



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO**

Rafael Lopes Gomes

**ROTEAMENTO BASEADO EM ESCOLHA DINÂMICA DE
MÉTRICAS EM REDES EM MALHA SEM FIO**

Belém
2009

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO**

Rafael Lopes Gomes

**ROTEAMENTO BASEADO EM ESCOLHA DINÂMICA DE
MÉTRICAS EM REDES EM MALHA SEM FIO**

**Trabalho de Conclusão de Curso Apresentado para a obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Antônio Jorge Gomes Abelém**

Belém
2009

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO

Rafael Lopes Gomes

**ROTEAMENTO BASEADO EM ESCOLHA DINÂMICA DE
MÉTRICAS EM REDES EM MALHA SEM FIO**

Trabalho de Conclusão de Curso apresentado para a obtenção do grau de Bacharel em Ciência da Computação.

Data da defesa: 29 de Junho de 2009

Conceito: Excelente

Banca Examinadora

Prof. Dr. Antônio Jorge Gomes Abelém
Universidade Federal do Pará - Orientador

Prof. Dr. Kelvin Dias Lopes
Universidade Federal do Pará - Membro

Prof. Dr. Eduardo Cerqueira
Universidade Federal do Pará - Membro

Dedico este trabalho à minha família e amigos.

AGRADECIMENTOS

Agradeço aos Professores da Faculdade de Computação pelo conhecimento que foi passado a mim e agradeço também aos amigos do curso que me ajudaram e compartilharam o conhecimento comigo.

Agradeço a minha família, minha mãe Rosangela Maria, meu pai Daniel Gomes e minha irmã Daniela Gomes, pelo apoio e ajuda nos momentos difíceis. Assim como minha namorada, Louise Barros.

Agradeço aos amigos do GERCOM (Grupo de Estudos de Redes de Computadores e Comunicação Multimídia) pelo apoio e ajuda que me possibilitaram desenvolver este trabalho de forma eficiente e coerente, assim como meu orientador Antônio Abelém.

”É melhor tentar e falhar, que preocupar-se e ver a vida passar; é melhor tentar, ainda que em vão, que sentar-se fazendo nada até o final. Eu prefiro na chuva caminhar, que em dias tristes em casa me esconder. Prefiro ser feliz, embora louco, que em conformidade viver ...”

Martin Luther King

Sumário

Sumário	vi
Lista de Figuras	viii
Lista de Tabelas	viii
Lista de Siglas	x
Resumo	xii
Abstract	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	4
1.3 Organização	4
2 Redes em Malha Sem Fio	6
2.1 Tipos de Nós em Malha sem Fio	6
2.2 Arquiteturas em Malha sem Fio	7
2.3 Diferenças entre redes Ad Hoc e Mesh	8
2.4 Soluções Pre-Mesh	9
3 Protocolos de Roteamento	11
3.1 Classificação dos Protocolos de Roteamento	11
3.2 Protocolos de Roteamento para as Redes Ad Hoc	11
3.3 Protocolos de Roteamento para as Redes Mesh	12
3.4 Optimized Link State Routing	13
3.4.1 OLSR-ETX	14
3.4.2 OLSR-ML	15
3.4.3 OLSR-MD	15
3.4.4 Vantagens e Desvantagens de Cada Métrica	16
4 Estudo Sobre os Protocolos TCP e UDP em Redes em Malha sem Fio	17
4.1 Estudo sobre o protocolo TCP	18
4.2 Estudo sobre o protocolo UDP	21
5 Extensão para o protocolo OLSR baseado na escolha dinâmica de métricas	25
5.1 Ajuste do cálculo da métrica ETX	25
5.2 Modificações no Algoritmo de Roteamento	27

5.3	Modificações na tabela de roteamento	29
5.4	Modificações no encaminhamento de pacotes	30
5.5	Seleção da métrica a ser utilizada para a seleção do conjunto de nós MPR .	31
6	Análise dos Resultados	33
6.1	Análise do Ponto de Vista da Rede (QoS)	35
6.2	Análise do Ponto de Vista do Usuário (QoE)	39
7	Conclusão	43
7.1	Trabalhos futuros	43
	Referências	45
A	Script de Simulação	49

Lista de Figuras

1.1	Experiência do Usuário Modelada em Forma de Árvore [Perkis et al., 2006]	3
2.1	Arquitetura Cliente [Akyildiz and Wang, 2005]	7
2.2	Arquitetura Infra-Estruturada [Akyildiz and Wang, 2005]	7
2.3	Arquitetura Híbrida [Akyildiz and Wang, 2005]	8
2.4	Classificação das redes sem fio de múltiplos saltos [Zhang et al., 2007]	8
3.1	Algoritmo de seleção de MPRs do protocolo OLSR	14
3.2	Exemplo de cenário utilizando a métrica ETX	15
3.3	Exemplo de cenário utilizando a métrica ML	15
4.1	Backbone da WMN na UFPA	18
4.2	Percentual de Descarte dos Fluxos TCP	19
4.3	Vazão dos Fluxos TCP	20
4.4	Probabilidade de Bloqueio dos Fluxos TCP	20
4.5	Percentual de Descarte dos Fluxos UDP	22
4.6	Vazão dos Fluxos UDP	22
4.7	Probabilidade de Bloqueio dos Fluxos UDP	23
4.8	Atraso dos Fluxos UDP	23
4.9	Jitter dos Fluxos UDP	24
5.1	Problema Identificado	27
5.2	Diferença no roteamento entre os protocolos	28
6.1	Backbone da WMN proposto para a UFPA	33
6.2	Probabilidade de Bloqueio dos Fluxos TCP	35
6.3	Probabilidade de Bloqueio das Chamadas VoIP	35
6.4	Probabilidade de Bloqueio da Transmissão de Vídeo	36
6.5	Vazão dos Fluxos TCP	36
6.6	Vazão as Chamadas VoIP	37
6.7	Vazão da Transmissão de Vídeo	37
6.8	Atraso das Chamadas VoIP	38
6.9	Atraso da Transmissão de Vídeo	38
6.10	Jitter das Chamadas VoIP	39
6.11	Jitter da Transmissão de Vídeo	39
6.12	Frame do Vídeo Original	41
6.13	Frames Transmitidos Pelos Protocolos	42

Lista de Tabelas

2.1	Diferenças entre as redes Ad Hoc e Mesh	9
3.1	Comparação das métricas	16
4.1	Parâmetros de Simulação	19
5.1	Rotas Seleccionadas pelos Protocolos para o Protocolo TCP	29
5.2	Rotas Seleccionadas pelos Protocolos Para o Protocolo UDP	29
6.1	Parâmetros das Simulações com 801.11g	34
6.2	Configuração dos Fluxos	34
6.3	Relação entre PSNR e MOS	40
6.4	Valores de VQM, SSIM, PSNR e MOS obtidos pelos protocolos	40

Lista de Siglas

CBR *Constant Bit Rate*

ETX *Expected Transmission Count*

FTP *File Transfer Protocol*

HTTP *Hyper Text Transfer Protocol*

IEEE *Institute of Electrical and Electronics Engineers*

IETF *Internet Engineering Task Force*

MAC *Medium Access Control*

MD *Minimum Delay*

MOS *Mean Opinion Score*

MPR *MultiPoint Relays*

NS-2 *Network Simulator*

OLSR *Optimized Link State Routing*

OLSR-DC *Optimized Link State Routing - Dynamic Choice*

OLSR-ETX *Optimized Link State Routing - Expected Transmission Count*

OLSR-MD *Optimized Link State Routing - Minimum Delay*

OLSR-ML *Optimized Link State Routing - Minimum Loss*

OWD *One Way delay*

PSNR *Peak Signal to Noise Ratio*

QoE *Quality of Experience*

QoS *Quality of Service*

SSIM *Structural Similarity Index*

TCP *Transmission Control Protocol*

TTL *Time To Live*

UDP *User Datagram Protocol*

UFF Universidade Federal Fluminense

UFPA Universidade Federal do Pará

VoIP *Voice over IP*

VQM *Video Quality Metric*

WLAN *Wireless Local Area Network*

WMN *Wireless Mesh Network*

Resumo

A crescente demanda das aplicações multimídia exige um novo comportamento dos protocolos de roteamento para as redes em malha sem fio (*Wireless Mesh Network* – WMN), por isso que além de suprir os requisitos mínimos de Qualidade de Serviço (*Quality of Service* - QoS), as WMN também precisam suprir os requisitos mínimos de Qualidade de Experiência (*Quality of Experience* - QoE).

Dentro deste contexto, este trabalho propõe uma extensão ao protocolo pro-ativo *Optimized Link State Routing* (OLSR) visando prover suporte diferenciado ao tráfego das aplicações que usam os protocolos *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP). Apresentando um análise não somente de um ponto de vista da rede (QoS), mas também em relação ao ponto de vista do usuário (QoE).

O desempenho da proposta, intitulada, *OLSR Dynamic Choice* (OLSR-DC), é analisada utilizando o simulador de redes *Network Simulator* (NS-2). Os resultados das simulações comprovaram o desempenho dos protocolos, analisando a vazão, a probabilidade de bloqueio dos tráfegos e a qualidade do vídeo recebido pelo usuário.

Palavras-chave: Redes em Malha sem Fio, OLSR, Roteamento, Escolha de Dinâmica de Métricas.

Abstract

The increasing demand of multimedia applications requires a new conduct of routing protocols for Wireless Mesh Networks (WMN). It is necessary to support the minimum requirements for Quality of Service (QoS), but the WMN also need to support the minimum requirements for Quality of Experience (QoE).

In this work we propose an extention to the proactive Optimized Link State Routing (OLSR) protocol in order to provide quality of service and support to applications which use TCP as their transport protocol. The proposal performance, called OLSR Dynamic Choice (OLSR-DC), is evaluated using the Network Simulator.

The simulations were performed to demonstrate the performance of the protocols, analyzing the throughput, the traffic blocking probability and the video received by the user.

Keywords: Wireless Mesh Networks, OLSR, Routing, Dynamic Choice of Metrics.

Capítulo 1

Introdução

Nos últimos anos, houve um grande avanço nas tecnologias de redes sem fio, provocando o surgimento de vários tipos de redes com o intuito de satisfazer diversas necessidades do mercado [Zhang et al., 2007].

Esse avanço gerou o surgimento das redes *Ad Hoc*, que são redes sem fio dinâmicas formadas espontaneamente, de forma que a comunicação entre os nós presentes na rede necessita de um estabelecimento de rotas de múltiplos saltos temporárias, sendo que isto ocorre sem a utilização de qualquer forma de infra-estrutura fixa [Liu and Singh, 2001].

Sendo assim, este tipo de rede forma um sistema autônomo onde os próprios nós são responsáveis em coordenar a comunicação via enlaces sem fio e, independente de serem transmissores ou receptores, espera-se que estes nós sejam capazes de rotear pacotes para os demais nós da rede.

Primeiramente, foram os militares que tiveram grande interesse neste tipo de rede, visto que esta poderia ser empregada imediatamente em campos de batalha para a comunicação entre soldados e veículos de guerra. Contudo, hoje em dia, a aplicabilidade deste tipo de rede vai além dos interesses militares, podendo ser utilizada por equipes de resgate em áreas afetadas por desastres, redes de sensores sem fio e entre outros [Moreira et al., 2008].

Junto a este cenário surgem as redes em malha sem fio, também conhecidas como *Wireless Mesh Network* (WMN), que são redes de múltiplos saltos, auto-configuráveis, com topologia variável, que possuem nós com comunicação feita por uma das variações do padrão desenvolvido pelo *Institute of Electrical and Electronics Engineers* (IEEE), IEEE 802.11, também conhecido como *Wireless Local Area Network* (WLAN), que possui roteamento feito dinamicamente.

As WMNs são um caso especial de rede *Ad hoc*, entretanto a principal diferença entre elas está no fato de que os nós das redes *Mesh* têm localização fixa, apesar de que essas localizações não sejam determinadas previamente.

Portanto, os algoritmos de roteamento têm muitas características em comum. Além disso, diferente das redes *Ad hoc*, os nós clientes em uma rede *Mesh* não precisam necessariamente atuar como roteadores, o que representa uma menor complexidade para os mesmos.

Segundo Bruno [Bruno et al., 2005], há muitas utilidades para as WMNs, algumas dessas utilidades são: aumento na área de cobertura de empresas e universidades; prover cobertura em locais onde o uso de cabeamento é algo inviável, devido ao custo e/ou de obstáculos físicos; prover comunicação em casos de emergência; gerar uma maior condição para se ter uma inclusão digital; e para dar suporte à operações militares e de salvamento.

Para Breuel [Breuel, 2007], dentre os vários motivos existentes, o que proporciona uma maior motivação para o uso de WMNs é a possibilidade de estender a conectividade à locais onde há dificuldade em instalar uma infra-estrutura cabeada, como áreas abertas, universidades, edifícios históricos e outros.

No entanto, apesar de WMNs serem consideradas muito úteis, ainda falta à elas uma padronização, situação essa que proporcionou o surgimento de diferentes soluções, sendo elas proprietárias ou não, que acabam levando a interoperabilidade entre essas soluções.

As soluções existentes que envolvem este tipo de rede são conhecidas como pré-mesh, visto que ainda não existe um padrão definido. Por conta disso, as soluções são baseados em outros padrões 802.1x [Moreira et al., 2008].

O grupo de trabalho IEEE 802.11 está trabalhando para especificar as funcionalidades das WMNs, as quais serão incorporadas ao padrão IEEE 802.11s [Moreira et al., 2008].

Independente do tipo, as redes sem fio têm certos aspectos a serem considerados no que diz respeito ao roteamento:

- Meio suscetível a erros;
- Problemas de terminais exposto e escondido;
- Topologias dinâmicas: nós podem se mover livre e randomicamente dentro de uma mesma rede ou entre redes;
- Recursos limitados como banda e energia;
- Segurança: nós não participantes da rede podem “escutar” as transmissões;
- Escalabilidade: devido a alta sobrecarga advindo do grande número de nós presente na rede são pontos importantes para o desenvolvimento de um protocolo de roteamento de funcionamento satisfatório.

Quando a rede em questão é uma rede *Mesh*, outros aspectos além da ausência de um padrão como a variedade de aplicações e diferentes formas de implementação dos protocolos de roteamento devem ser considerados para garantir o êxito na construção do protocolo.

1.1 Motivação

De acordo com Bruno [Bruno et al., 2005], os protocolos de roteamento para WMNs devem possuir as características de auto-gestão, auto-configuração e de auto-recuperação. Sendo assim, diferentes protocolos foram desenvolvidos, entretanto para Kowalik [Kowalik and Davis, 2006], estes protocolos visam suprir as necessidades das aplicações para as quais o protocolo foi focado, sendo que cada aplicação tem suas particularidades. Com isso os protocolos de forma geral não conseguem atender todas as necessidades das WMNs.

O crescimento da utilização da Internet através de redes sem fio motivou o desenvolvimento de aplicações e dispositivos que usam os recursos deste tipo de rede para o acesso à Internet. Deste modo, estes usuários de redes sem fio esperam obter os mesmos tipos de serviços que são oferecidos quando se encontram em redes cabeadas, assim como esperam o mesmo, quiçá superior, nível de qualidade [Chinta and Helal, 2003]. Com o crescimento das aplicações multimídia, as WMNs devem ser projetadas para transportar simultaneamente voz, dados e vídeo, que têm diferentes requisitos de desempenho.

Diferentes soluções foram desenvolvidas para proporcionar Qualidade de Serviço, ou *Quality of Service* (QoS), sendo propostos diferentes mecanismos e algoritmos, em sua maioria, para duas camadas do modelo OSI: aplicação e rede. O termo QoS refere-se à capacidade da rede garantir um certo nível de desempenho para um fluxo, em sua maioria um tráfego multimídia. A QoS faz uma perspectiva técnica do serviço, sendo um conceito técnico medido, expressado e compreendido em nível de rede e seus elementos, e geralmente tem pouco significado para o usuário.

Acima das tradicionais camadas é possível imaginar uma pseudo-camada de percepção, que preocupa-se com a experiência do usuário final. O conceito que expressa o desempenho desta camada de percepção é a Qualidade de Experiência, ou *Quality of Experience* (QoE).

O conceito de Qualidade de Experiência está relacionado à avaliação das aplicações multimídia do ponto de vista da percepção do usuário. O termo surgiu para suprir as “lacunas” deixadas pela avaliação tradicional realizada pelas métricas de QoS tais como: vazão, *jitter* e atraso, que avaliam o impacto das aplicações do ponto de vista da rede para suprir tais requisitos de desempenho.

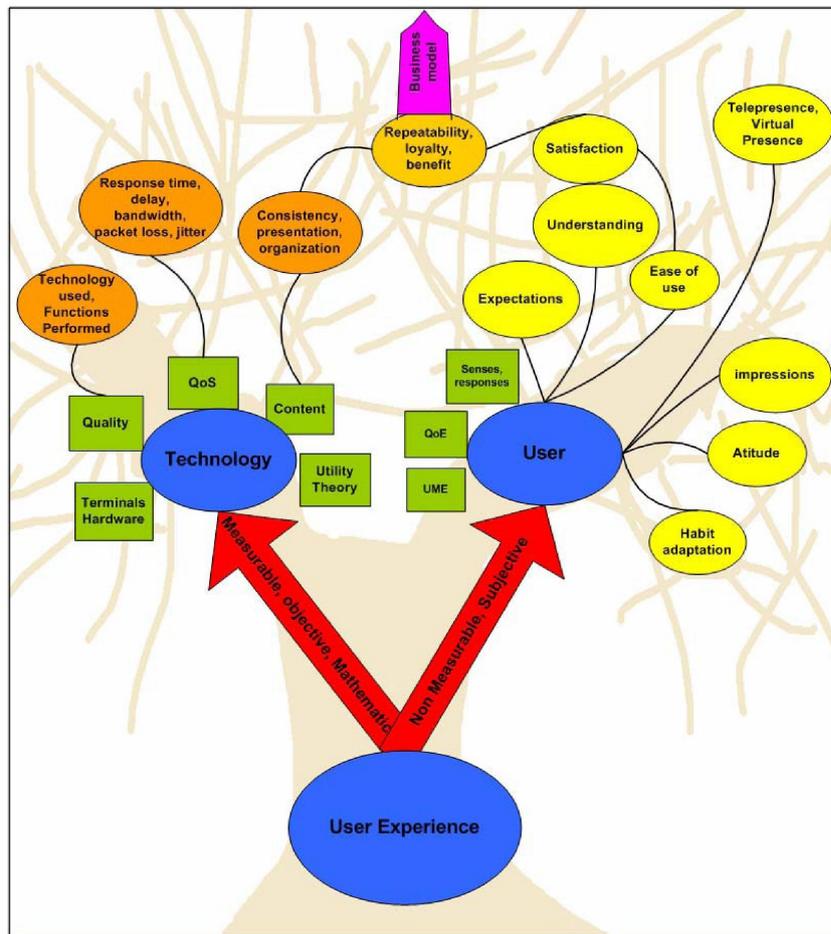


Figura 1.1: Experiência do Usuário Modelada em Forma de Árvore [Perkis et al., 2006]

As métricas de QoS não refletem a experiência do usuário diante de um vídeo, não permitindo afirmar se a qualidade do vídeo recebido pelo mesmo pode ser qualificada como boa. Pois se tratando de avaliações de aplicações multimídias, a sensibilidade humana é primordial [Jain, 2004] [Wang et al., 2002]. A Figura 1.1 mostra uma modelagem em árvore da experiência do usuário.

1.2 Objetivos

O principal objetivo deste trabalho é propor uma extensão para o protocolo *Optimized Link State Routing* (OLSR) baseada na escolha dinâmica de métricas, e testar sua viabilidade de implementação no cenário da WMN da Universidade Federal do Pará (UFPA), que se encontra parcialmente implantada. Esta escolha dinâmica de métricas deve ser feita a partir do tipo de pacote processado no momento, em nível de camada de transporte, objetivando assim atender as necessidades das aplicações de acordo com suas particularidades, sem que estas influenciem nas demais aplicações. Neste trabalho a proposta é intitulada *Optimized Link State Routing - Dynamic Choice* (OLSR-DC).

