.

Por se tratar de um ambiente externo, esta rede está suscetível a grandes variações causadas por mudanças climáticas, deslocamento das antenas pelo vento, entre outros fatores.

### 5.2 TOPOLOGIAS INTERNAS

Os testes de rede interna são baseados em uma rede montada ao longo de dois andares de um dos prédios da Escola de Engenharia da Universidade Federal Fluminense. Os nós foram espalhados por salas de professores e laboratórios. Ao longo do período em que foram realizados os testes, a rede interna teve que ser alterada por necessidades do projeto ReMesh. Desta forma, foram utilizadas duas topologias distintas, embora bastante semelhantes.

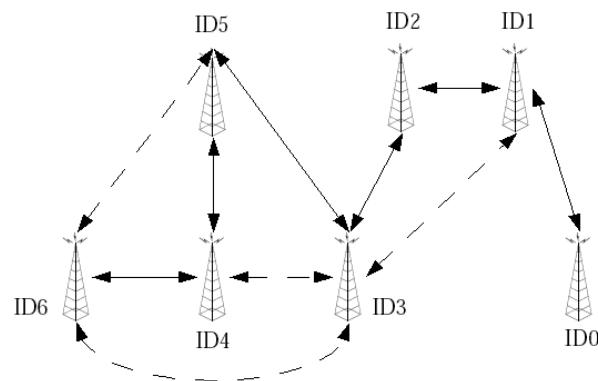




**Figura 7: Topologia externa utilizada nos testes.**

### 5.2.1 REDE INTERNA A

Esta topologia é formada por 7 roteadores, como mostrado na Figura 8. Através do uso desta rede, pode ser observada uma rota preferencial entre os dois extremos (os nós 0 e 6). Os enlaces que fazem parte desta rota estão representados pelas linhas contínuas. Já os enlaces secundários são mostrados através de linhas pontilhadas.

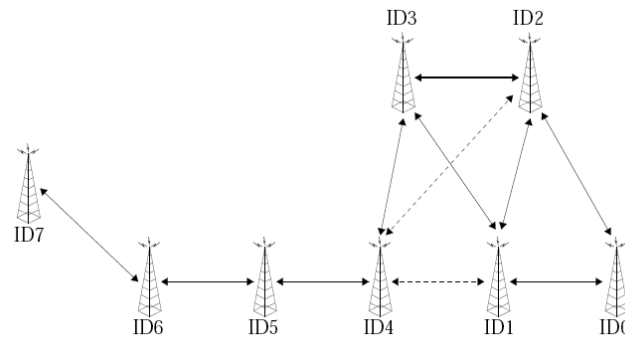


**Figura 8: Rede Interna A. Primeira topologia interna utilizada nos testes.**

### 5.2.2 REDE INTERNA B

A segunda topologia interna é formada por 8 nós. Uma representação da distribuição dos roteadores ao longo do prédio pode ser vista na Figura 9. Mais uma vez, os melhores enlaces foram destacados. Na figura, eles são representados pelas linhas contínuas, enquanto enlaces

de qualidade mais baixa são representados por linhas pontilhadas.



**Figura 9: Rede Interna B. Segunda Topologia interna utilizada nos testes.**

Por ser localizada no interior de um prédio, a rede interna não está exposta às mesmas variações da rede externa. No entanto, vários fatores prejudicam a comunicação entre os roteadores, como paredes, elevadores, outras redes na mesma faixa de frequência e até mesmo as pessoas que trabalham ou freqüentam o prédio.

Nas próximas seções, serão apresentados e discutidos os resultados obtidos nos testes de cada proposta apresentada no Capítulo 4.

### 5.3 TESTES COM A MÉTRICA ML

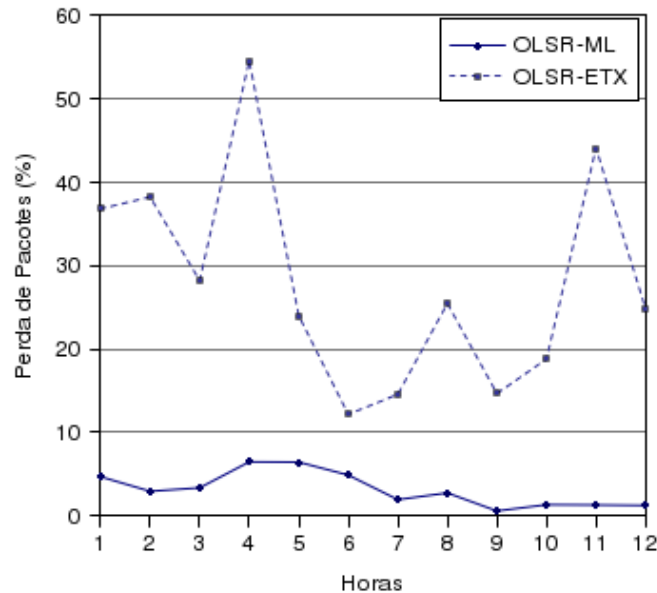
Para avaliar o desempenho da métrica ML foram realizados testes comparativos com a métrica ETX. Estes testes foram realizados na topologia externa e na rede interna A. Ambas as métricas foram testadas utilizando-se a implementação do protocolo OLSR de olsr.org [22]. Esta implementação já apresenta a opção de se utilizar a métrica ETX. Portanto, foi necessário implementar apenas a métrica ML, de acordo com as definições apresentadas no Capítulo 4.