Para desenvolver as características da proposta e mostrar o desempenho desta, este trabalho usa o *Network Simulator* (NS-2) [Fall and Varadhan, 2007], em sua versão 2.33, para se realizar simulações que proporcionaram dados, onde analisou-se a perda de pacotes, percentual de descarte, probabilidade de bloqueio, atraso e *jitter*.

A análise da proposta é feita sobre um ponto de vista não só da rede mas do usuário final também, em um cenário com grande concorrência entre os fluxos de dados, áudio e vídeo. O tráfego de vídeo foi avaliado sob a experiência que o usuário obteve diante da aplicação após a transferência, através das métricas de avaliação de QoE. Sendo o cenário utilizado nas simulações a WMN da UFPA, proposta em [Aguilar et al., 2007].

A WMN presente na UFPA, que tem como base os mesmos parâmetros utilizados no *testbed* da Universidade Federal Fluminense (UFF), mas com as particularidades e características da região amazônica. O desenvolvimento da rede *Mesh* da UFPA fez parte de uma das atividades da fase 2 do projeto ReMesh [ReMesh, 2007], sendo que a fase 1 foi o desenvolvimento da WMN da própria UFF.

O campus universitário da UFPA, situado na cidade de Belém, possui muitos edifícios os quais são cercados, em sua maioria por outros edifícios ou por obstáculos naturais, como grande número de árvores e pequenos e grandes rios. Agregado a este fato a cidade de Belém possui altos índices pluviométricos e de umidade atmosférica, o que faz com que o uso de tecnologias sem fio tenham maiores dificuldades com relação ao seu desempenho [Passos et al., 2006], sendo assim a implementação de uma rede *Mesh* não foge deste contexto.

O projeto ReMesh usa o OLSR como seu protocolo de roteamento, o qual é um protocolo pró-ativo padronizado pela *Internet Engineering Task Force* (IETF). Visando manter as rotas disponíveis atualizadas, os protocolos pró-ativos geralmente geram mais *overhead*, quando comparados aos protocolos por demanda e híbridos. Mas, como os roteadores nas WMNs não possuem movimento e nem restrições de energia, o roteamento baseado no estado dos enlaces está sendo muito utilizado nas WMNs, por isso o protocolo OLSR foi utilizado no projeto ReMesh [ReMesh, 2007].

1.3 Organização

Este trabalho está organizado da seguinte maneira: o Capítulo 2 explica as redes em malha sem fio existentes e suas características. O Capítulo 3 trata dos protocolos de roteamento em geral e do protocolo OLSR. O Capítulo 4 mostra um estudo feito acerca do desempenho dos protocolos *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP) com relação a WMN existente na UFPA. O Capítulo 5 apresenta o OLSR-DC, a extensão do protocolo OLSR proposta neste trabalho. O Capítulo 6 apresenta a análise dos resultados das simulações feitas no *Network Simulator* usando os protocolos

OLSR, *Optimized Link State Routing - Minimum Delay* (OLSR-MD), *Optimized Link State Routing - Expected Transmission Count* (OLSR-ETX) e o OLSR-DC. E, finalmente, o Capítulo 7 apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Redes em Malha Sem Fio

Devido ao seu potencial, as WMNs estão se tornando uma tecnologia promissora e terão papel importante na futura geração de redes móveis sem fio. As redes em malha sem fio são capazes de se organizar, configurar e restabelecer para permitir uma rápida implantação, fácil manutenção, baixo custo, alta escalabilidade e serviços confiáveis, assim como melhorar a capacidade e conectividade da rede [Zhang et al., 2007].

Diferente de sistemas sem fio centralizados, as redes em malha sem fio possuem como principal característica a tolerância à falhas, pois não há presença de entidades centrais, mesmo com a queda de vários enlaces, a disponibilidade dos serviços de comunicação [Zhang et al., 2007].

As redes *Mesh* são uma forma estendida das redes *Ad Hoc*. Nos últimos anos, houve um grande interesse em redes *Ad Hoc* devido ao potencial que estas podem oferecer em atividades comerciais e, principalmente, militares. Uma rede *Ad Hoc* é uma rede sem fio composta de equipamentos móveis que se comunicam através de ondas de rádio e não têm infra-estrutura fixa (uma administração central como um ponto de acesso). Estes equipamentos móveis também podem atuar como roteadores em virtude do alcance limitado de transmissão destes. Isto é, em dada situação os equipamentos terão que rotear um pacote para que o mesmo chegue ao destino [Held, 2005].

Mesmo possuindo características em comum, os protocolos e arquiteturas desenvolvidos para as redes *Ad Hoc* têm um fraco desempenho quando utilizados nas WMNs. Isto é explicado pela diferença que há entre estas redes no que diz respeito à aplicabilidade, objetivos com a implantação e limitação de recursos a qual elas são submetidas [Held, 2005].

2.1 Tipos de Nós em Malha sem Fio

Dois tipos de nós podem ser encontrados nas WMNs, no entanto estas aceitam a comunicação com outros tipos de redes e seus respectivos equipamentos [Akyildiz and Wang, 2005].

- Roteadores Mesh (*Mesh Routers*): Possuem as mesmas funcionalidades de roteadores convencionais, como *gateway* e *bridge*, porém com o suporte a WMN, que provê maior flexibilidade a rede, pois permite a comunicação com outros tipos de redes, como a cabeada, através do uso de múltiplas interfaces. Para que todos estes recursos possam ser executados de maneira satisfatória é necessário um maior poder computacional, necessitando normalmente de um computador para realizar o papel de nó central ou simplesmente fazer uso de um sistema embarcado.

- Clientes Mesh (*Mesh Clients*): Podem realizar o processo de encaminhamento de pacotes entre os demais elementos da rede, no entanto não podem exercer as funções de *bridge* ou *gateway*. Em contrapartida, este tipo de nó apresenta, geralmente, apenas uma interface de rede e um *hardware* bem mais simples.

2.2 Arquiteturas em Malha sem Fio

Através da combinação desses dois tipos de nós juntamente com a utilização de interfaces de redes não pertencentes a WMN, podem ser formadas três tipos de arquiteturas [Akyildiz and Wang, 2005]:

- Arquitetura Cliente: Apenas *Mesh Clients* são usados nesta arquitetura. Cada nó faz o papel de cliente e de roteador, como mostra a Figura 2.1. Eles se comunicam como em uma rede *peer-to-peer* formando uma estrutura muito próxima a de uma rede *Ad Hoc*, diferindo apenas na utilização de uma única tecnologia de transmissão.

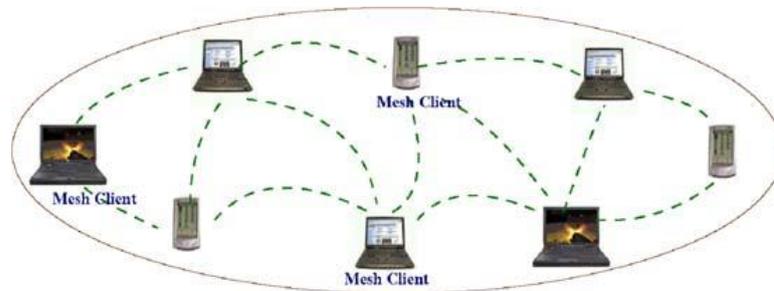


Figura 2.1: Arquitetura Cliente [Akyildiz and Wang, 2005]

- Arquitetura Infra-Estruturada: O *backbone* da rede é composto de *Mesh Routers* que fornecem a infra-estrutura básica para a conexão de clientes, vista na Figura 2.2. Através deste *backbone* formado é possível interligar diferentes redes com diferentes tecnologias de transmissão. Esse é o tipo de WMN mais usada, pois necessita de modificações apenas nos seus roteadores que normalmente utilizam duas antenas com canais distintos, uma para o *backbone* e outra para atender os clientes.

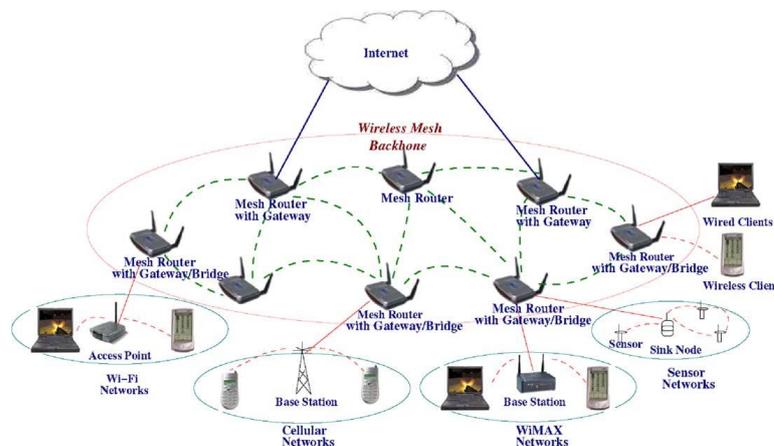


Figura 2.2: Arquitetura Infra-Estruturada [Akyildiz and Wang, 2005]

- Arquitetura Híbrida: Esta arquitetura faz o uso tanto dos *Mesh Routers* como dos *Mesh Clients*. Ela é a configuração mais completa, fazendo o uso de todas as possibilidades de comunicações que as WMNs oferecem, possibilitando que *Mesh Clients* e os clientes convencionais tenham acesso ao *backbone mesh* que oferece uma série de interligações com outras redes. A Figura 2.3 apresenta este tipo de arquitetura.

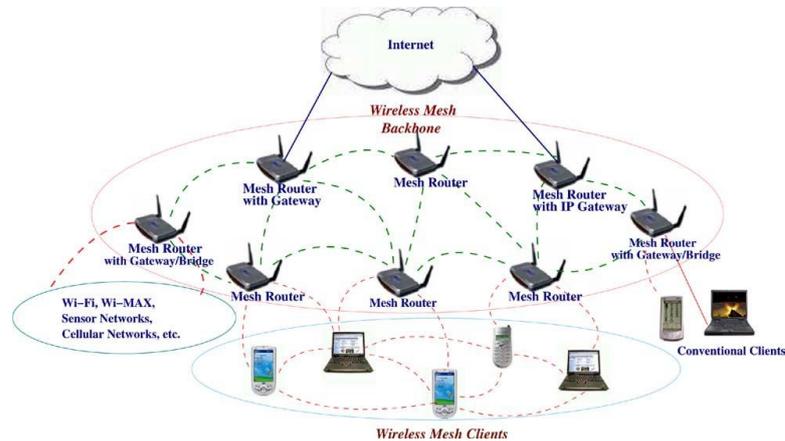


Figura 2.3: Arquitetura Híbrida [Akyildiz and Wang, 2005]

2.3 Diferenças entre redes Ad Hoc e Mesh

Zhang [Zhang et al., 2007] classifica as redes sem fio de múltiplos saltos em redes *Ad Hoc*, *Mesh*, de sensores e híbridas de acordo com a Figura 2.4.

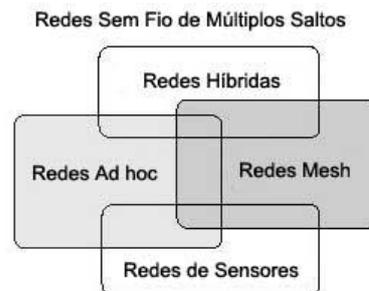


Figura 2.4: Classificação das redes sem fio de múltiplos saltos [Zhang et al., 2007]

A Tabela 2.1 mostra algumas diferenças entre as redes que interessam a este trabalho, *Ad Hoc* e *Mesh*. Apesar das redes *Mesh* terem origens nas redes *Ad Hoc*, elas diferem principalmente em aspectos como: mobilidade de nós e topologia da rede. As redes *Ad Hoc* são de alta mobilidade onde sua topologia muda dinamicamente. Já as redes *Mesh* são relativamente estáticas onde a maioria de seus nós são fixos, principalmente aqueles que compõem seu *backbone*.

A diferença topológica também influencia no roteamento por isso os protocolos utilizados devem ser de natureza reativa, pró-ativa ou hierárquica. Como as redes *Mesh* têm topologia estática composta de nós fixos, elas tendem a ter um eficiente consumo de energia que é um dos principais problemas existentes nas redes *Ad Hoc*.

Tabela 2.1: Diferenças entre as redes Ad Hoc e Mesh

Aspectos	Ad Hoc	Mesh
Topologia	Altamente dinâmica	Relativamente estática
Mobilidade dos nós	Média/Alta	Baixa
Restrição de energia	Alta	Baixa
Tipo de aplicação	Temporária	Semi-permanente/Permanente
Infra-estrutura	Inexistente	Parcial/Total
Retransmissão	Nós móveis	Nós fixos
Protocolos de roteamento	Reativo	Hierárquico/Pró-ativo
Implantação	Fácil	Requer planejamento

2.4 Soluções Pre-Mesh

O IEEE designou um grupo para trabalhar no padrão 802.11s que irá definir as funcionalidades das redes *Mesh*. Mas, este padrão ainda se encontra em desenvolvimento. Todavia há, disponíveis no mercado, soluções conhecidas como pré-mesh onde grandes empresas como Motorola e Cisco já oferecem soluções *Mesh* direcionadas aos mais variados ramos de atividades sejam estas acadêmicas, empresariais ou governamentais [Moreira et al., 2008].

A principal desvantagem destas soluções é justamente a falta de interoperabilidade entre as mesmas. Ou seja, o cliente destas soluções fica sem outras opções de equipamentos/softwarewares a não ser aqueles oferecidos pelo fabricante de quem ele inicialmente comprou a solução.

As universidades também começaram a explorar este tipo de rede e os resultados foram soluções baseadas em equipamentos 802.11b/g que já são populares por serem de fácil implantação e manutenção, assim como têm baixos custos.

O padrão no qual as soluções pré-mesh não comerciais são baseadas é o IEEE 802.11 que se refere a uma família de especificações de uma rede local sem fio (WLAN), que é um tipo de rede local que utiliza ondas de rádio de alta frequência ao invés de cabos para a comunicação entre clientes. Este padrão especifica uma interface de comunicação “pelo ar” entre um cliente móvel e um ponto de acesso ou entre dois clientes móveis [IEEE, 1999].

Este padrão é o mais popular para prover serviços de rede em lugares onde o cabeamento é dificultado ou muito custoso. Este padrão especifica as camadas física e *Medium Access Control* (MAC) adaptadas para redes sem fio e seus objetivos são prover conectividade a equipamentos e redes móveis, assim como padronizar acesso a uma ou mais bandas de frequência de rádio usadas na comunicação da rede local. As interfaces oferecidas às camadas mais altas pelo padrão 802.11 são as mesmas que os outros padrões 802.x oferecem. A camada MAC deve funcionar com diferentes camadas físicas utilizando diferentes técnicas de transmissão como infravermelho e espectro espalhado.

As redes sem fio utilizam canais de frequência de rádio como meio físico para a comunicação. Cada nó na rede envia informação que pode ser recebida por todos os outros nós dentro de sua faixa direta de transmissão. Como os nós enviam e recebem dados “pelo ar”, eles não necessitam estar conectados fisicamente a alguma rede. Assim, este tipo de rede oferece conectividade juntamente com mobilidade.

As redes 802.11 são compostas por quatro importantes componentes físicos: ponto de acesso (Access Point – AP), sistema de distribuição (Distribution System – DS), meio sem fio (*Wireless Medium*) e estações (*Stations*).

É possível identificar todos estes componentes em uma rede *Mesh*. Contudo, é válido mencionar que o AP deixa de ser uma entidade centralizadora como ocorre nas redes infra-estruturadas. Nas redes *Mesh*, o AP servirá como ponto de associação para clientes sem fio e terá principalmente a função de roteador, onde estará encaminhando informações entre os demais APs que compõem a rede *Mesh*.

Capítulo 3

Protocolos de Roteamento

Neste capítulo é apresentada uma classificação dos protocolos de roteamento e suas características necessárias a fim de garantir um roteamento eficiente. Também é apresentado o protocolo OLSR e algumas de suas extensões que visam melhorar o desempenho do mesmo, a partir de novas métricas para o roteamento.

3.1 Classificação dos Protocolos de Roteamento

De acordo com Murthy [Murthy and Manoj, 2004], os protocolos podem ser classificados considerando vários aspectos que podem ser baseados: no uso de informação temporal; na topologia; na utilização de recursos específicos; e no mecanismo de atualização de informação.

Protocolos baseados no uso de informação temporal constroem suas tabelas de rotas com base no histórico de qualidade dos enlaces. Se a topologia for considerada pelo protocolo, é o tipo de topologia que determina as ações do mesmo.

Já o protocolo baseado na utilização de recursos específicos considera, para determinação de suas rotas, recursos que devem ter sua utilização otimizada como o consumo de energia.

No mecanismo de atualização de informação, o protocolo de roteamento baseia-se em informações obtidas previamente ou sob demanda para construção da tabela de rotas.

Para este trabalho é considerada esta última classificação que divide os protocolos em roteamento:

- **Pró-ativo:** os nós trocam tabelas de roteamento periodicamente mantendo informações sobre toda a topologia, com cada nó conhecendo o menor caminho para cada outro nó da rede.
- **Reativo:** as rotas são estabelecidas sob demanda, ou seja, rotas são criadas quando solicitadas pelo transmissor.
- **Híbridos:** combinam as melhores características dos protocolos pró-ativos (roteamento dentro da mesma zona) e reativos (roteamento entre zonas diferentes).

3.2 Protocolos de Roteamento para as Redes Ad Hoc

Um protocolo de roteamento para redes *Ad Hoc* deve ser capaz de: trocar informações de roteamento; achar uma rota ótima até o destino com base em parâmetros como número

de saltos; colher informações sobre a queda de enlaces; restabelecer enlaces consumindo o mínimo de banda e poder de processamento; e utilizar banda mínima [Murthy and Manoj, 2004].

Há vários aspectos que devem ser considerados pelos protocolos de roteamento como:

1. Mobilidade: um dos aspectos mais importantes das redes *Ad Hoc* e resulta em quedas frequentes de enlaces, colisões de pacotes, *loops*, informações de roteamento desatualizadas e reserva de recursos dificultada;
2. Banda limitada: como o canal é compartilhado entre todos os nós, a banda é limitada pelo número de nós e tráfegos presentes em um dado enlace sem fio. Isto significa que os nós têm apenas uma fração da banda total disponível para seu uso;
3. Canal sujeito a erros: em canais sem fio, a taxa de erro de bits é muito elevada e deve ser cuidadosamente considerada nos protocolos de roteamento. Estado dos enlaces sem fio, razão sinal/ruído e perdas na rota são pontos importantes para definir a eficiência do protocolo;
4. Contenção: a contenção de um canal é diretamente proporcional ao número de nós presente no mesmo e resulta em um alto índice de colisões e consumo ineficiente da banda. É válido mencionar o problema de terminais exposto e escondido que também influenciam na contenção. Um bom protocolo dispõe de mecanismos de balanceamento de carga que evitam a formação de áreas de alta contenção.
5. Recursos limitados: poder de processamento, energia limitada e utilização de *buffers* também influenciam na eficiência do protocolo.

Portanto, as redes *Ad Hoc* demandam protocolos que saibam lidar com os aspectos previamente mencionados. Isto implica dizer que, para serem eficientes, estes protocolos devem atender requisitos como: atraso mínimo na obtenção de rotas; estabelecimento/restabelecimento rápido de rotas; roteamento sem *loops*; roteamento descentralizado; adaptabilidade a mudanças topológicas; sobrecarga de controle mínima; escalabilidade; provisão de qualidade de serviço; suporte a tráfego de tempo real; segurança e privacidade; e utilização eficiente de recursos escassos como banda, processamento, memória e energia.

3.3 Protocolos de Roteamento para as Redes Mesh

Os protocolos de roteamento para redes *Mesh* têm papel importante no que diz respeito à comunicação entre as partes envolvidas. Este tipo de rede é caracterizado pela presença de nós móveis e fixos interconectados através de enlaces sem fio formando uma rede de múltiplos saltos. Os nós em questão podem atuar como terminais e/ou roteadores para outros nós da rede estendendo, assim, a área de cobertura deste tipo rede.

Em virtude de os nós serem fixos e não serem operados á bateria, os protocolos de roteamento para as redes *Mesh* devem focar na confiabilidade e na melhoria do desempenho da rede ao invés de lidar com mobilidade ou minimizar consumo de energia.