Vale ressaltar que na rede externa o único teste executado sem a concorrência de usuários foi o de vazão. Isso porque esta é a rede de produção do projeto e, por isso, procurou-se evitar retirar os usuários da rede. Já na topologia interna, antes de cada teste, o acesso dos usuários era previamente bloqueado.

A seguir, são detalhados os testes realizados, bem como os resultados obtidos.

### 5.3.1 PERDA DE PACOTES

Os testes de perda de pacotes foram feitos utilizando a ferramenta *ping* do Linux. As Figuras 10 e 11 mostram os resultados obtidos nas topologias interna e externa, respectivamente.

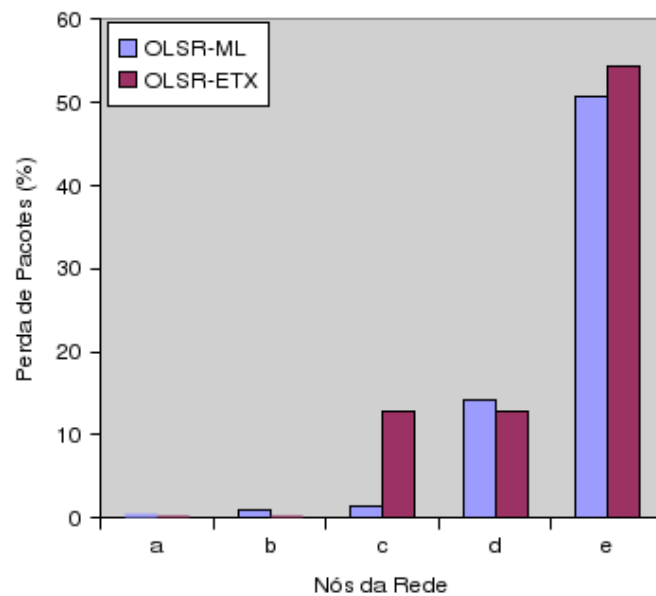


**Figura 10: Perda de pacotes na topologia interna A. Comparativo entre ETX e ML.**

No caso da topologia interna, o teste teve duração de 12 horas. A cada segundo, um *ping* era enviado de uma ponta a outra da rede (do nó 0 ao nó 6). Ao final de cada hora, a porcentagem de pacotes perdidos era calculada. Desta forma, pode-se observar a evolução da taxa de perda ao longo do dia. Como esperado, a Figura 10 mostra que a métrica ETX apresentou taxas de perda bem maiores durante todo o teste.

Quanto à topologia externa, foram realizadas 4 medições por dia, de 25 minutos, do *gateway* para cada nó, durante 2 dias. A Figura 11 mostra que em 4 dos 5 casos apresentados, os resultados das duas métricas foram bem similares. Porém, em relação ao nó *c*, a métrica ML apresenta uma grande vantagem.

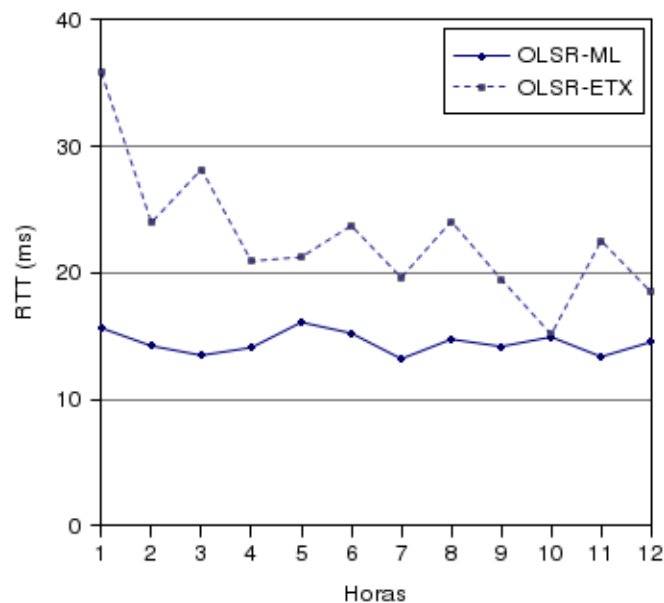
Este nó pode se comunicar com o *gateway* por 2 caminhos: diretamente, através de um enlace de qualidade mediana, ou passando pelo nó *b*, utilizando 2 bons enlaces. Neste caso, freqüentemente a métrica ETX dá preferência à rota direta, enquanto a métrica ML escolhe o caminho indireto. Desta forma, a probabilidade de perda de pacotes aumenta bastante com o uso da métrica ETX.



**Figura 11: Perda de pacotes na topologia externa. Comparativo entre ETX e ML.**

### 5.3.2 ATRASO

Para as medições de atraso foram utilizadas as saídas dos mesmos testes realizados para computar a perda de pacotes. Os valores de atraso considerados aqui na verdade são os valores do *Round-Trip Time* (RTT) reportados pela ferramenta *ping*. Ou seja, englobam o tempo de ida, processamento e volta do pacote ICMP.



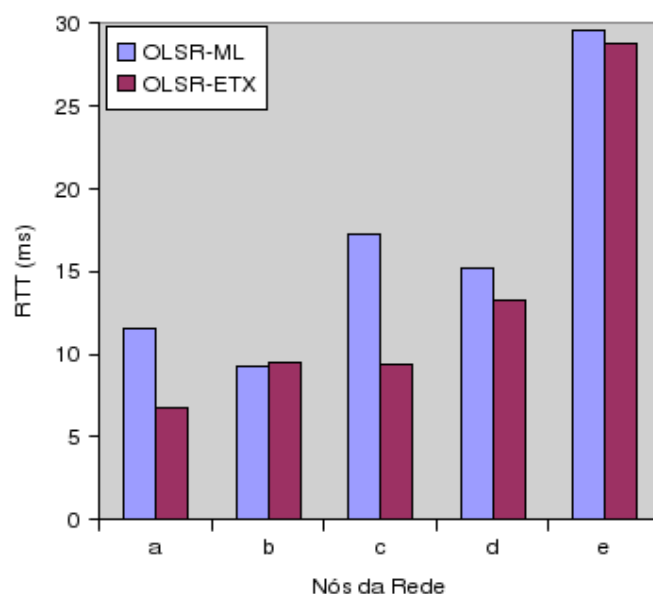
**Figura 12: Atraso na topologia interna A. Comparativo entre ETX e ML.**

No caso da rede interna, cujos resultados estão representados no gráfico da Figura 12, foi calculado o RTT médio de cada hora do teste. De forma surpreendente, o RTT médio com a métrica ETX foi sempre maior.

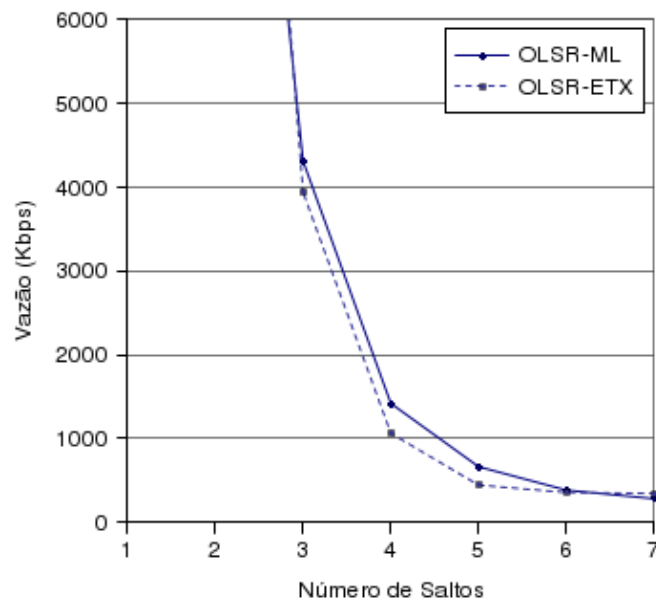
Uma possível explicação está no fato de que as probabilidades inferidas por esta métrica não são totalmente confiáveis, como discutido no Capítulo 3, o que pode levar a um número maior de retransmissões em nível 2 nas rotas escolhidas. Este resultado é bastante interessante, porque reforça as suspeitas de que os erros cometidos na inferência das probabilidades tenham um impacto significativo no desempenho da métrica. Já em relação à métrica ML, embora sofra da mesma imprecisão nas medidas, ela tende a escolher caminhos mais seguros. Ou seja, em geral, os enlaces que compõem a rota escolhida apresentam poucas perdas, acarretando em um baixo número de retransmissões na camada de enlace.

Outra possibilidade está relacionada ao modelo da métrica ETX que não considera o *backoff* exponencial. Quando um pacote é perdido na camada de enlace, existe um tempo de espera antes do reenvio do mesmo. Este tempo é escolhido aleatoriamente dentro de um intervalo que cresce exponencialmente a cada retransmissão. Desta forma, a cada retransmissão de um mesmo pacote realizada por um nó, o tempo será possivelmente maior. No entanto, a métrica ETX atribui o mesmo peso a todas as transmissões de nível 2, o que na realidade não ocorre.

Na topologia externa, os valores apresentados são relativos a média de todo o teste, para cada nó de destino da rede. Pelo gráfico mostrado na Figura 13, nota-se claramente que neste cenário a métrica ETX leva vantagem, escolhendo rotas mais curtas.



**Figura 13: Atraso na topologia externa. Comparativo entre ETX e ML.**



**Figura 14: Vazão na topologia interna A. Comparativo entre ETX e ML.**

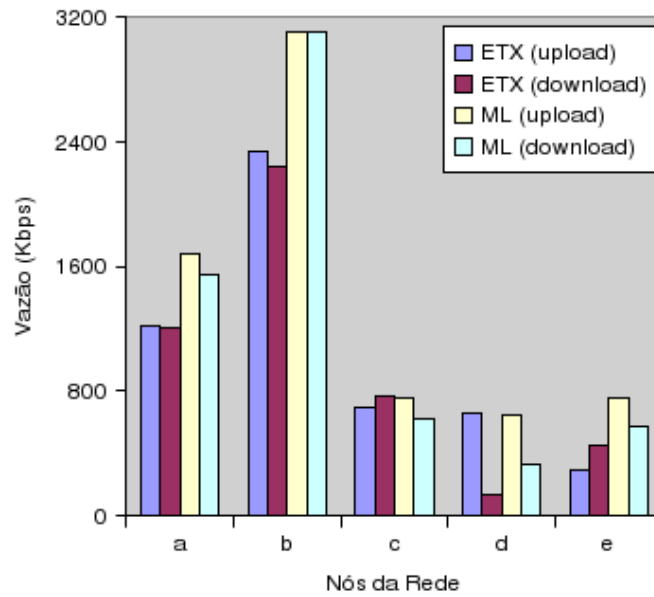
### 5.3.3 VAZÃO

Para medir a vazão obtida pelas métricas foi utilizado o programa *iperf* [21]. Nas duas topologias os testes foram realizados sobre transmissões confiáveis de dados utilizando o protocolo TCP.