Como as redes *Mesh* são consideradas uma extensão das redes *Ad Hoc*, seus protocolos devem ser capazes de se administrar, se configurar e se restabelecer [Bruno et al., 2005]. Para que os protocolos tenham um funcionamento eficiente, eles necessitam, principalmente, ter as seguintes características:

1. Tolerância à falhas: devido sua natureza, as redes *Mesh* devem garantir a robustez contra falhas de enlaces.
2. Balanceamento de carga: já que os roteadores são capazes de escolher caminhos eficientes para o tráfego de dados, esta se torna uma característica importante.
3. Redução de sobrecarga: para garantir o desempenho de redes sem fio é importante racionar o consumo de banda através da redução do *overhead*.
4. Escalabilidade: as redes *Mesh* surgiram justamente por serem capazes de suportar um elevado número de nós devido à ausência de entidades centralizadoras. Portanto, os protocolos devem saber como lidar com esta característica.
5. Suporte à qualidade de serviço: esta característica tem se tornado muito importante por conta da capacidade limitada do canal de comunicação, da suscetibilidade à interferência, do grande número de usuários que podem compor a rede e, principalmente, pelo surgimento de aplicações multimídia de tempo real.

3.4 Optimized Link State Routing

O protocolo OLSR [Clausen and Jacquet, 2003] é uma adaptação do algoritmo de estado de enlace tradicional voltado para redes *Ad Hoc*. Ele é um protocolo pró-ativo, que utiliza uma tabela de roteamento obtida por meio da troca de mensagens sobre informação da topologia da rede com outros nós.

Um único pacote do OLSR é capaz de transportar várias mensagens definidas pelo protocolo OLSR até que o tamanho máximo de pacote permitido pela rede seja atingido. Isso permite um menor *overhead* para o envio de diferentes mensagens por um mesmo nó da rede em um certo momento.

O que torna o protocolo OLSR adequado para as WMNs é o número limitado de pacotes de controle enviado por este. Essa otimização é alcançada a partir do conceito de *MultiPoint Relays* (MPR). Cada nó na rede seleciona nós para fazerem parte do seu conjunto de nós MPR, entre os seus vizinhos de primeiro salto que possuem comunicação simétrica (isto é, possuem comunicação bidirecional). A Figura 3.1 mostra o algoritmo de seleção do conjunto de nós MPR [Leguay et al., 2006].

A seleção do conjunto de nós MPR ajuda a evitar que a rede seja inundada por mensagens de controle geradas pelo protocolo. Para Munaretto [Munaretto et al., 2003] isso reduz o tamanho das mensagens de controle e minimiza o *overhead* oriundo da inundação de tráfego de controle.

Existe ainda outro tipo de mensagem de controle do protocolo OLSR, as mensagens de declaração de múltiplas interfaces de rede, que serve para associar vários endereços a um endereço principal. Ela é especialmente utilizada em ocasiões em que um único nó possui mais de uma interface sem fio participando da rede utilizando o protocolo OLSR.

De acordo com Leguay [Leguay et al., 2006], uma vantagem do protocolo OLSR do ponto de vista de QoS é que a sua natureza pró-ativa permite que rotas estejam disponíveis antes mesmo que a fonte precise iniciar um fluxo de pacotes para um nó de destino qualquer.

Essa característica é especialmente interessante em redes *Ad Hoc*, e também nas WMNs, que precisam oferecer suporte a aplicações multimídia, como por exemplo, voz

Algoritmo de seleção de MPRs do OLSR (i, N_i^1, N_i^2)

Entrada : Um nó i , os vizinhos de um salto de i N_i^1 e os vizinhos de dois saltos de i N_i^2 .

Saída : M_i , o conjunto de MPRs do nó i .

1 Início

2 Adicionar a M_i os nós em N_i^1 os quais são os únicos nós a proporcionar conectividade com um nó em N_i^2 .

3 Remover os nós de N_i^2 que já estão cobertos por um nó em M_i .

4 Enquanto $N_i^2 \neq \emptyset$ faça

5 Para cada nó em N_i^1 , calcular a alcançabilidade, isto é, o número de nós em N_i^2 que ele pode alcançar.

6 Adicionar a M_i o nó que proporciona a maior alcançabilidade. Em caso de múltiplas possibilidades, selecionar o nó que possui o maior número de vizinhos distantes 1 salto.

7 Remover de N_i^2 os nós que já são alcançados por algum nó em M_i .

8 Fim

Figura 3.1: Algoritmo de seleção de MPRs do protocolo OLSR

sobre IP, ou *Voice over IP* (VoIP), uma vez que a latência para o estabelecimento de uma comunicação é relativamente baixa.

Ainda de acordo com Leguay [Leguay et al., 2006], outra vantagem do OLSR enquanto protocolo que utiliza o algoritmo de estado de enlace, é que a computação das rotas é realizada utilizando o conhecimento sobre a topologia de toda a rede. Essa característica permite que o OLSR proporcione um melhor suporte a QoS que as propostas de protocolos de roteamento baseados no algoritmo de vetor de distância.

A proposta do OLSR tradicional utiliza o número de saltos (*Hop Count*) como métrica para o roteamento. O número de saltos é utilizado para computar a menor distância (e, portanto, para o protocolo a melhor rota) para um destino qualquer da rede.

Entretanto, para Aslam [Aslam et al., 2004], o critério de número de saltos definido na proposta original do OLSR não é capaz de fornecer suporte a QoS, uma vez que um caminho selecionado baseado no menor número de saltos pode não satisfazer aos requisitos de QoS determinados pela aplicação que fará uso da rede.

Restrições quanto à taxa de perdas de pacotes, atrasos, *jitter* e banda mínima não são possíveis de se garantir selecionando rotas que, apesar de se mostrarem menores em termos de número de saltos, podem ser extremamente instáveis.

Dentro deste contexto foram desenvolvidas algumas extensões para o protocolo OLSR, baseadas em outras métricas distintas do número de saltos, algumas destas, consideradas pro contexto deste trabalho, são apresentadas a seguir.

3.4.1 OLSR-ETX

Considerando a instabilidade de uma rede sem fio, há várias situações em que o menor número de saltos não é a melhor escolha. Em situações em que a rede sem fio é densa, pode ocorrer um grande número de rotas com o mesmo número de saltos, entretanto com diferentes qualidades de enlace. Podem inclusive ocorrer situações em que uma rota com um número maior de saltos apresente uma qualidade melhor que uma rota com um número menor de saltos.

Uma extensão ao protocolo OLSR utiliza uma nova métrica, o número esperado de

transmissões, do inglês *Expected Transmission Count* (ETX) [Couto et al., 2003], para selecionar as melhores rotas.

Esta extensão tem como objetivo encontrar rotas com o menor número esperado de transmissões, incluindo retransmissões, que são necessárias para que um pacote possa ser entregue e tenha seu recebimento confirmado pelo destino final. Para isso se escolhe as rotas que possuem a menor soma dos valores de ETX de cada enlace, como exemplificado no cenário mostrado na Figura 3.2.

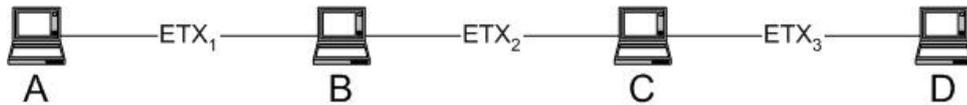


Figura 3.2: Exemplo de cenário utilizando a métrica ETX

Em casos onde há empate entre rotas, a rota com o menor número de saltos é escolhida, para que se possa manter a conformidade com [Clausen and Jacquet, 2003]. Deve-se ter em mente que esse valor é válido para ambas as direções do enlace, uma vez que em ambos os casos é necessário avaliar a ocorrência de um *round-trip* com sucesso.

3.4.2 OLSR-ML

De acordo com Passos [Passos et al., 2006], o uso do protocolo OLSR-ETX em redes *Ad Hoc* pode fazer com que a rede se torne instável e com muita perda de pacotes. [Passos et al., 2006] propôs uma nova alternativa para o cálculo da qualidade de enlace de uma determinada rota, visando escolher rotas com probabilidade mínima de perda de pacotes, a qual foi intitulada *Optimized Link State Routing - Minimum Loss* (OLSR-ML).

Nesta proposta, o ETX é interpretado como a probabilidade de que um *round-trip* ocorra com sucesso, diferentemente da proposta do OLSR-ETX, onde o valor de ETX reflete o número esperado de transmissões. A Figura 3.3 ilustra um cenário onde a métrica ML selecionaria a rota com maior número de saltos, mas com menor probabilidade de perda em cada enlace do caminho selecionado.

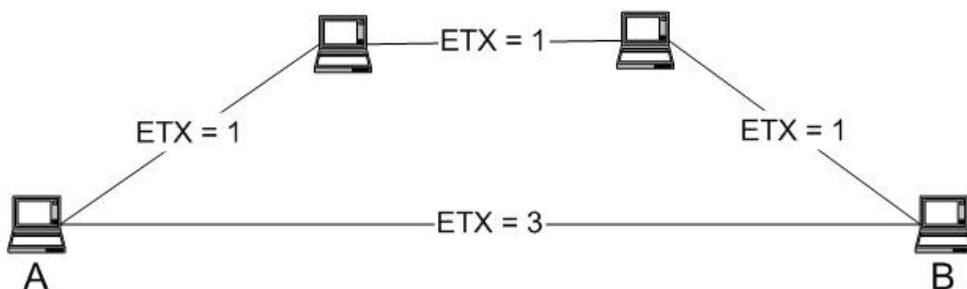


Figura 3.3: Exemplo de cenário utilizando a métrica ML

Nesta proposta, portanto, será selecionada a rota que apresentar a maior probabilidade de sucesso, ao contrário do que ocorre no OLSR-ETX, onde é selecionada a rota que apresenta o menor número esperado de transmissões de todos os saltos ao longo do caminho.

3.4.3 OLSR-MD

A idéia principal desta extensão, OLSR-MD, é medir o atraso dos enlaces, calculando-os através da técnica *AdHoc Probe*. Com isso, o cálculo da tabela de rotas podem ser feito com base no atraso calculado para cada um dos nós vizinhos [Cordeiro et al., 2007].

O algoritmo *AdHoc Probe* utiliza pares de pacotes para medir o atraso do enlace em um dos sentidos do mesmo, sendo esse atraso denominado *One Way delay* (OWD). A partir do cálculo do OWD é possível, portanto, determinar o retardo de transmissão e a capacidade do enlace. Sendo assim, a idéia é incluir a medida de retardo de enlace calculada pelo *AdHoc Probe* para se realizar o roteamento no protocolo OLSR-MD.

O *AdHoc Probe* necessita que pares de pacotes de tamanho fixo sejam enviados a uma taxa constante [Cordeiro et al., 2007], sendo que o tamanho fixo dos pacotes de controle permitirá o cálculo do OWD. Os demais pacotes de controle do OLSR serão enviados normalmente.

A técnica *AdHoc Probe* apenas permite calcular o OWD em um dos sentidos do enlace. Para que os nós possam saber o retardo de transmissão do enlace no caminho inverso, é proposta uma extensão da mensagem OLSR_HELLO do protocolo OLSR para carregar informações sobre o retardo de transmissão do enlace no sentido dos nós vizinhos ao nó que atualmente está enviando a mensagem OLSR_HELLO. O nó receptor da mensagem, portanto, passará a ter a informação do retardo de transmissão do enlace no sentido emissor para o receptor.

No protocolo OLSR-MD a seleção de rotas entre o nó atual e um nó qualquer na rede terá como critério o menor retardo de transmissão total, isto é, a menor soma do retardo de transmissão de todos os saltos que compõem o caminho.

3.4.4 Vantagens e Desvantagens de Cada Métrica

Esta subseção tem por objetivo comparar as métricas utilizadas nas extensões do protocolos OLSR mostradas anteriormente, onde a Tabela 3.1 apresenta uma comparação entre as métricas sumarizando suas vantagens e desvantagens.

Tabela 3.1: Comparação das métricas

Métrica	Vantagem	Desvantagem
Hop Count	Sua simplicidade, por ser baseada no menor número de saltos	Não considera as características dos enlaces
ETX	Reduz a sobrecarga relacionado a obtenção de controle	Sondas para obtenção de informação não sofrem com as mesmas taxas de perda devido ao seu tamanho
ML	Elimina rotas com altas taxas de perdas	Considera enlaces de baixa qualidade por ser baseada no produto da probabilidade total
MD	Considera a diferença na sincronização dos relógios dos nós comunicantes	Escolhe rotas que são compostas de nós que estão no mesmo domínio de colisão

Capítulo 4

Estudo Sobre os Protocolos TCP e UDP em Redes em Malha sem Fio

O desenvolvimento de extensões para o protocolo OLSR, tem o objetivo de aprimorar suas características e conseqüentemente seu desempenho. Essas extensões tem seguido várias vertentes, como modificações na seleção do conjunto de nós MPR [Badis et al., 2003], cálculo de novas métricas para o roteamento de pacotes, como em [Cordeiro et al., 2007] e [Couto et al., 2003], dentre outras estratégias.

A maior parte destes protocolos existentes visa prover QoS para as diversas aplicações multimídia, ou seja, visam garantir que a rede consiga atender a demanda necessária de recursos para atender as necessidades das aplicações em questão, sendo que essas necessidades estão diretamente relacionadas a medidas de desempenho da rede como atraso, largura de banda, *jitter*, bloqueio, vazão, taxa de perda de pacotes, taxa de erro de bits, entre outras. Essa relação direta ocorre, pois as aplicações multimídia utilizam, em sua maioria, como protocolo da camada de transporte o protocolo UDP.

Esse cenário de priorização à QoS se montou a partir do momento que ocorreu o crescimento do acesso à Internet via redes sem fio, que motivou o desenvolvimento de várias aplicações e pequenos dispositivos que usam os recursos destas redes. Deste modo, os usuários de redes sem fio esperam obter os mesmos tipos de serviços que são oferecidos para os usuários das redes cabeadas [Chinta and Helal, 2003].

Dentro desse contexto ocorre o fato de que a maioria das propostas existentes acaba deixando de fora do seu foco as aplicações mais clássicas das redes de computadores, aquelas que utilizam o protocolo TCP como protocolo de transporte. De acordo com Thomson [Thomson et al., 1997], cerca de 80% do tráfego existente na Internet é provido pelo protocolo TCP. Isso é justificável a partir do momento que se analisa que grande parte das aplicações da Internet como *File Transfer Protocol* (FTP), *Hyper Text Transfer Protocol* (HTTP) e outras se utilizam do protocolo TCP para proverem seus serviços.

A partir de um levantamento bibliográfico verificou-se que não se encontrou um protocolo de roteamento que vise atender as características e necessidades de cada tipo de pacote. Sendo assim neste trabalho se propõe uma extensão para o protocolo OLSR que visa atender não somente QoS para as aplicações multimídia, mas que além disso consiga garantir o desempenho das aplicações mais tradicionais e mais utilizadas nas redes de computadores.

A pesquisa se iniciou com o estudo do desempenho dos dois principais protocolos da camada de transporte, o UDP e o TCP nas redes sem fio, e os principais problemas enfrentados por estes nesse tipo de rede. Além de se procurar relacionar as características

de cada um desses protocolos com as métricas existentes, para assim se ter um melhor desempenho das aplicações existentes, cada uma com suas particularidades.

4.1 Estudo sobre o protocolo TCP

O protocolo TCP é classicamente conhecido por ser um protocolo *unicast* orientado a conexão, que possui como base para sua transmissão confiável de dados os mecanismos controle de fluxo e de congestionamento da rede, técnicas que proporcionam um controle eficiente da sua janela de congestionamento [Postel, 1981]. Sendo assim este se tornou o protocolo clássico da Internet por prover confiabilidade na transmissão dos dados fim-a-fim, característica que são de vital importância até os dias atuais.

TCP foi desenvolvido originalmente para redes cabeadas, onde estas têm pequenas taxas de perda de pacotes, pois possuem baixa taxa de erro de bits [Bakre and Badrinath, 1995]. Sendo assim, quando ocorre perda de pacotes o protocolo assume que a perda ocorreu por congestionamento da rede. As conexões sem fio têm altas taxas de erros de bits, com muita variação da qualidade do sinal e outros fatores que, conseqüentemente são herdados pelas WMNs.

Com essas características, o protocolo TCP se torna mais eficiente a partir do momento que ele possui um menor número de perda de pacotes, fato que resulta em uma menor saturação da rede, pois quanto menor o número de perdas, menor será o número de retransmissões [Belakrishnan et al., 1995].

Dentre as medidas de desempenho existentes as que mais tendem a se adequar a essas características são a vazão, bloqueio, taxa de erros de bits e taxa de perda de pacotes. Relacionando-se essas medidas as métricas existentes, percebe-se que, teoricamente, a métrica ETX [Couto et al., 2003] é a que mais se aproxima a essas necessidades, pois expressa o número de retransmissões necessárias para que o pacote seja entregue e confirmado. Para se comprovar essa maior eficiência foram realizadas simulações, onde o cenário utilizado foi o *backbone mesh* existente na UFPA, ilustrado na Figura 4.1.

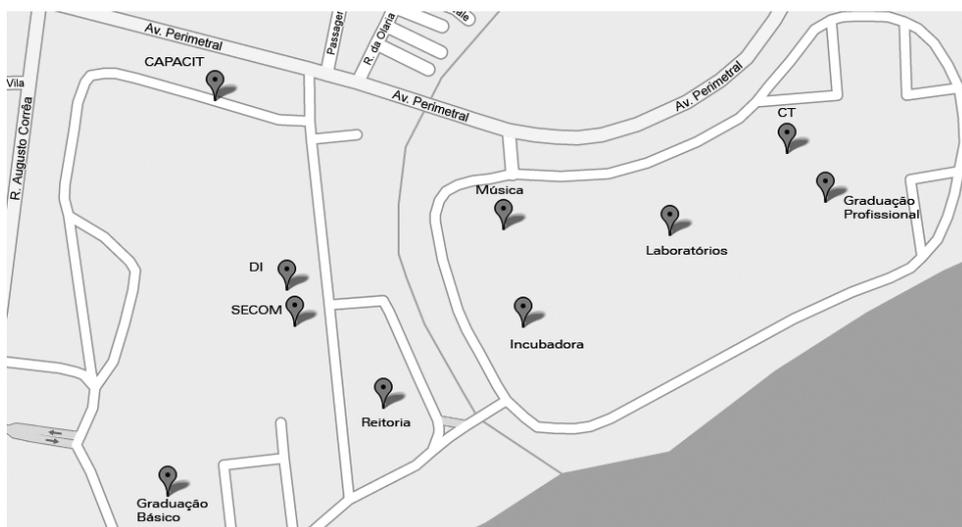


Figura 4.1: Backbone da WMN na UFPA

As simulações foram realizadas no NS-2 [Fall and Varadhan, 2007] utilizando diferentes sementes geradoras (*seeds*). Para a análise dos resultados foi utilizada a referência [Jain, 1991] como base para os cálculos. Dentre as várias versões do protocolo TCP proposta

na literatura, o TCP-Reno é o mais utilizado na Internet, sendo usado como protocolo padrão em vários sistemas operacionais, além disso, de acordo com Kim [Kim et al., 2006] o TCP-Reno é o TCP que apresenta os melhores resultados em redes que utilizam o protocolo OLSR. Portanto a partir desses fatos foi o utilizado o TCP-Reno nas simulações realizadas.

Tabela 4.1: Parâmetros de Simulação

Parâmetros	Valores	Descrição
Padrão IEEE	802.11b	Frequência no canal 3
Modelo de Propagação	Shadowing	Sombreamento
Antenas	Omnidirecional 18dB	Permite um maior alcance (360°)
Router's Carrier Sense Threshold	-76dBm	Pacotes recebidos acima deste limiar poderão ser interpretados
Router's Transmit Power	-80dBm	Pacotes acima deste limiar são recebidos
Frequência	2.422GHz (Canal 3)	Potencia utilizada nos roteadores WRT54G
Path Loss Exponent	1.59	Padrão de obstruções para ambiente outdoor, com sombreamento em áreas urbanas
Shadowing Deviation	5.4dB	Adequado a ambientes outdoor

A Tabela 4.1 mostra os parâmetros usados nas simulações, que tem por objetivo aproximar o máximo possível da rede real implantada na UFPA, representando as características da região e dos equipamentos. Os parâmetros de *Path Loss Exponent* e *Shadowing Deviation* foram utilizados de acordo com as medições apresentadas em [Moreira et al., 2008], os demais foram ajustados de acordo com os equipamentos existentes na rede e com o padrão 802.11 [IEEE, 1999].

A simulação foi constituída de quatro tráfegos TCP, envolvendo os seguintes pontos da rede em questão: Capacit → Graduação (Básico), Graduação (Profissional) → Música, Graduação (Básico) → Incubadora e Laboratórios → Secom. Cada simulação teve a duração de 50 unidades de simulação e foram efetuadas 50 vezes, para cada protocolo.

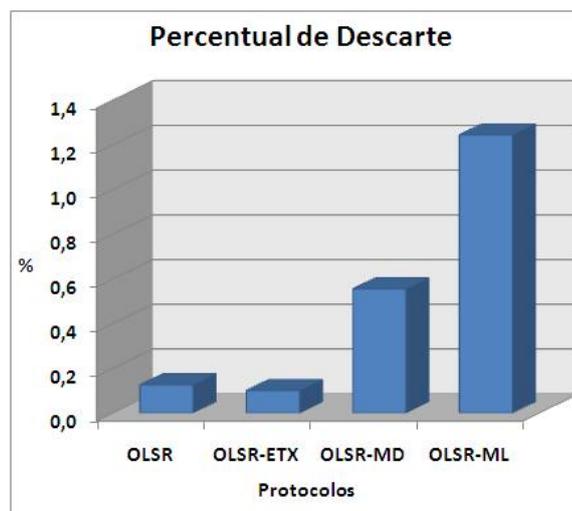


Figura 4.2: Percentual de Descarte dos Fluxos TCP

Os protocolos analisados a partir das simulações foram OLSR, OLSR-ETX, OLSR-MD e OLSR-ML. Para cada um dos protocolos analisou-se a probabilidade de bloqueio,

percentual de descarte e vazão.

A Figura 4.2 mostra os resultados sobre o percentual de descarte analisado de todos os protocolos, o que obteve o menor valor foi o protocolo OLSR-ETX, isto ocorreu devido ao fato do protocolo ter conseguido no geral efetuar um maior número de transmissões em relação às perdas ocorridas.

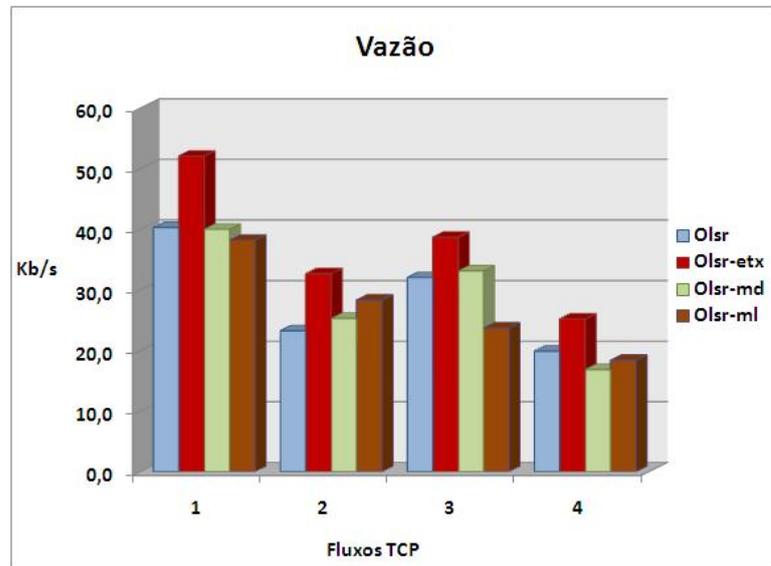


Figura 4.3: Vazão dos Fluxos TCP

A Figura 4.3 mostra a vazão que cada protocolo possui com relação a cada um dos fluxos efetuados, percebe-se que o protocolo OLSR original e a sua extensão utilizando a métrica ETX possuíram uma maior vazão com relação aos outros protocolos, sendo que no geral a extensão utilizando ETX obteve um resultado superior.

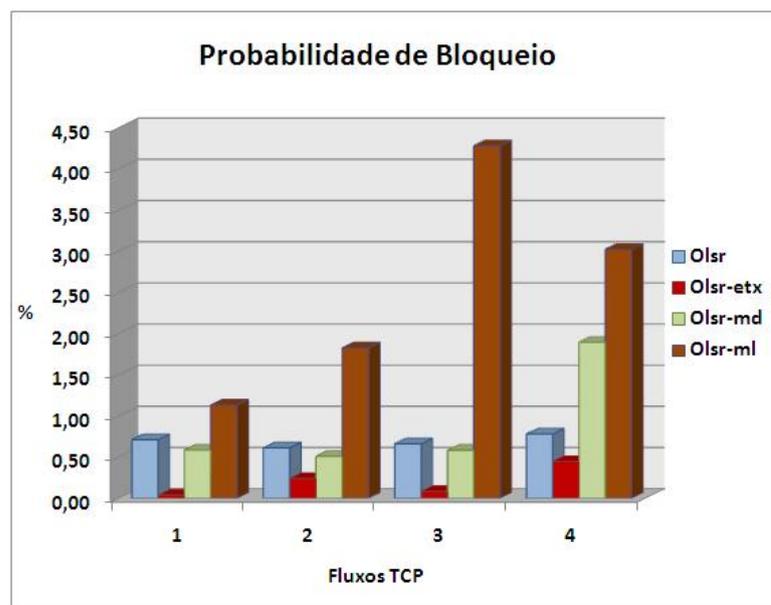


Figura 4.4: Probabilidade de Bloqueio dos Fluxos TCP

Os dados com relação a probabilidade de bloqueio são mostrados na Figura 4.4, onde se percebe que novamente as métricas OLSR e OLSR-ETX possuem um melhor resultado

com relação aos protocolos OLSR-MD e OLSR-ML, sendo que, assim como na análise da vazão, observa-se que o protocolo utilizando a métrica ETX alcança um desempenho melhor. Este melhor desempenho se deve ao fato de que o protocolo OLSR-ETX conseguiu escolher os enlaces na rede com melhor qualidade, com isso os tráfegos do OLSR-ETX possuíram um melhor desempenho em relação aos tráfegos efetuados pelos demais protocolos.

De um modo geral conclui-se, a partir dos dados coletados, que a métrica ETX possui um desempenho superior aos demais protocolos, quando se trata do protocolo TCP. Assume-se então que os pacotes que possuem como protocolo de camada de transporte o protocolo TCP devem ser roteados baseados na métrica ETX [Couto et al., 2003].

4.2 Estudo sobre o protocolo UDP

O protocolo UDP é conhecido por ser o protocolo da camada de transporte que visa ter o “melhor esforço”, ou seja, abre mão de mecanismo de confiabilidade, para proporcionar uma entrega de pacotes com maior rapidez. Alguns dos mecanismos não utilizados pelo protocolo são: controle de congestionamento, sequenciamento de pacotes, estabelecimento de comunicação e outros.

Ao deixar de utilizar esses mecanismos o protocolo UDP se torna um protocolo eficiente e ágil, mas perde a confiabilidade. Com isso ele se torna apropriado às aplicações que precisam da chegada dos pacotes o mais rápido possível e na “hora certa”, pois os pacotes que não se encaixarem nessas características serão descartados. Alguns exemplos de aplicação são Vídeo-conferências, VoIP, dentre outras.

O uso do protocolo UDP teve um grande crescimento nos últimos anos com o surgimento e popularização de aplicações multimídia na Internet e em redes locais, aplicações estas que cada vez mais tendem a se tornar mais comuns no dia-a-dia. Então surgiu uma preocupação com a capacidade das redes em oferecer QoS à essas aplicações [Ying et al., 2003].

Dentro deste contexto aplicações que utilizam o protocolo UDP se tornam mais eficientes a partir do momento que possuem os pacotes com baixo retardo, e com uma pequena variação destes, além de que ocorra uma perda de pacotes pequena, pois os mesmos não serão retransmitidos, apesar de que geralmente, estas aplicações são relativamente tolerantes à perdas.

Percebe-se então que as medidas de desempenho que mais tendem a se adequar às necessidades citadas são o atraso, o *jitter*, a vazão, o percentual de descarte e a probabilidade de bloqueio. Nota-se então que a métrica *Minimum Delay* (MD) [Cordeiro et al., 2007] é a que mais se aproxima dessas necessidades, pois esta lida diretamente com os valores de atraso dos enlaces.

Para se comprovar essa afirmação realizou-se simulações, onde o cenário utilizado foi novamente o existente na UFPA (ilustrado na figura 4.1). Foram usados os mesmos parâmetros citados nas simulações anteriores (mostrados na Tabela 4.1).

As simulações foram constituídas de 6 tráfegos UDP usando *Constant Bit Rate* (CBR), o que caracteriza 3 chamadas VoIP, sendo que estas utilizaram pacotes com 40 bytes de tamanho e uma taxa de transmissão de 8Kb/s, essa configuração foi usada para se adequar ao codec G.729 [Balam and Gibson, 2007]. No NS-2 as chamadas VoIP se caracterizam de 2 tráfegos UDP, por se tratar de uma aplicação bidirecional.

Os tráfegos produzidos foram entre os pontos: Capacit ↔ Incubadora, Graduação (Básico) ↔ Música e Graduação (Profissional) ↔ Secom.

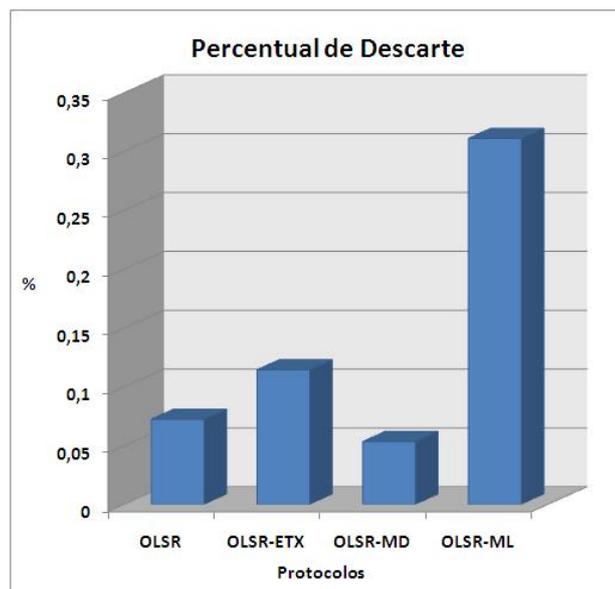


Figura 4.5: Percentual de Descarte dos Fluxos UDP

Na Figura 4.5 se têm os resultados sobre a percentual de descarte analisada de todos os protocolos. Observa-se que o protocolo OLSR-MD possuiu o menor índice dentre todos os protocolos, devido a sua maior eficiência nas transmissões com relação às perdas de pacotes ocorridas.

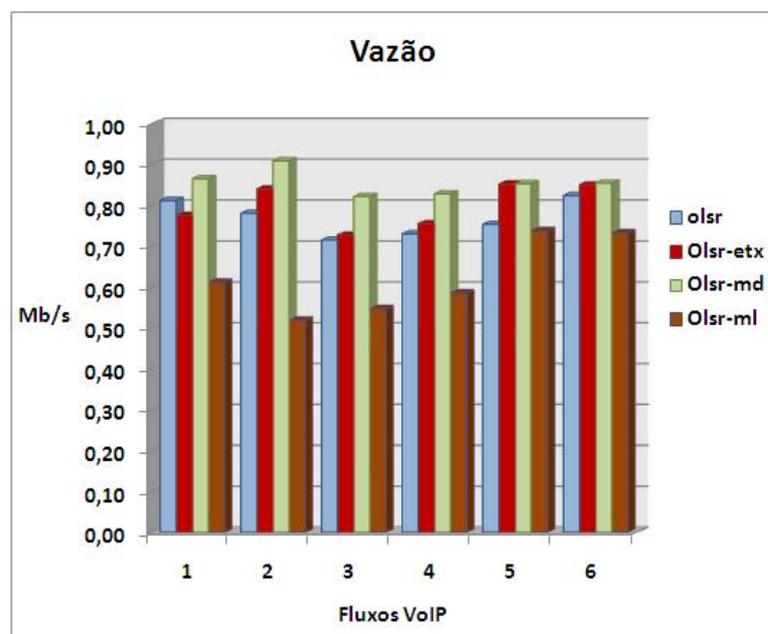


Figura 4.6: Vazão dos Fluxos UDP

A Figura 4.6 mostra-se o resultado da vazão que cada protocolo possui para cada um dos fluxos UDP, observa-se que, com exceção do protocolo OLSR-ML, todos os protocolos tiveram uma vazão considerável boa. Entretanto, o protocolo OLSR-MD obteve uma vazão relativamente maior diante dos demais protocolos.

Os dados de probabilidade de bloqueio são mostrados na figura 4.7, onde novamente percebe-se que, com exceção do protocolo OLSR-ML, todos os protocolos tiveram um

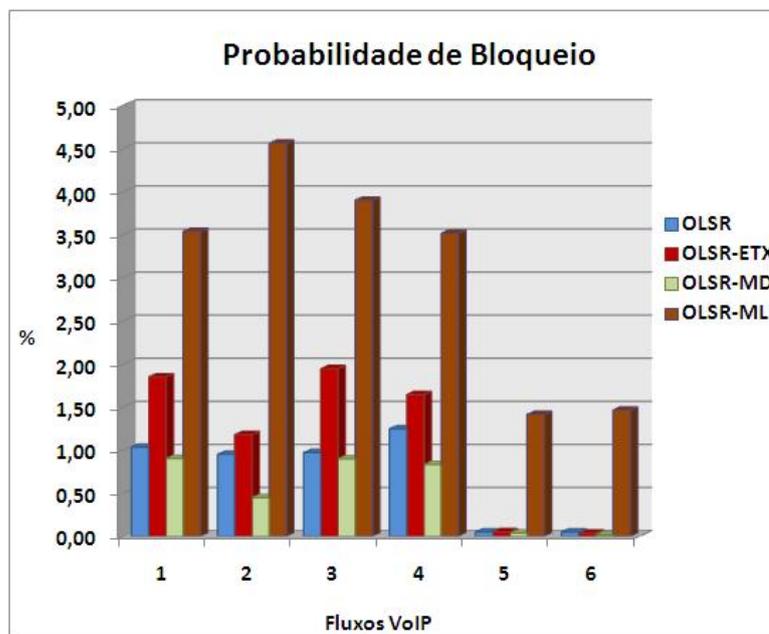


Figura 4.7: Probabilidade de Bloqueio dos Fluxos UDP

bloqueio semelhante, onde novamente o protocolo OLSR-MD obteve um desempenho superior.

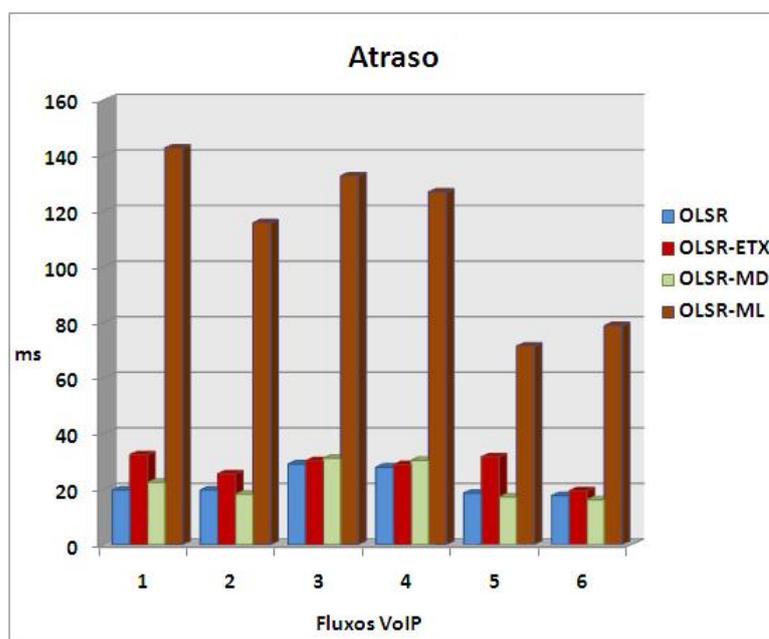


Figura 4.8: Atraso dos Fluxos UDP

As medições de atraso efetuadas para cada transmissão das chamadas VoIP são mostradas na Figura 4.8, mostrando que os protocolos OLSR, OLSR-ETX e OLSR-MD obtiveram resultados semelhantes, onde cada um obteve melhores resultados de acordo com o fluxo em questão.

Sendo que de modo geral o protocolo OLSR original obteve uma pequena vantagem com relação aos outros dois protocolos, devido as rotas escolhidas serem as com menor número de saltos, fato que reduz o atraso fim-a-fim.

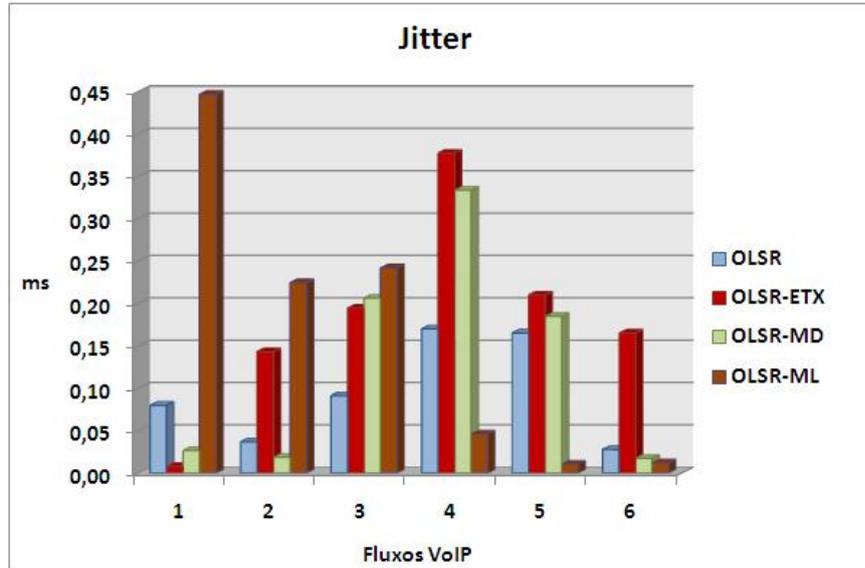


Figura 4.9: Jitter dos Fluxos UDP

Assim como ocorreu na análise sobre o atraso, os dados sobre o *jitter*, mostrados na Figura 4.9, nos mostram um desempenho relativamente parecido entre os protocolos, principalmente entre os protocolos OLSR-ETX e OLSR-MD, mas o que mais uma vez obteve um melhor desempenho foi o protocolo OLSR original, com uma menor variação do atraso.

Ao se concluir a análise do desempenho dos protocolos com relação às diversas métricas, percebe-se que o protocolo que obteve o melhor desempenho relacionado à entrega eficiente de pacotes UDP foi o protocolo OLSR-MD, enquanto que ao nível de atraso e variação deste (*jitter*) o protocolo OLSR original foi o que teve um melhor desempenho.

O fato de o protocolo OLSR ter tido esse melhor desempenho pode ser explicado pelo fato de que os pontos escolhidos para as transmissões possuíam distâncias que possibilitavam que com poucos saltos de distancia os pacotes pudessem ser entregues, o que se torna algo vantajoso para o protocolo.

Conclui-se que, com tráfegos de maior distância, o desempenho do protocolo não deve ser tão bom, como mostrado em [Moreira et al., 2008]. Além disso, deve-se perceber que o protocolo OLSR-MD obteve valores relativamente melhores que o protocolo OLSR referentes à entrega de pacotes e teve um desempenho muito próximo ao OLSR se considerado o atraso e *jitter*. A partir disso então se conclui que os pacotes que possuem como protocolo de camada de transporte o protocolo UDP devem ser roteados baseados na métrica MD.