Na topologia interna, transferências de 5 minutos foram realizadas do nó 0 a cada um dos outros nós. O gráfico da Figura 14 mostra os resultados obtidos. Nele, foram considerados os números de saltos até cada roteador, referentes à rota preferencial entre os nós 0 e 6, destacada na Figura 8.

Neste caso, as abordagens se mostraram equivalentes: de um lado rotas menores, levando a menos auto-interferência, enquanto do outro rotas mais seguras, sofrendo menor perda de pacotes.

Já na rede externa, os resultados apresentados são referentes a 14 dias de medições com cada métrica, sendo 4 medições de 5 minutos por dia. Os gráficos da Figura 15 mostram uma certa tendência de melhora com a métrica ML. Esta melhora se deve possivelmente às menores taxas de perda de pacotes, como mostrado na Seção 5.3.1, já que se trata de um teste utilizando o protocolo TCP.



**Figura 15: Vazão na topologia externa. Comparativo entre ETX e ML.**

#### 5.3.4 ESTABILIDADE DAS ROTAS

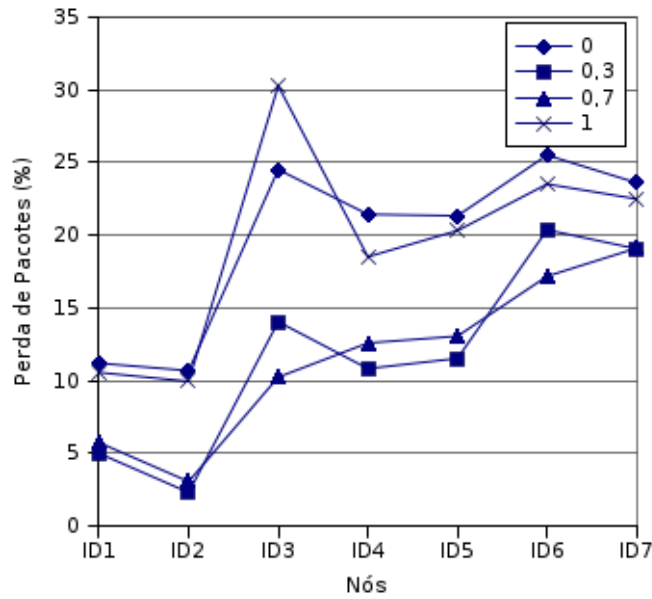
Utilizando a opção  $-R$  da ferramenta *ping*, que mostra os primeiros 9 saltos utilizados pelo pacote, pôde-se constatar que durante um teste de uma hora a métrica ETX realizou mais de 400 trocas de rotas, enquanto a métrica multiplicativa utilizou apenas uma rota.

### 5.4 TESTES COM A MÉTRICA MLAC

Como discutido no Capítulo 4, foram realizados testes comparativos com alguns valores do parâmetro  $\lambda$  da definição da métrica MLAC, relacionado ao custo da adição de um enlace a uma rota. Para estes testes, foi utilizada a rede interna B. Mais uma vez, a implementação foi realizada sobre o protocolo OLSR. Para simplificá-la, foi utilizada a segunda definição proposta da métrica MLAC, apresentada na equação 4.3.

Para efeito dos testes, foram utilizados quatro valores diferentes de  $\lambda$ : 0; 0,3; 0,7 e 1. A escolha destes valores teve como critério a distribuição linear de amostras no intervalo de 0 a 1. Vale ressaltar que quando  $\lambda$  é zero, a métrica MLAC se torna idêntica a métrica ML.

Para avaliar o desempenho de cada um destes valores, foram observados três parâmetros diferentes: o índice de perda de pacotes, o atraso e a vazão utilizando o protocolo TCP. Os resultados obtidos são mostrados nas próximas seções.



**Figura 16: Perda de pacotes com a métrica MLAC, variando  $\lambda$  para os valores 0, 0,3, 0,7 e 1.**

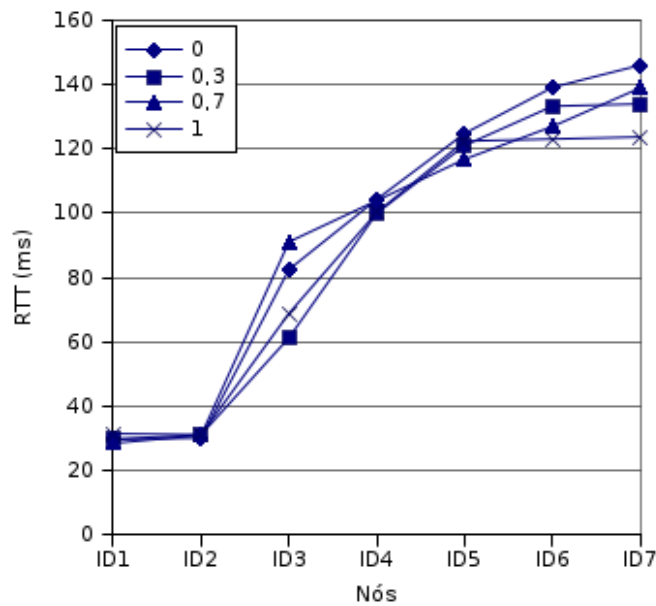
#### 5.4.1 PERDA DE PACOTES

Os testes de perda de pacotes foram realizados novamente através da ferramenta *ping*. Durante um dia, foram realizados 36 experimentos de 60 segundos com cada métrica. Cada experimento consistia do envio de 600 *pings* (1 a cada 100 ms) de 1024 bytes. Os pacotes foram originados de um computador conectado ao *gateway* da rede e os testes foram realizados para todos os demais nós.