A conclusão do estudo sobre o desempenho de cada um dos dois tipos de pacote, UDP e TCP de acordo com a métrica utilizada pelo protocolo, nos mostrou que a melhor opção para pacotes que usam o protocolo TCP é utilizar a métrica ETX para o roteamento dos mesmos, assim como a melhor opção para os pacotes que usam o protocolo UDP é o uso da métrica MD.

Capítulo 5

Extensão para o protocolo OLSR baseado na escolha dinâmica de métricas

Baseado nos resultados do capítulo anterior que comprovaram que o protocolo TCP possui melhor desempenho com a métrica ETX e que o protocolo UDP com a métrica MD consegue obter melhores resultados, desenvolveu-se OLSR-DC, para isso foram feitas as seguintes modificações no protocolo OLSR, mostradas nas subseções seguintes.

5.1 Ajuste do cálculo da métrica ETX

Originalmente, para fazer seu cálculo, o protocolo OLSR-ETX faz a medição da perda dos pacotes OLSR que foram recebidos dos vizinhos. À medida que as mensagens de HELLO dos vizinhos são recebidas, por padrão a cada 2 segundos [Clausen and Jacquet, 2003], o protocolo controla as perdas ocorrentes e assim determina a qualidade do enlace para o determinado nó.

Quando se completam 10 pacotes enviados se tem um número de pacotes suficiente para determinar a taxa de perda de pacotes existente entre o nó atual e cada vizinho. O protocolo usa um esquema de janela deslizante de 10 pacotes, ou seja, o cálculo do ETX é feito a partir dos últimos 10 pacotes que foram recebidos, ou que deveriam ter sido recebidos, computando assim os pacotes que foram recebidos com sucesso e os perdidos para gerar a qualidade do enlace.

Se, por exemplo, 3 em cada 10 pacotes são perdidos em seu caminho do vizinho até o nó atual, então se tem uma perda de pacotes $3 / 10 = 0,3 = 30\%$, ao mesmo tempo, 7 em cada 10 de pacotes que foram enviados com sucesso.

A estratégia utilizada pelo protocolo OLSR-ETX para analisar as mensagens recebidas é a seguinte: usasse uma estrutura de controle do número de identificação (ID) dos pacotes recebidos, ou seja, inicialmente o protocolo espera um pacote com o ID inicial, quando a primeira mensagem é recebida, o protocolo acessa seu cabeçalho e compara o ID do pacote com o ID esperado.

Caso o ID do pacote seja o esperado, o protocolo atualiza o ID esperado e processa o pacote em questão. Caso o ID seja diferente do esperado, o protocolo assume que o pacote com ID esperado foi perdido, então este adianta a sequência de pacotes esperado até encontrar o ID correspondente ao do pacote recebido, a fim de determinar o número de pacotes perdidos.

Se o ID não for encontrado até o limite estipulado (`OLSR_MAX_SEQ_NUM`), que por padrão é 65535 [Clausen and Jacquet, 2003], a sequência é reiniciada a partir do valor original, ID igual a 0. Após esta busca pelo ID, o número de pacotes perdidos é atualizado, com isso o valor da qualidade do enlace pode ser calculado e o pacote processado.

Exemplificando o cálculo explanado acima, se o protocolo espera um ID igual a 10 e é recebido um pacote com ID igual a 13, o protocolo incrementa o ID esperado até o recebido, identificando assim que foram perdidos 3 pacotes, os pacotes com ID 10, 11 e 12. Gera-se então o valor de ETX a partir dos pacotes 13 a 4, considerando 7 pacotes recebidos e 3 perdidos.

Por sua vez, a métrica MD, utiliza a técnica *Ad Hoc Probe* para o cálculo do retardo nos enlaces. No *AdHoc Probe*, pares de pacotes de tamanho fixo são enviados em uma direção única, com o tempo de envio registrado em cada pacote transmitido. Com a informação sobre o momento em que o pacote foi enviado, o destino pode então calcular o atraso do enlace, e por consequência a capacidade do enlace em um sentido. O receptor do pacote mede o retardo do enlace de cada pacote como a diferença entre o tempo de recebimento de cada um dos pacotes do par [Cordeiro et al., 2007].

Para controlar essa recepção de pares de pacotes o protocolo OLSR-MD utiliza, além do ID do pacote OLSR, um ID extra, para proporcionar ao receptor do pacote a capacidade de identificar qual pacote, do par, foi enviado primeiro. Sendo assim ambos os pacotes do par possuem o ID do pacote OLSR iguais, mas com ID extra diferentes.

Dentro do contexto das duas métricas explicadas acima, ao se utilizar o cálculo das duas ao mesmo tempo foi identificado um problema relacionado diretamente ao cálculo da métrica ETX, onde o cálculo da métrica MD impede o cálculo eficiente da métrica ETX.

Como citado acima, a métrica ETX utiliza uma estrutura para identificação de perda de pacotes baseada no ID do pacote OLSR recebido e o ID do pacote esperado, mas com a utilização da técnica *AdHoc Probe* para cálculo da métrica MD, os pacotes recebidos são duplicados, sendo assim eles possuem ID iguais.

Com isso no momento em que o segundo pacote do par é recebido, o protocolo espera um ID maior na sequência, ou seja, se em um determinado momento o segundo pacote do par que possui ID igual 0 chega, o protocolo espera o pacote com ID igual a 1. Quando esta situação ocorre o protocolo supõe, que o pacote esperado foi perdido, sendo assim o protocolo inicia uma busca pelo ID do pacote recebido, incrementando sequencialmente o valor do ID esperado.

Devido ao fato do pacote recebido possuir o ID igual ao anterior, a busca feita pelo protocolo não surte efeito, portanto o protocolo continua com a busca até o momento em que o valor limite estipulado for alcançado, então o protocolo reinicia a contagem e encontra o ID em questão.

Essa realidade torna o cálculo da métrica extremamente ineficiente, pois para todo o par de pacotes que são recebidos o fato explanado acima ocorre, ou seja, para cada par de pacotes no momento em que o segundo é recebido o protocolo realiza a busca, que dentro do contexto das duas métricas se torna inútil.

Exemplificando a situação acima, se um determinado par de pacotes com ID 0 for recebido, o primeiro pacote vai ser calculado corretamente, pois o receptor está esperando um pacote com ID igual a 0, número inicialmente esperado. Mas no momento em que o segundo pacote do par é recebido, o receptor espera um pacote com ID igual a 1, o que não ocorre, com isso o protocolo faz uma busca seqüencial até o valor limite, que por padrão é 65535, até que este reinicie a contagem, voltando o valor esperado a 0, somente

depois disso que o pacote será processado. A Figura 5.1 mostra a situação explicada.

Ilustração do Problema Identificado



Figura 5.1: Problema Identificado

Depois de identificado o problema citado, foram feitas estudos para identificar qual seria a solução mais viável, para que se pudesse não prejudicar o cálculo da métrica MD e ao mesmo tempo que esta não influenciasse de maneira negativa o protocolo em geral.

Decidiu-se então utilizar uma estrutura de controle extra para a sequência de pacotes recebidos. No momento em que um pacote vai ser processado para cálculo da métrica ETX, é verificado se este possui o mesmo ID do último pacote recebido. Caso este possua o mesmo ID o pacote não é considerado para o cálculo da métrica ETX, somente para a métrica MD. Caso o pacote possua um ID distinto do último recebido, este é processado normalmente.

A solução acima foi adotada devido esta conseguir solucionar o problema identificado e, além disso, não modifica o cálculo original da métrica ETX, mantendo sua forma original e não influenciando no cálculo da métrica MD.

5.2 Modificações no Algoritmo de Roteamento

Após as adaptações feitas no protocolo OLSR para se fazer os cálculos de ambas as métricas, ETX e MD, foram realizadas alterações no algoritmo de roteamento afim de que este possa realizar a busca dos melhores caminhos para ambas as métricas mencionadas.

O algoritmo utilizado como base foi o conhecido algoritmo de Dijkstra [Noto and Sato, 2000], amplamente utilizado para se determinar rotas com menor custo a partir de informações sobre a topologia da rede, neste caso a partir das informações de ETX e MD de cada enlace na rede.

O algoritmo de Dijkstra foi desenvolvido para calcular a rota com menor custo em grafos utilizando apenas uma medida para sua avaliação, fato que torna o algoritmo impróprio para a proposta deste trabalho. Vale lembrar que o OLSR-DC visa encontrar os melhores caminhos para os nós da rede, um baseado na métrica ETX e o outro na métrica MD. Sendo assim foram estudadas algumas possibilidades de adaptação para que se pudesse utilizar o algoritmo de Dijkstra no OLSR-DC, as quais serão explanadas a seguir.

A primeira possibilidade seria utilizar o algoritmo duas vezes, onde cada uma das vezes o algoritmo fosse calcular as rotas baseadas em uma métrica. Analisou-se então, que esta alternativa geraria um maior processamento de cálculo, além de que necessitaria de uma maior alocação de memória do sistema. Esses fatos poderiam tornar inviável a utilização do protocolo, futuramente, em um cenário real, pois este poderia provocar uma sobrecarga

nos equipamentos em redes muito densas. Apesar de que com a constante inovação de tecnologias de *hardware* e *software* isto poderia acabar não sendo um problema real.

A segunda opção seria realizar modificações no algoritmo de Dijkstra para que assim este pudesse calcular as melhores rotas para ambas as métricas, sem que fosse preciso executar o algoritmo duas vezes, sendo assim o cálculo gerado pela execução do algoritmo seria quase equivalente ao gerado pelo algoritmo originalmente, possuindo apenas mais verificações.

Dentre as duas opções citadas foi escolhida, apesar de ter uma maior complexidade para a sua implementação, a segunda opção, devido esta parecer mais viável para a implementação e desempenho em um cenário real. Para fazer com que o algoritmo de Dijkstra calculasse as rotas para os nós presentes na rede sem ser executado duas vezes, foram utilizadas listas e variáveis de controle, que auxiliaram na organização dos dados para que fosse possível ser feito esse cálculo de maneira eficiente.

Sendo assim, o algoritmo passou a possuir 2 lista de nós não verificados e 2 listas de nós já computados, uma lista para a métrica MD e outra para a métrica ETX. O número de *loops* do algoritmo se mantém os mesmos, pois o número de nós para ambas as métricas é o mesmo.

Além das alterações citadas acima, foi efetuada mais uma mudança, esta no próprio paradigma do algoritmo. Originalmente o algoritmo de Dijkstra quando se depara com rotas que possuem o mesmo valor, de acordo com a métrica utilizada, este escolhe a primeira que foi encontrada, o que significa que nas WMNs, que a será escolhida a rota com menor número de saltos.

Após estudos feitos com relação a rotas que possuem o mesmo valor de ETX no enlace, concluiu-se que caminhos com o mesmo valor de ETX e com número de saltos diferentes deve-se escolher aquele que possui o maior número de saltos. Esta escolha foi adotada, pois a partir do momento que uma rota com grande número de saltos possui o mesmo valor de ETX que uma com menor número, esta com maior número de saltos em cada um dos segmentos da rota tem uma baixa probabilidade de perda de pacotes, menor valor de ETX, o que se torna algo mais interessantes no foco da eficiência na entrega dos pacotes.

Ao se analisar o mesmo contexto para o foco do atraso, observa-se que os caminhos terem o mesmo valor de atraso com número de saltos distintos não influência de maneira direta, pois o atraso é uma métrica que não depende de probabilidades, é um valor que é calculado de maneira direta. Sendo assim adotou-se que quando forem encontrados caminhos com mesmo valor de atraso, será escolhido aquele que possui uma menor probabilidade de perda, ou seja, aquele que tem um menor valor de ETX.

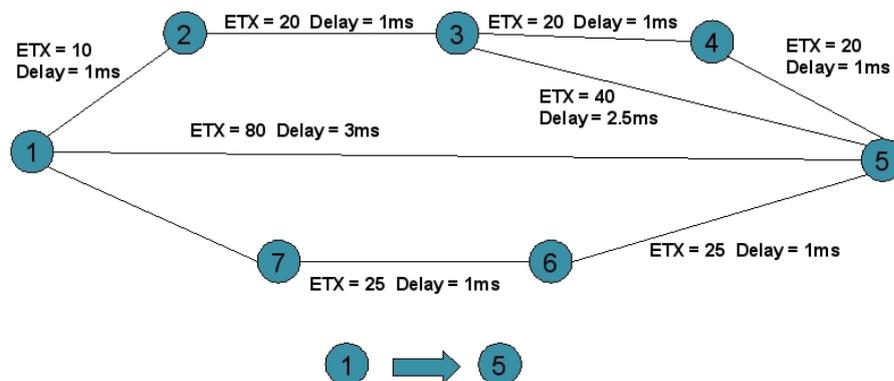


Figura 5.2: Diferença no roteamento entre os protocolos

Para ilustrar a diferença causada devido as alterações feitas no algoritmo, a Figura 5.2

mostra uma topologia com dados fictícios dos enlaces. As Tabelas 5.1 e 5.2 mostram como seria o roteamento feito pelos protocolos OLSR, OLSR-ETX, OLSR-MD e OLSR-DC, na situação mostrada na Figura 5.2 para os protocolos TCP e UDP.

Tabela 5.1: Rotas Seleccionadas pelos Protocolos para o Protocolo TCP

Protocolo	TCP	ETX	Delay
OLSR	1 → 5	80	3ms
OLSR-ETX	1 → 2 → 3 → 5	70	4.5ms
OLSR-MD	1 → 5	80	3ms
OLSR-DC	1 → 2 → 3 → 4 → 5	70	4ms

Percebe-se que, como mostra a Tabela 5.1, os protocolos OLSR-ETX e OLSR-DC escolheriam a rota com menor valor de ETX, entretanto a rota do protocolo OLSR-ETX, apesar de possuir um menor número de saltos, utiliza o enlace 3 → 5, que possui um alto valor de ETX, possuindo assim uma alta chance de perda do pacote. Por outro lado o protocolo OLSR-DC escolheria a rota com os enlaces 3 → 4 e 4 → 5 que possuem a metade da probabilidade de perda do pacote.

Tabela 5.2: Rotas Seleccionadas pelos Protocolos Para o Protocolo UDP

Protocolo	UDP	ETX	Delay
OLSR	1 → 5	80	3ms
OLSR-ETX	1 → 2 → 3 → 5	70	4.5ms
OLSR-MD	1 → 5	80	3ms
OLSR-DC	1 → 7 → 6 → 5	75	3ms

A Tabela 5.2, mostra que os protocolos OLSR, OLSR-MD e OLSR-DC, escolheriam as rotas com menor valor de atraso. Todavia, os protocolos OLSR e OLSR-MD escolheriam a que possui uma maior chance de perda do pacote (maior valor de ETX), por outro lado o protocolo OLSR-DC escolheria a rota que possui o mesmo atraso só que com uma menor chance de perda.

5.3 Modificações na tabela de roteamento

Originalmente o protocolo OLSR efetuava a inserção das rotas descobertas na tabela de roteamento de acordo com o número de saltos que cada uma delas efetuava para alcançar o nó correspondente ao seu destino na rede, assim mantendo a coerência usada [Clausen and Jacquet, 2003].

Esta estratégia de inserção de acordo com o número crescente de saltos continuou sendo utilizada no OLSR-DC, pois se verificou que esta não interferia negativamente na proposta. Primeiramente são inseridas as rotas correspondentes à métrica ETX e posteriormente são inseridas as rotas relativas à métrica MD.

A tabela de roteamento do protocolo OLSR originalmente possui quatro campos [Clausen and Jacquet, 2003]:

- R.dest_addr: nó a qual as informações são referentes;
- R.next_addr: nó da rede o qual é preciso enviar o pacote para que se possa alcançar o destino em questão;

- R_dist: número de saltos que são necessários para alcançar o destino em questão;
- R_iface_addr: endereço da interface de saída para o nó destino.

Na extensão proposta, o OLSR-DC, a tabela de roteamento manteria somente o campo R_dest_addr da tabela original, excluindo os demais, para assim acrescentar os seguintes campos:

- R_next_addr_tcp: nó da rede o qual é preciso enviar o pacote para que se possa alcançar o destino em questão, de acordo com a rota estabelecida que tenha o menor valor de ETX;
- R_next_addr_udp: nó da rede o qual é preciso enviar o pacote para que se possa alcançar o destino em questão, de acordo com a rota estabelecida que tenha o menor atraso, valor da métrica MD;
- R_dist_tcp: número de saltos que são necessários para alcançar o destino em questão, de acordo com a rota escolhida, que possui a melhor valor de ETX;
- R_dist_udp: número de saltos que são necessários para alcançar o destino em questão, de acordo com a rota escolhida, que possui o menor valor de atraso;
- R_iface_addr_tcp: endereço da interface de saída correspondente à rota com melhor valor de ETX para o nó destino;
- R_iface_addr_udp: endereço da interface de saída correspondente à rota com o menor atraso para o nó destino.

No OLSR-DC foi mantida a estratégia do OLSR original de recalculando a tabela de roteamento toda a vez que ocorrerem modificações nas informações locais dos enlaces assim como também no conjunto de informações sobre a topologia, já que a tabela é constituída de dados obtidos a partir dessas informações.

5.4 Modificações no encaminhamento de pacotes

O protocolo OLSR ao receber os pacotes efetua as seguintes etapas [Clausen and Jacquet, 2003]:

- Acessa-se o cabeçalho do pacote em questão, para retirar as informações que serão necessárias;
- Verifica-se se o pacote tem como emissor o próprio nó e se este já foi encaminhado alguma vez, fatos que caracterizariam um *loop* de roteamento, se o *loop* for detectado o pacote é descartado;
- Após a etapa anterior, verifica-se se o pacote é um pacote OLSR, pois estes têm um processamento específico, se o pacote em questão for um pacote de dados o processamento deste continua;
- Sendo o pacote em questão de dados, é decrementado o seu valor de *Time To Live* (TTL), sendo que se o valor resultante desse decremento for zero, o pacote é descartado;

- Depois de cumpridas as etapas anteriores, e o pacote ter atendido todas as verificações, este então passa para o processo de encaminhamento de pacotes de dados.

Até essa etapa a extensão proposta neste trabalho, a OLSR-DC, não fez modificações, utilizando a estratégia proposta pelo OLSR original. Já no processo de encaminhamento de dados foram realizadas adequações para a proposta deste artigo. A seguir serão mostrados os passos realizados originalmente pelo protocolo OLSR para o encaminhamento de pacotes [Clausen and Jacquet, 2003]:

- Acessa-se o cabeçalho do pacote em questão, para retirar as informações que serão necessárias para as etapas seguintes;
- A partir das informações retiradas, verifica-se se o pacote é endereçado para o nó em questão ou se é um pacote com destino *broadcast*, caso este seja para o nó em questão, o pacote então é enviado para as camadas superiores, caso não seja, o processo continua;
- Depois da verificação acima, procura-se na tabela de roteamento se o destino do pacote se encontra na tabela, caso o destino em questão não seja encontrado o pacote é descartado, caso haja o destino na tabela, o processamento continua;
- Neste ponto se retira as informações necessárias da tabela, como por exemplo, o próximo salto que o pacote tem q realizar, adicionando-as ao pacote em questão.
- Então finalmente o pacote é encaminhado para as camadas inferiores, onde será enviado ao seu próximo salto.

Dentro da realidade mostrada acima foram efetuadas modificações no intuito de fazer com que a escolha do próximo salto, e conseqüentemente do caminho, que o pacote irá percorrer seja feito de acordo com seu tipo de pacote, no nível de camada de transporte, e a métrica a ser utilizada, informações as quais já foram explanadas anteriormente.

Sendo assim foram realizadas basicamente duas modificações nas etapas mostradas anteriormente. Primeiramente, na etapa de retirada de informações foi feito com que, além das informações usadas originalmente, fosse acessado também o campo que informa o tipo de pacote utilizado na camada de transporte, para que assim essa informação pudesse ser utilizada adiante.

Posteriormente, na etapa de verificação do próximo salto referente ao destino, foi implementado para que o campo da tabela de onde se retira a informação seja acessado de acordo com o tipo do pacote verificado anteriormente, ou seja, pacotes que utilizam UDP como protocolo de camada de transporte utilizem o próximo salto referente à rota que possuir o menor atraso, o campo `R_next_addr_udp` da tabela, e os pacotes TCP usem o próximo salto com rota que tenha a melhor valor de ETX, o campo `R_next_addr_tcp`.