Para procurar amenizar o efeito da variação da qualidade da rede e assim permitir uma comparação entre os dados, os experimentos foram realizados de forma intercalada. Ou seja, para uma dada métrica, foram realizados 7 experimentos consecutivos, um para cada nó de destino. Em seguida, a métrica era alterada e mais uma sequência de 7 experimentos era realizada. O gráfico da Figura 16 mostra a média do percentual de pacotes perdidos nos experimentos em função do nó de destino.

Pode-se notar que para os valores intermediários (0,3 e 0,7) os índices de perda foram consideravelmente menores para todos os nós de destino. As maiores taxas de perdas de pacotes obtidas pelo parâmetro 1 podem ser justificadas pela tendência de escolha de caminhos muito curtos, como na métrica *Hop Count*, fazendo uso de enlaces longos demais, sujeitos a uma pior qualidade. Por outro lado, no caso do parâmetro 0, os índices podem ser explicados pela opção da métrica por caminhos mais longos, compostos de um número maior de enlaces, gerando maior auto-interferência na rota.





**Figura 17: RTT médio com a métrica MLAC, variando  $\lambda$  para os valores 0, 0,3, 0,7 e 1.**

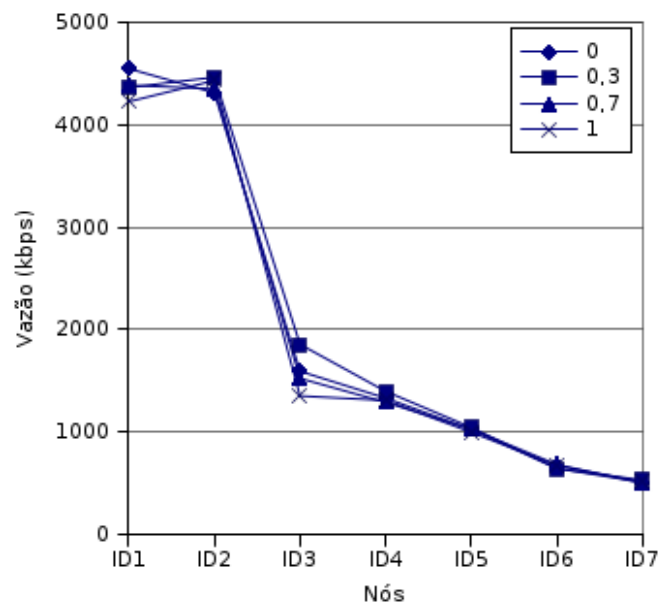
#### 5.4.2 ATRASO

Para as medidas de atraso, novamente foram utilizados os mesmos testes realizados para obter os dados de perda de pacotes. Foi calculada uma média dos valores de RTT reportados para cada pacote de *ping* enviado com sucesso. Vale ressaltar que, como as taxas de perda diferem entre os valores de  $\lambda$  e entre os destinos, o número de amostras utilizadas para cada resultado apresentado é diferente.

A Figura 17 mostra um gráfico do RTT médio obtido para cada valor de  $\lambda$  em função do nó de destino do teste. Pode-se observar que o desempenho de todas as métricas foi bastante similar. Apenas nos testes para os nós 5, 6 e 7 pôde-se notar uma tendência de atraso maior com a utilização do parâmetro  $\lambda = 0$ . Este resultado é bastante coerente com a teoria, que prevê a adoção de caminhos com um número maior de enlaces neste caso.

#### 5.4.3 VAZÃO

Para as medidas de vazão, foi utilizada a ferramenta *iperf*. Estes testes foram realizados de forma bastante similar aos testes apresentados nas duas seções anteriores. Ou seja, para cada nó de destino da rede, exceto o *gateway*, foram realizados 36 experimentos com cada uma das 4 variações propostas. Cada experimento consistiu de uma transferência confiável de dados sobre o protocolo TCP, durante 60 segundos. De maneira análoga aos testes realizados para perda de pacotes e latência, os experimentos ocorreram de forma intercalada.



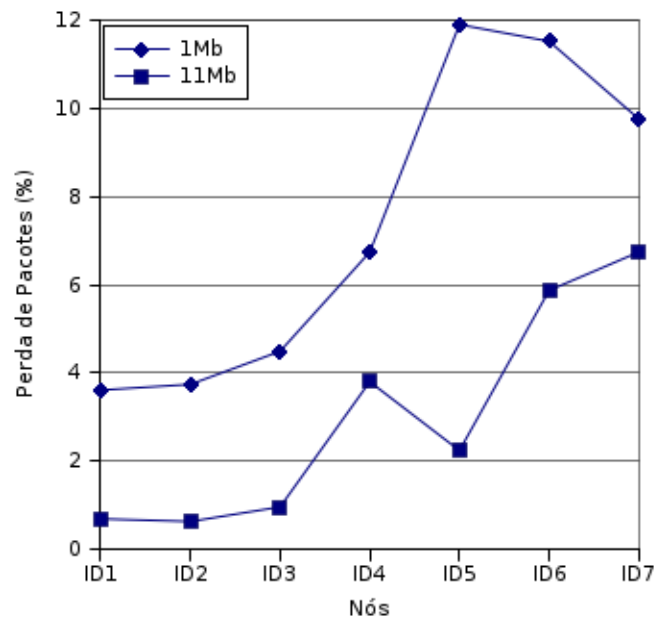
**Figura 18: Vazão média com a métrica MLAC, variando  $\lambda$  para os valores 0, 0,3, 0,7 e 1.**

A Figura 18 mostra a média dos resultados obtidos em cada experimento, em função do parâmetro  $\lambda$  e do roteador de destino. De forma surpreendente, os resultados foram bastante próximos. Dada a proximidade dos resultados obtidos para o atraso e a considerável diferença nos índices de perda de pacotes, era esperado que os resultados de vazão obtidos para os parâmetros 0,3 e 0,7 fossem melhores do que para 0 e 1. Para uma melhor visualização destes resultados, a Tabela 2 mostra os valores exatos.