Com isso, consegue-se que o pacote seja roteado de acordo com a métrica que possibilitará este obter um maior desempenho na rede, no nosso caso pacotes TCP com menor probabilidade de perda e pacotes UDP com um menor atraso ate o destino em questão.

5.5 Seleção da métrica a ser utilizada para a seleção do conjunto de nós MPR

Em sua versão original o protocolo OLSR, um nó seleciona os nós vizinhos que fazem parte do seu conjunto MPR de uma maneira que ele possa cobrir todos os seus vizinhos

de dois saltos de distância, que tenham comunicação simétrica com cada um dos nós seus nós vizinhos diretos, e que estejam dispostos a encaminhar pacotes em benefício da rede, característica esta determinada pelo parâmetro *willigness* [Clausen and Jacquet, 2003].

Um efeito negativo dessa abordagem é a probabilidade de nós que possam levar a rotas com maior estabilidade e vazão de dados serem excluídos do conjunto de nós MPR de outros nós, por não apresentar um parâmetro *willigness* favorável.

As mensagens encaminhadas através dos nós que fazem parte do conjunto MPR, as mensagens de controle, que são endereçadas a todos os nós da rede, são incorporadas a pacotes UDP para a sua transmissão através da rede, com isso herdamos algumas características do protocolo, como o cabeçalho, não possuem retransmissão nem confirmação dos pacotes transmitidos. Apesar de estas mensagens possuírem essas características citadas, estes não possuem a sensibilidade ao atraso, como a maioria das aplicações que utilizam pacotes UDP.

Ao se utilizar a métrica MD para ser incorporado a um dos critérios para se selecionar o conjunto de nós MPR, as informações obtidas dos nós teriam uma maior sincronização, assim os nós poderiam trabalhar com as informações mais atualizadas possível, já que esta métrica comprovadamente proporciona a escolha de rotas com menor atraso.

Apesar disso, a perda dessas informações é algo mais grave do que o fato destas chegarem desatualizadas, pois a partir do momento que uma mensagem é perdida, esta não será retransmitida, com isso esta passará por um longo período sem a atualização das informações, período este de espera até o envio de uma nova mensagem.

Portanto decidiu-se adotar a métrica ETX para fazer parte do critério de seleção do conjunto de nós MPR, pois esta consegue desempenhar um melhor papel na entrega das mensagens, como já foi mostrado anteriormente neste trabalho. Além disso as mensagens de controle tem prioridade, com isso não há possibilidade do pacote ser descartado por fila. Sendo assim os valores de ETX também serão utilizados no critério de seleção para o conjunto de nós MPR.

Capítulo 6

Análise dos Resultados

Este capítulo tem como objetivo mostrar o desempenho do protocolo OLSR-DC, proposto neste trabalho. Assim como feito nas seções anteriores, analisou-se o desempenho da proposta usando-se o NS-2 [Fall and Varadhan, 2007]. As simulações compararam o desempenho do OLSR e suas extensões para um cenário real. O cenário utilizado baseou-se no *backbone* da WMN que se encontra parcialmente implantada na Universidade Federal do Pará (UFPA) como ilustrado na Figura 6.1.



Figura 6.1: Backbone da WMN proposto para a UFPA

A Tabela 6.1 mostra os parâmetros usados nas simulações, que tem por objetivo aproximar a simulação o máximo possível da rede real, representando as características da região e dos equipamentos utilizados. Os parâmetros *Path Loss Exponent* e *Shadowing Deviation* foram utilizados de acordo com as medições apresentadas em [Moreira et al., 2008].

Os parâmetros de *Router's Carrier Sense Threshold* e *Router's Transmit Power* foram baseados no padrão 802.11 [IEEE, 1999]. Os demais valores foram usados para representar as antenas e os roteadores usados na WMN presentes na UFPA.

Tabela 6.1: Parâmetros das Simulações com 801.11g

Parâmetros	Valores
Padrão IEEE	802.11g
Modelo de Propagação	<i>Shadowing</i>
Antenas	Omnidirecional 18dB
<i>Router's Carrier Sense Threshold</i>	-76dBm
<i>Router's Transmit Power</i>	-80dBm
Potência de Transmissão	17 dBm (WRT54G)
Frequência	2.422GHz (Canal 3)
<i>Path Loss Exponent</i>	1.59
<i>Shadowing Deviation</i>	5.4dB

Foram realizadas vinte simulações utilizando diferentes sementes geradoras, para cada protocolo em questão: OLSR, OLSR-ETX, OLSR-MD e OLSR-DC, o protocolo OLSR-ML foi excluído dessa análise devido ao seu fraco desempenho no cenário utilizado, como foi mostrado nas seções anteriores. A Tabela 6.2 mostra a configuração dos fluxos utilizados nas simulações, todas as simulações tiveram duração de 100 unidades de simulação.

Tabela 6.2: Configuração dos Fluxos

Fluxo	Fonte	Destino	Início	Fim	Tráfego
1	7	3	17	67	TCP - Reno
2	9	4	18	68	TCP - Reno
3	0	5	5	55	UDP - CBR
4	5	0	5	55	UDP - CBR
5	8	1	6	56	UDP - CBR
6	1	8	6	56	UDP - CBR
7	3	6	10	90	Vídeo

Os fluxos UDP possuíram um *Bit Rate* de 8Kb/s e pacotes com tamanho de 40 bytes (RTP + UDP + Payload), afim de se representar o codec G.729 [Balam and Gibson, 2007]. Os fluxos TCP foram caracterizados como aplicações FTP seguindo o modelo de Pareto, possuindo uma taxa de 200k, pacote com 210 bytes de tamanho e 500 ms de duração da rajada [Moreira et al., 2008].

O tráfego de vídeo em questão foi feito a partir da ferramenta Evalvid [eva, 2009] que permite o controle da qualidade de vídeo em um ambiente de simulação e um vídeo real chamado “Bridge (close)” [cif, 2009]. Este vídeo possui 2000 *frames* no formato YUV, com uma dimensão de 352x288, o qual foi compactado através do codec MPEG-4 e enviado a uma taxa de 30 frame/s. Cada *frame* foi fragmentado em blocos de 1024 bytes, onde o pacote possuiu um tamanho de 1052 bytes.

As simulações mostram o impacto dos protocolos na qualidade do vídeo transmitido, avaliando os tráfegos não somente do ponto de vista da rede (QoS), mas também em relação ao ponto de vista do usuário (QoE). A análise baseada na percepção do usuário nos mostra o desempenho das aplicações multimídias na rede.

A configuração dos fluxos usada tem como objetivo mostrar o desempenho dos protocolos não somente do ponto de vista da rede, mas também para verificar o impacto dos protocolos com relação à percepção do usuário, para isto foi utilizada a transmissão de vídeo em concorrência com os demais fluxos, situação comum nas WMNs.

6.1 Análise do Ponto de Vista da Rede (QoS)

Os gráficos de probabilidade de bloqueio e vazão foram divididos, nas chamadas VoIP, fluxos TCP e transmissão de vídeo, para se facilitar a visualização dos mesmos. Já os gráficos de atraso e *jitter* são mostrados somente os dados referentes às chamadas VoIP e a transmissão de vídeo.

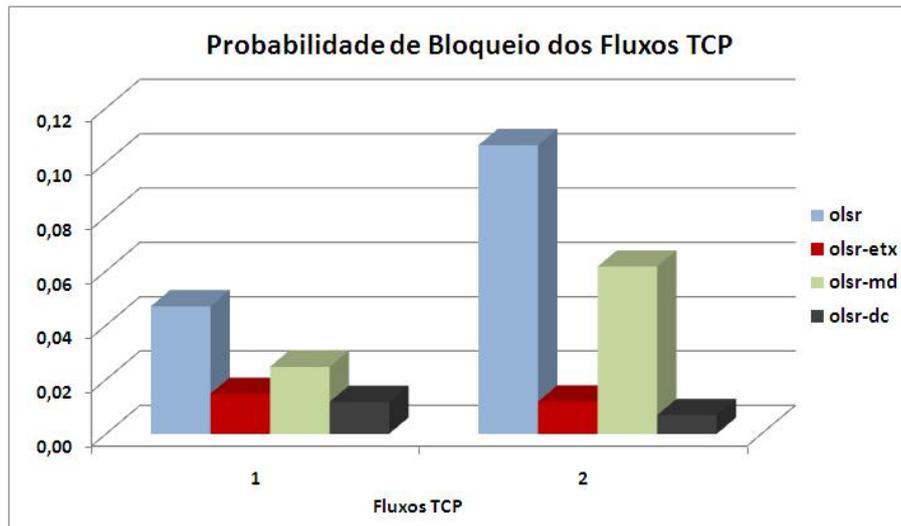


Figura 6.2: Probabilidade de Bloqueio dos Fluxos TCP

A Figura 6.2 mostra os resultados da probabilidade de bloqueio dos fluxos TCP. Percebe-se que o protocolo OLSR-DC possui a menor probabilidade de bloqueio, sendo que apenas o protocolo OLSR-ETX se aproxima dos resultados obtidos por este. Isso ocorre devido os dois protocolos usarem a mesma métrica para o roteamento dos pacotes TCP. Entretanto, a diferença se dá por mudanças no algoritmo de roteamento aplicadas no protocolo OLSR-DC.

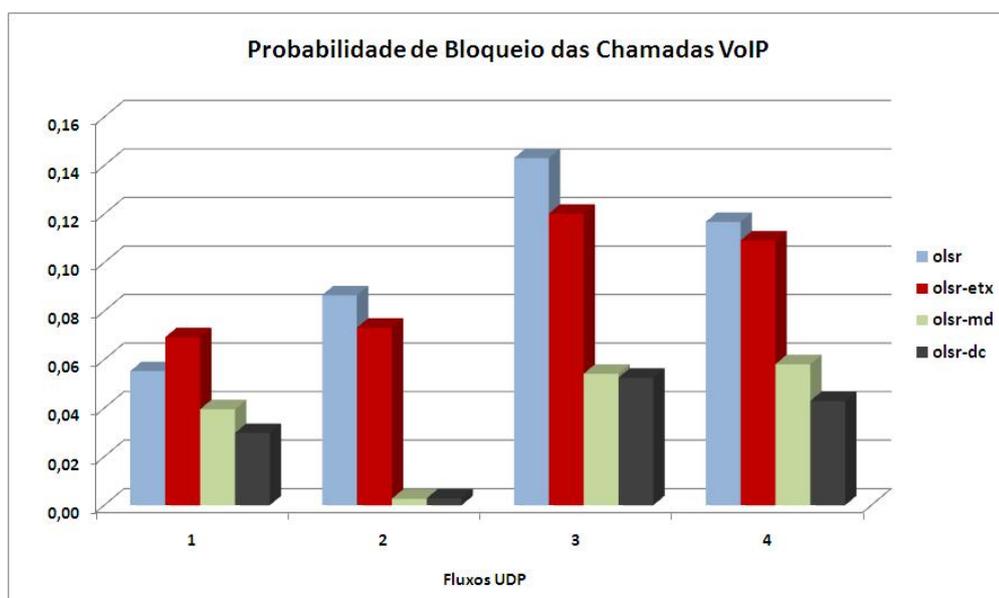


Figura 6.3: Probabilidade de Bloqueio das Chamadas VoIP

A Figura 6.3 mostra a probabilidade de bloqueio de cada uma das chamadas VoIP, sendo os fluxos UDP 1 e 2 correspondentes a primeira chamada, e os fluxos 3 e 4 correspondentes a segunda chamada. Na maioria dos fluxos UDP o protocolo OLSR-DC

possui a menor probabilidade de bloqueio junto com protocolo OLSR-MD, que em alguns momentos supera-o.

A diferença entre os valores da primeira e da segunda chamadas VoIP se dá pelo fato dos nós envolvidos na segunda chamada VoIP estarem mais distante e em uma zona de maior interferência dos demais nós, como ilustrado na Figura 6.1.

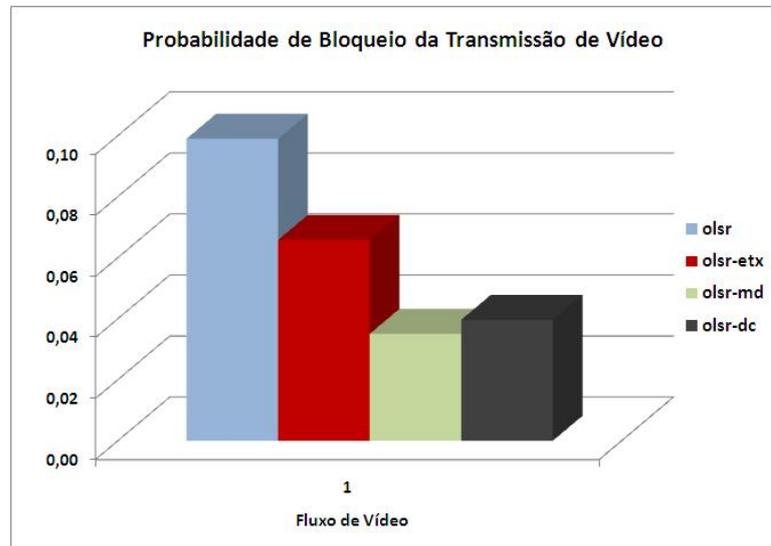


Figura 6.4: Probabilidade de Bloqueio da Transmissão de Vídeo

O protocolo OLSR-MD possui a menor probabilidade de bloqueio com relação à transmissão de vídeo, como é mostrado na Figura 6.4. A diferença entre os protocolos OLSR-DC e OLSR-MD acontece, pois o tempo que o protocolo OLSR-DC leva para determinar o tipo de pacote (TCP ou UDP) que irá ser roteado.

Nas chamadas VoIP e na transmissão de vídeo, o desempenho dos protocolos OLSR-DC e OLSR-MD são próximos devido os dois protocolos utilizarem a mesma métrica para o roteamento dos pacotes UDP. Sendo assim, com relação aos pacotes UDP, o protocolo OLSR-DC se aproxima do protocolo OLSR-MD, assim como com relação aos pacotes TCP, o protocolos OLSR-DC se aproxima do protocolo OLSR-ETX.

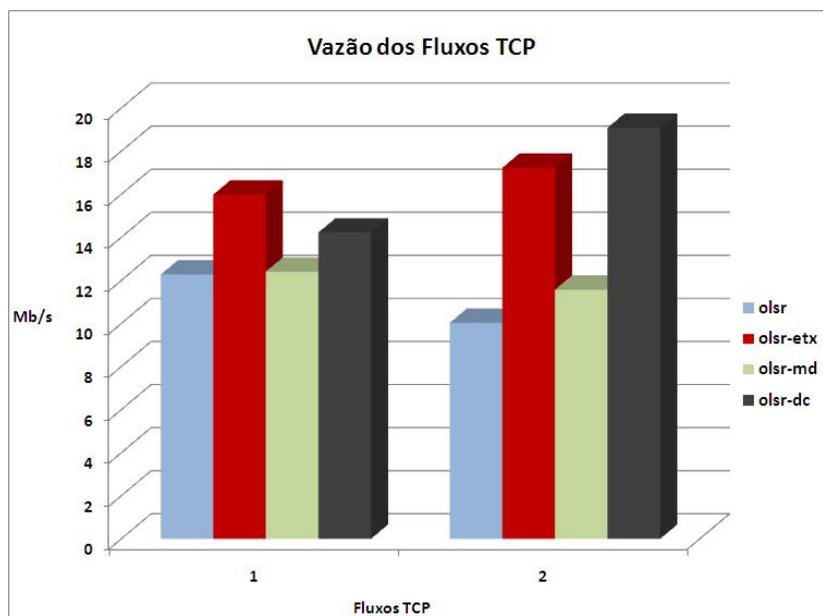


Figura 6.5: Vazão dos Fluxos TCP

A Figura 6.5 mostra a vazão dos fluxos TCP. Nota-se que o protocolo OLSR-ETX possui o melhor desempenho no fluxo 1, seguido pelo protocolo OLSR-DC. O protocolo OLSR-DC possui o melhor desempenho no fluxo 2. Isto ocorre, pois o fluxo 2 possui maior distância e conseqüentemente mais concorrência com os demais fluxos, portanto o protocolo OLSR-DC conseguiu rotear mais eficientemente os tráfegos.

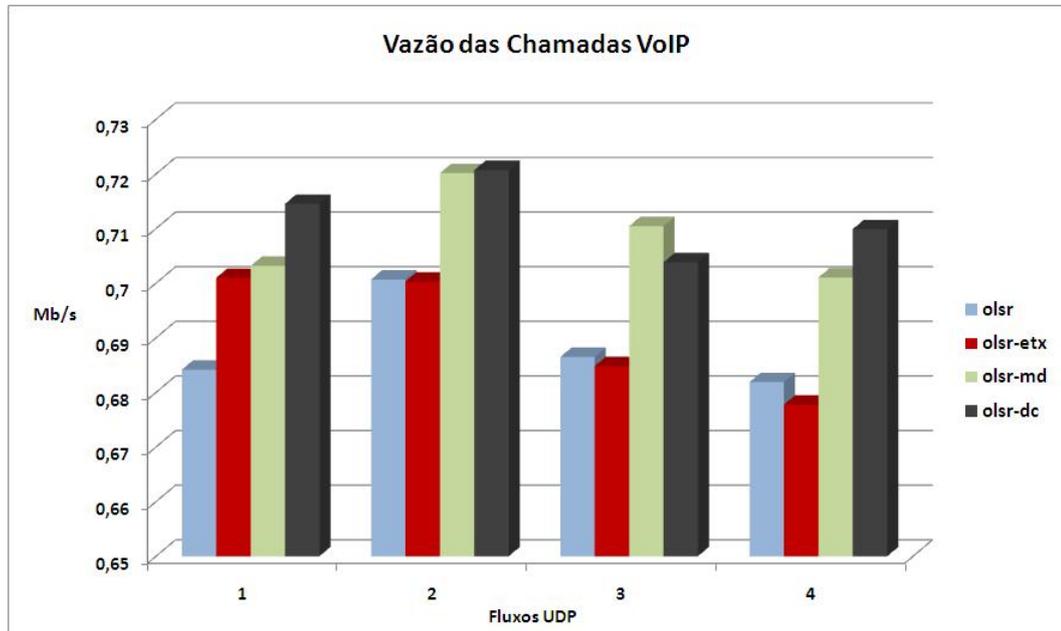


Figura 6.6: Vazão as Chamadas VoIP

A Figura 6.6 mostra a vazão dos fluxos UDP das chamadas VoIP. Percebe-se que o protocolo OLSR-DC possui o melhor desempenho, seguido do protocolo OLSR-MD.

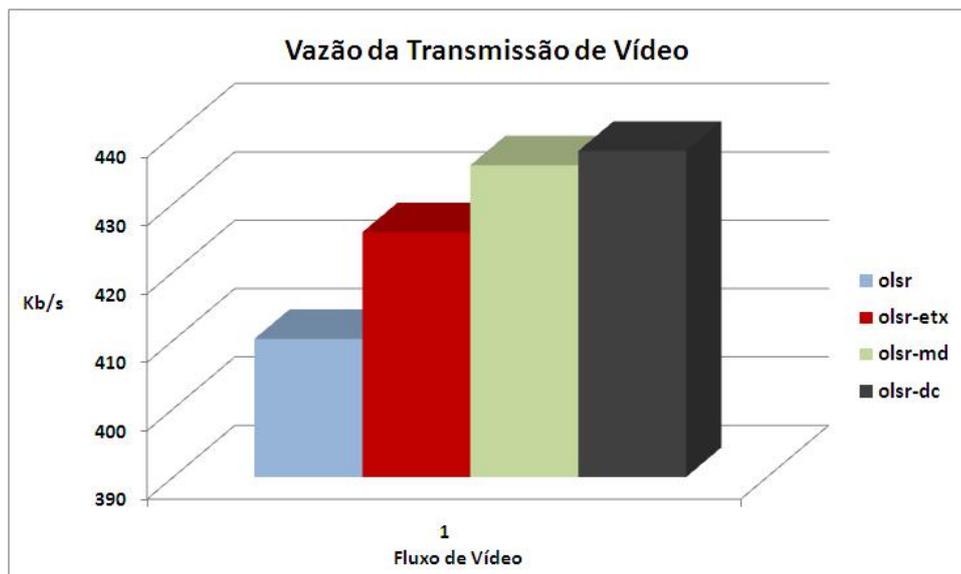


Figura 6.7: Vazão da Transmissão de Vídeo

Os protocolos que usam a métrica MD para rotear os pacotes de vídeo obtiveram os melhores valores de vazão, onde o protocolo OLSR-DC obteve uma pequena vantagem, como mostra a Figura 6.7.