**Tabela 2: Valores de vazão (em kbps) obtidos nos testes com a métrica MLAC.**

| $\lambda$ | ID1     | ID2     | ID3     | ID4     | ID5     | ID6    | ID7    |
|-----------|---------|---------|---------|---------|---------|--------|--------|
| 0         | 4547,75 | 4311,79 | 1593,85 | 1332,1  | 1026,4  | 646,03 | 508,93 |
| 0,3       | 4363,43 | 4463,88 | 1856,62 | 1398,25 | 1045,09 | 643,19 | 531,69 |
| 0,7       | 4385,79 | 4350,38 | 1524,47 | 1290,63 | 1028,25 | 683,03 | 501,03 |
| 1         | 4225,46 | 4431,99 | 1352,55 | 1309,61 | 998,67  | 674    | 513,4  |

Nota-se que para  $\lambda = 0,3$  foram obtidos os melhores resultados em relação aos nós 2, 3, 4, 5 e 7. No entanto, apenas para o nó 3 a diferença para a segunda melhor variação foi considerável (mais de 200 kbps).



**Figura 19: Perda de pacotes na rede interna A. Comparativo entre as métricas ML e MLURP.**

## 5.5 TESTES COM A MÉTRICA MLURP

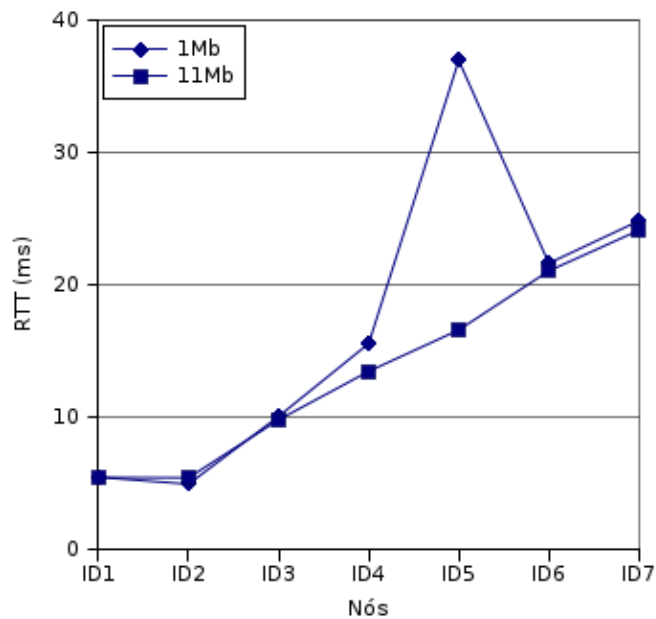
Os testes com a métrica MLURP foram realizados de forma bastante similar aos da métrica MLAC. Porém, como explicado no Capítulo 4, neste caso todos os roteadores tiveram sua taxa de transmissão para pacotes *unicast* fixada em 11 Mbps.

Para efeito de comparação, os testes foram realizados utilizando-se experimentos com a taxa de transmissão em *broadcast* de 11Mbps e de 1Mbps. Ou seja, o desempenho da métrica MLURP foi comparado com o da métrica ML.

### 5.5.1 PERDA DE PACOTES

Os experimentos realizados para inferir o índice de perda de pacotes são idênticos aos descritos na Seção 5.4.1. A diferença nos testes fica por conta apenas do número de variações avaliadas. Enquanto na Seção 5.4.1 foram comparadas 4 variações, aqui são consideradas apenas duas.

A Figura 19 mostra os resultados obtidos. Claramente a métrica MLURP foi melhor do que a métrica ML. Em relação a todos os nós da rede, os índices de perda obtidos com a métrica MLURP ficaram sempre consideravelmente abaixo dos da métrica ML. Este resultado reforça a suspeita de que os impactos causados pelos erros nas medidas das probabilidades, comentados



**Figura 20: Atraso na rede interna A. Comparativo entre as métricas ML e MLURP.**

nos Capítulos 3 e 4, são realmente grandes.

### 5.5.2 ATRASO

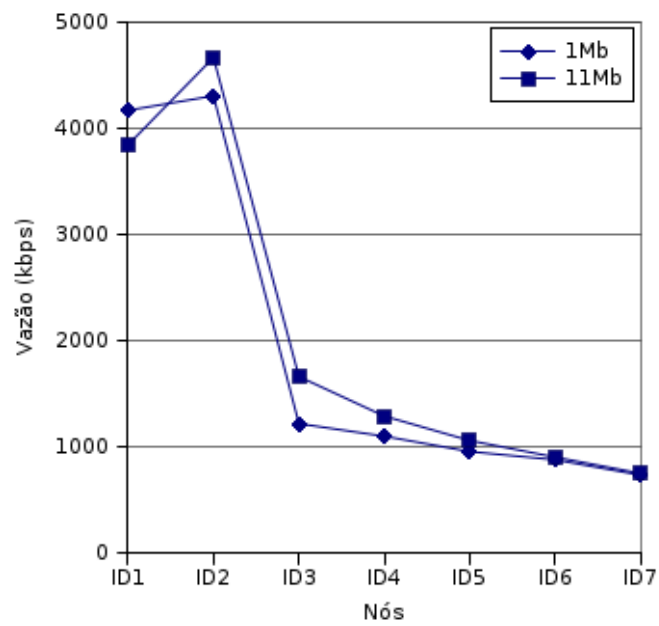
Também para esta medida de atraso, foram considerados os mesmos testes utilizados na Seção 5.5.1. Assim como nos testes anteriores, foi calculada a média dos valores de RTT reportados para cada *ping* transmitido com sucesso. A Figura 20 mostra os resultados obtidos.