A melhor vazão do protocolo OLSR-DC nos tráfegos multimídia se justifica pelo fato do mesmo distribuir os fluxos de maneira mais eficiente no período de alta concorrência,

distribuindo os tráfegos por caminhos que são mais vantajosos de acordo com o tipo de pacote processado no momento.

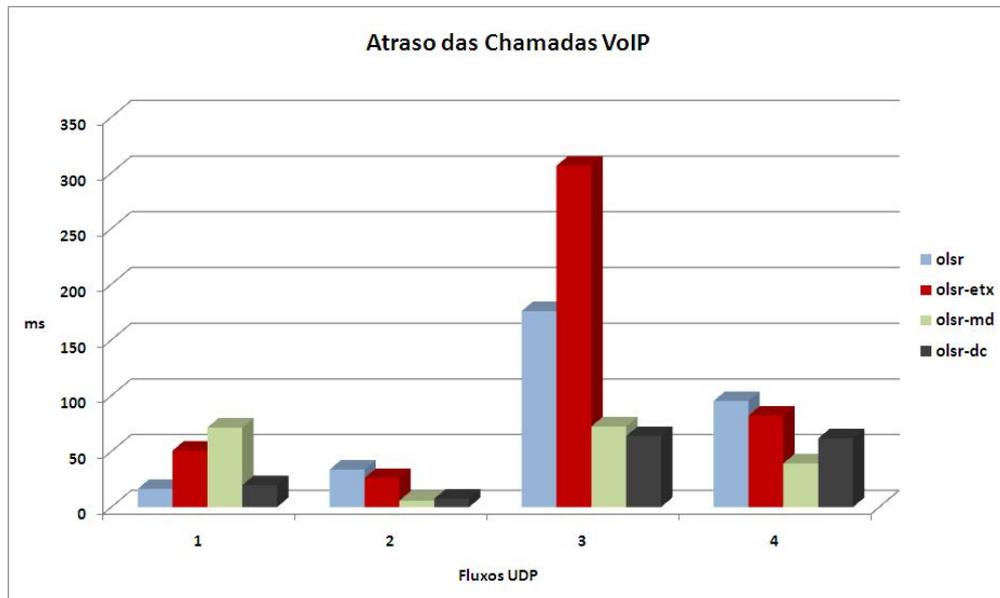


Figura 6.8: Atraso das Chamadas VoIP

No gráfico de atraso dos fluxos, mostrado na Figura 6.8, no geral, o protocolo OLSR-DC apresenta os melhores resultados em relação aos demais protocolos.

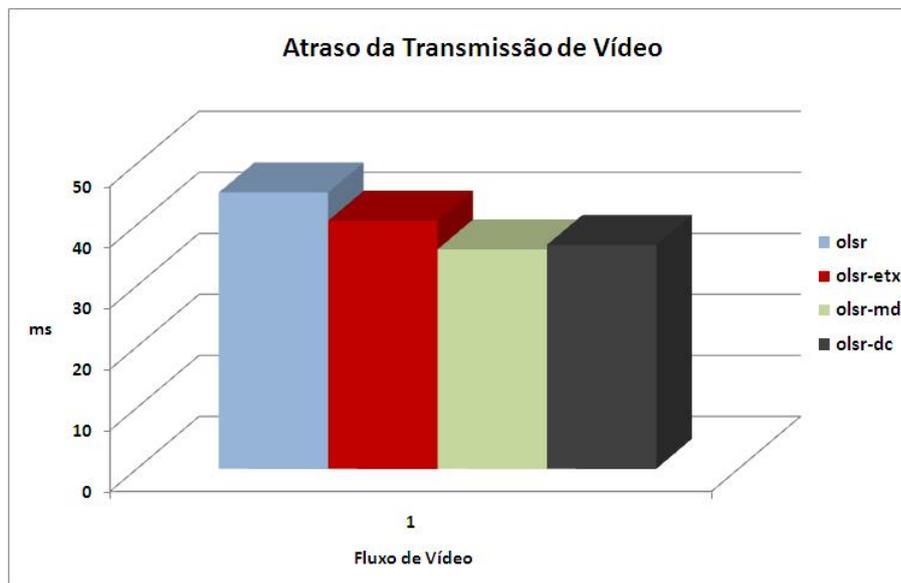


Figura 6.9: Atraso da Transmissão de Vídeo

A Figura 6.9 mostra os valores de atraso da transmissão de vídeo, onde se destacam os protocolos OLSR-DC e OLSR-MD, que mais uma vez, obtiveram os melhores valores, os quais forem similares.

O gráfico da Figura 6.10 mostra as taxas de *jitter* para cada um dos fluxos UDP das chamadas VoIP, nota-se que o protocolo OLSR-DC possuiu as melhores taxas de *jitter*, sendo superado em apenas no fluxo 4 pelo protocolo OLSR-MD.

Em relação a transmissão de vídeo, os protocolos OLSR-DC e OLSR-MD possuíram valores de *jitter* muito próximos, como mostra a Figura 6.11, sendo que o protocolo OLSR-MD obteve uma pequena vantagem.

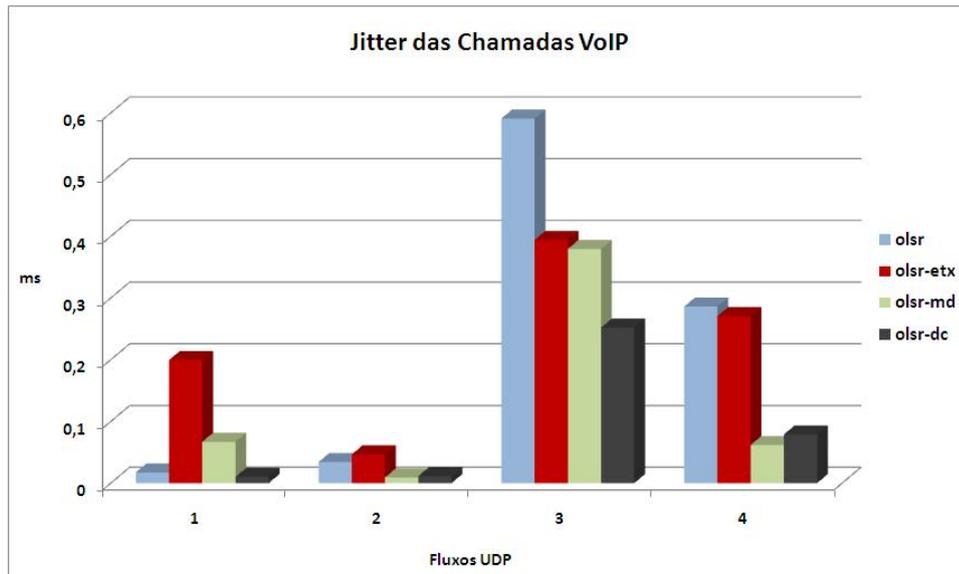


Figura 6.10: Jitter das Chamadas VoIP

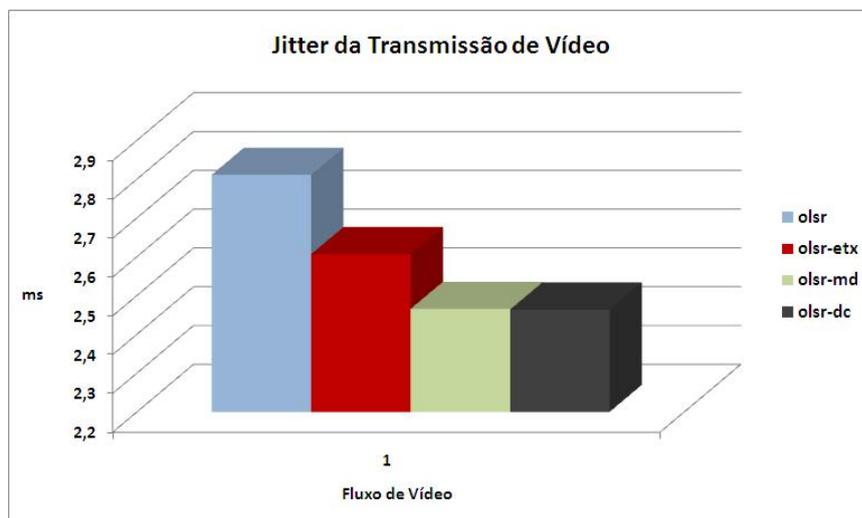


Figura 6.11: Jitter da Transmissão de Vídeo

De um modo geral, as simulações podem ser divididas em dois momentos, um de maior concorrência entre os fluxos, e outro onde somente o fluxo de vídeo foi transmitido. Os dados mostraram que no momento em que a rede possui uma maior concorrência entre os fluxos, alguns multimídia (VoIP e vídeo) e outros de dados, o protocolo OLSR-DC apresenta um melhor desempenho, pois visa realizar o roteamento dos fluxos baseados em suas particularidades, como atraso ou qualidade dos enlaces, conseguindo assim distribuir os tráfegos de forma mais eficaz.

6.2 Análise do Ponto de Vista do Usuário (QoE)

Levando em consideração a perspectiva do usuário em relação ao vídeo recebido pelo mesmo, nesta seção serão mostrados os *frames* resultantes da transmissão de vídeo de cada um dos protocolos e os correspondentes valores das métricas de QoE para cada um dos vídeos transmitidos pelos protocolos.

Utilizou-se métricas objetivas e subjetivas para se avaliar a qualidade do vídeo trans-

mitido [ITU-R, 1990]. As métricas subjetivas de QoE avaliam a qualidade das aplicações multimídias baseado na opinião do receptor. Como métrica subjetiva foi utilizada o *Mean Opinion Score* (MOS).

O MOS é uma escala de avaliação de percepção do vídeo que tenta quantificar a qualidade do vídeo baseada na impressão de qualidade humana. Essa qualificação é feita através de uma “nota” atribuída pelo usuário ao vídeo [Lambrecht and Verscheure, 1996].

As métricas objetivas de QoE estimam a qualidade do vídeo recebido pelo usuário através de modelos matemáticos quantitativos, cujos valores computados são mapeados em valores subjetivos de qualidade. As principais métricas objetivas são: *Peak Signal to Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM) e *Video Quality Metric* (VQM) [Lambrecht and Verscheure, 1996].

O PSNR é uma métrica objetiva que compara a qualidade do vídeo recebido pelo usuário em relação ao vídeo original, comparando *frame* por *frame*, obtendo valores entre 0 e 100 [Wang et al., 2004]. A métrica SSIM é uma medição da distorção estrutural do vídeo, tentando assim se ter uma melhor correlação com a impressão subjetiva do usuário, obtendo-se valores entre 0 e 1, sendo que quanto mais próximo de 1, melhor qualidade do vídeo [Wang and Bovik, 2004] [Wang and Bovik, 2002].

A métrica VQM mede os danos de percepção que o vídeo sofreu, incluindo em uma única métrica fatores como: embaçamento, ruído, movimentos desritmados, distorção de blocos e cores. Valores entre 0 e 5, sendo que quanto mais próximo de 0 melhor qualidade do vídeo [Pinson and Wolf, 2004].

Os vídeos foram analisados utilizando-se o software MSU Video Quality Measurement Tool Software [msu, 2009]. O valor de PSNR é expresso em dB (decibel). Para que um vídeo seja considerado de boa qualidade deve ter um valor médio de PSNR no mínimo de 30dB. Essa afirmação é baseada no mapeamento de valores de PSNR para MOS. Veja na Tabela 6.3 o mapeamento de PSNR para MOS [Lambrecht and Verscheure, 1996].

Tabela 6.3: Relação entre PSNR e MOS

PSNR (dB)	MOS
> 37	(5) Excelente
31 – 37	(4) Bom
25 – 30	(3) Regular
20 – 25	(2) Pobre
< 20	(1) Ruim

A Tabela 6.4 mostra os valores médios de PSNR, os correspondentes valores de MOS dos vídeos transmitidos nas simulações, os valores médios de VQM e os valores médios de SSIM dos vídeos transmitidos por cada um dos protocolos.

Tabela 6.4: Valores de VQM, SSIM, PSNR e MOS obtidos pelos protocolos

Protocolo	VQM	SSIM	PSNR	MOS
OLSR	4.9	0.75	15	Ruim
OLSR-ETX	2.7	0.86	27	Regular
OLSR-MD	1.6	0.91	32	Bom
OLSR-DC	1.7	0.90	31	Bom

Os dados da Tabela 6.4 mostram que os protocolos OLSR-MD e OLSR-DC possuíram os melhores valores com relação à qualidade geral do vídeo transmitidos, possuindo segundo a escala MOS uma qualidade “boa” de vídeo.



Figura 6.12: Frame do Vídeo Original

A seguir serão mostrados alguns *frames* dos vídeos transmitidos pelos protocolos. Os *frames* dão um respaldo visual e, conseqüentemente, um retorno sobre a qualidade do vídeo, permitindo comparar o desempenho de cada protocolo em relação as aplicações multimídia.

Os *frames* mostrados são correspondentes ao tempo de trinta segundos de transmissão do vídeo, “Bridge (close)” [cif, 2009]. No tempo de transmissão em questão a concorrência entre os fluxos se encontra no ápice, pois todos os fluxos estão sendo transmitidos. A Figura 6.12 mostra o *frame* original do vídeo no tempo de trinta segundos.

A Figura 6.13a exibe o *frame* resultante da transmissão pelo protocolo OLSR, percebe-se que o vídeo se torna de imperceptível, devido ao atraso, e principalmente, a quantidade de perdas resultante da concorrência com os demais fluxos.

A Figura 6.13b mostra o *frame* referente à transmissão pelo protocolo OLSR-ETX, o vídeo é perceptível ao usuário com uma qualidade regular, entretanto o mesmo possui um grande atraso quando comparado ao *frame* original.

A Figura 6.13c reproduz o *frame* referente à transmissão pelo protocolo OLSR-MD, o vídeo é perceptível ao usuário com uma qualidade boa, sendo este o qual obteve um menor atraso com relação ao vídeo original no *frame* escolhido, referente ao instante de trinta segundos de transmissão.

A Figura 6.13d mostra o *frame* referente à transmissão pelo protocolo OLSR-DC, o vídeo é perceptível ao usuário com uma qualidade boa, que possibilita uma percepção do usuário semelhante o do vídeo original.

A partir dos dados mostrados, conclui-se que no cenário mostrado, e nas condições em que foram realizadas as simulações o protocolo OLSR-DC é o que consegue se aproximar dos requisitos de ambos os tipos de tráfego, multimídia e de dados.

Deve-se levar em consideração que o cenário em questão visa retratar as condições da WMN presente na UFPA, um cenário com grande interferência natural e obstáculos, portanto um dos cenários onde a comunicação sem fio se torna um grande desafio [Aguiar et al., 2007].

Os fatos citados acima fazem com que o protocolo OLSR-DC seja mais adequado as WMNs que possuem os mais variados tráfegos, característica comum nas WMNs com



(a) Frame Transmitido pelo Protocolo OLSR



(b) Frame Transmitido pelo Protocolo OLSR-ETX



(c) Frame Transmitido pelo Protocolo OLSR-MD



(d) Frame Transmitido pelo Protocolo OLSR-DC

Figura 6.13: Frames Transmitidos Pelos Protocolos

muitos usuários finais que usam os mais diversos tipos de aplicações e esperam um serviço de boa qualidade.

Entretanto, o protocolo OLSR-DC acaba possuindo desvantagens: maior consumo de memória (aumento da tabela de roteamento) e um maior tempo para o encaminhamento do pacote (determinar se um pacote é TCP ou UDP). Sendo assim este pode ter problema de escalabilidade em WMNs extremamente grandes, devido a limitação dos *hardwares* utilizados nesse tipo de rede. Contudo, com os constantes avanços dos *hardwares* desenvolvidos para as redes sem fio este pode não se tornar um problema real.

Capítulo 7

Conclusão

Este trabalho apresentou uma extensão para o protocolo OLSR baseada na escolha dinâmica de métricas, a qual é feita a partir do protocolo de camada de transporte usado no pacote processado no momento. O objetivo com isso é tentar atender as necessidades das aplicações de acordo com suas particularidades.

Mostrou-se que o protocolo OLSR-DC consegue ter um desempenho muito próximo dos protocolos que utilizam uma métrica somente, como OLSR-ETX e OLSR-MD. O protocolo OLSR-ETX conseguem comprovadamente [Couto et al., 2003] atender mais as necessidades de aplicações que utilizam o protocolo TCP, devido as suas características já citadas nas seções anteriores, assim como o protocolo OLSR-MD consegue atender mais as necessidades das aplicações que tem como base o protocolo UDP.

Dentro desse contexto o protocolo OLSR-DC teve como objetivo extrair essa capacidade de cada um dos protocolos citados acima, ou seja, tentou-se ter um desempenho semelhante ao do protocolo OLSR-ETX com relação às aplicações que utilizam o protocolo TCP, e desempenho semelhante ao OLSR-MD com relação às aplicações que tem como base o protocolo UDP, geralmente aplicações multimídia.

Portanto o protocolo OLSR-DC possui um desempenho próximo ao do protocolo OLSR-ETX e maior do que o protocolo OLSR-MD, quando se trata de aplicações que utilizam o protocolo TCP. Da mesma forma o OLSR-DC conseguiu teve um desempenho, se tratando de aplicações que usam o protocolo UDP, semelhante ao do protocolo OLSR-MD, sendo algumas vezes até melhor que este, e superior ao protocolo OLSR-ETX.

Sendo assim este trabalho propôs uma extensão do protocolo OLSR, que apesar de gerar um pouco mais processamento e consumo de memória, consegue atender melhor as aplicações, por tentar suprir as necessidades de cada uma das aplicações, de acordo com suas características particulares.

7.1 Trabalhos futuros

Entre os trabalhos futuros pretende-se implementar a extensão proposta deste trabalho na WMN presente na UFPA [Aguiar et al., 2007], e assim realizar um *testbed* com este rede real, para assim comprovar por completo a eficiência mostrada nas seções anteriores.

Outra intenção é a de se estudar a possibilidade de adição de novas métricas, ou de substituição de algumas das métricas utilizadas, para prover o roteamento de determinados tipos de pacotes, uma das extensões que podem ser analisadas é [Moreira et al., 2008], dentre outras. Atualmente o desempenho do protocolo proposto esta ligado diretamente as métricas utilizadas nele, ou seja, a descoberta de uma métrica que melhora a perfor-

mance do protocolo TCP pode ser utilizada sem que esta influencie no desempenho do protocolo UDP, e vice-versa.

Além disso, o principal trabalho futuro previsto é um aperfeiçoamento do algoritmo de roteamento utilizado, visando suprir o processamento extra que foi gerado com este, algumas idéias para se melhorar o roteamento seriam as seguintes propostas [Sobrinho, 2004], [Noto and Sato, 2000], [Ikikardes et al., 2007] e [Hu et al., 2007].