Em termos de latência, os resultados das duas métricas foram bastante semelhantes. Este resultado não surpreende, já que o objetivo da métrica MLURP é minimizar a perda de pacotes, e não o atraso fim a fim.

### 5.5.3 VAZÃO

Os testes de vazão executados com a métrica MLURP foram realizados como explicado na Seção 5.4.3. No entanto, para estes testes foram comparadas apenas duas métricas.

Como mostrado no gráfico da Figura 21, em termos de vazão, a métrica MLURP não foi tão superior à métrica ML quanto em relação à perda de pacotes. No entanto, é notável que apenas em relação ao nó 1, a métrica MLURP foi pior. Nos outros casos, em especial para os nós 2, 3 e 4, pode-se notar uma tendência de melhora utilizando-se a abordagem proposta. Vale destacar que os testes foram realizados em momentos diferentes, embora muito próximos.



**Figura 21: Vazão na rede interna A. Comparativo entre as métricas ML e MLURP.**

Desta forma, é difícil obter uma comparação justa, especialmente para os nós menos distantes (que apresentam menor número de saltos até o *gateway*).

## CAPÍTULO 6 - CONCLUSÃO

Com a popularização das rede em malha, a necessidade do desenvolvimento de novas tecnologias para este tipo de rede se faz presente. As características particulares destas redes têm sido estudadas, porém ainda não foram completamente exploradas. Isto faz com que as redes em malha não desenvolvam todo seu potencial, em termos de desempenho e serviços oferecidos. Tendo em vista a diversidade de projetos que utilizam estas redes para promover a inclusão digital, este tipo de estudo torna-se ainda mais importante.

Neste trabalho procurou-se discutir uma das muitas áreas de pesquisa sobre este tipo de rede: as métricas de roteamento. Elas são de fundamental importância em redes em malha e redes *ad hoc*, pois é necessário estar constantemente avaliando os enlaces e rotas, sem, no entanto, interferir no desempenho da rede.

Foi realizada uma pesquisa com as principais propostas atuais nesta área. Cada métrica estudada foi apresentada de forma crítica, tendo suas vantagens e desvantagens comentadas. O aspecto de implementação também foi abordado. Em especial, foi analisada a métrica *Expected Transmission Count* (ETX), uma das mais utilizadas em implementações de redes em malha no mundo e base para diversas outras propostas da literatura.

Além deste estudo, foi proposta uma métrica multiplicativa, chamada *Minimum Loss*, com o objetivo de minimizar a perda de pacotes em uma transmissão fim a fim. Com base nesta métrica, foram propostas também duas variações que procuram corrigir deficiências da definição original. Uma destas extensões tem por objetivo aumentar a precisão das estatísticas de qualidade dos enlaces utilizadas pela métrica ML. No entanto, a mesma técnica proposta é aplicável a outras métricas, como a métrica ETX e outros trabalhos derivados.

Em termos de resultados, procurou-se demonstrar a validade da utilização de métricas multiplicativas em certas topologias de redes em malha sem fio. A avaliação de desempenho da métrica ML mostrou, em vários casos, melhora da utilização da rede em termos de perda de pacotes, estabilidade de rotas, atraso e, como consequência, vazão.

Outro resultado interessante foi o obtido com a métrica MLAC, a segunda proposta deste trabalho. Os testes mostram que variando o fator aditivo  $\lambda$ , é possível obter desempenho superior ao da métrica ML original, o que está de acordo com a teoria proposta.

Por fim, os testes realizados com a última proposta deste trabalho, a métrica MLURP, demonstram o quanto as falhas nas medidas de probabilidade afetam o desempenho das métricas baseadas neste tipo de estatística. Em especial, os testes de perda de pacotes, índice que as métricas propostas procuram minimizar, indicam uma vantagem significativa da métrica MLURP sobre a métrica ML. Embora existam obstáculos consideráveis na implementação da métrica MLURP, estes resultados reforçam a necessidade da elaboração de uma maneira mais coerente de obter informações sobre a qualidade dos enlaces.

## 6.1 TRABALHOS FUTUROS

Um dos objetivos deste trabalho é demonstrar a importância do estudo das métricas de roteamento para o desenvolvimento das redes *mesh*. Procurou-se demonstrar com os resultados que as métricas atuais ainda apresentam problemas conceituais que afetam de forma decisiva seu desempenho. Desta maneira, fica claro que esta área está aberta a pesquisa em diversos aspectos.

Entre os mais importantes, pode-se citar a definição de modelos mais completos e exatos. Na métrica ML, por exemplo, o modelo não leva em consideração a possibilidade de retransmissões no nível de enlace, possibilitando que, mesmo que uma transmissão neste nível falhe, o pacote não seja perdido. Modelos melhores possibilitariam um proveito maior dos recursos da rede.