Referências

- Bridge (Close)*. <http://www.tkn.tuberlin.de/research/evalvid/cif.html>, 2009.
- Evalvid*. <http://www.tkn.tu-berlin.de/research/evalvid/>, 2009.
- MSU Video Quality Measurement Tool Software*. http://compression.ru/video/quality_measure/index_en.html, 2009.
- E. Aguiar, P. Bittencourt, W. Moreira, and A. Abelém. Estudo comparativo de protocolos de roteamento para redes mesh na região amazônica. *25 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Maio 2007.
- I. Akyildiz and X. Wang. A survey on wireless mesh networks. *IEEE Communications Magazine, New York (USA)*, v. 43, n. 9, September 2005.
- N. Aslam, W. Phillips, , and W. Robertson. Composite metric for quality of service routing in olsr. *Canadian Conference on Electrical and Computer Engineering*, 2004.
- H. Badis, A. Munaretto, K. Agha, and G. Pujolle. Qos for ad hoc networking based on multiple metrics: bandwidth and delay. *Fifth IFIP International Conference on Mobile and Wireless Communications Networks.*, May 2003.
- A. Bakre and B. R. Badrinath. I-tcp: Indirect tcp for mobile hosts. *Distributed Computing Systems, Proceedings of the 15-th International Conference*, June 1995.
- J. Balam and J.D. Gibson. Multiple descriptions and path diversity for voice communications over wireless mesh networks. *IEEE Transactions on Multimedia*, pages 1073–1088, August 2007.
- H. Belakrishnan, S. Seshan, E. Amir, and R. Katz. Improving tcp/ip performance over wireless networks. *Proceedings of the 1st (MOBICOM), Berkeley, CA*, November 1995.
- C. Breuel. Redes em malha sem fios. *Instituto de Matemática e Estatística, Universidade de São Paulo, 2004. URL: http://grenoble.ime.usp.br/movel/Wireless_Mesh_Networks.pdf.*, 2007.
- R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 2005.
- M. Chinta and S. Helal. Ilc-tcp: An interlayer collaboration protocol for tcp performance improvement in mobile and wireless environments. *Proceedings of the Third IEEE Wireless Communications and Networking Conference (WCNC)*, March 2003.
- T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). *RFC 3626*, 2003.

- W. Cordeiro, E. Aguiar, W. Moreira, A. Abelem, , and M. Stanton. Providing quality of service for mesh networks using link delay measurements. *16th International Conference on Computer Communications and Networks*, 2007.
- D. S. J. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *9th Annual International Conference on Mobile Computing and Networking*, 2003.
- K. Fall and K. Varadhan. The network simulator – ns-2. URL: <http://www.isi.edu/nsnam/ns/>, 2007.
- G. Held. Wireless mesh networks. *1Aa ed. USA: Auerbach Publications*, June 2005.
- X. Hu, M. Lee, and T. Saadawi. Progressive route calculation protocol for wireless mesh networks. *Communications, 2007. ICC '07. IEEE International Conference*, June 2007.
- T. Ikikardes, M. Hoftauer, A. Kaelin, and M. May. A robust, responsive, distributed tree-based routing algorithm guaranteeing n valid links per node in wireless ad-hoc networks. *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium*, July 2007.
- ITU-R. *Recommendation BT 500-7. Methodology for the Subjective Assessment of the Quality of Television Pictures*. Technical Report, 1990.
- R. Jain. The art of computer systems: Techniques for experimental design, measurement, simulation, and modeling. *John Wiley & Sons, 1st edition*, 1991.
- R. Jain. Quality of experience. *IEEE Multimedia*, 11:90–98, June 2004.
- D. Kim, J. Cano, P. Manzoni, and C. Toh. A comparison of the performance of tcp-reno and tcp-vegas over manets. *Wireless Communication Systems, 2006. ISWCS '06. 3rd International Symposium*, September 2006.
- K. Kowalik and M. Davis. Why are there so many routing protocols for wireless mesh networks? *Irish Signal and Systems Conference*, 2006.
- C. Lambrecht and O. Verscheure. Perceptual quality measure using a spatio – temporal model of the human visual system. *Digital Video Compression: Algorithms and Technologies*, pages 450–461, 1996.
- J. Leguay, V. Conan, and Friedman. Qos routing in olsr with several classes of service. *4th Annual IEEE International Conference on Pervasive Computing and Communications Workshop*, 2006.
- J. Liu and S. Singh. Atp: Tcp for mobile ad hoc networks. *IEEE Journal. On Selected Areas on Communications*, vol. 9, no. 7, 2001.
- W. Moreira, E. Aguiar, A. Abelém, and Stanton M. Using multiple metrics with the optimized link state routing protocol for wireless mesh networks. *26 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, May 2008.
- A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle. Qolsr: Qos routing over olsr protocol. *5ème Rencontres Francophones sur les aspects Algorithmiques dès Télécommunications, Banyuls-sur-mer*, 2003.

- C. Murthy and B. Manoj. Ad hoc wireless networks: Architectures and protocols. *1^a ed. USA: Prentice Hall Professional Technical Reference*, June 2004.
- M. Noto and H. Sato. A method for the shortest path search by extended dijkstra algorithm. *Systems, Man, and Cybernetics, 2000 IEEE International Conference on Volume 3*, October 2000.
- Institute of Electrical and Eletronic Engineering. Ieee 802.11. information technology – telecommunications and information exchange between systems - local and metropolitan area network - specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *1Aa ed. USA: Auerbach Publications*, 1999.
- D. Passos, D. Teixeira, D. Muchaluat-Saade, L. Magalhães, and C. Albuquerque. Mesh network perfomance measurements. *5th International Information and Telecommunication Technologies Symposium*, 2006.
- A. Perkis, S. Munkeby, and Hillestad O. A model for measuring quality of experience. *Signal Processing Symposium*, pages 198–201, June 2006.
- M. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Transacions on Broadcasting*, 50:312–322, 2004.
- J. Postel. Internet protocol - darpa internet program protocol specification. *RFC 791, USC/Information Sciences Institute*, 1981.
- Projeto ReMesh. <http://mesh.ic.uff.br/>. *September*, 2007.
- J. Sobrinho. Algebra and algorithms for qos path computation and hop-by-hop routing in the internet. *Networking, IEEE/ACM Transactions on Volume 10, Issue 4*, August 2004.
- K. Thomson, G. J. Miller, and Wilder R. Wide-area internet traffic patterns and characteristics. *IEEE Network, Vol. 11, No. 6*, November 1997.
- Z. Wang and A. Bovik. Image quality assessment from error visibility to structural similarity. *IEEE Trans Image Processing*, pages 600–612, 2004.
- Z. Wang and A. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, pages 81–84, 2002.
- Z. Wang and J. Crowcroft. Qos routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, pages 1228–1234, 1996.
- Z. Wang, A. Bovik, and L. Lu. Why is image quality assessment so difficult? *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002.
- Z. Wang, L. Lu, and A. Bovick. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication, special issue on “Objective Video Quality Metrics*, February 2004.
- G. Ying, T. Kunz, and L. Lamont. Quality of service routing in ad-hoc networks using olsr. *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference*, January 2003.

Y. Zhang, J. Luo, and H. Hu. Wireless mesh networks: Architectures, protocols and standards. *Auerbach Publications, 1st edition, 2007.*

Apêndice A

Script de Simulação

Código Fonte A.1: Script de Simulação do Protocolo OLSR-DC

```
# =====  
# Define options  
# =====  
set opt(chan) Channel/WirelessChannel ;  
# channel type  
set opt(prop) Propagation/Shadowing ;  
# radio-propagation model  
set opt(netif) Phy/WirelessPhy ;  
# network interface type  
set opt(mac) Mac/802_11 ;  
# MAC type  
set opt(ifq) Queue/DropTail/PriQueue ;  
# interface queue type  
set opt(ll) LL ;  
# link layer type  
set opt(ant) Antenna/OmniAntenna ;  
# antenna model  
set opt(ifqlen) 50 ;  
# max packet in ifq  
set opt(nn) 10 ;  
# number of mobilenodes  
set opt(adhocRouting) OLSR ;  
# routing protocol  
set opt(x) 1000 ;  
# x coordinate of topology  
set opt(y) 1000 ;  
# y coordinate of topology  
set opt(seed) X  
set opt(stop) 100 ;  
# time to stop simulation  
# =====  
  
#  
# check for random seed
```

```

#
if {$opt(seed) > 0} {
  puts "Seeding_Random_number_generator_with_$opt(seed)\n"
  ns-random $opt(seed)
}

#Ganho das antenas
Antenna/OmniAntenna set Gt_ 18.0 ;
# Ganho das antenas utilizadas
Antenna/OmniAntenna set Gr_ 18.0 ;
# Ganho das antenas utilizadas

#####

#Camada fisica
#Phy/WirelessPhy set CStresh_ 2.511886431509582e-11 ;
# -76 dBm para dsss 802.11 de acordo com
#ANSI/IEEE Std 802.11, 1999 Edition (R2003)
#Phy/WirelessPhy set RXThresh_ 1.0000000000000001e-11 ;
# -80dbm para dsss 802.11 de acordo com ANSI/IEEE Std 802.11,
# 1999 Edition (R2003)
Phy/WirelessPhy set bandwidth_ 54Mb ;
# Banda oferecida
Phy/WirelessPhy set Pt_ 0.05011 ;
# Potencia de Transmissao em Watts (17dBm padrao ao WRT54G)
#17dBm <=> 0.05011 Watts
#16dBm <=> 0.03981 Watts
#15dBm <=> 0.03162 Watts
#12dBm <=> 0.01584 Watts
#10dBm <=> 0.01 Watts
#8dBm <=> 0.00630 Watts
#5dBm <=> 0.00316 Watts
Phy/WirelessPhy set freq_ 2.422e+9 ;
# Frequencia de 2.422GHz no Canal 3
# Receive sensitivity.
Phy/WirelessPhy set RXThresh_ 6.309573e-12
Phy/WirelessPhy set CStresh_ 6.309573e-12

#####

#Parametros obtidos das medicoes para o cenario em questao
Propagation/Shadowing set pathlossExp_ 1.59 ;
# Expoente de perdas
Propagation/Shadowing set std_db_ 5.4 ;
# Desvio padrao (dB)

#####

```

```

#Camada MAC
#802.11g:
Mac/802_11 set CWMin 15;
Mac/802_11 set CWMax 1023;
Mac/802_11 set SlotTime 0.000009;
Mac/802_11 set CCATime 0.000003;
Mac/802_11 set RxTxTurnaroundTime 0.000002;
Mac/802_11 set SIFSTime 0.000010;
Mac/802_11 set PreambleLength 96;
Mac/802_11 set PLCPHeaderLength 40;
Mac/802_11 set PLCPDataRate 6.0e6; ### 6Mbps
Mac/802_11 set MaxPropagationDelay 0.0000005; # // 0.5us
Mac/802_11 set dataRate_          54.0e6          ;
Mac/802_11 set basicRate_         20.0e6          ;
Mac/802_11 set RTSThreshold_ 3000;

#
# create simulator instance
#
set ns_ [new Simulator]

#
# control OLSR behaviour from this script -
# commented lines are not needed because
# those are default values
#
Agent/OLSR set use_mac_          true
Agent/OLSR set debug_            false
Agent/OLSR set willingness       3
Agent/OLSR set hello_ival_       2
Agent/OLSR set tc_ival_          5
Agent/OLSR set mpr_algorithm_    2
Agent/OLSR set routing_algorithm_ 2
Agent/OLSR set link_quality_     2
Agent/OLSR set fish_eye_         false
Agent/OLSR set link_delay_       true
Agent/OLSR set tc_redundancy_    3
Agent/OLSR set c_alpha_          0.4
Agent/OLSR set dynamic_choice_   true
Agent/OLSR set tcp_metric_       1
#define TCP_HOP_COUNT 0
#define TCP_ETX 1
#define TCP_ETX_OW 2
#define TCP_ML 3
#define TCP_MD 4
#define TCP_FUZZY 5
Agent/OLSR set udp_metric_       4

```

```

#define      UDP_HOP_COUNT          0
#define      UDP_ETX                 1
#define      UDP_ETX_OW              2
#define      UDP_ML                  3
#define      UDP_MD                  4
#define      UDP_FUZZY_ETX_OW_DELAY  5
#define      UDP_FUZZY_ETX_DELAY     6

#
# open traces
#
$ns_ use-newtrace
set tracefd [open wtrace.tr w]
set namtrace [open simulation.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

#
# create topography object
#
set topo [new Topography]

#
# define topology
#
$topo load_flatgrid $opt(x) $opt(y)
#include <typeinfo>
#
# create God
#
create-god $opt(nn)

#
# configure mobile nodes
#
$ns_ node-config -adhocRouting $opt(adhocRouting) \
                -llType $opt(ll) \
                -macType $opt(mac) \
                -ifqType $opt(ifq) \
                -ifqLen $opt(ifqlen) \
                -antType $opt(ant) \
                -propType $opt(prop) \
                -phyType $opt(netif) \
                -channelType $opt(chan) \
                -topoInstance $topo \
                -wiredRouting OFF \
                -agentTrace ON \
                -routerTrace ON \

```

–macTrace OFF

```
for {set i 0} {$i < $opt(nn)} {incr i} {  
    set node_($i) [$ns_ node]  
}
```

```
#  
# positions
```

```
$node_(0) set X_ 160.0 #CAPACIT  
$node_(0) set Y_ 485.0  
$node_(0) set Z_ 15.0
```

```
$node_(1) set X_ 305.0 #DI  
$node_(1) set Y_ 277.0  
$node_(1) set Z_ 15.0
```

```
$node_(2) set X_ 340.0 #SECOM  
$node_(2) set Y_ 226.0  
$node_(2) set Z_ 15.0
```

```
$node_(3) set X_ 270.0 #Grad Basico  
$node_(3) set Y_ 32.0  
$node_(3) set Z_ 15.0
```

```
$node_(4) set X_ 476.0 #Reitoria  
$node_(4) set Y_ 200.0  
$node_(4) set Z_ 15.0
```

```
$node_(5) set X_ 628.0 #Incubadora  
$node_(5) set Y_ 320.0  
$node_(5) set Z_ 15.0
```

```
$node_(6) set X_ 570.0 #Musica  
$node_(6) set Y_ 440.0  
$node_(6) set Z_ 15.0
```

```
$node_(7) set X_ 780.0 #LABS  
$node_(7) set Y_ 480.0  
$node_(7) set Z_ 15.0
```

```
$node_(8) set X_ 918.0 #CT  
$node_(8) set Y_ 597.0  
$node_(8) set Z_ 15.0
```

```
$node_(9) set X_ 968.0 #Grad Profissional  
$node_(9) set Y_ 550.0  
$node_(9) set Z_ 15.0
```

```

##### VIDEO
set frag_size 1024

set udp [new Agent/myUDP]
$ns_ attach-agent $node_(3) $udp
$udp set packetSize_ 1052
#add udp header(8 bytes) and IP header (20 bytes)
$udp set_filename sd_a01
$udp set fid_ 11
$udp set class_ 11
set null [new Agent/myEvalvid_Sink]
$ns_ attach-agent $node_(6) $null
$ns_ connect $udp $null
$null set_filename rd_a01

#video_in
set original_file_name st_a01
set trace_file_name video1.dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

set pre_time 0
set video_start 10.0

while {[eof $original_file_id] == 0} {
gets $original_file_id line

scan $line "%d%s%d%d%f" no_ frametype_ length_ tmp1_ tmp2_

set tmp2_ [expr double($tmp2_+ $video_start)]

set time [expr int(($tmp2_ - $pre_time)*1000000.0)]

if { $frametype_ == "I" } {
    set type_v 1
    set prio_p 0
}

if { $frametype_ == "P" } {
    set type_v 2
    set prio_p 0
}

if { $frametype_ == "B" } {
    set type_v 3
    set prio_p 0
}
}

```

```

if { $frametype_ == "H" } {
    set type_v 1
    set prio_p 0
}

puts $trace_file_id "$time_ $length_ $type_v_ $prio_p_ $frag_size"
set pre_time $tmp2_
}

close $original_file_id
close $trace_file_id
set video_stop $tmp2_
puts "$video_stop"

set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set video1 [new Application/Traffic/myEvalvid]
$video1 attach-agent $sdp
$video1 attach-tracefile $trace_file

$ns_ at $video_start "$video1_start"
$ns_ at $video_stop "$video1_stop"

$ns_ at [expr $video_stop + 1.0] "$null_closefile"
#$ns_ at [expr $video_stop + 1.0] "finish"

#After the simulation
#!/etmp4 sd_a01 rd_a01 st_a01 a01.mp4 a01e
#!/ffmpeg -i a01e.mp4 a01e.yuv
#!/psnr.exe 352 288 420 akiyo_cif.yuv a01e.yuv

#####

####    TCP
#1
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 1

set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(7) $tcp
$ns_ attach-agent $node_(3) $sink
$ns_ connect $tcp $sink
$tcp set packetSize_ 200
$tcp set fid_ 1

set ftp [new Application/FTP]
$ftp attach-agent $tcp

```

```

$ns_ at 17.0 "$ftp_start"
$ns_ at 67.0 "$ftp_stop"

#2
set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2

set sink1 [new Agent/TCP/Sink]
$ns_ attach-agent $node_(9) $tcp1
$ns_ attach-agent $node_(4) $sink1
$ns_ connect $tcp1 $sink1
$tcp1 set packetSize_ 200
$tcp1 set fid_ 2

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns_ at 18.0 "$ftp1_start"
$ns_ at 68.0 "$ftp1_stop"

#####

# setup UDP connection
#1
set udp2 [new Agent/UDP]
$udp2 set class_ 3
set null2 [new Agent/Null]
$ns_ attach-agent $node_(0) $udp2
$ns_ attach-agent $node_(5) $null2
$ns_ connect $udp2 $null2
$udp2 set fid_ 3

set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 40 # (40 bytes = RTP + UDP + Payload)
$cbr2 set rate_ 8Kb
$cbr2 attach-agent $udp2
$ns_ at 5.0 "$cbr2_start"
$ns_ at 55.0 "$cbr2_stop"

#2
set udp1 [new Agent/UDP]
$udp1 set class_ 4
set null1 [new Agent/Null]
$ns_ attach-agent $node_(5) $udp1
$ns_ attach-agent $node_(0) $null1
$ns_ connect $udp1 $null1
$udp1 set fid_ 4

set cbr1 [new Application/Traffic/CBR]

```

```

$ubr2 set packetSize_ 40      # (40 bytes = RTP + UDP + Payload)
$ubr1 set rate_ 8Kb
$ubr1 attach-agent $udp1
$ns_ at 5.0 "$ubr1_start"
$ns_ at 55.0 "$ubr1_stop"

# 3
set udp3 [new Agent/UDP]
$udp3 set class_ 5
set null3 [new Agent/Null]
$ns_ attach-agent $node_(8) $udp3
$ns_ attach-agent $node_(1) $null3
$ns_ connect $udp3 $null3
$udp3 set fid_ 5

set cbr3 [new Application/Traffic/CBR]
$ubr2 set packetSize_ 40      # (40 bytes = RTP + UDP + Payload)
$ubr3 set rate_ 8Kb
$ubr3 attach-agent $udp3
$ns_ at 6.0 "$ubr3_start"
$ns_ at 56.0 "$ubr3_stop"

# 4
set udp4 [new Agent/UDP]
$udp4 set class_ 6
set null4 [new Agent/Null]
$ns_ attach-agent $node_(1) $udp4
$ns_ attach-agent $node_(8) $null4
$ns_ connect $udp4 $null4
$udp4 set fid_ 6

set cbr4 [new Application/Traffic/CBR]
$ubr2 set packetSize_ 40      # (40 bytes = RTP + UDP + Payload)
$ubr4 set rate_ 8Kb
$ubr4 attach-agent $udp4
$ns_ at 6.0 "$ubr4_start"
$ns_ at 56.0 "$ubr4_stop"

#####

## Label the Special Node in NAM
$ns_ at 0.0 "$node_(0)_label_CAPACIT"
$ns_ at 0.0 "$node_(1)_label_Dep_Informatica"
$ns_ at 0.0 "$node_(2)_label_SECOM"
$ns_ at 0.0 "$node_(3)_label_Grad_Basico"
$ns_ at 0.0 "$node_(4)_label_Reitoria"
$ns_ at 0.0 "$node_(5)_label_Incubadora"
$ns_ at 0.0 "$node_(6)_label_Musica"

```

```

$ns_ at 0.0 "$node_(7)_label_Laboratorios"
$ns_ at 0.0 "$node_(8)_label_Centro_Tec"
$ns_ at 0.0 "$node_(9)_label_Grad_Profissional"

#
# define initial node position in nam
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 20
}

#
# tell all nodes when the simulation ends
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at $opt(stop).0 "$node_($i)_reset";
}

$ns_ at $opt(stop).0002 "puts_\\"NS EXITING...\\"_";_ $ns_ halt"
$ns_ at $opt(stop).0001 "stop"

proc stop {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}

#
# begin simulation
#
puts "Starting_Simulation..."

$ns_ run

```