Outro aspecto fundamental é a validade das informações obtidas dos enlaces. Como mostrado neste trabalho, a falta de precisão nestas informações pode invalidar o modelo considerado pela métrica, levando a desempenhos abaixo do esperado. Além de precisos, estes algoritmos de obtenção das informações devem ser tão transparentes para a rede quanto possível. Em outras palavras, métodos que utilizam em demasia os recursos da rede, prejudicando os fluxos de dados dos clientes, são inaceitáveis, independentemente da sua precisão. Uma solução para este problema pode passar pela utilização de monitoramento de tráfego para efeito da avaliação dos enlaces.

Especificamente em relação às propostas deste trabalho, mais testes com a métrica MLAC podem ser realizados em diferentes topologias, para que valores interessantes do parâmetro  $\lambda$  sejam inferidos. Sobre a métrica MLURP, podem ser estudadas maneiras mais portáteis de implementá-la, já que a mesma mostrou resultados bastante interessantes. Trabalhos futuros podem também avaliar o desempenho de outras métricas, como a métrica ETX, utilizando a mesma técnica de obtenção das probabilidades.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Ian F. Akyildiz, Xudong Wang e Weilin Wang. Wireless mesh networks: a survey. *Comput. Netw. ISDN Syst.*, 47(4):445–487, 2005.
- [2] John Bicket, Daniel Aguayo, Sanjit Biswas e Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. Em *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pp. 31–42, 2005.
- [3] Raffaele Bruno, Marco Conti e Enrico Gregori. Mesh networks: Commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3):123–131, 2005.
- [4] T. Clausen e P. Jacquet. Optimized link state routing protocol (OLSR). RFC Experimental 3626, Internet Engineering Task Force, outubro de 2003.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest e C. Stein. *Algoritmos*. Campus, 2002.
- [6] D. De Couto, D. Aguayo, J. Bicket e R. Morris. A high-throughput path metric for multi-hop wireless routing. Em *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 134–146, 2003.
- [7] CUWin. <http://www.cuwireless.net>, março de 2007. Acessado em 13-03-2007.
- [8] Richard Draves, Jitendra Padhye e Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. Em *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 114–128, 2004.
- [9] D. Johnson, D. Maltz e J. Broch. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. *Ad Hoc Networking*, capítulo 5, pp. 139–172. Addison-Wesley, 2001.
- [10] Can Emre Koksall, Kyle Jamieson, Emre Telatar e Patrick Thiran. Impacts of channel variability on linklevel throughput in wireless networks. Em *Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 51–62, 2006.
- [11] Debora Muchaluat-Saade, Célio Vinicius Neves de Albuquerque, Luis Claudio Magalhães, Diego Passos, Jairo Duarte e Rafael do Valle. Redes em malha: Solução de baixo custo para popularização do acesso à internet no brasil. Em *XXV Simpósio Brasileiro de Telecomunicações*, 2007.
- [12] OpenWrt. <http://openwrt.org/>, março de 2007. Acessado em 03-03-2007.
- [13] Diego Passos, Douglas Teixeira, Debora Muchaluat-Saade, Luiz Magalhães e Celio Vinicius Neves de Albuquerque. Mesh network performance measurements. Em *5th International Information and Telecommunication Technologies Symposium*, 2006.



- [14] Charles E. Perkins, Elizabeth M. Belding-Royer e Samir R. Das. Ad hoc on-demand distance vector (AODV) routing. RFC Experimental 3561, Internet Engineering Task Force, julho de 2003.
- [15] Charles E. Perkins e Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. Em *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, pp. 90–100, Washington, DC, EUA, 1999. IEEE Computer Society.
- [16] K. Ramachandran, M. Buddhikot, G. Chandranmenon, S. Miller, E. Belding-Royer e K. Almeroth. On the design and implementation of infrastructure mesh networks. Em *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.
- [17] ReMesh. <http://mesh.ic.uff.br>, junho de 2007. Acessado em 13-07-2007.
- [18] C. Santivanez e R. Ramanathan. Hazy sighted link state (HSLS) routing: A scalable link state algorithm, março de 2003. Em BBN Technical Memorandum, No. 1301. <http://www.ir.bbn.com/documents/techmemos/TM1301.pdf>. Acessado em 13-07-2007.
- [19] Solução *mesh* Cisco. [http://www.cisco.com/application/pdf/en/us/guest/products/ps6548/c1031/cdcont\\_0900aecd803664b4.pdf](http://www.cisco.com/application/pdf/en/us/guest/products/ps6548/c1031/cdcont_0900aecd803664b4.pdf), julho de 2007. Acessado em 13-07-2007.
- [20] Solução *mesh* Nortel. <http://www.nortel.com/wirelessmesh>, julho de 2007. Acessado em 13-07-2007.
- [21] Ajay Tirumala, Feng Qin, Jon Dugan, Jim Ferguson e Kevin Gibbs. <http://dast.nlanr.net/projects/iperf/>, março de 2007. Acessado em 03-03-2007.
- [22] Andreas Tønnesen. <http://www.olsr.org/>, fevereiro de 2007. Acessado em 03-02-2007.
- [23] N. Tsarmpopoulos, I. Kalavros e S. Lalis. A low-cost and simple-to-deploy peer-to-peer wireless network based on open source linux routers. Em *Proceedings of TRIDENT-COM'05*, pp. 92–97, 2005.
- [24] Y. Yang, J. Wang e R. Kravets. Designing routing metrics for mesh networks. *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*. IEEE Press, 2005.